PROCEEDINGS OF THE THIRD INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB 3rd '98)

Vol. 1

Jan. 19-Jan. 21, 1998
B-Con Plaza, Beppu, Oita, JAPAN

Editor : Masanori Sugisaka

ISBN4-9900462-8-5

Proceeding of The Third International Symposium on

ARTIFICIAL LIFE AND ROBOTICS

(AROB 3rd '98)

January 19-21, 1998
B-Con Plaza, Beppu, Oita, JAPAN

Editor: Masanori Sugisaka

ISBN4-9900462-8-5, 1998

THE THIRD INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS (AROB 3rd '98)

ORGANIZED BY

Oita University
Under The Sponsorship of Ministry of Education,
Science, Sports, and Culture, Japanese Government

CO-SPONSORED BY

Robotics Society of Japan(RSJ, Japan)
Santa Fe Institute(SFI, USA)
The Institute of Electrical Engineers of Japan(IEEJ, Japan)
The Society of Instrument and Control Engineers(SICE, Japan)

CO-OPERATED BY

Japan Robot Association(JARA, Japan)
The Institute of Electrical and Electronics Engineers,
Tokyo Section(IEEE, USA)
The Institute of Electronics, Information and
Communication Engineers(IEICE, Japan)
The Institute of System, Control and Information
Engineers(ISCIE, Japan)

SUPPORTED BY

Asahi Shimbun
Beppu Municipal Government
Kyushu Bureau of International Trade and Industry, MITI
NHK Oita Station
Nihonkeizai Shimbun
Nikkan Kogyo Shimbun
Oita Asahi Broadcasting
OBS Broadcast Company
Oita Chamber of Commerce and Industry
Oitagodo Shinbunsya
Oita Industrial Group Society
Oita Municipal Government
Oita Prefectural Government
Oita System Control Society
Science and Technology Agency
TOS Broadcast Company

HONORARY PRESIDENT

M.Hiramatsu (Governor, Oita Prefecture)

ORGANIZING COMMITTEE

K.Abe (Tohoku University, Japan)
K.Akizuki (Waseda University, Japan)
S.Arimoto (Ritsumeikan University, Japan)
W.B.Arthur (Santa Fe Institute, USA)
C.Barrett (Los Alamos National Laboratory, USA)
Z.Bubunicki (Wroclaw University of Technology, Poland)
J.L.Casti (Santa Fe Institute, USA) (Vice Chairman)
T.Christaller (GMD-German National Research Center for Information Technology, Germany)
J.M.Epstein (Brookings Institution, USA)
T.Fujii (RIKEN, Japan)
S.Fujimura (The University of Tokyo, Japan)
Y.Fujita (The Japan Academy, Japan)
T.Fukuda (Nagoya University, Japan)
M.Gen (Ashikaga Institute of Technology, Japan)
T.Gomi (AAI, Canada)
H.Hagiwara (Kyoto School of Computer Science, Japan)
I.Harvey (University of Sussex, UK)
P.Husbands (University of Sussex, UK)
M.Ito (RIKEN, Japan)
D.J.G.James (Coventry University, UK)
T.Jinzenji (Sanyodenki Co. Ltd., Japan)
J.Johnson (The Open University, UK)
R.E.Kalaba (University of Southern California, USA)
H.Kashiwagi (Kumamoto University, Japan)
O.Katai (Kyoto University, Japan)
S.Kauffman (Santa Fe Institute, USA)
J.H.Kim (KAIST, Korea)
H.Kimura (The University of Tokyo, Japan)
S.Kitamura (Kobe University, Japan)
H.Kitano (Sony Computer Science Laboratory Inc., Japan)
K.Kyuma (Mitsubishi Electric Co., Japan)
C.G.Langton (Santa Fe Institute, USA)
J.J.Lee (KAIST, Korea)
C.Looney (University of Nevada-Reno, USA)
G.I.Marchuk (Russian Academy of Sciences, Russia)
G.Matsumoto (RIKEN, Japan)
K.Mastuno (MITI, AIST, Japan)
H.Miura (The University of Tokyo, Japan)
T.Mushya (Brain Function Laboratory, Inc., Japan)
T.Nagata (Inst. of Sys. and Inf. Tech. / Kyushu, Japan)
M.Nakamura (Saga University, Japan)

H.H.Natsuyama (Advanced Industrial Materials, USA)
Y.Nishikawa (Osaka Institute of Technology, Japan)
R.G.Palmer (Santa Fe Institute, USA)
R.Pfeifer (University of Zurich-Irchel, Switzerland)
M.Raibert (MIT, USA)
S.Rasmussen (Santa Fe Institute, USA)
T.S.Ray (ATR, Japan)
P.Schuster (Santa Fe Institute, USA)
T.Shibata (MITI, MEL, Japan)
K.Shimohara (ATR, Japan)
C.Sommerer (ATR, Japan)
L.Steels (VUB AI Laboratory, Belgium)
M.Sugisaka (Oita University, JAPAN) (Chairman)
K.Tamura (MITI, AIST, Japan)
H.Tanaka (Tokyo Medical & Dental University, Japan)
Y.Tokura (ATR, Japan)
N.Tosa (ATR, Japan)
K.Tsuchiya (Kyoto University, Japan)
S.Ueno (Kyoto School of Computer Science, Japan)
Y.Ueno (Professor emeritus of Kyoto University, Japan)
A.P.Wang (Arizona State University, USA)
W.R.Wells (University of Nevada-Las Vegas, USA)
T.Yamakawa (Kyushu Institute of Technology, Japan)
Y.G.Zhang (Academia Sinica, China)

STEERING COMMITTEE

M.Asada (Osaka University, Japan)
H.Asama (RIKEN, Japan)
J.L.Casti (Santa Fe Institute, USA)
S.Fujimura (The University of Tokyo, Japan)
T.Fukuda (Nagoya University, Japan)
H.Kashiwagi (Kumamoto University, Japan)
J.J.Lee (KAIST, Korea)
M.Nakamura (Saga University, Japan)
T.S.Ray (ATR, Japan)
K.Shimohara (ATR, Japan)
M.Sugisaka (Oita University, Japan) (Chairman)
H.Tanaka (Tokyo Medical & Dental University, Japan)
K.Ueda (Kobe University, Japan)
S.Ueno (Kyoto School of Computer Science, Japan)
Y.G.Zhang (Academia Sinica, China)

PROGRAM COMMITTEE

K.Aihara (The University of Tokyo, Japan)
S.Arimoto (Ritsumeikan University, Japan)
J.L.Casti (Santa Fe Institute, USA)

T.Fujii (RIKEN, Japan)
S.Fujimura (The University of Tokyo, Japan) (Chairman)
T.Fukuda (Nagoya University, Japan)
M.Gen (Ashikaga Institute of Technology, Japan)
I.Harvey (University of Sussex, UK)
T.Hasegawa (Kyushu University, Japan)
Y.Hori (The University of Tokyo, Japan)
P.Husbands (University of Sussex, UK)
H.Inooka (Tohoku University, Japan)
Y.Kakazu (Hokkaido University, Japan)
J.H.Kim (KAIST, Korea)
S.Kitamura (Kobe University, Japan)
H.Kitano (Sony Computer Science Laboratory Inc., Japan)
H.Kobatake (Tokyo University of Agriculture & Technology, Japan)
K.Kumamaru (Kyushu Institute of Technology, Japan)
J.J.Lee (KAIST, Korea)
T.Nagata (Inst. of Sys. and Inf. Tech. / Kyushu, Japan)
M.Nakamura (Saga University, Japan)
Y.Nishikawa (Osaka Institute of Technology, Japan)
T.Omori (Tokyo University of Agriculture & Technology, Japan)
K.Okazaki (Fukui University, Japan)
R.Pfeifer (University of Zurich-Irchel, Switzerland)
T.S.Ray (ATR, Japan)
T.Sawaragi (Kyoto University, Japan)
T.Shibata (MITI, MEL, Japan)
K.Shimohara (ATR, Japan)
S.Shin (The University of Tokyo, Japan)
M.Sugisaka (Oita Unuversity, Japan)
H.Tamaki (Kobe University, Japan)
H.Tanaka (Tokyo Medical & Dental University, Japan)
N.Tosa (ATR, Japan)
K.Tsuchiya (Kyoto University, Japan)
K.Ueda (Kobe University, Japan)
S.Ueno (Kyoto School of Computer Science, Japan)
X.Wang (Oita Unuversity, Japan)
K.Watanabe (Saga University, Japan)
T.Yamakawa (Kyushu Institute of Technology, Japan)
X.Yao (The University of New South Wales, Australia)

LOCAL ARRANGEMENT COMMITTEE

T.Endo (Oita University, Japan)
T.Ezaki (Research and Development Center, Oita University, Japan)
T.Hano (Oita University, Japan)
H.Hikawa (Oita University, Japan)
H.Ikeuchi (Oita University, Japan)
K.Kinoshita (Oita University, Japan)
A.Kuriyakawa (Oita University, Japan)

E.Kusayanagi (Oita University, Japan)
T.Matsuo (Oita University, Japan)
Y.Morita (Oita University, Japan)
T.Nabeshima (Oita University, Japan)
T.Nakano (Oita University, Japan) (Chairman)
K.Ohta (Oita-Aist Joint Research Center, Japan)
M.Sugisaka (Oita University, Japan)
A.Tominaga (Oita University, Japan)
K.Yoshida (Oita University, Japan)

TOPICS

Hardware Oriented Topics are welcome
in the fields given by

Artificial Brain
Artificial Intelligence
Artificial Life
Artificial Livings
Artificial Mind
Brain Science
Chaos
Cognitive Science
Complexities
Computer Graphics
Evolutionary Computations
Fuzzy Control
Genetic Algorithms
Human Care Robotics
Innovative Computations
Intelligent Control and Modeling
Micro Machine
Micro-Robot World Cup Soccer Tournament
Neural Networks
Neurocomputers
Neuro Computing Technologies
and Its Application for Hardware
Robotics
Robust Virtual Engineering
Virtual Reality
Related Fields

COPYRIGHT

MATHEMATICS AND COMPUTATION (North-Holland).

All correspondence relating to the symposium should be addressed to:

Prof. Masanori Sugisaka
Chairman of International Symposium on Artificial Life and Robotics (AROB)
AROB Secretariat
Dept. of Electrical and Electronic Engineering
Oita University
700 Dannoharu, Oita 870-11
JAPAN
TEL 001-81-975-54-7831
FAX 001-81-975-54-7841
E-MAIL msugi@cc.oita-u.ac.jp

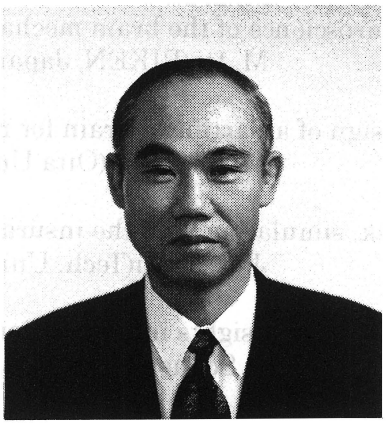
WWW Home Page http://AROB.cc.oita-u.ac.jp/

AROB 3rd '98 is supported financially by the following companies

All Nippon Airways Co., Ltd.
Japan Air System
Mitsubishi Heavy Industries, Ltd.
NEC Corporation
Nishi Nippon Electric Wire Cable Co., Ltd.
Oita Gas Co., Ltd.
Sanwa Shurui Co., Ltd.
Shintsurukai Kosan Co., Ltd.
Sony Oita Corporation
Sugahara Industry Co., Ltd.
Yanai Denki Kogyo Co, Ltd.
Yasukawa Electric Co., Ltd.
Yatsushika Sake-Brewing Co., Ltd.

PREFACE

Masanori Sugisaka
Chairman of AROB
(Professor, Oita University)



It is our great pleasure to invite you all to The Third International Symposium on Artificial Life and Robotics(AROB 3rd '98), organized by Oita University under the sponsorship of Ministry of Education, Science, Sports, and Culture(Monbusho), Japanese Government and co-sponsored by Santa Fe Institute(SFI), USA, SICE, RSJ, and IEEJ, Japan. This symposium invites you all to discuss development of new technologies concerning Artificial Life and Robotics based on simulation and hardware in 21st century. It is also our great honor to welcome several famous scientists and engineers as members of the International Organizing, Steering, Program Committees from this year.

This symposium is also financially supported by not only Monbusho but also other private companies. On behalf of the International Steering Committee we would like to express our gratitude to Honorary President Mr. Hiramastu(Governer, Oita Prefecture), members of the International Organizing Committee, the International Program Committee, the Local Arrangement Committee, Session Chairmans and Co-Chairmans and all the presenters. We acknowledge also the strong support of Monbusho and the sponsoring organizations in promoting the event. Finally we thank all who have contributed towards the sucess of this symposium.

We hope that AROB 3rd '98 will become a celebration to the establishment of our international joint research institute on artificial life, complexity, and robotics for twenty first century by the support of Monbusho's program of center of excellence. we hope that you will obtain fruitful results by exchanging ideas through discussions during the symposium and also will enjoy your stay in Beppu, Oita.

Masanori Sugisaka
M. Sugisaka

January 12, 1998

TECHNICAL PAPER INDEX

Special lecture(Invited Talk)

S-B1:Bio-Informatics(Invited Session)

S-B2:Evolutionary Computations and Applications (Invited Session)

S-C1:Robot Motions(Invited Session)

S-C2:Design of Emergence(Invited Session)

S-D1:Locomotion Robots(Invited Session)

S-D2:Why is Intelligent Control?(Invited Session)

S-E1:Robotics I(General Session)

S-E2:Evolutionary Computations(General Session)

S-A1:Hardware Evolution(Invited Session)

S-A2:Brain Computing and Brainware(Invited Session)

S-A3:Neuro Computing Technology(General Session)

S-B3:Agricultural Robot(Invited Session)

S-B4:Artificial Brain(Invited Session)

S-B5:Genetic Algorithms and Genetic Programming(General Session)

S-C3:Modeling and Complexity I(Invited Session)

S-C4:Sensor and Vision(General Session)

S-C5:Modeling and Complexity II (Invited Session)

S-C6:Neural Network(General Session)

S-D3:Intelligent Control and Modeling (General Session)

S-D4:Robotics II(General Session)

S-D5:Micro Machine and Micro Soccer Robot (General Session)

S-E3:Artificial Life, Brain, and Organs(General Session)

S-E4:Genetic Algorithms and Neural Network(General Session)

S-E5:Artificial Mind, Chaos, and Complexties(General Session)

S-A4:Complex Adaptive System I(Special Session)

S-A5:Robot and Mechatronics Control via Softcomputing (Invited Session)

S-B6:Complex Adaptive System II (Special Session)

S-B7:Control in Robotics(Invited Session)

S-C7:Artistic Communication for Artificial Life and Robotics (Invited Session)

S-C8:Communication(General Session)

S-C9:Vision Application(Invited Session)

S-D6:Life and Complexity(Invited Session)

S-D7:Robotics(Invited Session)

S-E6:Fuzzy Control and Virtual Reality (General Session)

S-E7:Evolutionary Computation and Neural Network (Invited Session)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Neuroscience of the brain mechanisms of mind and behavior

Masao Ito
Brain Science Institute (BSI)
The Institute of Physical and Chemical Research (RIKEN), Japan
+81-48-462-1111, E-mail: masao @ postman.riken.go.jp

The manuscript of this invited talk will be distributed at the symposium.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Design of an artificial brain for robots

Masanori Sugisaka
Department of Electrical and Electronic Engineering
Oita University
Oita, 870-11 Japan
e-mail msugi@cc.oita-u.ac.jp
tel 001-81-975-54-7831, fax 001-81-975-54-7841

Abstract This paper proposes a new information processing and control system, which is called artificial brain(ABrain), for robotics using neurocomputers and a Von-Neumann type microcomputer, and interfaces operating these computers. We introduce three robotic systems with ABrains developed in our laboratory recently. The one is an ABrain in order to recognize the moving object in a robot system for recognition and tracking. The other are an ABrain for controlling the duty ratio in the robot system for recognition and tracking and an ABrain for controlling steering angle in an intelligent mobile vehicle. Based on our results obtained we present a realization of general type of ABrain in order to recognize a moving pattern and track it simultaneously in this paper.
Keywords ABrain, Pattern recognition, Tracking, Neurocomputer

1. Introduction

We have already developed a primitive artificial brain[1] (denoted ABrain), which was constructed by both one neurocomputer and a Von-Neumann type NEC PC-9801 microcomputer, for tracking a moving object, utilizing our previous result[2]. The system consists of one CCD video-camera, two DC motors with encorders, one on the horizontal axis and one the vertical axis, a pulse-width modulation(PWM) driving unit, a 32-bit microcomputer NEC PC-9801, the RICOH neurocomputer RN-2000[3,4], and their interfaces. The control was produced by the RN-2000 neurocomputer, which is able to learn various control laws instantly, in order to track a moving object within a predetermined range of errors.

We have also developed the fast pattern recognition system based on the moment invariants by using the primitive ABrain[5,6]. The fast pattern recognition system consists of a CCD video camera, an image processing system named FDM, a monitor, two stand lights, a NEC PC-9801 microcomputer and a RICOH neurocomputer RN-2000 which is considered as the primitive ABra in. The experimental studies in order to recognize five dynamic patterns of Japanese chestnuts were performed. From the studies high speed of both learning and recognition has been achieved compared with the former pattern recognition system based on the software of artificial neural networks developed by us[7]. In addition, we have also developed an intelligent navigation technique of a mobile vehicle[8-10] for smooth running using the primitive ABrain[11].

Based on the systems developed in our laboratory stated above, the configulation of a new ABrain is proposed in this paper. First, we give the general idea of the ABrain[12-15]. Second, we consider approaches how to make the ABrain. Third, we give the preliminary results obtained from our recent studies in order to design the general type of ABrain. Then, we will propose a general structure of the ABrain in order to perform various behavioral functions such as, learning, recognition, decision etc. based on adap-

tion, evolution, self-organization, and emergence for robotics.

2. What is the ABrain?

In this section, we show what the ABrain is. Before we explain it, we state briefly the reason why the ABrain is necessary. If a new ABrain like human brain is developed and is imbedded into a robot or a machine, it will become soft and friendly for human being. If the emotion emerges from the ABrain of the robot, the robot is able to behave like human being. In other word, it means that we give an artificial life(denoted ALife) into the robot by installing the ABrain. This is a dream which many engineers want to achieve for a long time.

Let us forcus our attention on the functions of the ABrain. What is the ABrain? Figure 1 shows the flow of information and learning in the ABrain. The stimulus from a sensor is transfered into the neural network system for recognition, which is called the ABrain for recognition(denoted R-ABrain), in the brain nervous system. The results processed by the R-ABrain are transfered into the neural neural network system for thinking and based on the results processed by the neural network system a decision will be made for what kind of behavior should be taken. This neural network system is called the ABrain for thinking and decision(denoted T,D-ABrain). The output processed by T,D-ABrain is transfered into the neural network system for behavior or action, which is called the ABrain for behavior(denoted B-ABrain) in the brain nervous system. Finally, the stimulus processed by the B-ABrain is transfered into the actuators and drivers in order for the systems or the robots to achieve an object predetermined.

We illustrate an example of the functions of the ABrain. When human being see a tiger, human being runs away in order to avoid danger. On the other hand, when human being see a cake, human being tries to eat it. This example suggests both what the ABrain is and what we should make.

The learning process is included in the ABrain(R-ABrain, T,D-ABrain, B-ABrain) as illustrated in Fig.1. After the learning process has finished, the results learned are memorized as a posteriori knowledge. The results will be partially memorized into gene as a priori knowledge in the evolution process. There exist clearly the differences between adult's brain and baby's brain and, however, we forcus our attention on buiding the ABrain like adult's brain.

The ABrain for a robot consists of three partial ABrains stated above; (1) R-ABrain (2) T,D-ABrain (3) B-ABrain. Conseuently, the ABrain in the robot is able to recognize the objects, to think how it behave and to make a decision, and to take an action. The structures or network systems of three ABrains are illustrated briefly in Figs. 2, 3, and 4, respectively.

3. How to make the ABrain

It is one way for us to build the ABrain for a robot by utilizing the functions of human brain. The human brain whose functions are not still clarified is regarded as a complex adaptive system. The human brain and nervous system in total body consists of a huge amount of neural network systems which is aggregation of neurons. The ultrastructure of the neuron which is an unit for information processing is shown in Fig. 5[16–18].

A new research field for building the ABrains for robots has been born recently by mixing brain science and engineerings such as VLSI hardware and computer software. Realization of an ABrain by hardware architecture with self-organizing functions etc. stated above is a dream of human being in the twenty first century. Our efforts toward this object will yield a better life for human being. Before proposing the general structure of the ABrain for robot, we will show the structure of the primitive ABrain developed recently in our laboratory in the succeedings. The primitive ABrain was used for both controlling motors[1,11] and recognizing patters of an object[5,6]

4. ABrains developed

The ABrain(denoted ABrain-1) developed in our laboratory on Februrary 2, 1995[1] was installed into the recognition and tracking system

for moving objects[2]. This ABrain is able to learn how to control two DC motors for tracking a moving object in three-dimensional space by itself instantly. Thereafter we developed the ABrain (denoted ABrain-2) on February 28, 1995[5,6] to recognize five dynamic pattern of Japanese chestnuts. Recently, we have developed the ABrain(denoted ABrain-3) on February 17, 1997 for controlling the angle of steering wheels of a mobile vehicle(denoted MV). We will explain each ABrains briefly in the following.

4.1 ABrain-1[1]

The ABrain-1[1] consists of the 32-bit Von-Neum ann type microcomputer NEC PC-9801βA, the RICOH neurocomputer RN-2000(called Yamato), the interfaces, the software(called Nadeshiko) for both constructing a neural network and learning a control law by it, and the software for tracking a moving object.

ABrain-1 for the recognition and tracking system of moving objects is shown in Fig.6. Detailed descriptions of the system have been given previously[19–22]. The solid line in Fig.6 shows the configulation of the primitive ABrain. We succeeded in tracking both the desired values(sinusoidal, circular, and elliptical movements) of encoders on both the X(horizontal) and Y(vertical) axes and a light from an electric lamp moved by hand.

The structure of the neural network in the RN-2000 neurocomputer in the tracking experiments is shown in Fig.7. There are four neurons in the input layer, sixteen neurons in each of the first and second intermediate layers, and one neuron in the output layer. The inputs to the neural network are errors between the center of the CCD camera and the desired values of the encoders on the X and Y axes, or the point of highest intensity of the light from an electric lamp moved by hand, as shown in Fig.8. The teaching signals are the duty ratios of the pluse width modulation(PWM). There are two reasons why we use four neurons in the input layer. Please see the reference[1].

As the learning or taining data for the neural network shown in Fig.7 we used the propotional control data. Both the real training data and the corresponding data used for the neural network in the RN-2000 neurocomputer are shown in Table 1. In this table the input data to the neurons are the deviations(0r errors) of e_x and e_y, which are equal to 1,2,...,10, and the teaching data are the duty ratios of the DC motors on both the horizontal and the vertical axes, which are equal to 0.1,0.2,...,1.0. The input data are transformed into four bit digits by using a bit transformation technique. The teaching signals are normalized so that the maximum value 1.0 and minimum value 0 in Table 1 correspond to 127 and 0, respectively.

In the tracking experiment we used the neural network shown in Fig.7, which is constructed by Nadeshiko software. The configulation of the neural network using the neurocomputer is illustrated in Fig.9. The duty ratio for the two DC motors on the X and Y axes are produced sequentially from one neural network. In other words, first the error between the desired value of X(denoted $X_{desired}$ in Fig.9) and the value of the encoder of the DC motor on the X axis(denoted X motor in Fig.9) is processed by the neural network in order to obtain the duty ratio of the X motor. Then the same process is repeated in order to obtain the duty ratio of the Y motor.

By taking account of the hardware specifications, the sampling time employed in the experiments is 100 ms. The procedure described above[19–22] are repeated at each sampling time throughout the tracking experiments. We describe briefly how to detect the position of the point of highest intensity of light from a lamp being moved by hand, and then explain how to track it. The detection procedures are as follows.

1. Read 6-bit image data of the light (15×12= 180 pixels) using a CCD camera.
2. Write the image data into the memory of the NEC PC-9801βA.
3. Detect the position of the point of highest intensity of the moving light.

The tracking procedures are as follows.

4. Calculate the duty ratio for the X and Y mo-

tors sequentially using the trained neural network in the RN-2000 neurocomputer.
5. Move the system in order to make the center of the CCD video-camera coinside with the position of the point of the hightest intensity of the light using the duty ratios calculated above.

We show one of the results of tracking a light being moved by hand in Fig.10, where the distance between the moving light and the CCD video-camera was approximately 60cm. In this figure the loci from 8s to 25s are illustrated. The locus of the X and Y coordinates is shown by the solid line, where the dotted line is the locus of the point of highest intensity. The result obtained from tracking a light being moved by hand is satisfactory for the following reasons.

The neural network structure constructed in the RN-2000 neurocomputer is simple.
The control law used to train the neural network is a simple proportional law.

4.2 ABrain-2[5,6]

The ABrain-2 was developed for the pattern recognition system shown in Fig. 11, which consists of a CCD video camera, an image processing system named FDM, a monitor, two stand lights, an NEC PC-9801 microcomputer and a RICOH RN-2000 neurocomputer where these two different types of computers constitutes an artificial brain. Experimental studies to recognize five dynamic patterns of Japanese chestnuts were performed. From the studies, a high speed of both learning and recognition has been achieved.

The simulation of the proposed system with the ABrain-2 is carried out in the following steps. The first step uses the image scene, which is compressed into 256(16×16) pixels by preprocessing from the original image scene to the input pattern. In the second step, seven moment invariants are computed for each pattern. In the third step, three moment invariants instead of seven moment invariants are used as the inputs of the artificial neural network constructed in the neurocomputer for all samples because of the hardware specifications of the neurocomputer. In the fourth step the training begins, using the backpropergation method by the neurocomputer. By ending the artificial neural network training in the neurocomputer, the output of the network is used to recognize the dynamic patterns.

We will show the moment invariants and the neural network for recognition. The moment invariants Q_1, Q_2, Q_3 are invariant under translation and rotation[23]

$$M_{pq} = \sum_{i=1}^{16}\sum_{j=1}^{16} i^p j^q P_{ij} \tag{1}$$

$$\mu_{pq} = \sum_{i=1}^{16}\sum_{j=1}^{16} (i-\hat{i})^p (j-\hat{j})^q P_{ij} \tag{2}$$

$$Q_1 = \mu_{20} + \mu_{02} \tag{3}$$

$$Q_2 = (\mu_{20} - \mu_{02}) + 4\mu_{11}^2 \tag{4}$$

$$Q_3 = (\mu_{30} - \mu_{12}) + (3\mu_{21} - \mu_{03})^2 \tag{5}$$

where P_{ij} is a digital 0 or 1 function, i and j have integer values and $\hat{i} = M_{10}/M_{00}$ and $\hat{j} = M_{01}/M_{00}$.

The moment invariant above is used as the inputs for the neural network in the RN-2000 neurocomputer. In our pattern recognition system only the three moment invariants Q_1, Q_2, and Q_3 are used instead of seven moment invariants because of the hardware specification in the neurocomputer. The artificial neural network system constructed in the neurocomputer is shown in Fig.12. It consists of four layers: the input layer with 15 neurons, the first intermediate layer with 16 neurons, the second intermediate layer with 16 neurons, and the output layer with 5 neurons, which is equal to the number of patterns to be recognized.

The experimental studies on the learning of five dynamic patterns of Japanese chestnuts were performed with the image processing system shown in Figs. 11 and 12. Figure 13 shows the patterns that were used in the learning process and

gives examples of the five different patterns. The results obtained confirm the effectiveness of the proposed system and show improvement of both the learning and recognition speeds, which is the main point of the ABrain-2[5,6]

ABrain-3

The ABrain-3 has been developed recently for controlling steering angle of the front wheels of the MV called neuro mobile vehicle (denoted NM V)[8-11] shown in Fig.14. The configulation of the electrical part of the NMV is shown in Fig.15 where the part of the solid line is the primitive ABrain which is belong to the same category of the B-ABrain as the ABrain-1. The neural network used in the ABrain-3 consists of four layers; the input layer with 8 neurons, the first intermediate layer with 16 neurons, the second intermediate layer with 16 neurons, and the output layers with 1 neuron. The input to the neural network is the error of angle between the bottom of the center line of the CCD camera and the center of the gravity of the object detected. The output from the neural network is the value which corresponds to the number of the pulse of the stepping motor to control the steering angle[11]. The experimental results are satisfactory.

In this section, we explained two defferent types of ABrains. The one is the R-ABrain and the other is the B-ABrain. We did not show the structure of the T,D-ABrain because at this stage the functions of it are performed using the information obtained from the experiments by skillful engineers. However, it is important to develop the structure of the neural network in the T,D-ABrain in order to design the general ABrain for robotics. This problem is now under consideration.

5. General structure of ABrain

In this section, we propose the general structure of the ABrain based on the results obtained concerning the primitive R-ABrain, T,D-ABrain, and B-ABrain. The configulation of a general ABrain for robots is shown in Fig.16[24]. This configulation has a general structure for the ABrain used for robots and intelligent machines or computers. The ABrain consists of the following artificial organs:

1. various sensors which receive different information,
2. a self-organizing circuit which organizes the structure of the artificial neural networks and numbers of both neurons and neurocomputers in the ABrain,
3. several parallel neurocomputers and a Von-Neumann type computer,
4. a circuit for generating internal peformance criterion in order to optimize the total neural network systems including the parallel neurocomputers and a Von-Neumann type computer,
5. and decision circuit which determines control actions to drivers or actuators.

The general hardware structure of the ABrain shown in Fig.16 is able to perform the functions of information processing stated in the preceedings. In other words, each ABrain such as R-ABrain, T,D-ABrain, and B-ABrain is embedded into this general ABrain hardware.

The R-ABrain consists of a neural network of the neurocomputer in which the inputs to the neurons in the input layer are feautures(e.g., moment invariants) of the object. The T,D-ABrain consists of a neural network of the neurocomputer in which the inputs to the neurons in the input layer are genetic information and/or knowledge obtained from the experiment and learning. The B-ABrain consists of a neural network of the neurocomputer in which the inputs to the neurons in the input layer are the errors or differences between the present states and the desired states. The information received from sensors is processed squentially or in parallel in the general ABrain which consists of the three ABrains . The problem how to design the T,D-ABrain have to be solved. The design of both the general ABrain hardware and operating software is now under investigation.

6. Conclusions

In this paper, we considered how to design the

general sturucture of the ABrain which consists of three type of primitive ABrains(R-ABrain, T,D-ABrain, B-ABrain). Also, we explained both the R-ABrain[5,6] and the B-ABrain[1,11] which were developed in our laboratory recently. Since we have obtained the guidline for designing the general ABrain based on the results developed by our group, we will apply this guidline for constructing ABrains for intelligent machines and robots in various engineering fields related to human being.

References

[1] Sugisaka M (1997) Neurocomputer control in an artificial brain for tracking moving objects. Artif Life Robotics **1**(1):47-51

[2] Sugisaka M , Tonoya N (1996) Neuro control for recognition and tracking sytem of moving object. Systems Science **22**(2):59-67

[3] Eguchi H, Furuta T, Horiguchi H, Oteki S (1991) Neural network hardware with learning function utilizing pulse-density modulation. Trans Inst Electron Inf Commun Eng J74-C-II:369-376

[4] Oteki S, Hashimoto A, Furuta T, Motomura S, Watanabe T, Stork DG, Eguchi H (1993) A digital neural network VLSI with on-chip learning using stochastic pulse encording. Proceeding of the 1993 International Joint Conference on Neural Networks, Nagoya, Japan, pp 3039-3045

[5] Sugisaka M (1996) Pattern recognition using neurocomputer. In: Sugisaka M(ed) Proceedings of the International Symposium on Artificial Life and Robotics(AROB 1st), Beppu, Oita, Japan, Feb. 18-20, 1996, pp. 248-251, 1997

[6] Sugisaka M (1997) Fast pattern recognition using a neurocpmputer. Artif Life Robotics(to appear)

[7] Sugisaka M, Teshnelab M (1993) Fast pattern recognition by using moment invariants computation via artificial neural networks. Control Theory and Advanced Technology 9:877-886

[8] Sugisaka M, Hara M, Tonoya N, Ike Y, Kim SW, Lee JJ (1994) Running test of a mobile vehicle. Proceedings of the Asian Control Conference, Tokyo, Japan, July 27-30, 1994, pp439-442

[9] Sugisaka M (1997) Development of a mobile vehicle (in Japanese). Kikainokenkyu(Science of Machine) **49**(6):5-15

[10] Sugisaka M, Wang X, Lee JJ (1997) Control strategy for an intelligent mobile vehicle. Artif Life Robotics(to appear)

[11] Sugisaka M, Wang (1998) Artificial brain for a mobile vehicle. International Symposium on Artificial Life and Robotics(AROB III '98), Beppu, Oita, January 19-21, 1998(to appear)

[12] Sugisaka M (1997) Artificial brain in robotics (Invited talk). Proceeding of Joint Conference on Control, Automation and Robotics, Seoul, Korea, March 14-15, 1997, pp 5-10, 1997

[13] Sugisaka M (1997) Artificial brain in robotics (Invited talk). Seminar/Workshop on Artificial Life and Evolutionary Robots, Beijing, China, September 16-18, 1997, pp 47-52, 1997

[14] Sugisaka M (1997) Neurocomputer control in artificial brain for tracking of moving objects (Invited talk). Seminar/Workshop on Artificial Life and Evolutionary Robots, Beijing, China, September 16-18, 1997, pp 53-69, 1997

[15] Sugisaka M (1996) Design of artificial brain. Lecture Note, The University of Tokyo, 21 March, 1996

[16] Levitan IB, Kaczmarek LK (1997) The neuron-cell and molecular biology. Oxford University Press, Oxford, pp 3-32

[17] Nicholls JG, Martin AR, Wallace BG (1992) From neuron to brain-a cellular and molecular approach to the function of the nervous system. Sinauer Associates, Massachusetts

[18] Ito M (1995) Nou no sekkeizu(in Japanese). Tyuokouronsya, Tokyo

[19] Sugisaka M (1992) Recognition and tracking system of moving objects. Patent Tokkaihei 5-2333062 (in Japanese)

[20] Sugisaka M, Kaita H, Hara M, Nakao K (1992) Recognition and tracking system of moving objects based on artificial neural network and PWM control. Proceedings of the IFToNM-jc International Symposium on Theory of Machines and Mechanics, Nagoya, Japan, vol.1, pp 104-107

[21] Sugisaka M (1992) Recognition and tracking system of moving objects based on artificial

neural network. Proceedings of the 1992 Korean Automatic Control Conference, Seoul, Korea, pp 573-574
[22] Sugisaka M, Hara M, Tonoya N (1994) Neuro-fuzzy control for recognition and tracking system of moving objects. Proceedings of the 3rd International Workshop on Advanced Motion Control, Berkeley, USA, pp 1091-1098
[23] Widrow B, Winter RG, Baxter RA (1988) Layered neural nets for pattern recognition. IEEE Trans Acousic Speech and signal processing ASSP-**36**(7):1109-1118
[24] Sugisaka M (1997) Neural networks for control in an artificial brain of recognition and tracking system. Proceedings of International Symposium on Artificial Life and Robotics(AROB '97), Beppu, Oita, Japan, 1997, pp 99-102.

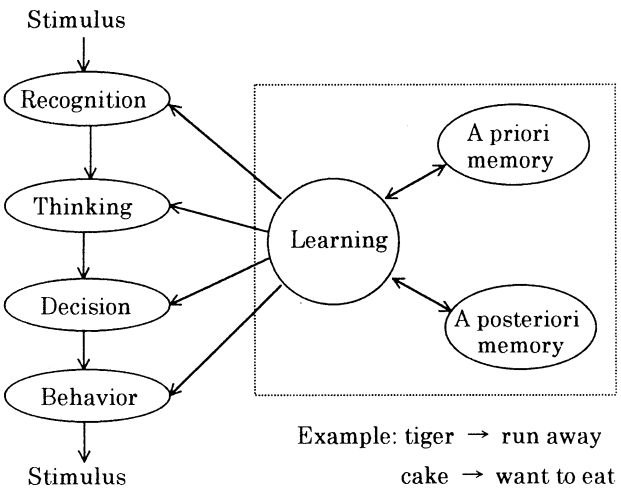


Fig. 1 The flowchart of information processing in the ABrain

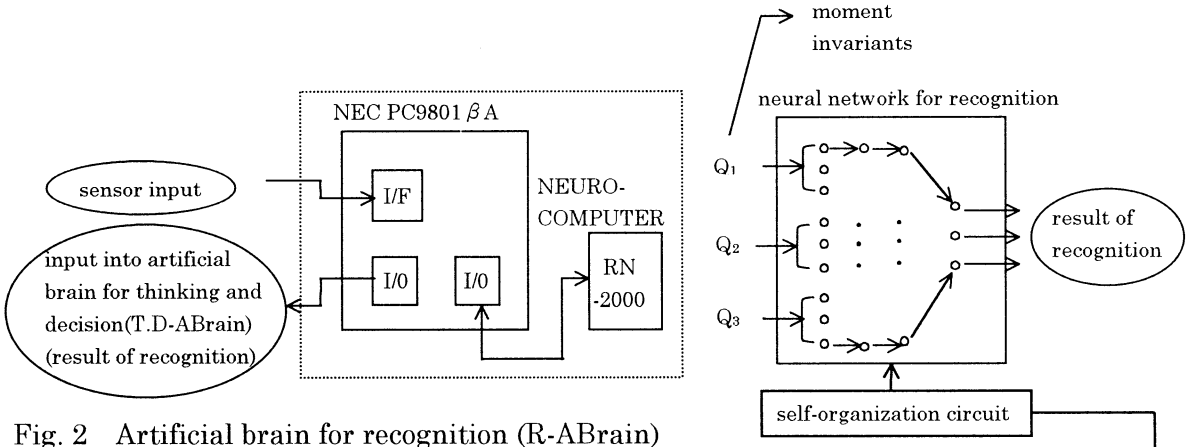


Fig. 2 Artificial brain for recognition (R-ABrain)

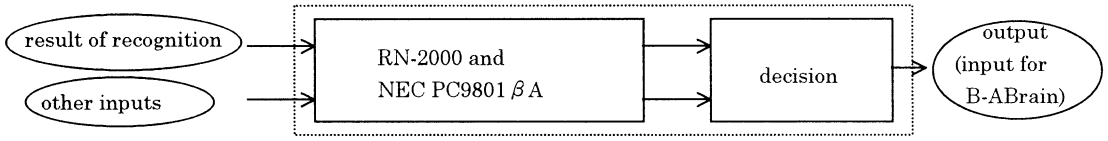


Fig. 3 Artificial brain for thinking and decision(T,D-ABrain) (self-learning and self-thinking)

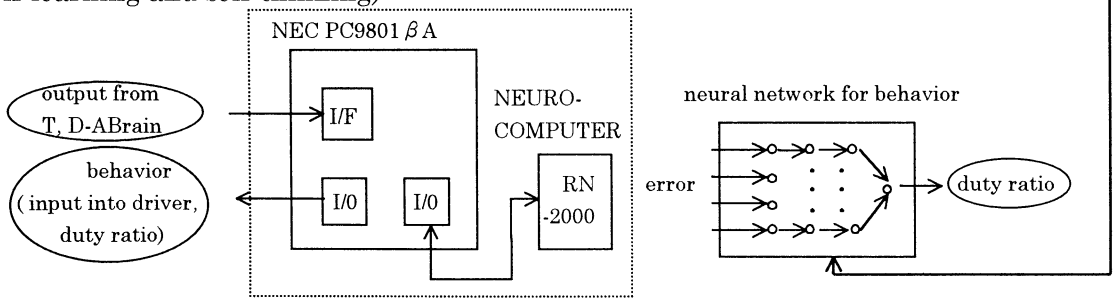


Fig. 4 Artificial brain for behavior(B-ABrain)

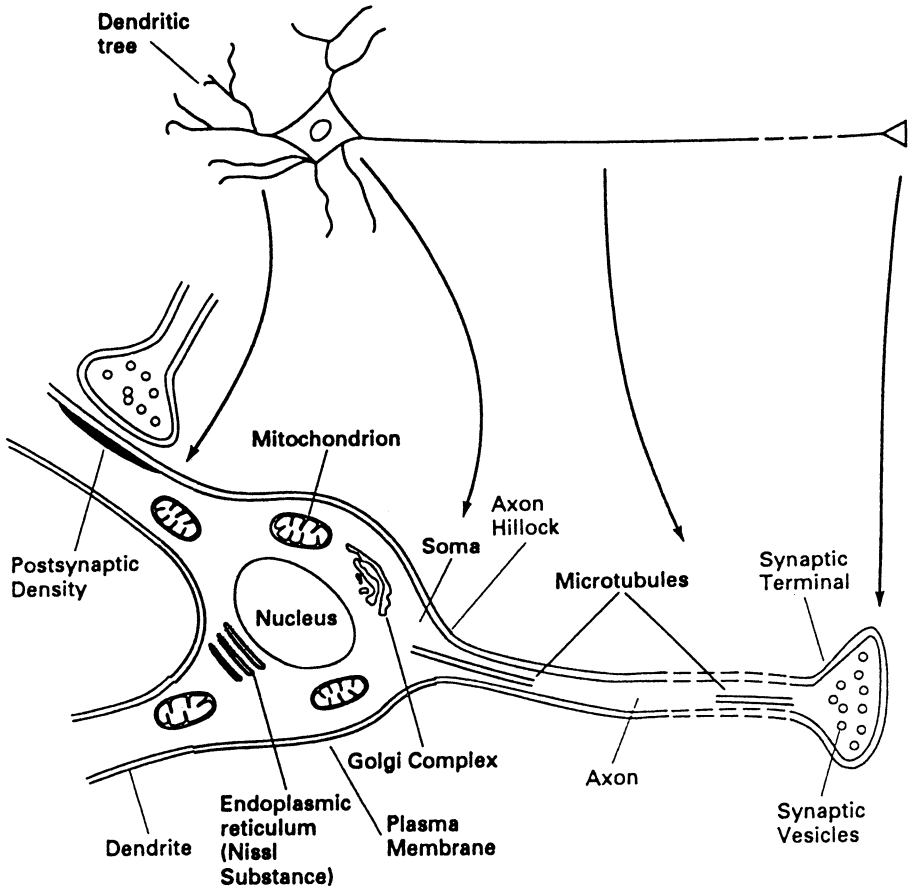


Fig. 5 Ultrastructure of the neuron

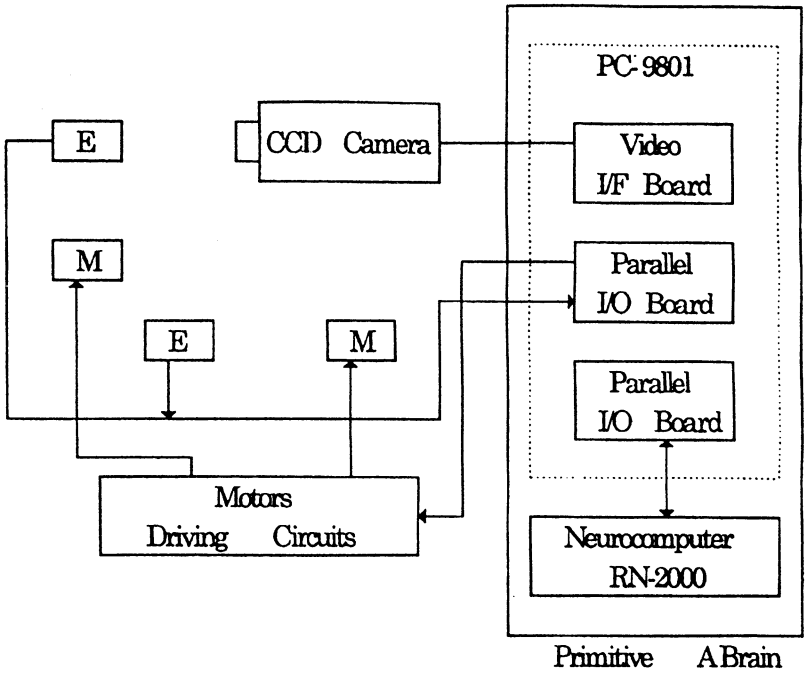


Fig. 6 Artificial brain (ABrain) for recognition and tracking. *M*, motor; *E*, encoder

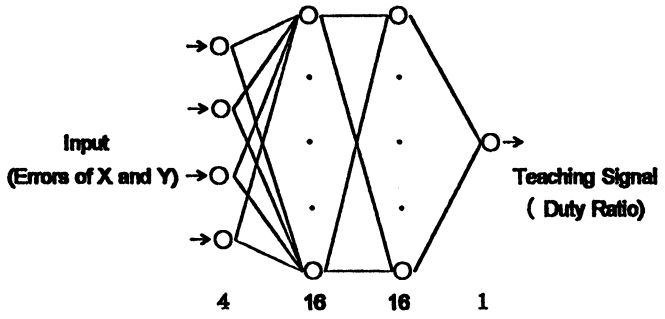


Fig. 7 Structure of the neural network in RN-2000

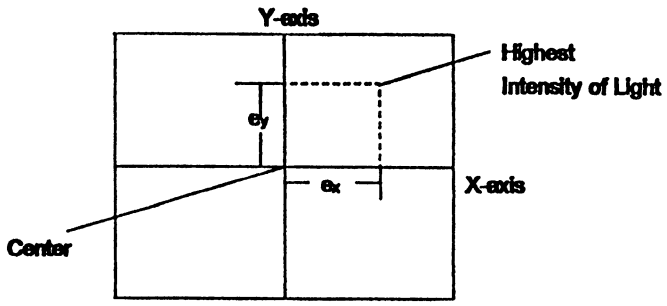


Fig. 8 Errors or deviations

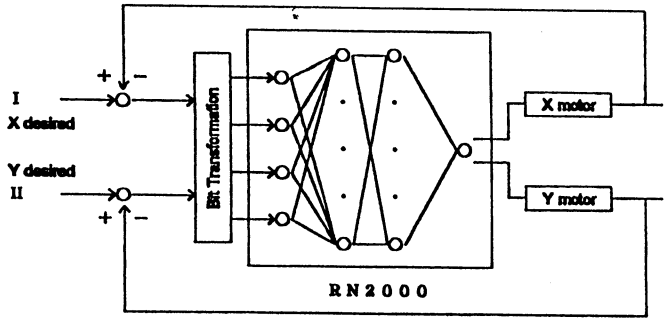


Fig. 9 Configuration of the neurocontroller (I and II show the order of processing)

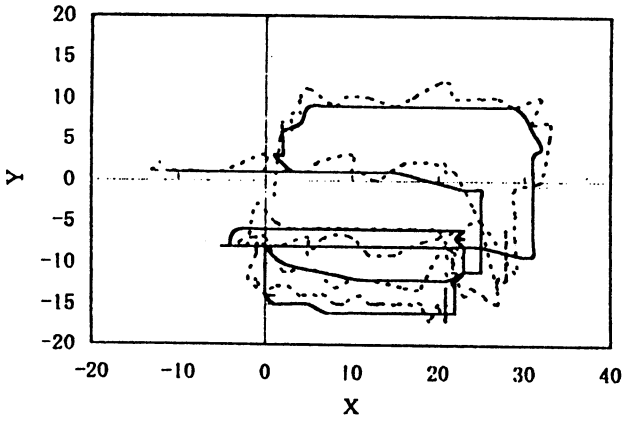


Fig. 10 Results of tracking a moving light (8-25s). Loci of the light and X and Y coordinates.

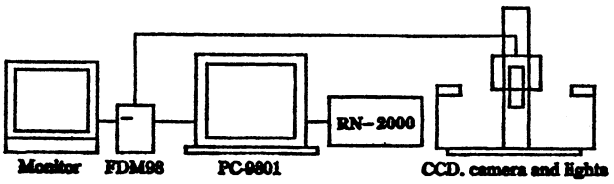


Fig. 11 New pattern recognition system using RICOH RN-2000 neurocomputer

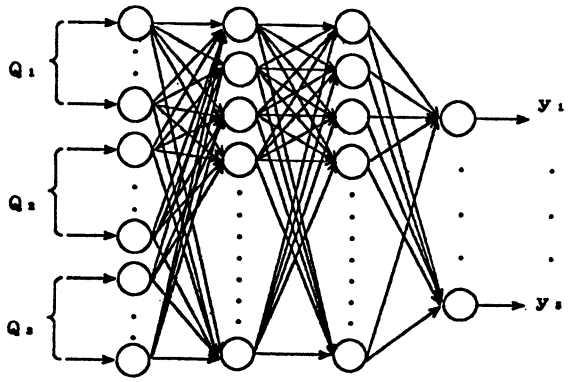


Fig. 12 Neural network constructed in RN-2000 neurocomputer

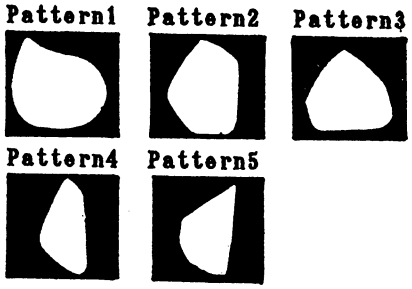


Fig. 13 Examples for testing recognition of five various patterns

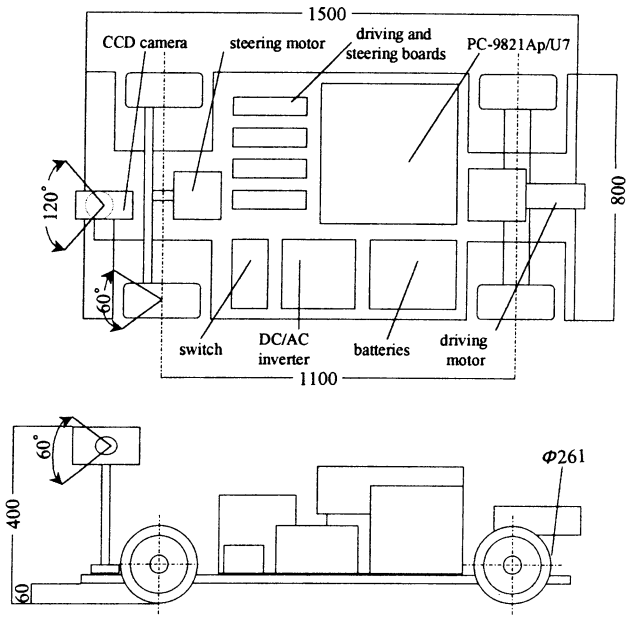


Fig. 14 The mechanical configuration of the mobile Vehical

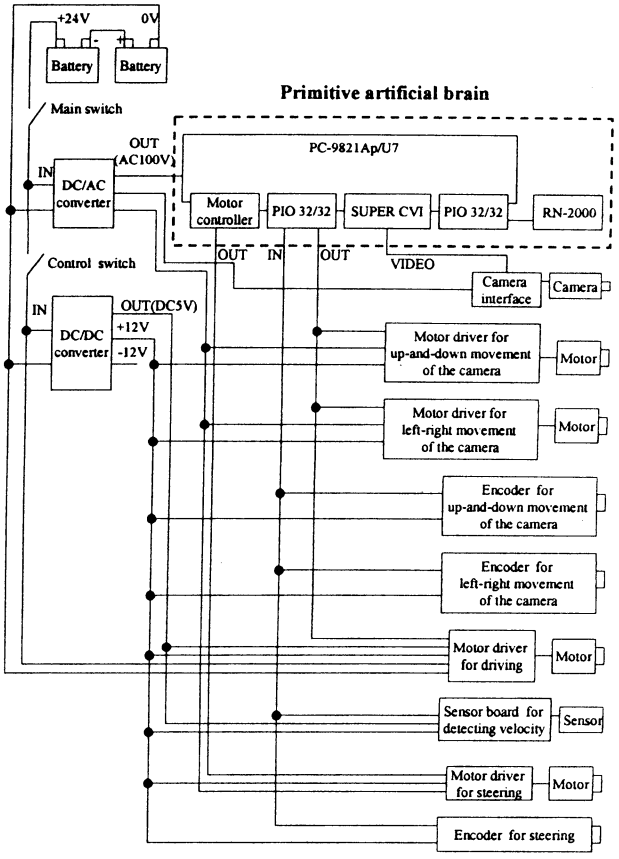


Fig. 15 Configuration of the mobile vehicle

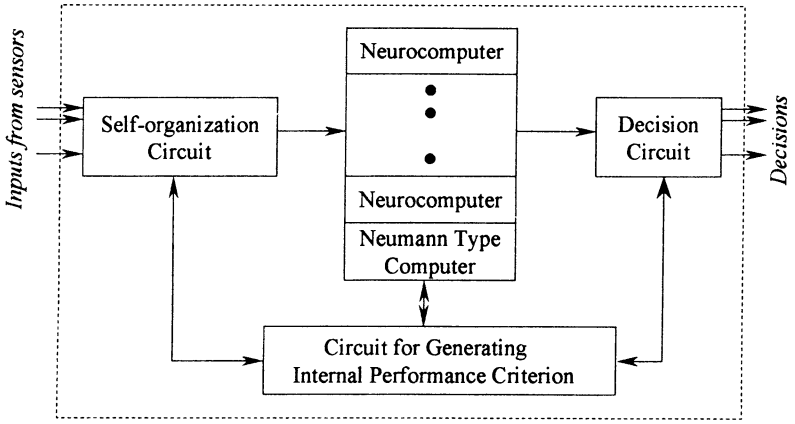


Fig. 16 The configuration of the artificial brain

Table 1. Data for training

Errors (deviations)		Teaching signals		Learned results
0	0000	0	0	0
1	0001	10	13	15
2	0010	20	25	27
3	0011	30	38	31
4	0100	40	51	53
5	0101	50	64	63
6	0110	60	76	72
7	0111	70	89	85
8	1000	80	102	108
9	1001	90	114	112
10	1010	100	127	120

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Risk, Simulation, and the Insurance Industry

John L. CASTI
Institute for Econometrics, OR & System Theory
Technical University of Vienna
A-1040 Vienna, Austria

AND

Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501, USA

Statistics, Laws and Nature

In late August of 1992, Hurricane Andrew roared through the east coast of the United States, causing nearly twenty billion dollars worth of damage. For insurance companies stuck with this gargantuan bill, Andrew was but the tip of the iceberg in what seems to be an ever-increasing exposure to greater risks from all types of cataclysms---floods, volcanic eruptions, earthquakes, tidal waves, hurricanes and just about any other whim thrown up by Mother Nature. So it's natural to wonder whether the newly-emerging science of complexity can be of any help in enabling us to understand more clearly the likelihood of these kinds of disasters.

It's certainly no secret to insurance underwriters that the distilled essence of risk assessment and management lies in identifying the relative frequencies of different levels of disaster. Basically, the guiding principle of risk management is if you can plan for it, you can adjust the rates for your policies to accommodate it. So the heart of the risk question comes down to the problem of prediction.

The approaches scientists use for predicting the doings of nature come in two quite different flavors. Let me term them the *phenomenological* and the *foundational.* Phenomenological methods are based upon identifying statistical regularities in natural processes. For example, when you look at the climatic patterns depicted in the "Drought Clock" shown in Figure 1, it's hard to believe that we don't live in interesting times. The hot/cold and wet/dry periods clearly seem to repeat themselves in a very cyclical manner-at least when your unit of time is centuries rather than weeks or months. This illustrates the statistical way of forecasting climatic change: gather lots of historical data and look for regularities in the occurrence of dramatic events.

Foundational methods, on the other hand, focus upon the basic physical processes giving rise to things like earthquakes, floods and volcanic eruptions. Using the laws of physics, chemistry, and geology, we try to build mathematical and computer models of, say, the atmosphere that we can use to predict what nature will be doing next. The daily weather forecasts we see on television and in the newspapers are generated in exactly this manner.

Both the phenomenological and the foundational approaches to prediction have been dramatically influenced over the past few years by developments in what has come to be termed, "the science of complexity." Put compactly, complexity theorists have discovered that everyday common sense just doesn't always work when it comes to predicting and/or explaining the behavior of many types of systems. And just about every natural process ranging from the processes of weather formation to the human heartbeat fall into this category, as do a lot of social and behavioral systems like speculative markets and the outbreak of warfare. So here I'll give a brief glimpse of what these developments have to say about issues affecting risk assessment.

Randomness and Predictability

Suppose we're given a sequence of observations $x_0, x_1, x_2 \ldots, x_T$ of some system taken at times $t=0, 1, 2, \ldots, T$. We might think of these numbers as

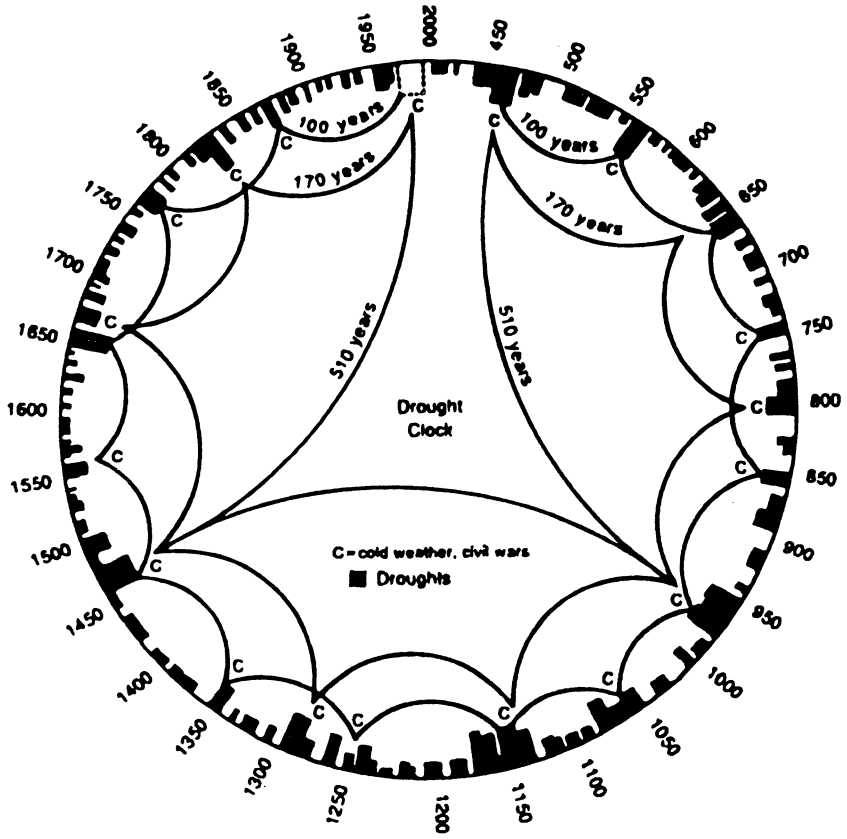


Figure 1. The Drought Clock

measurements of, say, the water levels in a reservoir or the daily closing prices of Exxon stock. If we're in the prediction business, certainly one of the first things we'd like to know is whether or not these observations are completely random. If so, then there is no pattern to be found and no way to predict the next observations from the past; if not, then we at least have a fighting chance of ferreting out a rule that will help us to do better than just guess the next observation, x_{T+1}.

One of the principal lessons taught by the modern theory of chaotic processes is that a sequence of numbers can **appear** totally random, yet may in fact be the result of following a fixed, completely deterministic rule. For example, if the numbers above happened to have been generated by the simple "logistic rule": the next number equals four times the preceding number minus four times the *square* of the preceding number, or for the mathematically inclined,

$$x_{t+1}=4x_t(1-x_t),\ t=0,1,\ldots,T-1$$

In this case, none of the classical statistical methods would be able to discern the fact that the set of numbers arose from a completely deterministic rule; they would all render the verdict that the sequence is purely random. But methods of modern nonlinear dynamics say differently. Here's how.

First, we plot the number x_{t-1}against x_t. If the sequence is truly random, then x_t. is completely independent of any of its predecessors. Consequently, we would expect to obtain a scatter diagram looking something like that of Figure 2(a). But when we make such a plot for the set of numbers generated by the logistic rule, we obtain what's called the *logistic parabola* shown in Figure

This is a one-dimensional curve living in a two-dimensional space, suggesting there is some very definite relationship linking any two adjacent numbers in the sequence; the numbers are not independent, hence not random. This graphical procedure is a kind of poor man's version of a much more sophisticated test called the *correlation dimension*, which provides us with a powerful analytical tool for distinguishing the random from what we might term the "pseudorandom."

To see how this general idea works in a specific situation, let's turn to a closely-related procedure developed originally by the hydrologist H.E.. Hurst to study the movement of water reservoir levels for flood control.

Nature's Memory

Consider the water reservoir depicted in Figure 3, and suppose the number x_t above represent the inflow to the reservoir in year t. Denote the standard deviation of the inflow over this period by S_T, and let R_T represent the *range* of inflows, i.e.. R_T is simply the difference between the largest and smallest numbers in the sequence. Hurst discovered a remarkable fact about the dimensionless ratio R_T/S_T.

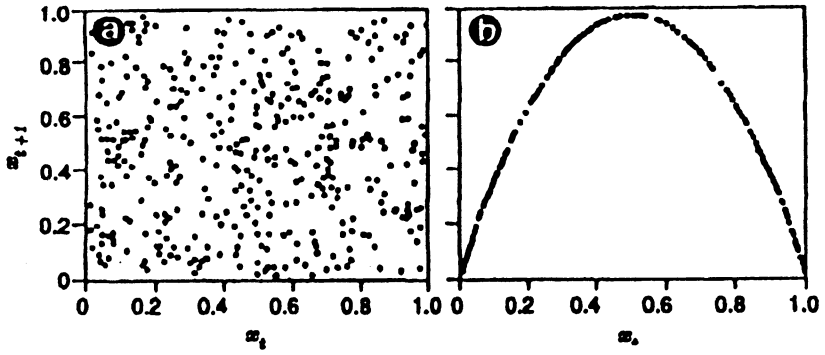


Figure 2. (a) Random scatter, (b) The logistic parabola

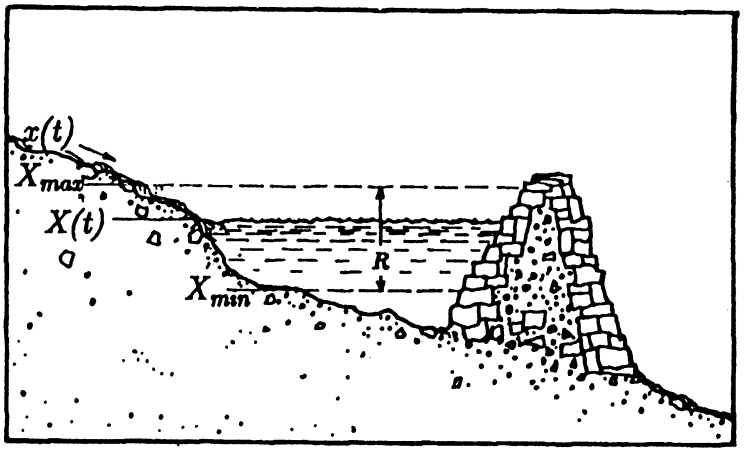


Figure 3. A water reservoir with inflow X_t and range R_T

Upon calculating the *R/S* ratio for a very large number of quite natural phenomena ranging from flood levels on the Nile to trends in global temperature variation. Hurst found that they all seemed to obey the empirical relation $R_T/S_T = T^H$, where H is a number now called the *Hurst* exponent. Furthermore, it can be shown that when the elements of the original data set are independent from one time period to the next, the value of H tends to 1/2 when the number of data points T is great enough. But the surprising conclusion of Hurst's efforts is that H differs greatly from 1/2 for most natural processes. So if you believe the data, then it seems that processes as diverse as sunspot fluctuations, river discharges and rainfall levels all have some kind of long-term "memory effect." More specifically, since most of these natural processes have values of H greater than 1/2, fluctuations above the mean tend to be followed by fluctuations that are also larger than average. This fact is of great interest and value in explaining why big hurricanes, big floods and big stock market crashes tend to occur in clusters.

Current work in complex, adaptive systems has turned up many other tools that are of use in predicting dramatic events. Let me mention just two that relate more to the foundational way of prediction than to the phenomenological: the theory of catastrophes, which enables us to characterize mathematically shifts of stability of exactly the sort that cause earthquakes, and the theory of self-similar phenomena, which leads to fractal descriptions of the scale of everything from forest fires to epidemics. Both of these ideas are explored formally in references [1] and more informally in [2, 3]. In conclusion, these examples show that complex system theory is destined to play an important role in enabling analysts to get a better handle on the problem of managing risk from natural disasters. Now let me look briefly at some of the work being done in multiagent simulation [4] to come to terms with the peculiarities of these systems.

Insurance World

We are currently in the process of constructing a computer model of the insurance industry, using the theory of complex, adaptive systems to understand how the interactions among insurers, customers and the environment lead to the emergence of prices, capacity and structuring in the industry. This study will begin with a simplified model of the insurance industry focusing on property catastrophe reinsurance. This model will contain all the generic elements of insurance needed to later extend the approach to more specific questions and types of insurance.

The Global Insurance Industry

As a crude, first-cut, the insurance industry can be regarded as an interplay among three components: *firms*, which offer insurance, *clients*, who buy it, and *events*, which determine the outcomes of the "bets" that have been placed between the insurers and their clients. The overall situation is depicted in Figure 4, which is a *very* high-level view of the insurance industry. But it points out clearly the fact that to understand the dynamics of the industry, it is necessary to take into account the interactions among all three components of the system; just as with the celebrated 3-Body Problem of celestial mechanics, the linkages among the interacting bodies cannot be broken without losing the essence of what makes the problem a "Problem". So it is too with the insurance industry.

By taking a lower-level, more detailed look at each of these three interacting components, we obtain different views of the industry. For example, thinking of "insurers" as being catastrophe reinsurance companies, "clients" as primary insurers, and "events" as things like hurricanes, government regulations, the capital markets, and geographic distribution of risks, we obtain a picture of the world property catastrophe reinsurance industry. But however one slices these three components, what is manifestly evident is that each of these many insurance industries constitutes what in modern parlance we call a *complex, adaptive system.*

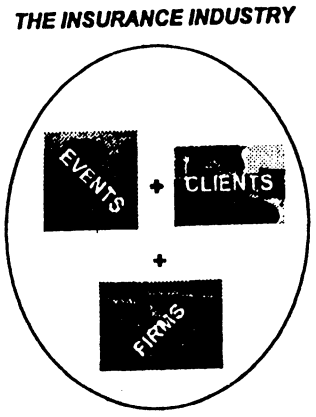


Figure 4. A global view of the insurance industry.

Such systems generically take the form of a large number of interacting agents, each agent taking decisions on the basis of limited information that comes its way. Moreover, the agents have the ability to learn from their past experiences and thus to modify the rules that they employ as the decision process unfolds. As a consequence of these myriad interactions among the agents, the overall system displays global behavioral patterns that usually cannot be understood simply by knowing about the agents in isolation. Rather, these patterns "emerge" out of the interactions among the agents. Examples of such systems include price movements in speculative markets, road-traffic patterns and the appearance and disappearance of species in an ecological ecosystem. We can comfortably add insurance industries to this list.

Since it has proved to be fiendishly difficult to formulate most complex, adaptive systems in conventional mathematical terms, the method of choice for the analysis of these systems is computer simulation. We represent each agent as a piece of computer code, and let these chunks of code interact inside the machine to create new pieces of code by the rules of interaction. In this fashion, we create a virtual counterpart of the real-world system, its agents and environment. This proposal suggests the creation of just such a world for the study of the global property casualty reinsurance industry.

Risk and Insurance

In the "would-be world" proposed here, the agents consist of primary casualty insurers and the reinsurers. The events are then natural hazards, such as hurricanes and earthquakes, as well as various external factors like government regulators and the global capital markets.

The construction of Insurance World has both a purely scientific, as well as applied, purpose. Quite independently of the specific context of the insurance industry, there are basic questions in the theory of risk and organizations that Insurance World should provide a tool to illuminate. Among these general scientific questions are:

●***Optimal Uncertainty***: While insurers and reinsurers talk about getting a better handle on uncertainty, so as to more accurately assess their risk and more profitably price their product, it's self-evident that certain knowledge of natural hazards (a single-point atomic distribution) would spell the end of the insurance industry. On the other hand, complete ignorance of hazards (a uniform distribution) is equally bad for the industry. This suggests that there is some optimal level of uncertainty at which the insurance---but perhaps not their clients---can operate in the most profitable and efficient fashion. What is that level? Does it vary across firms? Does it vary between reinsurers, primary insurers, and/or end consumers?

●***Industry Structure***: In terms of the standard metaphors used to characterize organizations---a machine, a brain, an organism, a culture, a political system, a psychic prison---which type(s) most accurately represents the insurance industry? And how is this picture of the organization shaped by the specific ``routines" used by the decisionmakers in the various components making up the organization?

●***Portfolio Theory***: The efficient frontier concept in modern portfolio theory provides a theoretical framework for evaluating the risk-return relationship in a heterogeneous group of assets. Traditionally, the risk is evaluated by a distribution derived from a time-series of values of the asset and the correlations among these time-series. Catastrophic losses and other types of episodic phenomena are difficult to place in this framework, in large part because the historical pattern of value bears little relationship to the true risk on reasonable time-scales.

For some classes of assets (insurance and reinsurance in particular), the spatial and temporal correlations of the episodic risks mean that the behaviors of and interactions among asset-holders influence the overall risk. In these same types of systems, the behaviors of the asset-holders/risk takers also influence the expected return through both pricing of risks and common perceptions of risk based on the behavior of the rest of the market. Finally, these behaviors have a feedback on the overall flow of capital in and out of markets. All of these behavioral characteristics of the application of portfolio theory to episodic phenomena indicate the potential for dynamical processes to influence the overall structure of the market and the validity of portfolio for this aspect of an economy. The Insurance World model will provide a framework for studying the importance of this dynamical aspect of portfolio management in the context of simulated risk-takers, asset-holders and episodic loss events. In particular, the model will be used to shed light on the question of how an entire industry optimally balance a portfolio of risky ventures when those ventures are natural hazards like hurricanes, earthquakes, and floods rather than man-made risks.

From the more specific viewpoint of the insurance industry itself, the goal of Insurance World is to understand how behaviors of both insurers and reinsurers emerge as a consequence of their interactions, both with each other and with the external events. The model will also provide a tool for posing more specific questions about the behavior of the property casualty insurance industry. Questions of this sort include

●What controls the flow of capital into and out of the reinsurance firms individually, and to and from the market in general?

●What is the price-setting mechanism for reinsurance contracts?

●How is the flow of capital into and out of firms with individual business strategies related to the strategies chosen by other firms?

●What type of alternative mechanisms are there for providing reinsurance coverage, and how do they fare in markets composed of different players and/or conditions?

●How does the timing and magnitude of external events like hurricanes impact pricing and capacity?

In closing, it's worth noting that the ideas underlying Insurance World can also be used to create silicon copies of systems in many other areas of business and economics. For example, work is currently underway to create an electronic copy of a supermarket, its customers and supply chain, in an effort to understand the dynamics of both customer satisfaction and inventory control.

References

[1] Casti, J. *Reality Rules--I, II.* New York: Wiley, 1992.

[2] Casti, J. *Searching for Certainty.* New York: Morrow, 1991.

[3] Casti, J. *Complexification.* New York: HarperCollins, 1994.

[4] Casti, J. *Would-Be Worlds.* John Wiley & Sons, New York, 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Molecular Insights into Evolution

Peter Schuster
Institut für Theoretische Chemie
Universität Wien
A-1090 Wien, Austria

Abstract

Evolution of ribonucleic acid (RNA) molecules in the test-tube provides a possibility to study evolutionary optimization and adaptation to the environment on time scales accessible to human observers. Diversity of genotypes, however, is prohibitive for a complete experimental recording of the process on the molecular level. The number of RNA sequences and structures is too large to be determined by means of currently available techniques. Computer simulation, in contrary, is able to handle large numbers of individual sequences and has no major problem with data retrieval. However, it can deal only with simplified relations between genotypes and phenotypes, being RNA sequences and structures, respectively. Based on a course-grained notion of structure, as represented by RNA secondary structures, for example, a comprehensive model of evolution has been developed that allows to follow optimization at full molecular resolution. This model describes the course of *in vitro* selection experiments and provides a straightforward explanation for the occurrence of steps observed in evolution. It initiated the development of new mathematical concepts which analyse evolution as a complex process viewed simultaneously in concentration space, sequence space and shape space.

1 Introduction

Biological evolution is an exceedingly complex process which for several reasons is commonly unaccessible to experimental investigations: (i) generation times are much too long for the recording of thousands of generations and thus prohibitive for observations of evolutionary phenomena in nature, (ii) unfolding of phenotypes from gentotypes is much too complex to be modeled comprehensively at the current state of knowledge, and (iii) diversity of genotypes is too large to be dealt with exhaustively (The number of genotypes, being 4^n sequences of chain length n, exceeds by far all accessible terrestrial resources as we have about 10^{150} or 10^{4000} polynucleotide sequences with chain lengths typical for viroids or viruses, respectively). In order to be able to analyse evolutionary phenomena by means of methods applied in physics or chemistry reduction to essentials and time-lapse techniques speeding up generation times are inevitable.

The first experiments on evolution of molecules in the test tube were performed by Sol Spiegelman and his coworkers [1] and showed that RNA molecules are capable of exhibiting all essential features of Darwinian evolution in environments sustaining RNA replication. Molecular diversity is created by mutation and a selection constraint is introduced into the system by replacing consumed materials through transfer of small quantities into fresh stock solution. This setup leads to an optimization of replication rates observed in a manyfold increase in the RNA production of populations in series of up to 100 transfers. This serial-transfer technique introduced into *in vitro* evolution by Spiegelman has been developed further since. Nowadays, fully automatized machines are available which allow to design RNA molecules fulfilling additional criteria like, for example, resistence to cleavage by RNases [2]. Other approaches to the evolutionary design of RNA molecules are based on selection through intervention. An example is the well-known SELEX technique which selects candidates for further improvement through binding to chromatographic columns. Recent examples are the development of optimal binders, so-called RNA aptamers, to predefined target molecules [3] or the design of catalytic RNA molecules (For a recent publication see: [4]). Despite the success of the experimental approach towards cell-free optimization there is still a lack of a comprehensive theory of evolution. Just as chemical engineering requires a solid basis in chemical reaction kinetics and material science in order to be successful, evolutionary biotechnology is doomed to fail without a powerful theory of molecular evolution that allows to predict optimal conditions for selection experiments.

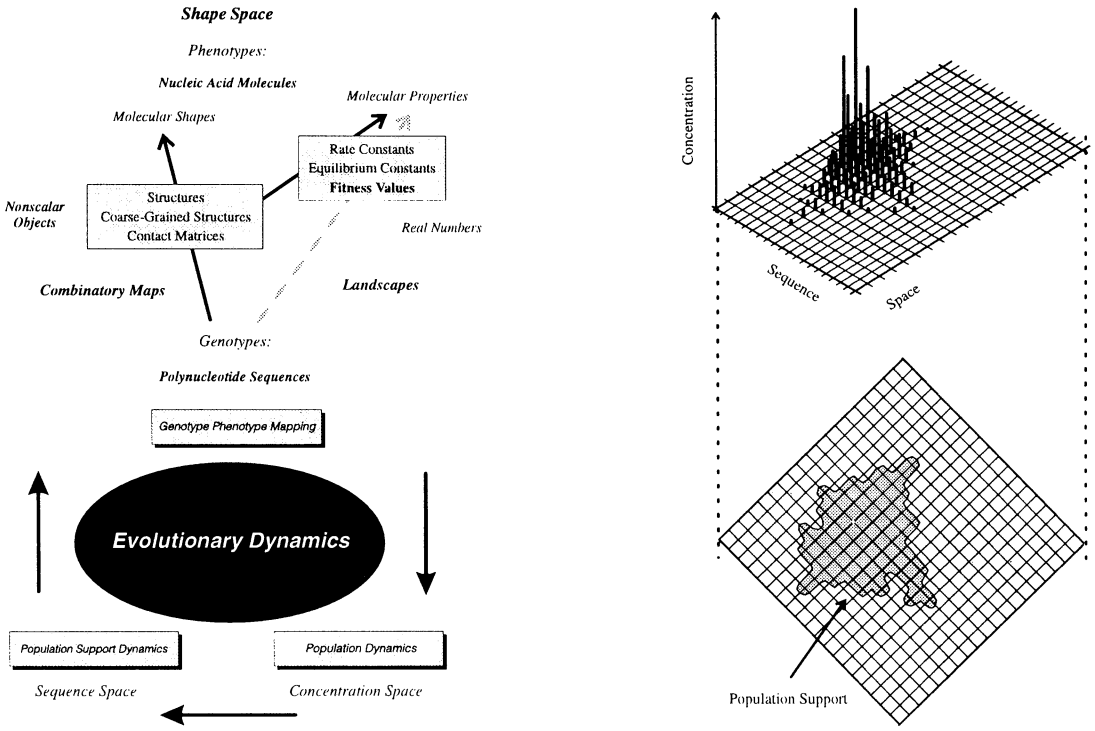


Figure 1: A comprehensive model of evolutionary dynamics. The complex evolutionary process is partitioned into three simpler phenomena (l.h.s.): (i) population dynamics, (ii) support dynamics, and (iii) genotype-phenotype mapping. Population dynamics is tantamount to population genetics of asexually reproducing individuals. It is based on chemical kinetics of replication, mutation, and selection. Support dynamics describes the migration of populations in sequence space. Population forms quasispecies like distributions of genotypes (r.h.s.) which consist of a most frequent master sequence and its closely related mutants. Genotype-phenotype mapping eventually deals with the unfolding of genetic information. Two classes of mappings are distinguished: (i) combinatory maps relating genotypes to phenotypes which are elements of another vector space, for example, spatial structures of molecules, and (ii) landscapes mapping genotype space into the real numbers. In molecular evolution landscapes provide rate constants, equilibrium constants, and other scalar properties of phenotypes [5, 6]. Landscapes are often (but not necessarily) constructed in two steps: (i) a mapping from sequence space into molecular structures and (ii) a mapping from the space of structures into real numbers being quantitative measures of molecular properties.

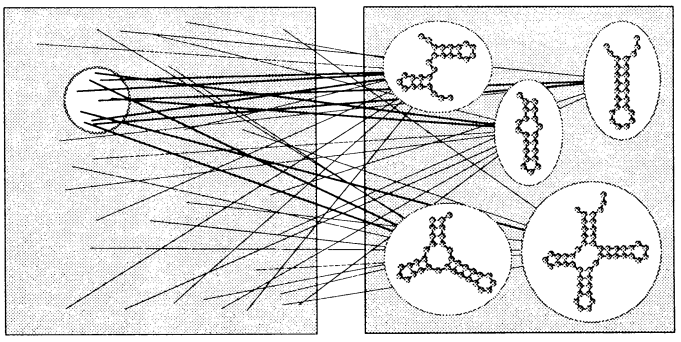


Figure 2: Shape space covering. Only a (relatively small) spherical environment around any arbitrarily chosen reference sequence has to be searched in order to find RNA sequences folding into any common secondary structure.

2 A theory of molecular evolution

The development of a theory of molecular evolution was initiated in 1971 by the publication of Manfred Eigen's seminal article on self-organization of biological macromolecules [7]. His concept is based on chemical kinetics and considers replication and mutation as parallel processes. Suitable models, for example the uniform error-rate model of point mutations, allow formally to deal with the full diversity of sequence space. Later on, the original concept was extended and completed to yield the theory of molecular quasispecies [8, 9]. Quasispecies theory found a successful application in molecular virology [10] and became the basis for molecular evolution experiments [11].

Conventional population genetics as well as almost all other theoretical concepts of evolution are describing distributions of gentotypes and their changes with time. Thus, they are dealing with selection rather than evolutionary optimization or adaptation. The properties of phenotypes, for example rate or equilibrium constants, appear only as parameters of the corresponding dynamical equations and the relations between genotypes and phenotypes are not considered explicitly. The restriction to selection of genotypes is problematic whenever evolutionary processes are shaped by the properties of the accesible phenotypes [12]. Genotype-phenotype dichotomy, however, is genuine to evolution because variation occurs on genotypes, whereas selection is always operating on the corresponding phenotypes.

More recently [13, 14] a comprehensive model of evolutionary dynamics has been suggested which simplifies evolution by reducing it to three simpler processes (figure 1): (i) population dynamics, (ii) support dynamics, and (iii) genotpye-phenotype mapping. Population dynamics is a generalization of population genetics in the sense that no assumptions concerning the targets of selection are made. They may be loci (genes), genotypes or families of genotypes. Support dynamics describes how populations migrate through sequence space. Relevant questions are, for example, whether populations approach stationary mutant distributions which are represented by molecular quasispecies, or whether populations drift randomly through sequence space in the sense of a diffusion process. Genotype-phenotype mapping completes the dynamics of evolution. It provides the input for population dynamics since it determines the parameters of the dynamical equations. It allows to detect redundancy known as selective neutrality of genotypes. In addition, it defines the phenotypes which are accessible from a given phenotype through mutation.

Although genotype-phenotype mapping is a core issue of evolutionary dynamics it is commonly so complex that it is very hard if not impossible to model. Phenotypes are sophisticated, highly organized and often hierachically ordered, structured objects with an enormously rich and diverse repertoire of functions. Here, we shall restrict our considerations to the simplest conceivable case: evolution of RNA molecules in the test-tube. Then, genotype and phenotype are two aspects of the same molecule [1], the sequence and the structure, respectively, and genotype-phenotype mapping boils down to the relation between sequences and structures of RNA molecules. Sequence-structure relations of biopolymers are difficult to explore too, but a coarse-grained notion of structure, the so-called RNA secondary structure, is sufficiently simple for mathematical analysis and studies by means of computer simulation and, at the same time, captures the essential features of realistic genotype-phenotype mappings.

3 RNA genotypes and phenotypes

Relations between RNA sequences and secondary structures are understood as mappings from sequence space into a space of shapes. They have been studied by means of a mathematical model based on random graph theory [15], by means of exhaustive folding of all sequences of given chain length [16] as well as by statistical analysis of appropriately chosen samples [17, 18]. These investigations revealed four regularities:

(i) The number of sequences exceeds the number of structures by several orders of magnitude and hence, sequence-structure maps are many to one.

(ii) Relatively few common structures are contrasted by many rare structures which usually play no role in evolution. In the limit of long chains we have almost all sequences folding in a tiny subfraction of all structures.

(iii) In order to find a given sequence forming any of the common structures one need not explore whole sequence space. It is sufficient to search a relatively small spherical neighborhood of an arbitrarily chosen reference sequence (**shape space covering**, see figure 2 and [18]).

(iv) Common structures are characterized by a high degree of neutrality expressend by the fraction of nearest neighbors (λ) that are neutral with respect to selection. The sets of sequences folding into them, i.e. their preimages, form extended **neutral networks** in sequence space.

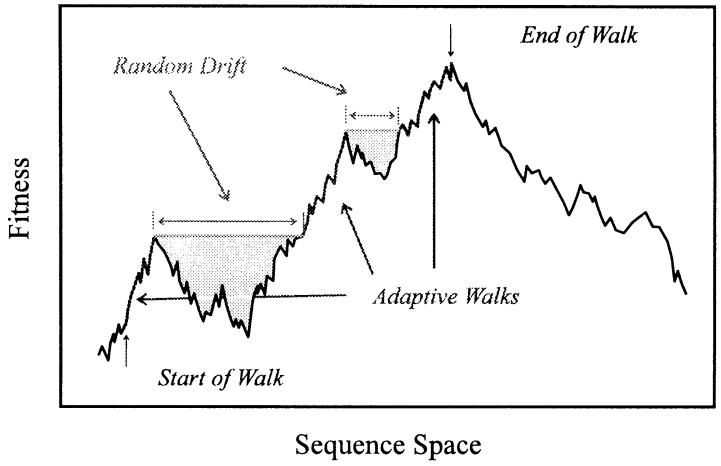


Figure 3: Neutral networks and evolutionary optimization.

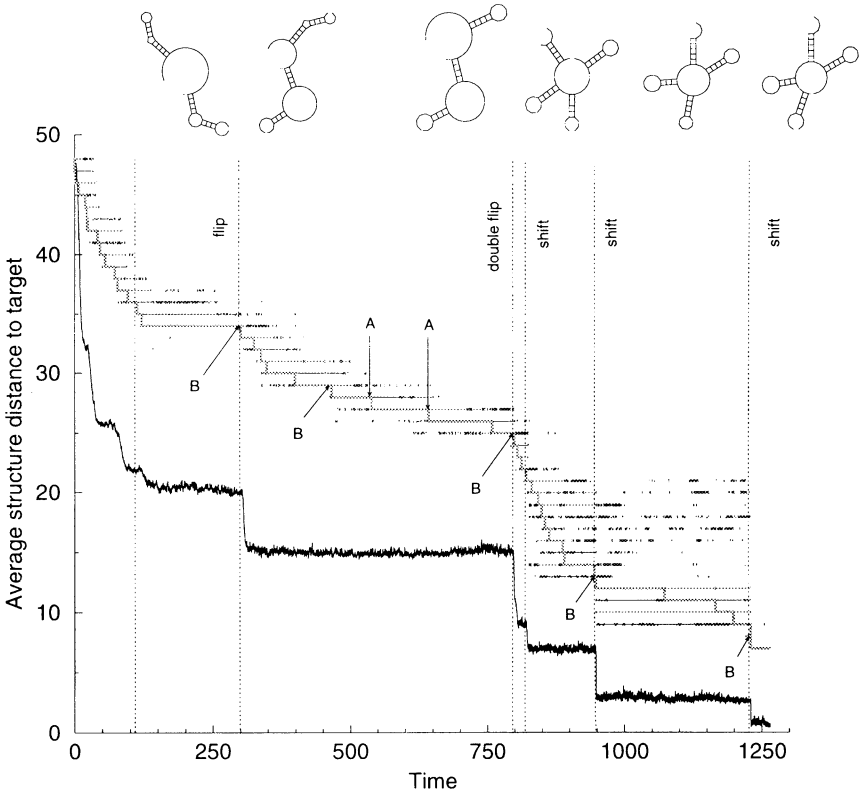


Figure 4: Evolutionary optimization of RNA structure. Fitness of structures is computed as a function of the distance to the target structure that was chosen here to be the tRNA clover-leaf ($n = 76$). The distance to the target structure is averaged over the whole population consisting of about 1000 RNA molecules and plotted against time. A mutation rate of $p = 0.001$ per site and replication was applied. A relay series of 41 phenotypes has been retrieved. Six most important ones are shown on top of the figure. Two classes of transitions are distinguished: (i) continuous transitions (for example those marked by **A**) and (ii) discontinuous transitions (**B**). Discontinuous transitions lead to major changes in the structures which are followed by cascades of minor fitness improving steps.

The four properties are highly relevant for evolution. The existence of neutral networks, for example, is crucial for the efficiency of evolutionary searches (figure 3 and [19]) because it avoids populations to be caught in local fitness optima, so-called evolutionary traps. Shape space covering provides a powerful tool for the design of efficient protocols for searches in sequence space [20].

4 Molecular images of RNA evolution

The results on sequence-structure mapping of RNA reported in the previous section call for independent tests by means of computer simulations or experiments. The first simulations based on a realistic model of RNA structures were reported about ten years ago [21]. Like in later works a population of RNA molecules undergoing replication and mutation is subjected to the constraints of a flow reactor which keeps the population size N constant within fluctuations of $\sqrt{N}$-size. The replication rates are determined according to some predefined fitness criterion. More recently, simulations of this kind were used to show that evolution on the neutral network of a tRNA-structure corresponds to a diffusion process where the diffusion coefficient is proportional to the mutation rate [19]. In this simulation as in the computer experiments described below replication rates (r_α) were assumed to depend on structure (independently of the sequence folding into it and thus fulfilling the neutrality condition). In particular, a function of the kind $r_\alpha = (\delta + d(\alpha, \tau)/n)^{-1}$ was used, where δ is some constant, n the chain length of the RNA, and $d(\alpha, \tau)$ the distance between structure α and the target structure τ. General results, however, were found to be largely independent of the specific choices of constants and fitness functions.

Optimization of RNA structures was studied through simulations of a population size in the flow reactor. The approach towards the target structure which happened to be a tRNA clover-leaf occurs in steps. Periods of fast decrease in distance to the target are interrupted by long quasi-stationary phases of almost constant average fitness (figure 4). The course of the evolutionary optimization process has been reconstructed by computing the **relay series** of phenotypes. The relay series is the uninterrupted sequence of structures which eventually leads to the formation of the target structure. The full relay series of the computer experiment shown in figure 4 contains 41 structures, six particularly important ones are shown on top of the figure. The transitions between structures fall into two classes:

- **continuous transitions** representing minor structural changes and leading to globally frequent structures in the neighborhood of the neutral network of the intial structure and

- **discontinuous transitions** representing major stuctural changes and leading to globally rare but locally frequent structures (They are marked by **B** in figure 4).

Discontinous transitions occur at the ends of the quasi-stationary phases (There is one exception around time $t \approx 500$ which represents a "silent" discontinuous transition since it does not change fitness). Every major transition is followed by a cascade of minor transitions which are accompanied by substantial fitness increase. Then the population approaches the next plateau along which neutral evolution occurs at approximately constant fitness until it reaches sequences where a fitness-improving discontinuous transition is locally frequent. As indicated in the sketch shown in figure 3 evolutionary optimization on realistic landscapes with high degree of neutrality proceeds on two time scales: fast periods containing cascades of adaptive changes are interrupted by long quasi-stationary phases of neutral evolution during which populations drift randomly on neutral networks until they reach a local neighborhood that sustains the next major transition.

5 Generalization to evolution

The RNA model has been used to develop a statistical notion of nearness in genotype space that can be formulated as a novel kind of topology [12] which allows straighforward generalizations to other evolutionary systems. Nearness is restricted to frequent occurence in the neighborhood of neutral sets. Transitions between near phenotypes are continuous and occur readily. Discontinuous transitions occur between globally distant phenotypes in those regions of genotype space where these phenotypes are locally frequent neighbors. Evolution is viewed as a sequence of continuous and discontinuous transitions where the latter depend on special genotype requirements. The role of neutral evolution is to search for these special genotypes through random drift.

Acknowlegements

The work reported here has been supported finacially by the Austrian *Fonds zur Förderung der wissenschaftlichen Forschung* (Projects P-10578 and P-11065), by the Commission of the European Union (Contract Study PSS*0884), and by the Santa Fe Institute.

References

[1] S. Spiegelman, "An approach to the experimental analysis of precellular evolution," *Quart.Rev.Biophys.*, Vol. 4, pp. 213-253, 1971.

[2] G. Strunk, T. Ederhof, "Machines for automated evolution experiments *in vitro* based on the serial-transfer concept," *Biophys.Chem.*, Vol.66, pp.193-202, 1997.

[3] A.D. Ellington, "Aptamers achive desired recognition," *Current Biology*, Vol.4, pp.427-429, 1994.

[4] B. Zhang, T.R.Cech, "Peptide bond formation by *in vitro* selected ribozymes," *Nature*, Vol.390, pp.96-100, 1997.

[5] P. Schuster, "Landscapes and molecular evolution," *Physica D*, Vol.107, pp.351-365, 1997.

[6] P. Schuster, "Genotypes with phenotypes. Adventures in an RNA toy world", *Biophys.Chem.*, Vol.66, pp.75-110,1997.

[7] M. Eigen, "Self-organization of matter and the evolution of biological macromolecules," *Naturwissenschaften*, Vol.58, pp.465-523, 1971.

[8] M. Eigen, P. Schuster, "The hypercycle. A principle of natural self-organization. Part A: Emergence of the hypercycle," *Naturwissenschaften*, Vol.64, pp.541-565, 1977.

[9] M. Eigen, J. McCaskill, P. Schuster, "The molecular quasispecies," *Advances in Chemical Physics*, Vol.75, pp.149-263, 1989.

[10] E. Domingo, C. Escarmis, N. Sevilla, A. Moya, S.F. Elena, J. Quer, I.S. Novella, J.J. Holland, "Basic concepts in RNA virus evolution," *FASEB J.*, Vol.10, pp.859-864, 1996.

[11] S. Brakmann, "On the generation of information as motive power for molecular evolution," *Biophys.Chem.*, Vol.66, pp.133-143, 1997.

[12] W. Fontana, P. Schuster, "Shaping space. The possible and the attainable in RNA genotype-phenotype mapping", *J.Theo.Biol.*, submitted. Published as Santa Fe Institute Preprint No. 97-11-081, 1997.

[13] P. Schuster, "Artificial life and molecular evolutionary biology." In: F. Morán, A. Moreno, J.J. Merelo, P. Chacón, Eds., "Advances in Artificial Life," *Lecture Notes in Artificial Intelligence*, Vol.929, pp.3-19. Springer-Verlag, Berlin, 1995.

[14] P. Schuster, "How does complexity arise in evolution?", *Complexity*, Vol.2, pp.22-30, 1996.

[15] C. Reidys, P.F. Stadler, P. Schuster, "Generic properties of combinatory maps. Neutral networks of RNA secondary structures", *Bull.Math.Biol.*, Vol.59, pp.339-397, 1997.

[16] W. Grüner, R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I.L. Hofacker, P.F. Stadler, P. Schuster, "Analysis of RNA sequence structure maps by exhaustive enumeration", *Monatsh.Chem.*, Vol.127, pp.355-389, 1996.

[17] W. Fontana, D.A.M. Konings, P.F. Stadler, P. Schuster, "Statistics of RNA secondary structures", *Biopolymers*, Vol.33, pp.1389-1404, 1993.

[18] P. Schuster, W. Fontana, P.F. Stadler, I.L. Hofacker, "From sequences to shapes and back. A case study in RNA secondary structures", *Proc.Roc.Soc.(London)B*, Vol.255, pp.279-284, 1994.

[19] M.A. Huynen, P.F. Stadler, W. Fontana, "Smoothness within ruggedness. The role of neutrality in adaptation", *Proc.Natl.Acad.Sci.USA*, Vol.93, pp.397-401, 1996.

[20] P. Schuster, "How to search for RNA structures. Theoretical concepts in evolutionary biotechnology", *Journal of Biotechnology*, Vol.41, pp.239-257, 1995.

[21] W. Fontana, P. Schuster, "A computer model of evolutionary optimization", *Biophys.Chem.*, Vol.26, pp.123-147, 1987.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

BRAIN COMPUTING

Gen MATSUMOTO
Director, Brainway Group
Brain Science Institute (BSI)
The Institute of Physical and Chemical Research (RIKEN)
2-1Hirosawa, Wako, Saitama 351-01 Japan
phone:+81-48-467-9635/ fax:+81-48-467-9643/ e-mail: gen@brainway.riken.go.jp

The brain is an automatic algorithm acquisition system. Acquired algorithm is represented both as configurational and as activity changes in the neuron networks. One of the essential strategies for the brain to acquire algorithm is learning. New in-computing information is used as a sort of trigger for activating some of the existing neuronal circuits, which enables the brain to provide output in the form of behaviour. Giving an output will exercise a learning effect, to rewrite the algorithm. That is to say, the brain compiles a conversion table in advance on the basis of learning, and the in-coming information allows to select one of pre-arranged responses, which has the highest correlation with itself as an output. Giving an output results in a learning effect, and permits to modify responses in the repertory in accordance with the effects of output.

Let us examine the nature of learning, which is genetically provided as a brain strategy for acquiring algorithm. The learning effect is induced when a neuron or a neural network or a brain receives a "significant" stimulus. Here, the matter concerns what a significant stimulus is, and what a learning effect is. The learning effect may be defined as a factor to cause and the "significant" stimulus may be defined as an input information causing a neuron or a neural network or a brain launch an output.

A neuron checks the amount of synaptic memory at the input end in the moment an output is given out, and wither reinforces or attenuates the coupling depending upon the memory amount. In this way, signals coming into a neuron on temporally separated manner are integrated when the neuron gives an output. The rule of time-sequence learning is dependent on output, and creates an asymmetrically coupled neuron circuit. As the activity in the neuronal circuit induced in time sequence propagates uni-directionally, it seems that putting things in the temporary order is "brain-compatible" based on this principle. The rule of time-sequence learning is very interesting when considering how the brain perceives time. When the physical time span is the same for both the young and the aged, the quantity of information received by a young brain as "significant stimuli" is far greater than that received by an aged brain, inducing much more intensive learning effect and impressing much greater amount of relevant information in the brain. If an aged person feels that a year in the past was too short, it means that the amount of information if which correlation had been established through learning and which had been stored in the brain as memory of experience was very small. Living "fruitful" in time depends upon how far the learning effect has been enhanced through moving or impressive experience, and how much "engram" has been engraved in the brain in a limited span of physical time, but as acquiring much greater amount of information in the course of human existence and inducing by far the more abundant learning effects in the brain within that time span of physical time.

What is an input information entitled to be a "significant" stimulus? A sensory stimulus of higher physical strength can be an input of higher intensity. However, though a physically strong stimulus can induce an output in the initial stage, soon the accommodation steps in to reduce the effectiveness of the stimulation. A strong stimulus can be accepted as an effective stimulus in sustained manner only when it is related to emotion. For instance, whether or not a loud sound input is perceived as a strong stimulus depends not on its physical loudness, but on its

emotional effect, such as hatred or scare.

The brain activity is controlled in the manner of feed-forward. The input information to the brain is handed in parallel by dual information processing systems: a cognitive information system in the cerebral cortex and an emotional information system involving amygdaloid body of the brain lymbic system.

The emotional information system decides whether the input information is pleasant or unpleasant, or in other words, valuable or valueless for the existence of an individual. This is a judgement of poor resolution, but gets to the conclusion very quickly. On the other hand, the cognitive information system is characterized by high precision, but requires longer time for processing. If an input information is provided to be emotional pleasant, and valuable on the basis of judgement including results of cognitive processing, the brain activity is enhanced through the humoral control mechanism such as diffusible transmitter substance, to induce learning effects. The enhanced brain activity makes it ready to launch an output, which in its turn exercises the learning effect, consequently. That is, the brain constructs automatically a circuit to process such an information of which value has been recognized.

It may be no exaggeration to state, therefore, that the brain activity is controlled most effectively by the emotional or value information, and all the information handed by a human being is ascribed to the emotion. For instance, when a little boy tells to his mother, "Daddy gave me 10 bucks, and I'm going to buy a picture book.", the most important information the boy wishes to convey his mother is his happiness caused by the interaction with his father, rater than the fact that he gets money or his plan how to use it. If the information is defined as things responsible for the brain activity, the information significant for the brain should be things concerning the emotion underlying the activity, rather than things is the value information.

Since the information processing in the brain is characterized by constructing a conversion table through learning, involving a pre-arranged set of responses, or repertoires, and letting the input information to pick up one of them, there must be a strategy to derive an appropriate solution very effectively. The strategy being employed by the brain in this respect is a hypothesis – verifying system. That is, just like the emotional evaluation based on rough concepts in the thalamus-lymbic system, the cognitive information processing system in the cerebral neocortex is supposed to be controlled in the direction of backing up the conceptual logic at an earlier stage. This mechanism will allow to activate neuronal circuit to be selected in the neocortex for a particular response, and the cognitive information system works so as to provide a logical backup for the initial rough assumption. There is an important device to support this mechanism: focal attention system. It has been known in the visual cognitive mechanism that a projection system innervates from the pulvinar nucles of the thalamus in the radiating manner the primary visual area, the pre-visual area and the temporal association area, to selectively activate a particular neuronal pathway in the visual recognition circuits. Let us consider a typical example of pattern recognition in the visual cognitive processing. I would like to propose a stepwise recognition mechanism in place of the conventional thinking of package recognition. In the latter, the brain analyzes the input pattern sequentially and attains to a recognition that the pattern currently being viewed is that Miss A's face. According to my thinking, the first step is a rough categorization that the object being viewed is a human face. Them, the primary cognitive system provides a number of potential candidates: Miss A's face, Miss B's face, and so on. Recently, a feedback pathway from the temporal association area to the primary visual area via the pre-visual area has been demonstrated. Following a rough conceptualization of cognition on the trial basis, iterative verification of the assumption is repeated until a final pattern recognition is reached after having eliminated every trace of self-inconsistency.

The brain has adopted the hypothesis-verifying strategy as a means of effectively selecting a response it needs out of a set repertoires acquired through learning. When you convince yourself that a thing can be achieved, the cognitive information system is put

into a full activity to provide a logical backup for your conviction. I this way, what you believe to be realizable is realized without fail. On the contrary, if you have nothing to believe in, the brain loses the motivational orientation for information processing, and is put into confusion. You need a conviction on how to lead yourself, or your information processing in regard to what you want or what you have not experienced. It seems that this is a possible origin of belief, and the desire for beliefs is fully "brain-compatible". "Letters to the Hebrew" Paragraph 1, Chapter 11 of the New Testament defines the belief in this way: "The belief is to convince of what you want and to confirm what you have never seen. "When we encounters an experience of acquiring unexpected power or achieving a great success by conveying of something, we feel the existence of non-perceptual control. This feeling leads to the joy of believing, and having faith in the existence of almighty. It seems that the system of religion evolved in this way as the civilization grew. Science also has the same way of developing starting with the birth of civilization, aiming at understanding the nature through pile-up of logic and achieving what the man wants. The decisive difference between science and religion is that the religion relies upon the predisposed conviction, while in science the conviction is acquired as a result of investigating facts. The difference which has separated the two from each other, just like water and oil, is vividly reflecting the difference in the mode of information processing between the computer and the brain. If the intuitive conviction in the brain should lead the information processing in the brain to a wrong direction, an erroneous logical backup would be provided, resulting in a fallacious behavioral output. This is a sort of uncontrollable convicted criminal. However, the information processing of higher order, such as creativeness, owes much to such a fallible mechanism of information processing in the brain. The boundary between the genius and the madness is as thin as a sheet of paper.

If it could be clarified through the brain research what a man is, our way of living and the objective of our life would be provided with a solid ground. The objective of human life may be described as realizing oneself. Then, what realizing oneself means? As sated in the above, realizing oneself is to establish one's own value system. A society in which individuals having heterogeneous value system coexist while mutually respecting other person's value systems, may be regarded as a fruitful society. The organizational pattern of value system is identical for both individuals and societies, in spite of hierarchical difference. If societies having their own value systems or cultures could coexist while mutually respecting other societies' value system or cultures, it would be possible to achieve the global peace. If a particular society claims exclusive righteousness of its own value system, a war will ensue.

The religion and the science have been repulsive to each other as antipodal entities. In consideration of the fact that the religion has been and still is playing a significant role as a mental support for living, and that the brain is also providing a support for the mental activity in the inner world, it is a matter of utmost importance for teh human being, I believe, to define the position of religion from the viewpoint of brain science and to clarify the relation between religion and science.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Some theoretical issues on computer simulations[1]

C.L. Barrett
Los Alamos National Laboratory
TSA/DO-SA
87548 New Mexico
Mailstop: TA-0, SM-1237, MS M997

C.M. Reidys
Los Alamos National Laboratory
TSA/DO-SA
87548 New Mexico, USA
Mailstop: TA-0, SM-1237, MS M997

November, 1997

[1] short version, extended version in press in: Appl. Math. and Comp.

Abstract

The subject of this paper is the development of mathematical foundations for a theory of simulation. Sequentially updated cellular automata (sCA) over arbitrary graphs are employed as a paradigmatic framework. In the development of the theory, we focus on the properties of causal dependencies among local mappings in a simulation. The main object of study is the mapping between a graph representing the dependencies among entities of a simulation and a graph representing the equivalence classes of systems obtained by all possible updates.

1 Introduction

Computer simulations are "composition engines", that compose the actions of "entities" or "actors" with respect to a certain support-structure and are typically used for prediction and analysis. Consequently, a key question for a "theory of simulation" is to analyze *how* this composition is actually works, meaning: how does it depend on the actors/agents and on the actual dependency structure among them? Sequential dependencies, for example, are an issue in non-cooperative game theory [16]. Accordingly the primary focus of a theory of simulation is on the properties that result from composition of causal local interactions among individual entities. In order to assure generality, given a collection of local mappings and an update schedule, by hypothesis we assume entities to exist inherently—whether or not the mappings are specifically associated with software agent/actors, sites in a lattice, or otherwise. Accordingly, we make no essential reference in the theory to any particular method of local representation in a simulation. For mathematical convenience and without loss of generality relative to simulation viewed as ordered composition of local maps, in this paper we utilize sequential updates of cellular automata (sCA) [21] over graphs as a paradigmatic framework for locality and sequentiality. We actually define a sCA over a graph (c.f. [23] for a similar approach that considers CA over Cayley graphs), X, by associating to each vertex i, a *state* $x_i \in \{0,1\}$ and a *local map* f_i defined on the states of the i-neighbors and x_i, and which returns the state y_i. Application of local maps f_{i_k} in a particular order then induces a sequential CA (sCA) over a graph. This setting allows us to express the locality by adjacency in X and sequentiality by the different orders in which the vertices can be updated. For example, update sequences can be introduced for standard (i.e., a CA updated in parallel according to a nearest neighbor rule and closed boundary conditions) CA [18, 7, 1]. This yields an sCA over Circ_n (the circle graph on n vertices). In fact, every CA (over Circ_n) can be derived from a sCA over $2n$ vertices.

The line of thought of the paper as it relates to simulation is as follows. First basic terminology and the ansatz are introduced. Then the dependency graph and its local mappings are introduced. The dependency graph induces another graph (an update graph) wherein vertices are permutations of mapping update orders, and components imply functionally equivalent update orders. Relationships between these two graphs are proven where the dependency graph is viewed as a random graph (a probability space) that induces a random update graph. Hamiltonian paths in one graph imply isolated points in the other, and a single threshold function exists for the occurrence of both a Hamiltonian path *and* connectivity in the dependency graph [15, 5, 4, 3, 12]. This corresponds to the occurrence of isolated points in the update graph.

The interpretation relevant to simulation is that if the local mappings have more than very sparse dependencies in an update time interval, then, with probability one, unique update sequences, nearly a notion of a specific script, will be necessary to produce a given dynamical system. The number of components in the update graph is studied which obviously corresponds to the number of dynamical systems that a collection of co-dependent mappings can generate. The existence and the length of a path in the random update graph $U(G_{n,p})$ between two update orderings, e.g., update schedules, is analyzed. In fact, this equivalence can be tested in polynomial time. Further, the average size of a $U(G_{n,p})$-component is computed. The relevance to simulation is that given two update orders, one can determine if they are in the same component; that is, if they produce the same dynamical system. We emphasize that this determination is made based on analysis of the structure of of the causal dependencies among local mappings and not by measurements in the state space of those mappings or their composition. Last, the update is viewed from a categorical point of view rather than as a random variable. These results address the essential foundational issues of computer simulation, generation of system dynamics by composition of local maps and congruence of simulations, in a very general way and points to strategies for development of the theory. Full proofs of all results can be found in [9].

2 Terminology

Let X be a finite undirected graph with vertex set $\mathbf{v}X$ and edge set $\mathbf{e}X$. Two adjacent vertices P, Q are called *extremities* of an edge y and a subgraph $Y < X$ is called *induced*, if $P, P' \in \mathbf{v}Y$ are adjacent in Y if and only if P, P' are adjacent in X. A *path* in X is a multi-set $(Q_1, y_1, Q_2, \ldots, y_n, Q_{n+1})$, where $Q_i \in \mathbf{v}X$, $y_i \in \mathbf{e}X$, Q_i and Q_{i+1} are extremities of y_i. The *length* of a path is the number of edges in the multi-set $(Q_1, y_1, Q_2, \ldots, y_n, Q_{n+1})$; the *distance* $d_X(P, Q)$ between X-vertices P, Q is the minimal length of a X-path connecting P, Q or ∞ if there is no such path. Suppose $|\{Q_1, \ldots, Q_n\}| = n$; the path $(Q_1, y_1, Q_2, \ldots, Q_n, y_n, Q_1)$ corresponds up to isomorphism to a graph, which we refer to as Circ_n, the circle graph on n vertices. A graph X is called *connected* if any two vertices occur in an X-path. X-subgraphs induced by maximal connected subsets of vertices are called *components*. Let Y be a subgraph of X; the set of $X \setminus Y$-vertices that are adjacent to some vertices of Y, is called the *Y-vertex-boundary*, $d_X Y$. We denote the *ball* of radius k in X by

$$\mathcal{B}_k(i, X) = \{\, j \,|\, d_X(i, j) \leq k \,\}. \qquad (2.1)$$

A *random graph* [14, 15, 5, 2] is a finite probability space consisting of subgraphs of a base graph X. E.g. take $X = K_n$, the complete graph over n vertices, and consider subgraphs $Y < X$ with the underlying probability $\mu_{n,p}(Y) = p^{|\mathbf{e}Y|}[1-p]^{\binom{n}{2}-|\mathbf{e}Y|}$. We denote the random graph $(\{Y < K_n\}, \mu_{n,p})$ by $G_{n,p}$. Its elements are obtained by selecting each K_n-edge with independent probability p. Let $p_n, q_n : \mathbb{N} \to [0,1]$ be probabilities. A *threshold function* of a property $(\#)$ of a random graph is a probability p_n such that: for n tending to ∞, the probability of the set of subgraphs having $(\#)$ fulfills a 1-0–law: for $\lim_{n\to\infty}(q_n/p_n) = 0 : \lim_{n\to\infty} \mu_{q_n}\{H \models (\#)\} = 0$ and for $\lim_{n\to\infty}(p_n/q_n) = 0 : \lim_{n\to\infty} \mu_{q_n}\{H \models (\#)\} = 1$. A n-cube $\mathcal{Q}_2^n$ is a graph whose vertices are n-tuples $(x_1, \ldots, x_n)$, $x_i \in \{0,1\}$ and two vertices are adjacent if they differ in exactly one coordinate. We will abbreviate *almost surely* by a.s. and write for functions $f : \mathbb{N} \to \mathbb{R}$, $g : \mathbb{N} \to \mathbb{R}$: $f = O(g)$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ exists and $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c > 0$, in particular $f \sim g$ for $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$ and $f = o(g)$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ exists and $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Further we make use of the following result on sums of independent random variables, $Z_n = \sum_{i=1}^n z_i$, where each z_i has values in $\{0,1\}$ [8, 2]. Then, $\forall c > 0\, \exists c_1 > 0$:

$$\mu_n\{\,|\, Z_n - \mathbb{E}[Z_n] \,|\, > c\,\mathbb{E}[Z_n]\,\} \leq e^{-c_1 \mathbb{E}[Z_n]}. \qquad (2.2)$$

Let S_n be the symmetric group over n letters. We write $(1, 2, \ldots, n-1, n)$ as $\mathbf{id}$ and set

$$\forall\, \pi \in S_n : \quad \mathrm{Inv}(\pi) = |\{\, (i,k) \,|\, i < k \wedge \pi(i) > \pi(k) \,\}| \qquad (2.3)$$

that is $\mathrm{Inv}(\pi)$ is the number of *inversions* of the permutation π. For a pair (i,k) with $\pi(i) > \pi(k)$ we say (i,k) is an *inversion-pair* (i.p.) of π [13].

3 Sequential cellular automata over graphs

A *cellular automaton* (CA) [21, 18] of size n (with closed boundary conditions, i.e. cell i depends on $i-k$ mod n, $\ldots$, $i+k$ mod n) is a dynamical system

$$g : \mathbb{F}_2^n \to \mathbb{F}_2^n, \qquad g[(x_1, \ldots, x_n)] = (y_1, \ldots, y_n) \quad (3.1)$$

where $\mathbb{F}_2^n$ is a n-dimensional vector space over $\mathbb{F}_2$. The map g is induced by *parallel application* of a *local rule* f, that is, a mapping

$$f : \mathbb{F}_2^{2k+1} \to \mathbb{F}_2, \quad f(x_{i-k\equiv n}, \ldots, x_{i+k\equiv n}) = y_i . \tag{3.2}$$

Here, local means that the coordinate y_i depends only on $x_{i-k\equiv n}, \ldots, x_{i+k\equiv n}$ for small k. In particular, the locality of a nearest neighbor CA corresponds to the adjacency relation of the *circle graph* Circ_n with vertex set $\{1, \ldots, n\}$ and $k = 1$. Next we define sequential CAs over regular graphs. In certain analogy with a classic CA with nearest neighbor rule, where $X \cong \mathrm{Circ}_n$, a sequential CA (sCA) is a quadruple $(X, \mathbb{F}_2, (f_{i,X}), \pi)$, where

- X is a regular graph with $\mathbf{v}X = \{1, \ldots, n\}$ and vertex degree m
- $\mathbb{F}_2$ is the set of *states* of an X-vertex
- $f_{i,X}$ is a *local map*: $f_{i,X} : \mathbb{F}_2^{m+1} \to \mathbb{F}_2$

  $f_{i,X}(x_{i_1}, \ldots, x_{i_{m+1}}) = y_i \quad i_1 < i_2, \ldots, i_m < i_{m+1}$

  $i_k \in \mathcal{B}_1(i, X)$ where $i_1, \ldots i_{m+1}$ are X-vertices and $\mathcal{B}_1(i, X)$ is defined by (2.1)
- π is a permutation, or ordering $\mathfrak{O}(\pi)$ according to which the local maps $f_{i,X}$ are applied.

The mapping $(x_{i_1}, \ldots, x_{i_{m+1}}) \mapsto y_i$ is referred to as the *update* of x_i. The local maps $f_{i,X}$ induce maps $F_{i,X} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ in a natural way:

$$F_{i,X}((x_j)_j) = (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n) . \tag{3.3}$$

where $y_i = f_i(x_{i_1}, \ldots, x_{i_{m+1}})$. The induced maps $F_{i,X} : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $F_{j,X} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ allow the formation of products $F_{i,X} \circ F_{j,X}$. Now, choosing an ordering $\mathfrak{O}$ of the X-vertices a complete update of $\mathbf{v}X$ corresponds to a product

$$\prod_{i=1}^{n} F_{\pi(i),X} : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n , \quad \pi \in S_n . \tag{3.4}$$

Suppose now we have a subgraph $Y < X$ such that $\mathbf{v}Y = \mathbf{v}X$. Although Y is not regular we can define maps $f_{i,Y}, F_{i,Y}$ analogous to $f_{i,X}, F_{i,X}$ if

- $f_{i,Y} : \mathbb{F}_2^{m+1} \to \mathbb{F}_2$, $f_{i,Y}(x_{i_1}, \ldots, x_{i_{m+1}}) = y_i$, $i_1 < i_2, \ldots, i_m < i_{m+1}$, $i_k \in \mathcal{B}_1(i, X)$, and
- $f_{i,Y}$ does *not* depend on $X \setminus Y$-neighbors.

Note that in the extreme case $\mathbf{e}Y = \emptyset$, the maps $f_{i,Y}$ only depend on the vertex i itself. In complete analogy to (3.3) we then induce the maps $F_{i,Y}$ and the time evolution of Y-vertices is obtained by iterating $\prod_{i=1}^n F_{\pi(i),Y} : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$, $\pi \in S_n$. Each factor $F_{\pi(i),Y}$ is a mapping that updates the state of vertex $\pi(i)$ as a function on all $\pi(i)$ neighbors in Y and that leaves all other Y-vertex states invariant.
Let us next assume that $Y < X$ and $\mathcal{F} = \{F_{1,Y}, \ldots, F_{n,Y}\}$ are fixed. Then it is natural to ask which permutations leave a given product $\prod_{i=1}^n F_{\pi(i)}$ invariant.
Given two permutations $\pi, \pi' \in S_n$, it is important to note that the equality

$$\prod_{i=1}^{n} F_{\pi(i)} = \prod_{i=1}^{n} F_{\pi'(i)} \tag{3.5}$$

is due to either (a) symmetries of the maps F_i or (b) the structure of X. In this paper we will not address (a), instead we will elaborate (b) in some more detail. First ,$\forall \{\pi(i), \pi(j)\} \in \mathrm{e}[Y]$ $\exists F_{\pi(j),Y}, F_{\pi(i),Y}$

$$F_{\pi(j),Y} \circ F_{\pi(i),Y} \neq F_{\pi(i),Y} \circ F_{\pi(j),Y} \tag{3.6}$$

and second $\forall F_{\pi(j),Y}, F_{\pi(i),Y} : \{\pi(i), \pi(j)\} \notin \mathrm{e}[Y]$

$$F_{\pi(j),Y} \circ F_{\pi(i),Y} = F_{\pi(i),Y} \circ F_{\pi(j),Y} . \tag{3.7}$$

Taking an abstraction of this situation we may then replace a product $\prod_{i=1}^n F_{\pi(i),Y}$ by the corresponding permutation π. Moreover, we can introduce a neighborhood relation by saying that two permutations $\pi = (i_1, \ldots, i_n)$ and $\pi' = (i'_1, \ldots, i'_n)$ are adjacent iff there exist two consecutive indices i_k, i_{k+1} such that $i_k = i'_{k+1}$, $i_{k+1} = i'_k$ and $i_j = i'_j$, otherwise. This approach leads to the definition of the *update graph* $U(Y)$ whose components imply certain *equivalence classes* of sCA over Y. We will formally define update graphs in the next section.

4 Update graphs

In this section (see [9] for proofs) we establish a mapping U, that assigns to an arbitrary graph Y with $\mathbf{v}Y = \{1, \ldots, n\}$ an *update graph*, $U(Y)$, whose vertices are permutations, $(i_1, \ldots, i_n)$, $i_k \in \mathbb{N}_n$. It is straightforward to show that every transposition of consecutive vertices i_k, i_{k+1} with $\{i_k, i_{k+1}\} \notin \mathbf{e}Y$ leaves the product of arbitrary local maps $\prod_{i=1}^n F_{i,Y}$ invariant.

Proposition 1 *Let Y be a subgraph of a regular graph X with $\mathbf{v}Y = \{1,\ldots,n\}$ and $\pi = (i_1,\ldots,i_n)$, $\pi' = (i'_1,\ldots,i'_n)$ two permutations of its vertex set. Suppose there exists a sequence of transpositions $(s_k, s_l, \cdots, s_h)$, where $s_j = (j, j+1)$ and $m+1 = |(s_k, s_l, \cdots, s_h)|$, such that*

$$\begin{aligned} s_k\pi &= (i_1,\cdots,i_{k+1},i_k,\cdots,i_n) = \pi^1 \\ s_l\pi^1 &= (i^1_1,\cdots,i^1_{l+1},i^1_l,\cdots,i^1_n) = \pi^2 \\ &\cdots \\ s_h\pi^m &= (i^m_1,\cdots,i^m_{h+1},i^m_h,\cdots,i^m_n) = \pi' \,, \end{aligned}$$

with $\{i_k,i_{k+1}\},\{i^1_l,i^1_{l+1}\},\ldots,\{i^m_h,i^m_{h+1}\} \notin \mathbf{e}Y$ Then we have

$$\prod_{r=1}^{n} F_{\pi(r),Y} = \prod_{r=1}^{n} F_{\pi'(r),Y} \,.$$

Accordingly, our construction is an adequate framework in which to study the equivalence class problem for sCAs. We next give the formal definition of update graphs $U(Y)$ for arbitrary graphs Y:

Definition 1 *Let Y be a graph with $\mathbf{v}Y = \{1,\ldots,n\}$. The update graph $U(Y)$ over Y has vertex set S_n and two vertices $(i_1,\ldots,i_n)$, $(h_1,\ldots,h_n)$ are adjacent iff*

$$\begin{aligned} \exists j \in \mathbb{N}_{n-1} : (i_j,i_{j+1}) &= (h_{j+1},h_j),\ i_m = h_m,\ \textit{else} \\ \wedge\{i_j,i_{j+1}\} &\notin \mathbf{e}Y \,. \end{aligned}$$

For a graph Y we refer to the graph having $\mathbf{v}Y$ as vertex set and an empty edge set as $Y_\varnothing$. Obviously, $U(Y_\varnothing)$ is a $n-1$ regular graph with diameter $\binom{n}{2}$ and we have $U(Y) < U(Y_\varnothing)$. Further, let $Y < X$ be a subgraph of a regular graph X, and let $C^{(1)}_n, C^{(2)}_n$ be two $U(Y)$-components, then there exists a Y-edge $\{i,k\}$ such that $C^{(1)}_n$-vertices can be written as $\pi_1 = (\ldots,i,\ldots,k,\ldots)$ and $C^{(2)}_n$-vertices $\pi_2 = (\ldots,k,\ldots,i,\ldots)$ respectively. According to (3.6) there exist local maps $F_{Y,i}, F_{Y,k}$ such that $F_{i,Y} \circ F_{k,Y} \neq F_{k,Y} \circ F_{i,Y}$. If, for example, all maps $F_{r,Y}$ are bijective we conclude from this

$$\prod_{r=1}^{n} F_{\pi_1(r),Y} \neq \prod_{r=1}^{n} F_{\pi_2(r),Y} \,.$$

That is, for any two $U(Y)$-components, $C^{(1)}_n, C^{(2)}_n$, there exist local maps such that induced sCA over Y w.r.t. $C^{(1)}_n, C^{(2)}_n$-vertices are different.

Lemma 1 *Let Y be a graph and $U(Y)$ be the update graph, defined as above. Then, $U : \{Y < X\} \longrightarrow \{U(X) < U(X_\varnothing)\}$ is injective.*

The assignment $Y \mapsto U(Y)$ allows us to study the relation between actual dependencies of a family of local maps and the corresponding induced dynamics, without exhaustively enumerating rule space.
A natural and general framework for this question is to assume that $X = K_n$, the complete graph over n vertices, and to consider the random graph $G_{n,p}$. U can be viewed as a random variable over $G_{n,p}$, which induces a random graph $U(G_{n,p})$ as follows

$$U : G_{n,p} \to (\{U(Y)\},\mu_U) \,, \quad Y \mapsto U(Y) \,, \qquad (4.1)$$

where $\mu_U(U(Y)) = \mu_{n,p}(Y)$. According to lemma 1 μ_U is well defined.

Proposition 2 *The following assertions hold*

$$\begin{aligned} &\textit{for } p = 0 \quad \mu_{n,0}\{U(G_{n,0}) \textit{ is connected }\} = 1 \,. \\ &\textit{for } p = 1 \quad \mu_{n,1}\{\mathbf{e}U(G_{n,1}) = \varnothing\} = 1 \,. \end{aligned}$$

These observations lead to the question: what is the evolution of $U(Y)$ as p increases from 0 to 1? For this purpose it is useful to recall an important theorem on Hamiltonian paths in $G_{n,p}$, which was a long standing open problem of Erdős and Rényi [14, 15], solved by Bollobás [4, 3] and independently by Komlós and Szemerédi [12].

Theorem 1 *Suppose $p = \frac{\ln(n)+\ln(\ln(n))+c}{n}$ with $c \in \mathbb{R}_+$, then*

$$\lim_{n\to\infty} \mu_{n,p}\{G_{n,p} \textit{ is Hamiltonian }\} = e^{-e^{-c}} \,. \qquad (4.2)$$

As we will see later, $\frac{\ln(n)}{n}$ is the threshold function for the emergence of a Hamiltonian path *and* connectivity of $G_{n,p}$ [10, 14, 15]. Theorem 1 in particular implies that $p = \frac{c\ln(n)}{n}$ is a threshold function for the existence of isolated vertices in $U(G_{n,p})$.

Corollary 1 *Let $p = \frac{\ln(n)+\ln(\ln(n))+c}{n}$ with $c \in \mathbb{R}_+$, then*

$$\lim_{n\to\infty} \mu_{n,p}\{U(G_{n,p}) \textit{ has an isolated vertex }\} = e^{-e^{-c}} \,. \qquad (4.3)$$

Now we turn to the component-structure of $U(G_{n,p})$ as p tends to 1. Again, we can report a duality between the random graphs $G_{n,p}$ and $U(G_{n,p})$. In this context it is worth recalling the main epochs in the evolution of $G_{n,p}$ [10, 14, 15, 5].

Theorem 2 *Suppose* $p = \frac{c}{n}$ *with* $c \in \mathbb{R}_+$, *then for*

$$c < 1, \quad \lim_{n\to\infty} \mu_{n,p}\{\textit{all components} \leq O(\ln(n))\} = 1,$$
$$c = 1, \quad \lim_{n\to\infty} \mu_{n,p}\{\exists \textit{ a component} \geq O(n^{\frac{2}{3}})\} = 1,$$
$$c > 1 \quad \lim_{n\to\infty} \mu_{n,p}\{\exists \textit{ a component} \geq O(n)\} = 1.$$

For $p = \frac{c\ln(n)}{n}$ *with* $c \in \mathbb{R}_+$

$$\lim_{n\to\infty} \mu_{n,p}\{G_{n,p} \textit{ is connected}\} = e^{-e^{-c}}. \tag{4.4}$$

We can now state first properties of U:

Theorem 3 *For the random graph* $G_{n,p}$ *let* $U(G_{n,p})$ *be the induced random graph. Then* $\forall\ 0 < a < 1$*:*

$$U(G_{n,p}) \text{ has a component of size } \geq \frac{n!}{2^{[1+a]p\binom{n}{2}}} \textit{ a.s.} \tag{4.5}$$

In particular, for $p = o(\frac{\ln(n)}{n})$ *and* $0 < \epsilon < 1$ *we have*

$$U(G_{n,p}) \textit{ has a giant component} \geq [n!]^{1-\epsilon} \textit{ a.s.} \tag{4.6}$$

Proposition 3 *Let the random graphs* $G_{n,p}$ *and* $U(G_{n,p})$ *be given. Then we have*

$$\mu_{n,p}\{\pi, \mathbf{id} \in U(G_{n,p}) \textit{ are connected}\} = (1-p)^{\mathrm{Inv}(\pi)}. \tag{4.7}$$

Note that the random variable $\mathrm{Inv} : S_n \to \mathbb{Z}$ is in the limit Gaussian distributed with $\mathbb{E}[\mathrm{Inv}] = \frac{n(n-1)}{4}$ and $\mathbb{V}[\mathrm{Inv}] \sim \frac{n^3}{36}$ [13]. In view of Proposition 3 this observation could allow us to derive results on $U(G_{n,p})$-components for $p = \frac{c\ln(n)}{n}$ and beyond. A first result in this context analyzes the random variable $C_{\mathbf{id}} : U(G_{n,p}) \longrightarrow \mathbb{Z}$:

$$C_{\mathbf{id}}(U(Y)) = |\{\pi \mid \pi, \mathbf{id} \text{ are connected}\}|. \tag{4.8}$$

Proposition 4 *Let the random graphs* $G_{n,p}$ *and* $U(G_{n,p})$ *be given. Then we have*

$$\mathbb{E}(C_{\mathbf{id}}) = p^{-n} \prod_{i=1}^{n} [1 - (1-p)^i]. \tag{4.9}$$

In particular we have for $p = 1$, $\mathbb{E}(C_{\mathbf{id}}) = 1$ *and for* $p = 0$, $\mathbb{E}(C_{\mathbf{id}}) = n!$.

We finally give a categorical interpretation of U:

Proposition 5 *Let* Y *be a graph and* Y^c *be the graph with* $\mathbf{v}Y^c = \{1, \ldots, n\}$ *and* $\mathbf{e}Y^c = \mathbf{e}K_n \setminus \mathbf{e}Y$. *Let* ϕ *be a surjective graph morphism, then we have the commutative diagram*

$$\begin{array}{ccc} Y_1^c & \longrightarrow & U(Y_1) \\ \downarrow{\scriptstyle \phi} & & \downarrow{\scriptstyle U(\phi)} \\ Y_2^c & \longrightarrow & U(Y_2) \end{array}$$

and for surjective graph morphisms ϕ, ψ *holds*

$$U(\phi \circ \psi) = U(\phi) \circ U(\psi) \tag{4.10}$$
$$U(\mathbf{id}_{Y^c}) = \mathbf{id}_{U(Y)}. \tag{4.11}$$

5 Discussion

An abstraction of computer simulations viewed as compositions of local mappings that produce global system dynamics has been developed. This perspective allows analysis without detailed formal reference to particular local mappings beyond the fact that they can depend on each other. We employ sequential cellular automata (sCA) over random graphs as a paradigmatic framework for elements of a theory of simulation and show that dynamical systems generated by simulations can be classified by equivalence classes on update sequences. Simulation is only beginning to be seen as an area to be developed formally in computer science, but due to the widespread use of computer simulation, the need for development of theoretical foundations is recognized. The main aspects of a theory of simulation we are concerned with here are locality and sequentiality. Our analysis has focused on the relation between the basic dependency graph Y, whose vertices are updated by local maps (local here being defined by the adjacency in Y), and its update graph $U(Y)$ whose components imply certain equivalence classes of sCA over Y. What distinguishes the approach we take here from what we see occurring in the literature might be summarized by our specific intent to exclude any detail from consideration of the local rules aside from dependencies, the use of random graphs to broadly characterize properties of simulation, the focus on generation of dynamics as primary, and the derivation of mimicry from the basic generative considerations. Mathematically, a rather different, but related view can be found in Z. Róka [23], which reveals interesting insights concerning simulations between CA on different Cayley graphs. The conceptual similarity between Róka's approach and ours consists of a recognition of the im-

portance that structural properties of the representation are influential to the generated dynamics and that equivalent systems can be generated over different structures under certain conditions. However, our setting is not restricted to Cayley graphs over finite groups and their morphisms.
We have discussed in detail how our approach relates to [11, 6, 19, 22]. Moreover, it may be of interest that the mathematical properties observed in this analysis of simulation are related to random graph models of sequence to structure mappings in biomolecules [17], and topics in theory of parallel computing [20]. In particular our theory has interesting implications for gene-regulatory networks, modeled by random Boolean networks. Our results show to what extend a specific order matters, according to which certain genes are activated.

Acknowledgments. We want to thank N.A. Baas, Y.C. Chen, S.M. Fraser, J.H. Johnson and G.C. Rota for stimulating discussions. Also thanks to Henning Mortveit and Riko Jakob for helpful suggestions. Special thanks and gratitude to Darrell Morgeson and Al Sattelberger for their support. This research is supported by Laboratory Directed Research and Development under DOE contract W-7405-ENG-36 to the University of California for the operation of the Los Alamos National Laboratory.

References

[1] Wuensche A. and Lesser M.J. *The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata.* Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, 1992.

[2] N. Alon, J. Spencer, and P. Erdős. *The Probabilistice Method.* Wiley Interscience Discrete Math and Optimization, 1991.

[3] Bollobás B., Fenner T.I., and Frieze A.M. An algorithm for finding Hamiltonian cycles in random graphs. *to appear*, 1985.

[4] Béla Bollobás. Almost all regular graphs are Hamiltonian. *Europ. J. Comb.*, 4:97–106, 1983.

[5] Béla Bollobás. *Random Graphs.* Academic press, 1985.

[6] P. Cartier and D. Foata. *Problemes combinatoires de commutation et reárrangements.* Lecture Notes in Math 85, Springer, 1969.

[7] Langton C.G. *Computation at the edge of chaos.* PhD thesis, Univ. of Michigan, 1990.

[8] H. Chernoff. A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[9] Barrett C.L. and Reidys C.M. Elements of a theory of simulation I: sequential CA over random graphs. *Appl. Math. and Comp.*, 1997. in press.

[10] Erdős. Graph theory and probability. *Canad.j.Math.*, 11:34–38, 1959.

[11] Viennot G.X. Heaps of pieces I: Basic definitions and combinatorial lemmas. *Annals New York Academy of Sc.*, pages 542–570, 1986.

[12] Komlós J. and Szemerédi E. Limit distributions for the existence of Hamilatonian cycles in a random graph. *Discr. Math.*, 43:55–63, 1983.

[13] Comtet L. *Advanced Combinatorics.* D. Reidel Publishing Comp., 1974.

[14] Erdős P. and Rényi A. On random graphs I. *Publ. Math. Debrecen.*, 6:290–297, 1959.

[15] Erdős P. and Rényi A. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.*, 5:17–61, 1960.

[16] E. Rasmusen. *Games and Information.* Blackwell, Oxford UK, Cambride US, 1989.

[17] C.M. Reidys, P.F. Stadler, and P.K. Schuster. Generic properties of combinatory maps and neutral networks of RNA secondary structures. *Bull. Math. Biol.*, 59(2):339 – 397, 1997.

[18] Wolfram S. Computation theory of cellular automata. *Commun. Math. Phys.*, 96:15–57, 1984.

[19] R. Stanley. Acyclic orientations of graphs. *Discrete Math.*, 5:171–178, 1973.

[20] Diekert V. *Combinatorics on Traces.* Springer, Lecture Notes in Computer Science: 454, 1990.

[21] von Neumann J. *Theory of Self-Reproducing Automata.* University of Illinois Press, 1966.

[22] Shi J. Y. The enumeration of coxeter elements. *J. of algebr. Comb.*, 6:161–171, 1997.

[23] Róka Z. *Cellular automata on Cayley graphs.* PhD thesis, Ecole Normale Superieure de Lyon, France, July 1994.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Artificial Stock Market

R.G. Palmer, W. Brian Arthur, John H. Holland, and Blake LeBaron
Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501, USA

Abstract

The Santa Fe Artificial Stock Market consists of a central computational market and a number of artificially-intelligent agents. The agents choose between investing in a stock and leaving their money in the bank, which pays a fixed interest rate. The stock pays a stochastic dividend and has a price which fluctuates according to agent demand. The agents make their investment decisions by attempting to forecast the future return on the stock, using genetic algorithms to generate, test, and evolve predictive rules.

The artificial market shows two distinct regimes of behavior, depending on parameter settings and initial conditions. One regime corresponds to the theoretically-predicted rational expectations behavior, with low overall trading volume, uncorrelated price series, and no possibility of technical trading. The other regime is more complex and corresponds to realistic market behavior, with high trading volume, high intermittent volatility (including GARCH behavior), bubbles and crashes, and the presence of technical trading. One parameter that can be used to control the regime is the exploration rate, which governs how rapidly the agents explore new hypothesis with their genetic algorithms. At low exploration rate the market settles into the rational expectations equilibrium. At high exploration rate it falls into the more realistic complex regime. The transition is fairly sharp, but close to the boundary the outcome depends on the agents' initial "beliefs"—if they believe in rational expectations it occurs, and is a local attractor; otherwise the market evolves into the complex regime.

1 Introduction

Today's standard economic theory, general equilibrium theory, or *rational expectations*, says in its shortest statement that agents—traders, firms, individuals, etc—*deduce* their optimum behavior by logical processes from their circumstances. It further assumes that the agents have complete information, that they're perfectly rational, that they have common expectations, and that they know that everyone else has these same properties too.

One of the consequences of this approach is that almost everything is decided at time zero. The agents first work out how the whole future should be, and then the world just plays itself out. There is no dynamics, no learning, and no evolution.

When this rational expectations approach is applied to a stock market [5], it implies that there shouldn't be anything like market moods or psychologies. There shouldn't be bubbles, crashes, or bursts, and volatility should be low. There shouldn't be much trading volume; the only reason one person would trade with another would be if something happened externally, changing the assets available for investment. There shouldn't be money to be made by *technical trading*, simply extrapolating patterns in a time series of price, because any regularity in the price series should have already been arbitraged away by the rational agents.

These ideas don't fit the facts of real stock markets very well. There do seem to be bubbles, crashes, and moods of the market. The volume and volatility are much higher than can be accounted for by external changes. And people on Wall Street do seem to make money by technical trading (see e.g., [4]).

There are some ways to modify the theory or its application to attempt to come to terms with these discrepancies, including a number of ideas of *bounded rationality*, and theories involving *noise traders* [7]. However none of these theories seem wholly convincing [2], and we describe here a very different approach.

In starting this project in 1989 [1], we asked what agents *really* do in markets, and more generally in the world. Our answer then, and now, is that they (a) classify whatever they see; (b) notice patterns; (c) generalize and form internal models, ideas, or rules of thumb; and (d) act on the basis of those internal models. We decided to build a stock market along these lines in the computer, whereby agents would notice the patterns

in the price (and in any other data they had access to), form models, and then trade on that basis.

Of course, the agents have to evaluate and adapt their internal models after seeing how well they work. Actually, each agent has a number of different ways of predicting the future, and is continually evaluating and comparing them. The ones that work well gain more weight and are used more often. The ones that fail are eventually thrown out and replaced.

The agents buy and sell stock in the market, and thereby affect the stock price. What the agents do affects the market. What the market does affects the agents. So the market behavior emerges from the collective behavior of the agents, who are themselves co-evolving.

From an economics viewpoint, the aim of our work is to examine a market of interacting agents that can learn with an open set of possibilities, and see whether it converges to a rational expectations equilibrium or to something else. The core result is that there are two equilibria. The model can show rational expectations behavior, and it can show realistic market behavior, but they are two separate equilibria.

2 Structure of the Market

The basic structure of the model is N agents ($i = 1, 2, \ldots, N$) interacting with a central market. Typically $N = 50$–100. There may be several types of agents. In contrast to many other interacting agent models, the agents don't interact directly with each other, but only via the market.

In the market there is a single stock, with price $p(t)$ per share at time t. Time is discrete ($t = 1, 2, \ldots$); period t lasts from time t until $t + 1$. The stock pays a dividend $d(t+1)$ per share at the end of period t. The dividend times series $d(t)$ is itself a stochastic process defined independently of the market and the agents' actions. We normally use a simple random process with persistence called an AR-1 or Ornstein-Uhlenbeck process, given by

$$\hat{d}(t+1) = \rho\hat{d}(t) + \alpha\eta(t)$$

where $\hat{d}$ means the offset of the dividend from a fixed mean $\bar{d}$, so $d(t) = \bar{d} + \hat{d}(t)$, ρ and α are parameters, and $\eta(t)$ is a Gaussian random variable, chosen independently at each time t from a normal distribution with mean 0 and variance 1.

There is also a fixed-rate asset, *the bank*, which simply pays a constant rate of return r per period. The agents have to decide how much money they want to put into the stock (which has a fixed total number of shares—if somebody buys, someone else has to sell), and how much they want to leave in the bank. At any time t, each agent i holds some number of shares of stock $h_i(t)$ and has some amount of cash $M_i(t)$ in the bank. Its total wealth is then

$$w_i(t) = M_i(t) + h_i(t)p(t).$$

At the end of the period, one time step later, this portfolio becomes worth

$$\bar{w}_i(t+1) = (1+r)M_i(t) + h_i(t)p(t+1) + h_i(t)d(t+1),$$

where the three terms are the money in the bank, with interest, the new value of the stock, and the dividend pay-out. $\bar{w}(t+1)$ and $w(t+1)$ are not the same, because trading occurs in between, potentially moving assets between the bank and the stock.

The trading process is managed by a *specialist* inside the market. The specialist also has the job of setting $p(t+1)$. Its fundamental problem at each time step is that the number of bids to buy and offers to sell may not match, and yet the total number of shares of stock is fixed. We have explored several approaches to this issue, including rationing of bids or offers [6], having the specialist maintain a buffering inventory itself, and holding an auction in which the price at a given time is adjusted until the bids and the offers match closely. Only the last approach, an auction, is described further here. If there are more bids than offers, then the price is raised, so the bids drop and the offers increase, until they match closely.

One more thing that is defined at the level of the overall market structure is the information that is made available to the agents for use in their decision making. In principle this information set (which we call *the world*) consists of the price, dividend, total number of bids, and total number of offers, at each past time step. There are other variables that we have tried adding too, including a predictor of the future dividend (which can be done in the computer, by running the stochastic process forward, but not in the real world!), and a random "sunspot" variable around which the agents might coordinate their actions.

However, we usually condense most of this information into a string of *world bits*. At any given time, the world that the agents see consists of a string of 80 or so bits, and some recent price and dividend information. Some examples of these bits, each of which is either *true* or *false* at each time t, are as follows:

1. $rp(t)/d(t) > \frac{3}{4}$
2. $rp(t)/d(t) > 1$

3. $rp(t)/d(t) > \frac{5}{4}$
4. $p(t) > \mathrm{MA}_{10}\{p(t)\}$
5. $p(t) > \mathrm{MA}_{100}\{p(t)\}$
6. $p(t) > \mathrm{MA}_{500}\{p(t)\}$
7. Always *true*

Here $\mathrm{MA}_n\{p(t)\}$ means a moving average over the most recent n steps of $p(t)$, i.e.,

$$\mathrm{MA}_n\{p(t)\} = \frac{1}{n}\sum_{k=0}^{n-1} p(t-k)$$

The quantity $rp(t)/d(t)$ would be 1 in a simple equilibrium notion of fundamental value, so the deviation from 1 gives a sense of how much the stock is underpriced or overpriced.

We classify bits into 3 categories: technical, fundamental, and control. Technical bits, by definition, just depend on the past price series, and are the only ones that a strict technical trader would use. Bits 4–6 in the above list are technical bits. Control bits are useless ones that we include as experimental controls, like bit 7 in the above list. Fundamental bits are anything else, generally involving the dividend series in some way. Bits 1–3 above are fundamental.

3 Structure of the Agents

Fundamentally the agents have to decide whether to invest in the stock or the bank. If at any time step, they conclude that they want to invest more in the stock than previously, then they submit a bid to buy more shares. Conversely, they may submit an offer to sell shares.

We have examined many types of agents, and our software can mix different types in the same market (see [6] for a description of *condition-action agents*). But this paper only treats *forecasting agents* that use number of *predictors*, each of which attempts to predict the future return (price plus dividend). By seeing how well their predictors work, the agents can estimate their accuracy (prediction variance) and update or replace poor ones. Because they know the variance of their overall predictions, the agents can also perform a risk aversion analysis called CARA, constant absolute risk aversion. This is a standard computation, based on an exponential utility function and used in portfolio analysis, that gives an optimal division of funds between two possible assets when the mean and variance of the expected return is known for each asset. If agent i's estimate of the mean return is $E_i[p(t+1)+d(t+1)]$ with variance σ_i^2, then under CARA (and an additional Gaussian assumption) the ideal number of shares to hold is given by

$$h_i^{\mathrm{desired}}(t) = \frac{E_i[p(t+1)+d(t+1)] - p(t)(1+r)}{\lambda\sigma_i^2}$$

where λ is a parameter, the degree of relative risk aversion.

The agents' predictors actually consist of two parts, a *condition part* and a *forecast part*. The condition part determines when each particular predictor is *activated*, as explained below. Only activated predictors produce forecasts, using their forecast part. The forecast part, in the simplest case, is just a linear rule

$$E_{ij}[p(t+1)+d(t+1)] = a_{ij}\big(p(t)+d(t)\big) + b_{ij}$$

where $E_{ij}[]$ means the expected (predicted) value for agent i's j'th predictor, and a_{ij} and b_{ij} are the coefficients that constitute the forecast part of this predictor. Although this is itself a very simple linear form, the condition parts make the overall prediction only piecewise linear.

Every time a predictor is activated, the agent checks to see how well it performed when the period is over. This is used to maintain a variance σ_{ij}^2 for each predictor, as a weighted moving average of its past squared errors.

There are several ways to combine a set of predictions and variances, $E_{ij}[p(t+1)+d(t+1)]$ and σ_{ij}^2, for each activated predictor j, into the single overall forecast and variance, $E_i[p(t+1)+d(t+1)$ and σ_i^2, needed for the CARA calculation. The simplest, used for all the results described here, is to use the currently best predictor, the one with the smallest σ_{ij}^2 across all activated j's.

The condition part of each predictor is implemented with a *classifier system*, in which the condition part is represented by a ternary string of the symbols $\{0, 1, \#\}$, one for each of the world bits that the agent can observe (we can restrict agents to see only a subset of all the world bits). A condition symbol of `0` means that the corresponding world bit must be *false* for the condition part to match, while conversely `1` requires *true*. A condition symbol of `#` is a *don't care*, and matches either *true* or *false*. For example, the condition string `##1###0#` matches the world bits `01110100` (where `0` stands for *false* and `1` for *true*), but not `10100110`.

Some of an agent's predictors may give good predictions when they are activated, while others may not.

A genetic algorithm is used to adjust and evolve a better set of predictors. For each agent at each period we run the genetic algorithm with probability $1/K$, where K is a parameter. The genetic algorithm eliminates some of the worst predictors, those that have the highest variance, and generates some new ones to replace them. Typically we replace 20 out of 100 predictors.

To generate new predictors we first clone some of the best existing ones in the current population. Then we either perform *mutation* or *crossover* (or sometimes both) on those cloned predictors. Mutation means changing a few condition bits, and modifying the a_{ij}'s and b_{ij}'s by a random amount. We use parameterized distributions of such changes; see [2] for details. Crossover means selecting two parent predictors, taking some condition bits from each, and interpolating their a_{ij}'s and b_{ij}'s. It is not clear whether crossover has any positive effect beyond causing large jumps in the space of condition bits. We also sometimes perform *generalization*, changing some of the fixed bits (0's and `1's`) to don't cares (`#`'s) in predictors that have not been activated for a long time.

In choosing predictors for replacement and cloning, we mainly select according to variance; low variance means high fitness. But we also impose a small penalty for each bit that is not a `#`, giving a little pressure not to condition the predictors on too many bits.

4 Results

Our main initial goal in this project was to look for realistic market behavior. We asked in particular whether the price series of our stock would look like a real stock price series. We realized that goal several years ago now, finding reasonably realistic market behavior as shown by time series analysis.

In 1995, we started a second phase of experiments by asking whether our agents could also show rational expectations behavior. If homogeneity is assumed, so that all the agents have the same beliefs, then the rational expectations equilibrium for this market can be computed, and has a simple form in which the price is linear in the dividend. Thus rational expectations behavior is within the framework of our agents' linear forecasts.

We tried giving the agents initial beliefs in the rational expectations result by setting the initial conditions for a_{ij} and b_{ij} to the calculated rational expectations values. We found that they stayed there, and that the rational expectations equilibrium is in fact a local attractor—when we started the agents initially fairly close to it, they went to that state. It resulted in a very stable market, just as the theory says, with very little trading going on, and homogeneous agent behavior.

On the other hand, when we started the agents with almost any other conditions, they never settled into rational expectations behavior even after millions of periods, and the system behaved much more like a real market. Thus we had found two regimes of behavior, which we call the "rational expectations regime" and the "complex regime" respectively.

In the rational expectations regime, we see relatively low trading volume, with very little information in the price series that can be exploited for prediction. The forecast parameters—the a_{ij}'s and b_{ij}'s—all converge to be the same; the agents end up becoming homogeneous. The technical bits are not useful, and are dropped from use.

In the complex regime, we find that the agents remain heterogeneous and continuously co-evolve. One of the tests we did was to take a successful agent out of the market, freeze it, and then reinsert it thousands of periods later. We found that it didn't do well at all. Even though the market looks statistically much the same, an agent that was trained in one period doesn't work well later, because the detailed information that it is picking up in its bits and forecasts is changing over time.

The trading volume remains much higher in the complex regime. It varies greatly, and has GARCH behavior. It is auto-correlated, and there are correlations between volume and volatility. These are all features found in real markets. There are sometimes bubbles and crashes, fairly minor ones usually, and over-reactions. And the agents *do* use the technical bits, despite the small cost to do so.

More recently, in a third phase of experiments reported in [2], we looked at what happens as we change certain parameters. We found that we can force the market into either regime with appropriate parameter values, using the same random initial conditions. But with intermediate parameter values, the initial conditions dictate which regime is reached, as in the second phase of experiments.

In most experiments, we varied just the parameter K that dictates how often the genetic algorithm is run. That controls how often the agents explore new possible ways of predicting the future. We frequently compared two values, $K = 250$ and $K = 1000$, that we call *fast exploration* and *slow exploration* respectively. Fast exploration puts us in the complex regime, while slow exploration gives the rational expectations

regime. We always used random initial conditions in these experiments.

An example of our time series analysis was a fit of the price series to a linear recurrence relation

$$p(t+1) = A + Bp(t) + \varepsilon(t+1)$$

where $\varepsilon(t)$ represents the residual variation after fitting the best values for A and B. In a rational expectations equilibrium the residuals $\varepsilon(t)$ ought to be independent identically-distributed Gaussian variables, because the price is driven by the AR-1 dividend series. So we tested the $\varepsilon(t)$ series for normality and correlations. The following table shows some results for fast and slow exploration, from averages over 25 runs of each case.

	Fast Exploration	*Slow Exploration*
σ	2.147 ± 0.017	2.135 ± 0.008
κ	0.320 ± 0.020	0.072 ± 0.012
ρ_1	0.007 ± 0.004	0.036 ± 0.002
$\rho_i^{(2)}$	0.064 ± 0.004	0.017 ± 0.002

The first line, σ, shows the standard deviation of $\varepsilon(t)$. The second line, κ, shows the excess kurtosis, a measure of non-Gaussian behavior based on the fourth moment. Fast exploration gives much larger deviations from normality, in the direction of the "fat tails" seen in real market data. The third and fourth lines show two measures of single-step autocorrelation, $\rho_1 = \langle \varepsilon(t)\varepsilon(t+1) \rangle$, and $\rho_1^{(2)} = \langle \varepsilon(t)^2 \varepsilon(t+1)^2 \rangle - \sigma^4$.

The ARCH(1) test [3], gives about 37 for the fast exploration case, compared to 3.2 for the slow exploration case.

This sort of analysis can be extended by including additional terms in the recurrence relation. For example to test whether the $rp(t)/d(t) > \frac{3}{4}$ technical bit is actually of any use in predicting the price series, we write

$$p(t+1) = A + Bp(t) + CI_{rp/d>\frac{3}{4}}(t) + \varepsilon(t+1)$$

where $I_{rp/d>\frac{3}{4}}(t)$ is 0 or 1 as appropriate at each t. We then fit the coefficients as before. In this case we found $C = -0.44 \pm 0.10$ for the fast exploration case, compared to $C = 0.05 \pm 0.09$ for the slow exploration case. Thus this bit is useful in the fast but not the slow case.

There is much more yet to explore with this model. Given a whole computational market in the computer, we can experiment with what happens upon changing the market structure, the specialist, the dividend series, and so on. Some other future plans include:

1. Multiple stocks. It is not clear whether introducing more than one stock will fundamentally change our results, but the experiment seems worthwhile.

2. Impact of wealth. As the agents get more wealthy, they don't actually have more influence in the market under our CARA assumptions. There are several ways to change those assumptions.

3. Improved prediction. There are many ways to improve our agents' prediction methods. We have experimented briefly with neural network predictors, but did not find significantly different results.

4. Transition details. The transition between the two behavior regimes deserves detailed study. Is it really a sharp transition, or is it gradual? What are the sizes of the basins of attraction of the two regimes? How do they scale with the number of agents, and with other parameters?

5. Information control. It is possible to give different agents different information sets and thus explore the effect of private information. We can also provide information that is released periodically to all agents, like news reports.

6. Strategic behavior. We can have a longer time horizon so that the agents can look further ahead, rather than just one period at a time. Then we need to allow the agents to have strategic behavior over multiple periods.

Acknowledgements

We thank Paul Tayler and Brandon Weber for their collaboration in this project. This work was supported in part by the Santa Fe Institute's Economics Program, including grants from Citicorp, Coopers and Lybrand, McKinsey and Company, the Russell Sage Foundation, and the Walker Foundation; and by core funding to the Santa Fe Institute from the John D. and Catherine T. MacArthur Foundation, the National Science foundation, and the U.S. Department of Energy; and by gifts and grants from individuals and members of the Institute's Business Network for Complex Systems research.

References

[1] W.B. Arthur, "On Learning and Adaptation in the Economy," Santa Fe Institute Paper 92-07-038, 1992.

[2] W.B. Arthur, J.H. Holland, B. LeBaron, R.G. Palmer, and P. Tayler, "Asset Pricing Under Endogenous Expectations in an Artificial Stock Market," pp. 15–44 in *The Economy as an Evolving Complex System II*, edited by W.B. Arthur, D. Lane, and S.N. Durlauf, Addison-Wesley, 1997.

[3] T. Bollerslev, R.Y. Chou, N. Jayaraman, and K.F. Kroner, "ARCH Modeling in Finance: A Review of the Theory and Empirical Evidence," *Journal of Econometrics*, Vol. 52, pp. 5–60, 1990.

[4] J.A. Frankel and K.A. Froot, "Chartists, Fundamentalists, and Trading in the Foreign Exchange Market," *AEA Papera and Proceedings*, Vol. 80, pp. 181–185, 1990.

[5] R.E. Lucas, "Asset Prices in an Exchange Economy," *Econometrica*, Vol. 46, pp. 1429–1445, 1978.

[6] R.G. Palmer. W.B. Arthur, J.H. Holland, B. LeBaron, and P. Tayler, "Artificial Economic Life: A Simple Model of a Stockmarket," *Physica D*, Vol. 75, pp. 264–274, 1994.

[7] A. Shleifer and L.H. Summers, "The Noise Trader Approach to Finance," *Journal of Economic Perspectives*, Vol. 4, pp. 19–33, 1990.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Virtual engineering and complex system

J. Doyle
Control and Dynamical Systems California Institute of Technology
107-81, Pasadena, CA 91125, USA
+1-626-395-4808, E-mail: doyle @ cds.caltech.edu

Abstract

Perhaps the most ubiquitous trends in science and engineering involve an increased emphasis on 1) complexity: including higher levels of automation and integration in the systems we build and study, and 2) virtuality: use of computer-based modeling, analysis, and simulation together with supporting information technology, including graphics and visualization. While these trends are widely discussed and promoted the details are often badly misunderstood.

Complexity in engineering and biological systems fundamentally arise for the same reason: they have been designed to operate in an uncertain environment, and must be far more robust than the components from which they are built. In the design of engineering systems, we are now typically less limited by our ability to fabricate materials and components, and more limited by our ability to manage and control the complexities and uncertainties of highly interconnected, heterogeneous systems. It is almost a cliche that engineering and biology are becoming more alike. What is less well-known, and is the focus of this talk, is how the issues of robustness and uncertainty are the key to understanding complexity, and the theoretical and computation tools required to exploit this understanding.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Intelligent Robotic System
- Adaptation, Learning and Evolution -

Toshio FUKUDA

Center for Cooperative Research in Advanced Science and Technology,
Dept. of Mechano-Informatics and Systems &
Dept. of Micro System Engineering, Nagoya University
1 Furo-cho, Chikusa-ku, Nagoya 464-01, Japan

Naoyuki KUBOTA

Dept. of Mechanical Engineering
Osaka Institute of Technology
5-16-1 Omiya, Asahi-ku, Osaka 535, Japan

Abstract - Living creatures have evolved and formed ecological system by adapting to their dynamic environment. Robots also need the adaptability to dynamic environment. This paper presents methodologies for adaptation, learning and evolution of robotics. Further, the intelligence of a robot emerges as the result of the synthesis of simultaneous processing of perception, decision making and action. A robotic system requires the whole structure of intelligence, and acquires skill and knowledge through the interaction with dynamic environment by computational intelligence including neural network, fuzzy system and genetic algorithm.

Keywords: Intelligent Robotics, Motion Planning, Learning, Structured Intelligence

1. Introduction

Recently, intelligent robotic systems are often required in critical and dangerous areas for human beings. To accomplish given tasks in such areas, robots collect or receive information concerning its environment, and make decisions and actions to the environment by itself. Consequently, robotic systems require capabilities such as inference, learning, planning and prediction which have been discussed as artificial intelligence (AI)[1]. The intelligence as a whole robotic system emerges from the close linkage of the capabilities. Computational intelligence including neural networks (NN), fuzzy systems (FS) and genetic algorithms (GA), have been applied to realize intelligence on the robotic systems[3-6]. Each technique plays the specific role in intelligent robotic systems. NN has been applied for nonlinear mapping of objective problems[2]. FS has been applied for representing human linguistic rules[5]. GA has been applied for solving optimization problems[7].

In addition, behavior-based AI (BBAI) including reactive behaviors, has been discussed as learning methods dependent on environmental information[1,8]. This kind of reactive behavior is also know as sensor driven behavior. A robotic system with sensor driven behavior makes action according to the local information of the environment, and therefore, it is easy to fit with the dynamic environment. The BBAI stresses the importance of the interaction between robot and environment, while classical AI is based on the representation and manipulation of explicit knowledge. Recently, behavior analysis, model-based learning and others, have been proposed [9,10]. However, robotic systems require structured intelligence to obtain higher intelligence as a whole system. Therefore, we have proposed robotic system with structured intelligence[11].

This paper is organized as follows. Section 2 discusses adaptation, learning and evolution for intelligent systems, and section 3 proposes a robotic system with structured intelligence.

2. Intelligent Robotic System

2.1. Adaptation, Learning and Evolution in Robotics

This subsection discusses intelligent system from the viewpoints of evolution, adaptation and learning. Human beings have evolved and formed ecosystem under various environments by adapting to dynamic environment, by searching for livable environment and by rebuilding environment. From the evolutionary points of view, the living creatures have the following characteristics: 1) genetic information to be inherited, 2) interaction between genetic information and 3) structure emerged by the interaction (Fig.1). Living creatures have many interactions in various levels such as organs, cells and genes. Basically, living creatures have evolved by change of genetic information and natural selection by competition between individuals, or between species.

In addition, living creatures can hear, see, smell and touch for perceiving their environments from the viewpoint of adaptation. The information acquired by sensing from their environments is very important for surviving under their dynamic environments. By using the acquired information, living creatures can adapt to their dynamic environments. It is generally recognized on the engineering that '*adaptation*' is defined as change over

short time, and '*evolution*' is defined as change over long time. However, this paper defines *adaptation* and *evolution* as follows.

- Adaptation is to improve the condition through the interaction with environment. Adaptation is evaluated under the evaluation criterion and environmental factors (Fig.2.a).

$$Evaluation = f_{criterion}(form \circ environment) \quad (1)$$

- Evolution is to maintain diversity of form and to change the form globally. Evolution is evaluated concerning form (Fig.2.b).

$$Evaluation = f_{criterion}(form) \quad (2)$$

For example, it is often said that a form evolved, when a simple form changed into a more complex one. Here the form includes figure, pattern, function and organization. Adaptation emphasizes the improvement, while evolution emphasizes the change. The driving force for adaptation and evolution is mainly learning and inheritance (Fig.3). Learning is a process of gaining knowledge or skill. Adaptation includes improvement by learning and simple adjustment. In general, adaptation is used with learning on the engineering, and the learning is fundamentally performed as necessary change based on error function and others. On the other hand, inheritance is a process of receiving characteristics from the previous form. In nature, mutation rarely changes genetic information as transmitting errors, and as a result, the phenotype of a living creature also changes. Consequently, evolution can be defined as resultant/accidental change by the inheritance from the previous one. Also, evolutionary change has a possibility to change the form globally. Therefore, evolutionary concept is used for global search in robotics.

Learning includes learning by discovery, learning by show from observers, and reinforced learning through the interaction with the environments. In the case of human being, the acquired knowledge can be inherited from parents to children by communicating information such as teaching and showing. The linguistic information processing enables the easy inheritance of knowledge to offspring. Especially, symbolic information processing is easy to logically understand. However, human being does information processing unconsciously, and the most of human information processing is intuitive information processing on the human brain. The logical information processing enables the exact inference, but is weak at vagueness, fuzziness and paradox. On the other hand, the intuitive information processing can deal with fuzzy, vague and incomplete information, but is weak at logical inference and synthesis. Thus, human being overcomes their disadvantages by integrating the logical and intuitive information processing on the brain at the same time.

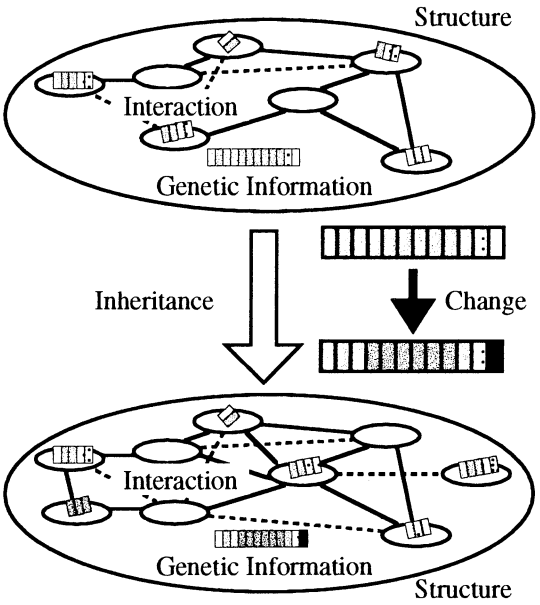
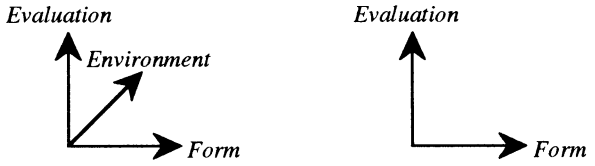


Fig.1 Structure emerging through the interaction between genetic information



(a) Adaptation (b) Evolution

Fig.2 Evaluations for adaptation and evolution

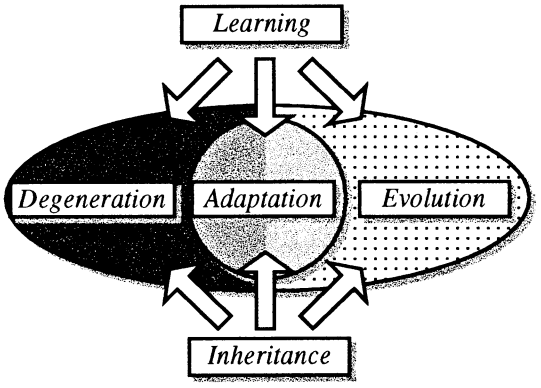


Fig.3 Evolution and adaptation by learning and inheritance

Consequently, the whole structure of information processing is very important.

2.2. Intelligent technologies for Robotics

The recent research fields concerning intelligence and life, include brain science, soft computing, artificial life and computational intelligence[1-7]. Here we describe them briefly.

(1) The brain science: Its aim is to understand the biochemical and physical mechanism of human brain and to construct a highly interconnected neural

network like human brain[1-4].

(2) Soft computing: Soft computing proposed by Zadeh, is a new concept for information processing, and its objective is to realize a new approach for analyzing and creating flexible information processing of human being such as sensing, understanding, learning, recognizing and thinking[5].

(3) Artificial life (A-life): A-life means 'life made by humans rather than nature[6]. A-life has three types of approaches: wetware system from the molecular level, software system from the cellular level and hardware system from the organism level.

(4) Computational Intelligence: In the strictest sense, computational intelligence depends on numerical data and does not rely on knowledge[3,4]. Bezdek discussed intelligence from the three level: artificial, biological, and computational.

Based on the above discussion, we also summarize this. CI aims to construct intelligence from the viewpoints of biology, evolution and self-organization. CI tries to construct intelligence by internal description, while classical AI tries to construct intelligence by external (explicit) description.

The human brain processes information super-quickly and super-accurately like a network. NN and FS are based on the mechanism of human brain. NN simulating the biological brain can be trained to recognize patterns and to identify incomplete patterns. The neurodynamics plays the role of non-linear mapping from input to output. NN is composed of many interconnected neurons with input, output, synaptic strength and activation. NN has applied for system identification, control and pattern matching (mapping from environmental information to conceptual information). While NN simulates physiological features of human brain, fuzzy theory simulates psychological features of human brain. Fuzzy theory[3-5] expresses a degree of truth, which is represented as a grade of a membership function. The fuzzy logic is a powerful tool for uncertainty and incompleteness. Fuzzy controller has been applied for various systems that are complicated, ill-defined and time-varying.

In addition, the GAs provide the evolution mechanism for population dynamics, robot society and A-life[6]. From the view point of simulated evolution, GAs can maintain the genetic diversity in a population to adapt to dynamic environment. Therefore, GAs are often called adaptive systems. GAs have been applied for parameter identification and structural optimization. In addition, reinforcement learning basically optimizes state - action pair or sensory information - action pair according to success or failure from the environment. The reactive models based on the reinforcement learning can perform on-line learning through the interaction with the environment. Temporal difference learning is applied for solving delayed-reward prediction problems.

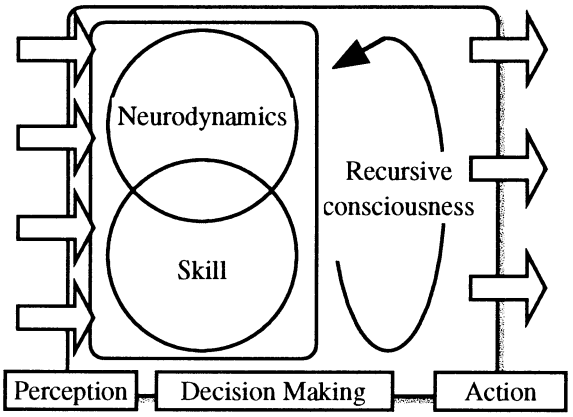


Fig.4 Architecture of structured intelligence

To realize higher intelligence on the robotic systems, the emerging synthesis of various techniques is required as a whole system. Nuero-fuzzy computing has developed for overcoming their disadvantages. In general, the neural network part is used for its learning and classifying, while the fuzzy logic part is used for inference and crisp output. GA is applied for tuning and structure optimization of NN and FS. To summarize, an intelligent system can adapt quickly to dynamic environment by NN and FS with back propagation method and delta rule, and furthermore, the structure of intelligent system can evolve globally by GA.

3 Robotic System with Structured Intelligence

3.1 Structured Intelligence

The intelligence of robot system depends on the architecture of hardware and software as a whole system. This section proposes a robotic system with structured intelligence constructed from the viewpoint of cognitive psychological features. The structured intelligence basically has three features: 1) neurodynamics, 2) skill and 3) recursive consciousness[11] (Fig.4). As mentioned before, human being uses the logical and intuitive information processing on the brain at the same time.

The skill part is defined as generalized knowledge and motion. To represent skill, we apply fuzzy inference system as the logical information processing. Consequently, the skill is explicit representation of human intelligence. The neurodynamics is used as the intuitive information processing. Nonlinear property of NN has been applied for representing various relationships to be learned. NN makes intuitive output based on the learned relationship and human intelligence often needs the intuitive outputs to generate new output. Various outputs rise and fall in human mind. However, the

nonlinear property sometimes makes ineffective output. To control ineffective outputs, we need meta-level controller, that is, recursive consciousness (see Fig.4). The recursive consciousness simulates the high level of human mind, and plays the roles of binding some outputs and selecting output. We need two types of evaluations for binding some output at least. The first one is external evaluation through the interaction with environment. The other is internal evaluation for binding output recursively from the meta-level. Evolutionary concept is useful for generating output. GAs are applied for generating output.

In the perception level, the structured intelligence receives input (sensory information) from environment (Fig.4). Each sensor has specific sensing range and threshold, and makes output based on them. Therefore, we propose sensory network (Fig.5). A sensor is fired by sensory input and other sensors' output, and the sensing range and threshold are changed through the interaction. Thus, the sensory network outputs qualitative information to the structured intelligence. Based on the input (qualitative information), NN and fuzzy system make various outputs (candidate outputs) in parallel. Next, GA generates final outputs based on the candidate outputs. After that, the NN and fuzzy system are trained by learning based on the generated output and its evaluation from the environment. In this way, NN, Fuzzy and GA play the roles of intuitive inference, deductive inference and generation of final output, respectively. In fact, when we solve a problem or generate an output, NN and Fuzzy system limit the solution space a priori, and GA searches the limited solution space for the best solution[11].

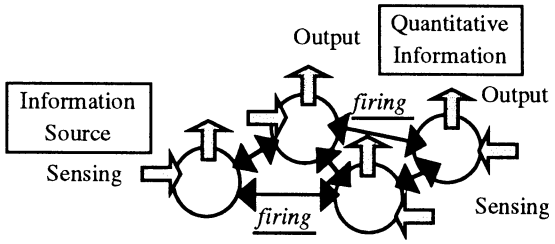


Fig.5 Sensory network for perceiving environment

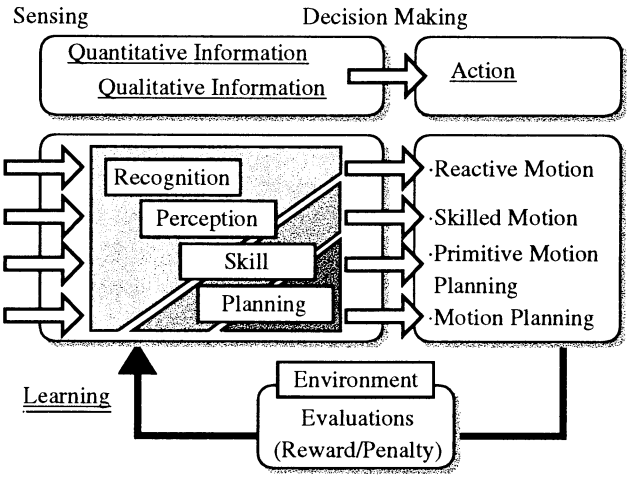


Fig.6 Architecture of a robot with structured intelligence

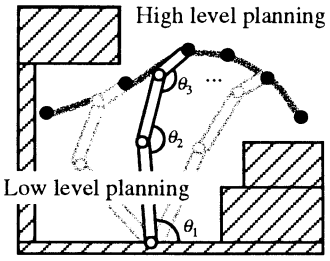


Fig.7 Hierarchical trajectory planning for robot manipulator

3.2. Structured Intelligence for Robotic System

An intelligent robot requires close linkage of perception, decision making and action. Figure 6 shows the architecture of a robotic system with structured intelligence. A robot receives sensory information, and recognizes quantitative information of the environment by the sensory network (Fig.6). Next, the robot perceives its external environment through the interpretation into qualitative information by sensor fusion/integration and focus/release of selective attention. Based on perceptual information from environment, the robot makes action from four levels in parallel (Fig.6). The action levels are reactive motion, skilled motion, primitive motion planning and motion planning. When the robot recognizes dangerous quantitative information from environment, the robot makes reactive motion (reflex). This action level requires no decision making. In the skilled motion level, the robot performs fundamental motion such as locomotion and transportation based on the simple decision making. In addition, when the robot has no skill, the robot generates new motion. If the robot can easily combine its simple skill, the robot generates its new motion by combining skills. However, if the robot has no motions and skill, the robot generates new motions by the motion planning based on inverse kinematics and inverse dynamics. Thus, a robot has no skill at initial state, but gradually acquires skill and motion through the interaction with environment.

3.3 Redundant Manipulator with Structured Intelligence

The robot basically generates new motions by GA. We have proposed various trajectory planning methods for redundant manipulators[11]. One of them is hierarchical trajectory planning composed of configuration generation as low level planning and trajectory generation as high level planning (Fig.7). The hierarchical trajectory generation method for intelligent robots is based on the concept of external and internal evaluations (Fig.8). The external evaluation is basically used for generating

primitive motions (configurations) through the interaction with environment. The internal evaluation is a criterion for binding the robot's primitive motions recursively. This criterion dynamically changes according to the total evaluation, that is, the criterion for selecting a configuration becomes high, as the best trajectory gives high performance. In addition, the total evaluation for evaluating trajectories is formed by the recognition part for detecting dangerous state of the robot manipulator. This means that the conditions for reactive motion against dangerous states are the constraints for dangerous states in the motion planning.

To achieve a given task, the robot manipulator, in, general, performs the following steps: (1) find obstacles (perception), (2) generate a collision-free trajectory (decision making), and (3) trace the trajectory actually (action). Here we focus on the decision making with hierarchical trajectory planning, and next, consider learning based on the generated motions. The hierarchical trajectory planning can easily generate a collision-free trajectory by combining some configurations of the redundant manipulator (Figs.7 and 9). A trajectory is represented as a series of configurations ($\theta i,t$) where i denotes joint number and t denotes descrete time. The low level planning generates collision-free configurations as primitive motions. The high level planning generates a collision-free trajectory binding some intermediate configurations generated in the configuration generator. In fact, the best trajectory is evaluated by internal simulator.

The best trajectory is used for the learning of the primitive motions by using NN. Figure 10 shows the architecture of primitive motion generation and motion planning. First, a task is given to the redundant manipulator. Next, NN makes outputs of a series of configurations as a trajectory ($\theta init$) for performing the given task. GA generates initial candidate solutions of trajectories according to the outputs from NN. Through GA optimization, motion planning is performed until satisfying the aspiration level for evaluating the feasibility of trajectory generation. Finally, GA outputs the best trajectory (θ *), and simultaneously, the NN is trained by the backpropagation method according to θ *. The input variables are the target direction and current configuration of θ i,t and the output variables are $\Delta\theta$ i,t. The initial trajectory is generated according to $\Delta\theta$ i,t.

3.4 Simulation Results of Motion Planning and Learning

This subsection shows the simulation results of motion planning and motion learning. NN learns the relationship of initial configuration of a redundant manipulator and a series of intermediate configuration to the final configuration according to a given task. We give a same task 5 times (trials) shown as Fig.11. The objective is to

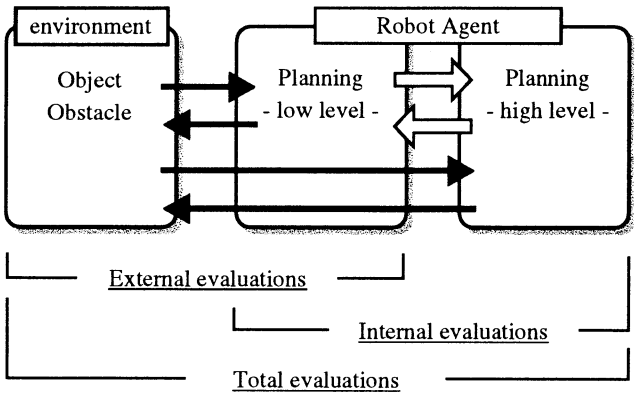


Fig.8 External evaluation and internal evaluation for motion planning

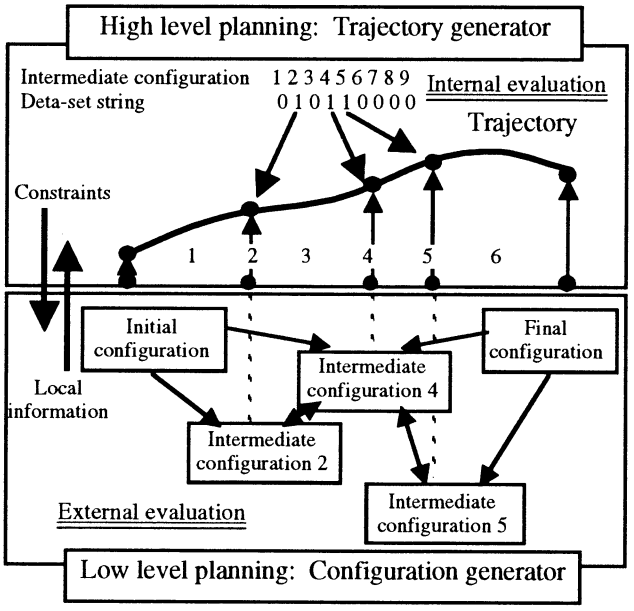


Fig.9 Hierarchical trajectory planning

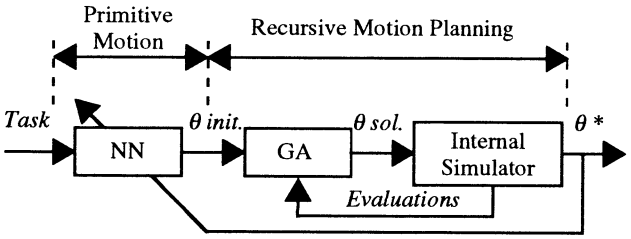
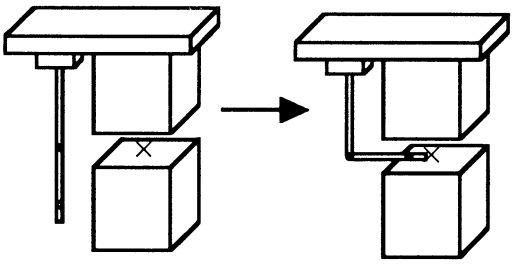


Fig.10 Architecture of primitive motion generation and motion planning

generate a collision-free trajectory to reach the final configuration. Figure 12 shows the fitness value of the best trajectory against the evaluation times in the trajectory generator. In the first trail, NN has no primitive motions, the initial candidate solutions of trajectories are no good. Therefore, GA must generate a trajectory without primitive motions. Consequently, GA requires many evaluation times to generate feasible trajectories. Next, NN learns a series of intermediate configurations of the best trajectory where the learning times is 100



(a) Initial configuration (b) Final configuration

Fig.11 Simulation example of 7 DOF manipulator

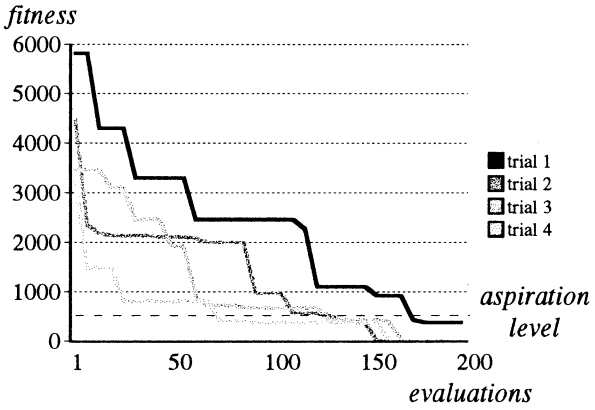
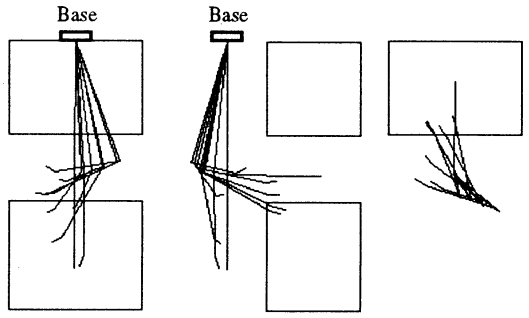
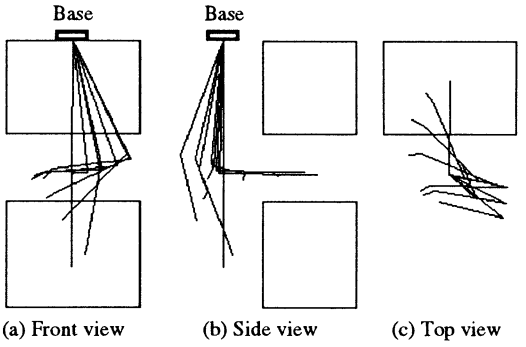


Fig.12 Simulation results of GA optimization



(A) the 4th trial



(B) the 5th trial

Fig.13 output of NN in the 4th and 5th trials

(relatively short). Consequently, NN can not completely learn them. Figure 13 shows the output of 4th trial and 5th trial from the NN, respectively. The side view in each figure shows the collision with the obstacles. In the 4th trial, the redundant manipulator collides with obstacles, but in the 5th trial, the redundant manipulator avoids collision with obstacles. Therefore, GA optimization is not required in the 5th trial, since NN makes a collision-free trajectory correctly.

4. Summary

This paper proposed a robotic system with structured intelligence constructed from the viewpoint of cognitive psychological features. Simulation results show that the learning by NN and motion planning by GA can be simultaneously performed against a given tasks. In future subject, we must develop the sensory network for perception and apply the proposed system to a real robots.

References

[1] Russell S., Norvig P. (1995) Artificial Intelligence. Prentice-Hall, Inc.

[2] Anderson J., Rosenfeld E. (1988) Neurocomputing - Foundations of Research. The MIT Press

[3] Zurada J., Marks II R.J., Robinson C.J. (1994) Computational Intelligence - Imitating Life. IEEE Press

[4] Palaniswami M., Attikiouzel Y., Marks II R., Fogel D., Fukuda T. (1995) Computational Intelligence - A Dynamic System Perspective. IEEE Press

[5] Jang J., Sun C., Mizutani E. (1997) Neuro-Fuzzy and Soft Computing. New Jersey, Prentice-Hall, Inc.

[6] Langton C. (1995) Artificial Life -An Overview. The MIT Press.

[7] Fogel D. (1995) Evolutionary Computation. IEEE Press

[8] Brooks R. (1986) A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, RA-2-1, pp.14-23

[9] Colombetti M., Dorigo M., Borghi G. (1996) Behavior Analysis and Training - A methodology for Behavior Engineering, IEEE Transaction on Systems, Man, And Cybernetics, Part B: Cybernetics, Vol.26, No.3, pp.365-380

[10] Tani J. (1996) Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective, IEEE Transaction on Systems, Man, And Cybernetics, Part B: Cybernetics, Vol.26, No.3, pp.421-436

[11] Kubota N., Arakawa T., Fukuda T., Shimojima K. (1997) Motion Planning for A Robotic System with Structured Intelligence, Proc. of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 60-65.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Future of observable universe

Yoshio Fujita
Presidnt
The Japan Academy

Address : 7-32 Ueno Park, Taito-ku, Tokyo, 110 Japan
Tel : +81-3-3822-2101 Fax : +81-3-3822-2105

Abstract

In the summer of 1936, when I was a student of Department of Astronomy, Faculty of Sciences, at the University of Tokyo before graduation, I was taking pictures of stars using the 8 inches refracting telescope equipped in the room to complete my graduation project. It was a time that we could easily take pictures of relatively dark stars by locating the telescope to the same direction during about one hour in Azabu near the center of Tokyo. At that time, could I imagine the great progress of the astronomy like this?

Although, at that time, the United States of America had already operated the 100 inches reflecting telescope on the mount Wilson and it was in actively used, we did not have such an excellent telescope in our country. In 1960 after 30 years passed, we had a 1.8 meter reflecting telescope on the mount Tikurin in Kamogata near Okayama and took part in the member of research groups on astronomy in the world. After more 40 years has passed, our 8.3 meter reflecting telescope will be built on the top of the mount Mouna Care in Hawaii in next year 1998 and the activity of the researches will be expected in accordance with the 10 meter reflecting telescope in America which is now in the operation and the telescope in the Union of European countries. The optical telescopes stated above played an important role in elucidating the world of stars, and the image of the universe considered now has emerged according to the progress of development of observation devices which capture the radiant energy from the heavenly body in wide wavelength region. The visible range is a part of the universe. According to r-rays, x-rays, ultraviolet rays, infrared rays, and radio waves, the corresponding observation devices are built, respectively. In addition, as we could not expect it in the old day, the observation devices are in a satellite which is not affected from the atmosphere in the earth. The size of a radio telescope among them becomes large because it is not affected so much from the effect of the atmosphere in the earth. As to optical telescopes, the Habble space telescope especially is now working effectively although it did not work at first stage and plays an important role in order to elucidate the universe. The results of developments of these observation devices yield the elucidation of the state of stars in universe, the Galaxy which is the collection of a huge amount of stars, and a group of the Galaxy, and furthermore reaches the investigation on the birth and death of stars and black holes, etc.. The problems on the spread of the universe and the age of the universe show the interesting developments. At the third general meeting of IAU in Kyoto, August last year, the numerical value for the age of the universe raged from 8 billion years to 10 billion years was shown according to the discussions from the observation and the theory based on the measured data. Depending on the development of the observation, it will be an interesting problem whether the expanding universe still continue or not. Besides, it is a deep concern of home work for human being how the search for the civilization outside the earth evolves.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Origins of order
in biology, economics, and technology

S. Kauffman
Santa Fe Institute
1399 Hyde Park Road Santa Fe, NM 87501,USA
+1-505-984-8800, E-mail:stu@santafe.edu.

The manuscript of this invited talk will be distributed at the symposium.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Development of Micro Robot System for Playing Soccer Games

Ju-Jang Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373 - 1 Kusong-dong, Yusong-gu, Taejon 305 - 701, KOREA
TEL: +82-42-869-3432 E-mail : jjlee@ee.kaist.ac.kr

Abstract — This paper surveys the recent trends in developing a micro robot soccer system and presents the design procedure for soccer-playing robots, focusing on our system based on the centralized approach. The robot soccer game has a lot of challenging problems such as coordination between robots, motion planning of robots, visual recognition of objects, and so on. Considering the results of the MIROSOT contests, the centralized approach may be more powerful than the distributed approach in order to implement such functions.

Our attempt is to develop a micro robot system with a remote-brainless control architecture. After that new techniques were applied to the chasis design, actuators, the radio link, the vision system software, and the control strategy software. Using a fast vision system, we obtain the configuration of each robot, then the host computer computes the desired motion and commands each robot directly via RF communication. We describe the technical tips for developing the robots in detail here and explain our strategy for obtaining the scores. This article also includes most of recent improvements in robot hardware and software.

1. Introduction

A robot soccer game comes in the field of multi-agent systems since the first MIROSOT[1] (Micro-Robot World Cup Soccer Tournament) contest was held in 1996. A robot soccer game has lot of challenging problems such as coordination between robots, motion planning of robots, visual recognition of objects, and so on. The orchestration of such functions is the key factor in winning a game.

In [1], the authors explain three control methods of soccer-playing robots: a remote-brainless control, a vision-based control, and a robot-based control. Such categorization of control structure depends if each robot has its own CPU and who determines the robot's motion using the visual information, as shown in Table 1. The designer may choose one method according to his/her research interest.

There has been a controversial discussion about which control method is more appropriate for a robot soccer game between a centralized control and a distributed control [6]. Although the robot-based control (i.e. the distributed control) is a very interesting field of research, most of teams decided to use the centralized control structure because it is relatively easy to make the robots and they have

[1] http://www.mirosot.org

a very fast vision system. Considering the results of the MIROSOT contest, we can find a centralized control structure to have more advantages than a distributed approach, for example, there is no problem in sharing information between robots. Also, it is very hard to implement an ideal robot-based control method because putting vision systems on each robot is impractical with current technologies, due to size constraints.

In this paper, we present the first attempt to build up a micro robot system based on the remote-brainless control architecture. Our robot system is very easy and cheap to develop so that we can expect the reliablity of the robot system. During the development, new techniques were applied to the chasis design, the actuators, the radio link, the vision system software, and the control strategy software and significant improvements were made.

This paper is organized as follows: Section 2 will give technical tips for a hardware architecture of micro robot soccer system in detail. In Section 3, we will explain a control strategy, focusing on our software which has the hierarchical structure with two levels. Finally, we will draw some conclusions in Section 4.

2. System Architecture

As shown in Fig. 1, a soccer robot system consists of robots, a host computer, a radio link from the host computer to each robot, and a vision system. We will explain each part in the following section and then give an example of our robot system in detail.

2.1. Host Computer

According to the rule of the MIROSOT contest, each team is allowed to use only one host computer such as IBM PC compatible, UNIX workstation and whatever. The role of host computer is to recognize what happens on the playground using a vision system and to give the team robots some information such as what they have to do, the current state of robots and so on. The content of information depends on the designer's choice, that is, the control architecture adopted.

In our case, we use the remote-brainless control method to implement our soccer robot system so that the success of our system depends on the performance of host computer, which is responsible for controlling and cooperating the robots. Our host computer receives the tracking data of the ball and the robots, over a serial port from the vision board, computes the desired motion of the robots, and sends the control signal to the robots through the radio control (R/C) transmitter. We chose a desktop personal computer with a 200MHz Intel Pentium-Pro microprocessor as our host computer.

2.2. Vision System

The results of the MIROSOT contests show the importance of the performance of the vision system; the faster the spatial sensing, the faster the feedback loops for controlling robot positions, and, thus, the faster the robots can travel [2]. We think that the updating frequency of vision system should be at least 30Hz to win the game. The overall vision system is composed of two components; a camera and an image grabbing board. To track the objects on the playground, we recommend to recognize color patches on the robots because an edge detection or a gray level detection is very hard to distinguish the objects with similar shapes or brightness.

In our case, fast and accurate spatial sensing is very crucial to the success of controlling the robot's motion, because our robots have no processor to control their own motion. Using visual informa-

tion, the host computer, which is responsible for controlling the robots, computes the desired velocity of the robot's each wheel and sends the corresponding control signal to the robots.

Our vision system consists of a video camcoder and the Cognachrome vision system[2]. The Cognachrome vision system is able to track several objects in the field of view at a full 60Hz frame rate, with performance degrading to 30Hz when the number of tracking objects is increased. The system can be trained to recognize up to three colors simultaneously, and there may be many objects of each color present in the field of view. For each object, the system calculates various statistics of the recognized bulbs such as centroid, area, and orientation of major and minor axes when desired. The user can obtain the tracking data in a user-defined format through one of the board's two asynchronous serial ports [3].

Although the Cognachrome vision system has a good performance in speed and accuracy, we have some problems in distinguishing two robots when they move very close to each other. We think that this problem can be solved by improving our software because the team from the Newton Labs won the first place twice using the same vision system.

2.3. Communication Between Host Computer and Robots

In general, there are two ways of implementing the communication link between the host computer and the robots; an infrared (IR) remote control method and a radio frequency (RF) communication. The IR remote control method is very simple and easy to build, however it is too sensitive to the lighting condition that the communication link can be often unstable during a game. On the contrary, the RF communication method is not influenced by the environmental condition. Also, we can buy the commercial products easily [3]. To solve the frequency sharing problem between teams, it is recommended to prepare two RF modules with different frequencies. The RF communication methods can be divided into two groups; a digital modem for the vision-based robots and an analog transceiver for the remote-brainless robots. In the case of the digital modem, the designer should assign the decoding and encoding protocol for the data packets.

We employ one-way radio link from the host computer to the robots using a commercial R/C transmitter and receiver. A conventional R/C transmitter has a trainer mode, which enables a student to control the radio control model such as R/C car, airplane, and so on. To control the robots directly, we set the trainer mode on and let the host computer send the *pulse-code modulated control signal* to the transmitter through a trainer jack.

Using the programmable timer 8253, we developed a interfacing board between the host computer and the R/C transmitter. All the modes of 8253 are used to generate a train of pulses of varying widths. After observing real signals from R/C transmitter, we programmed our interfacing board to have the same protocol. The pulses are updated at a given period, typically set to $20ms$. This updating frequency matches the cycle frequency of vision system well so that motion control of robots are made possible based on visual feedback.

2.4. Robot Hardware

Depending on the control architecture, the designer must select the proper robot hardware. In the case of the robot-based control architecture, the robot

[2]http://www.newtonlabs.com/ cognachrome

[3]http://www.radiometrix.co.uk

hardware is featured by its sensor system such as infrared sensors. However, the vision-based robot and the remote-brainless robot do not need to have such sensor system necessarilly. Now we explain each part of the robot hardware in what following.

2.4.1. Mechanical Part and Actuators

To drive the robot, most of the teams employed differential drive mechanism using a pair of actuators. Since the first MIROSOT contest in 1996, there was only one team to use omnidirectional robots [4]. Although their robots had the advantage of full directionality, their motion was not robust and reliable against the playground condition. When the slippage between the omnidirectional wheel and the ground occured, their motion was affected more easily than the differential drive mechanism.

To select the actuators, we can consider two kinds of actuators; a DC motor and a stepping motor. Although the stepping motor is easy to control, the size constraint of the robot leads us to choose DC motors as the actuators. Adjusting the gear ratio, we can obtain the proper torque and speed of the robot. To control the motion precisely, an encoder signal can be used as a feedback input.

2.4.2. Robot-Based Control

In the robot-based control architecture, each robot is responsible for determining its own motion and strategy in order to perform many kinds of autonomous behaviors. To react to the local environment, the robots have usually their own sensor system such as infrared sensors. Using sensory inputs, a robot is able to obtain the following information; whether the robot possesses the ball or not and the existence of obstacles around. The signal modulation technique is widely used in transeiving the infrared sensor signal, as in a remote controller, in order to diminish noise effect.

As stated earlier, putting vision system on each robot is impractical with current technologies, due to size constraints. Thus, the host computer provides only the current states of robots and ball through RF communication.

2.4.3. Vison-Based Control

In the vision-based robot system receives a velocity command from the host computer, then controls its motion using the on-board processor. To control the robot's motion precisely, most of the teams put the motion control chip such as LM629 series on their board.

2.4.4. Remote-Brainless Control

We will now explain the remote-brainless control architecture, focusing on our case. Our robots consist of very simple components: batteries, the hacked R/C servo motors, the wheels, and the R/C receiver module. These components are very cheap and easy to obtain in the hobby shop. Using double-sticky tape, we fix these components on the robot's body together.

To drive the robot, we employ the differential drive mechanism using a pair of Hitec HS-80 servo motors, which were modified to spin all the way around and to have a lower gear ratio. A key technique in modification is to remove the potentiometer from the servo motor. A unmodified servo motor has an integrated circuit and a potentiometer to implement a closed-loop position control system. After removing the potentiometer, the closed-loop position control system is changed to the closed-loop velocity control system. Now, the velocity of the servo motor depends on a *pulse-code modulated signal* sent by the radio. In [5], you can see more explanations on how to make the R/C servo motor continuously revolvable.

The robot's wheels are made to have a diameter

of $32mm$ with standard rubber O-rings. The robot is powered by four 1.2 volt NiCd batteries wired in series. The power line is connected to two RC servo motors via the receiver. Fig. 2 and Fig. 3 show the components of the robot and our soccer robot respectively.

3. Control Architecture

What is the best policy to win the game is always the primary concern for the software developers. A good strategy enables the robot to improve its position in order to attack and block the opponent. Also, a well coordinated and accurate motion is essential in the control strategy. The robustness and reliablity can be another issue. To satisfy such issues, there can be differential types of strategies (See [6]). Thus, in this section, we will present one of the ways to develop the strategy, focusing on our case.

Our software for controlling the robots has a hierarchical structure with two levels. The low level control software is responsible for controlling the speeds of two wheels of robots and implements the basic motions such as rotation, translation, kicking motion, and so on. The high level takes the current states of the robots and the ball, and specifies the desired motion of robots, using basic motions. All these softwares run on our host computer.

3.1. Low Level Control with Basic Motions

Our robot with the differential drive belongs to the nonholonomic system that is known to have some difficulties in controlling a motion [7]. To minimize such difficulties, we use a simple rotate-and-go strategy to put our robots on a target position.

To implement a rotate-and-go motion, we use a proportional gain controller structure. The gain has a different value according to an error range. The rotate-and-go motion enables us to put each robot on a target position without considering a final orientation. As stated earlier, fast and accurate spatial sensing is very crucial to achieving a robot's desired motion.

In kicking motion, the direction of attack is determined from the orientation of a robot when a contact between robot and ball occurs. However, the rotate-and-go motion does not consider a final orientation so that we devise a virtual target algorithm. The main idea of virtual target algorithm is as follows: initially, a virtual target is away from the robot, so that the target drags the robot. As the robot moves close to the virtual target using the rotate-and-go motion, the target converges to the desired position. Dragging effect enables us to put the robot on the desired position while modifying its orientation to fit the specified orientation. If a precise motion control of robots is possible, there is no benefit in having the virtual target instead of simply the direction. However, most systems have some difficulties in adjusting control parameters due to the nonlinearity of motors, so that it is difficult to implement a precise motion controller. It turns out that the virtual target algorithm can compensate for these difficulties during the contest.

Fig. 4 shows a geometrical representation on how to generate a virtual target. In this figure, the robot is desired to follow the arc-shaped path which satisfies the specified orientation at the end of this path.

During the game, returning motions of robots are very important to prevent our robots from dribbling the ball into our own goal area. By modifying the virtual target algorithm a little, we can obtain a smooth returning motion. As shown in Fig. 5, we use a switching strategy between a kicking motion and a returning motion. We divide a playing area

into two: one for kicking motion and the other for returning motion. When the robot enters the area of returning motion, we set the attraction point in the area of kicking motion. Also, the orientation at the attraction point is assigned depending on the distance between the ball and the robot. We found that the returning motion enables the robot to avoid the ball. Indeed, if we assign the attraction point properly, our switching motion can be used to defend the opponent's attack.

In spite of our powerful algorithm, we have some troubles in implementing the symmetric back and forth motion because the hacked R/C servo motors have a hard nonlinearity. However, we think that these minor problems are easy to overcome in future contests.

3.2. High Level Control

High level controller is responsible for the strategy based on visual information. Whenever the ball position is updated, the high level plans the three robots' motion and sends the *pulse-code modulated signal* through the interfacing board. The goalie robot has a simple strategy to block the ball and the other two robots use more complex strategies to score the goals.

3.2.1. Goalie's Strategy

A goalie has a quite simple strategy as follows: it stays between the ball and the goal line, so that the back and forth motion is quite enough to block the ball with its side. The simple motion does not consider the goalie's kicking motion because the probability of losing the points can be increased during the kicking motion. The goalie's motion is limited so that the goalie always stays between goal posts.

3.2.2. Field Players' Strategy

The objective of field player robots is as follows: the field player robot kicks the ball into the opponent's goal and when such a kicking is not possible or the opponent's attack is aggressive, it will try to improve its position in the playing ground. To achieve this objective, it is very important to know what is going on in the playing ground.

After dividing the playing ground into four areas, based on the center of the ball, we establish a rule-based strategy using some basic motions. The rules are determined depending on where the field players are located. Fig. 6 shows us the division of the playing ground. The opponent's goal post and the end points of our goal line are used as the fixed points when we draw these lines.

The basic principle of our rule-based strategy is quite simple as follows: when our robot is in area 1, this robot should move out of area 1 in order not to disturb the other robot's attack. In areas 2 or 3, the robots need to pass the ball to the other robot or return to area 4 for blocking the ball. The more the robots in the area 4, the more chances our robots get to of attack. When our robot is in area 4, it performs the kicking motion using the virtual target algorithm. Although our strategy is relatively simple, we were at times surprised by the results, which show the complex behavior of our robots.

In future works, we will apply the evolutionary technique to generate the rules of our strategy. Also, we need to develop more sophisticated strategy to coordinate the robots as well.

4. Conclusions

In this paper, we surveyed the recent trends in developing the micro robot soccer system and described the technical tips for building up the

remote-brainless controlled system in detail. Also we explained the control strategy, focusing on our software. The remote-brainless control method has some advantages that it is very easy and cheap to make the robots and the control strategy can be implemented in high level.

The micro robot soccer system consists of robots, a host computer, a radio link from the host computer to each robot, and a vision system. We described how to develop the overall system in detail and give our case as an example.

Our software architecture has the hierarchical structure with two levels. The low level is responsible for controlling the robots' motion and implementing some basic motions such as kicking and returning. The high level takes the current states of the robots and the ball, and specifies the desired motion of robots based on the rule-based strategy.

In future works, we will apply the evolutionary technique to generate the strategy rules. Also, we hope to participate in future contests with more enhanced robots and to verify the performance of our strategy.

References

[1] J. -H. Kim, H. -S. Shim, H. -S. Kim, M. -J. Jung, I. -H. Choi, and J. -O. Kim, "A Cooperative Multi-Agent System and Its Real Time Application to Robot Soccer," *Proc. of IEEE Conf. on Robotics and Automation*, pp. 638-643, Albuquerque, New Mexico, Apr., 1997.

[2] Randy Sargent, Bill Bailey, Carl Witty, and Anne Wright, "Use of Fast Vision Tracking for Cooperating Robots in the MIROSOT Micro-Robot World Cup Soccer Tornament," *Proceedings of Micro-Robot World Cup Soccer Tournament*, pp. 155-157, Taejon, Korea, Nov., 1996.

[3] Anne Wright, Randy Sargent, and Carl Witty, *Cognachrome Vision System User's Guide*, Newton Research Labs, 1996.

[4] Remy Blank, Jean-Marc Koller, Michel Lauria, and Francois Conti, "Omnidirectional robot and fast positioning systems," *Proceedings of Micro-Robot World Cup Soccer Tournament*, pp. 85-86, Taejon, Korea, Jun., 1997.

[5] Joseph L. Jones and Anita M. Flynn, *Mobile Robots Inspiration to Implementation*, A. K. Peters, 1993.

[6] *Proceedings of Micro-Robot World Cup Soccer Tournament*, Taejon, Korea, Nov., 1996.

[7] Jean-Paul Laumond, "Controllability of a Multibody Mobile Robot," *IEEE Trans. on Robotics and Automation*, Vol. 9, No. 6, pp. 755-763, Dec., 1993.

[8] Sun-Gi Hong, Tae-Dok Eom, Choon-Young Lee, Min-Soeng Kim, M. Sugisaka and Ju-Jang Lee, " Designing Soccer-Playing Robot Team Based on the Centralized Approach," *Proceedings of Micro-Robot World Cup Soccer Tournament*, pp. 119-123, Taejon, Korea, Jun., 1997.

Table 1. Categorization of Control Structure

	CPU	motion planner
remote-brainless	×	host computer
vision-based	○	host computer
robot-based	○	each robot

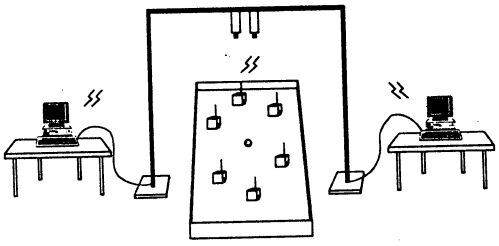


Fig. 1. Overview of overall micro robot soccer system

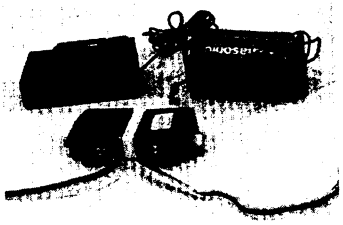


Fig. 2. The components of soccer robot

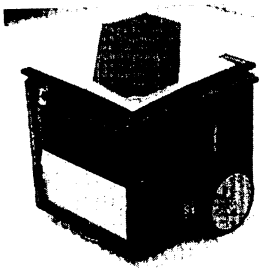


Fig. 3. A soccer robot

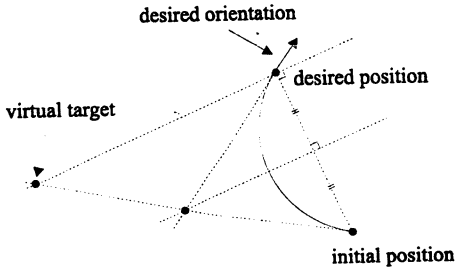
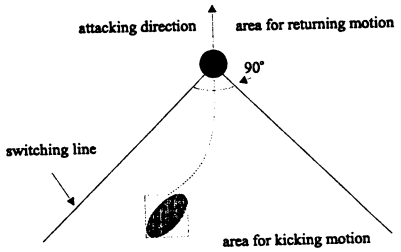
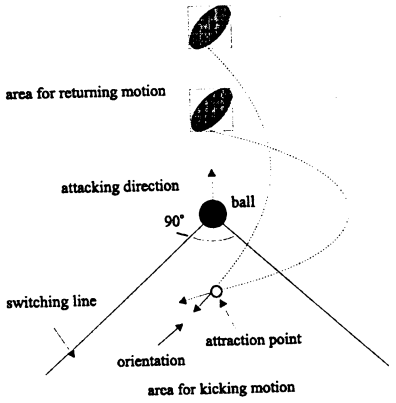


Fig. 4. Geometrical representation for virtual target



(a) Kicking motion



(b) Returning motion

Fig. 5. Switching between kicking motion and returning motion

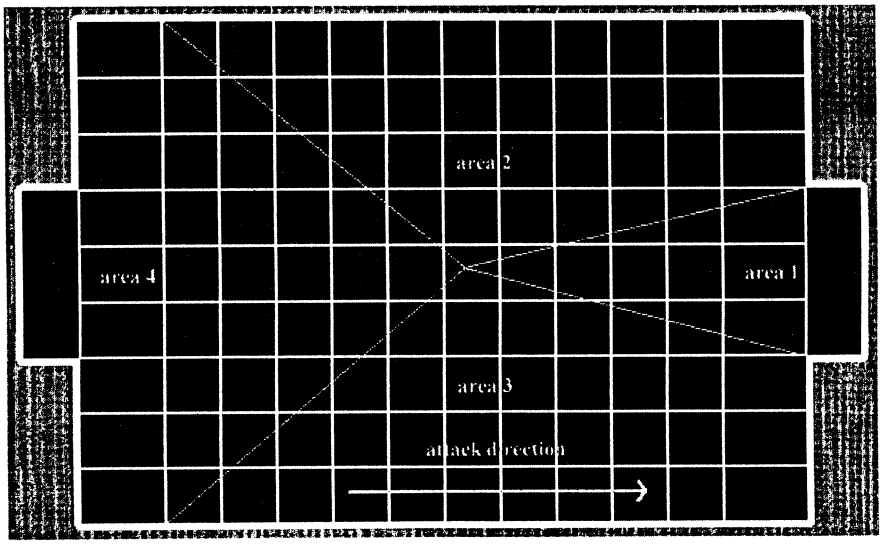


Fig. 6. Areas used in rule-based strategy

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Bio-imformatic Activity Modelling for Human-Artifacts Symbiosis under Resource-Boundedness

Tetsuo Sawaragi and **Osamu Katai**
Dept. of Precision Mechanics, Graduate School of Engineering, Kyoto University
Yoshida Honmachi, Sakyo, Kyoto 606-01, Japan.

Email. sawaragi@prec.kyoto-u.ac.jp

Abstract

We investigate into a biologically-inspired design of an interface agent, that is embedded inside the human-artifact interactions rather than as an external observer and has to work as an intelligent associate for a human user/operator in a time-critical situation like at an emergency. First recent paradigmatic shift of artifacts design principles are discussed from an interdisciplinary viewpoint. Then, after the idea of Clancey's activity modelling, we discuss about the design principles of a situated interface agent. That is, different from the conventional supervisory agent's task of seeking for optimizing the isolated control task, such an agent has to be able to maintain its identities as an organism living within the multiple contexts looking inwards to consider the the nature of memory and perception, and looking outwards to consider the nature of social action with a human operator. Some ongoing works by the authors along such a design principle are presented.

1 Introduction

Recent highly advanced industrial automation gives the human operator a challenge of unknown properties and there is a great concern for the side-effects of a higher degree of automation. So far many researchers have pointed out such brittleness of automation systems such as *automation-induced surprises* [20], *clumsy automation* [22] and *ironies of automation* [1]. To tackle with these automation-induced problems, a concept of *human-centered automation design* has been launched [16]. The ability to accept responsibility and to find innovative ways to fulfill these responsibilities is a unique human characteristic. Herein the design philosophy is "people are in charge" or "human-in-the-loop". The key question is then how to design an artifact system in which a human and an automated system can coexist and collaborate with one another.

An interface agent is a semi-intelligent computer programs that can learn by continuously "looking over the shoulder" of the user as he/she is performing actions against some complex artifacts and is expected to be capable of providing the users with adaptive aiding as well as of alternating the activities instead of a human [11], [12]. In this sense, an agent has to coexist with a human user so that it can evolve by itself as a human user's proficient level improves and can stimulate a human user's creativeness coordinately by changing its role dynamically as a human's associate, rather than to replace the human user with itself.

2 Bio-informatic Activity Modelling in an Interactive Domain

Currently there occurs a rapidly developing radical paradigmatic or epistemic shift from the conventional *rationalistic* orientation toward an *ecological* view that the proper units of knowledge or sources of intelligence are primarily *concrete, embodied, lived*, concentrating on how to deal with the domain of *immediate present* where the concrete actually lives [26], [25]. This emerging view is commonly shared by interdisciplinary fields; *cognition in practice* in cognitive science [10], *ethnomethodology* in sociology [4], *situated and embedded action* [24] in human-machine systems, etc.. The common interests wherein are in a *consensual* domain which unconscious, structurally-coupled autonomous entities do produce through dynamical mutual interactions, rather than based on an individual's reflection and introspection isolated from the practice. A similar paradigmatic shift is also ongoing in the field of social cybernetics over the last few decades [5]; a shift from *first-order* toward *second-order* cybernetics. The latter has concentrated increasingly on the analysis of interacting process, including even observers of the processes in the systems to be studied, and generally deals with *living* systems ranging from simple cells all the way up the evolutionary scale to human beings.

Researchers advocating the importance of situatedness have put an emphasis on the role of the environment, the context, the social and cultural setting, and the situations in which actors find themselves [14]. Thus, "situated" means a state of being embedded within the interactions with both internal and external processes. The key issue for a situated agent is its ability of *activity modelling*. An activity is not just something we do, but a manner of interacting and of participating. Viewing activities as a form of engagement emphasizes that the conception of activity constitutes a means of coordinating action, a manner of being engaged with other people and things in the environment, what Clancey calls a *choreography* [2]. Our conception of what we are doing, and hence the context of our actions is always social and is defined with respect to our activities. In a word, activities provide the background for constructing situations and are the grounding of intentionality.

The above-mentioned activity-oriented approaches

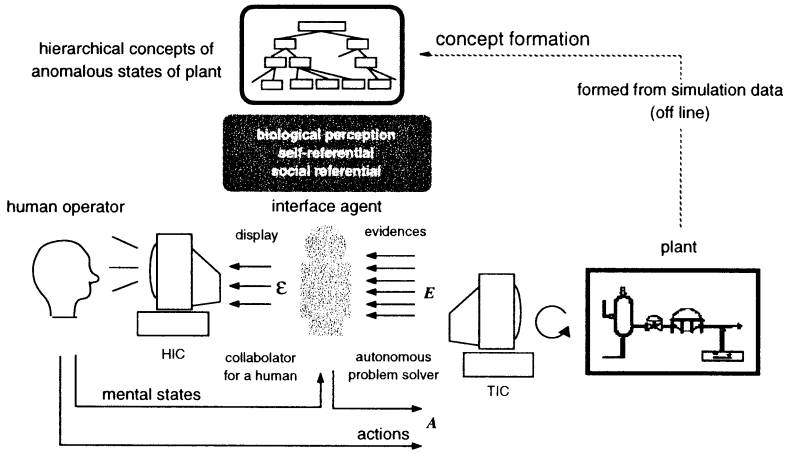


Figure 1: An interface agent situated within the human-system interactions.

are quite different from conventional principles for designing "intelligence" as follows. First, from the descriptive modeling perspective, the encompassing and composed nature of activities has been often missed because the modeler starts by choosing one activity to model and the the point of view of one role fulfilling one predefined goal within this activity. Second, in the information processing view, there is a sharp line between the individual and the environment, such that the relation is of input-output. On the contrary in the situated cognition view, the functional distinction between conscious and subconscious is emphasized, and the line between culture and individual experience is less distinct. Finally, in the rationalistic orientation, the idea of rational action suggests that goals and perspective rules control actions. By this view, tasks are isolated things to do and context is just the given world of data to operate upon; behavior is conditional on the facts. The contrary situated view is that defining goals, claiming what constitute the facts, and following plans and policies all occur within *nested* activities. In this sense, all action is situated in the actor's conceptions of what they are supposed to be doing, that is, norms, values, and roles.

In the following of this paper, we investigate into a biologically-inspired design of an interface agent, that is embedded inside the human-artifact interactions rather than as an external observer and has to work as an intelligent associate for a human user/operator in a time-critical situation like at an emergency. This is schematically illustrated in Fig.1. Herein, different from the conventional supervisory agent's task of seeking for optimizing the isolated control task, such an agent has to be able to maintain its identities as an organism living within the multiple contexts looking inwards to consider the the nature of memory and perception, and looking outwards to consider the nature of social action with a human operator.

Major tasks of an interface agent are divided into the following three kinds of activities. First, in order to provide with affordance-rich information to a human operator, an agent must be capable of perceptual recategorization and reconceptualization [25]. Second, an agent must be able to reason about the self; to have conception of what it is doing, and hence its own state, structure and process. Such conceptual constraints enable value judgments about how it uses its time, what it builds, what it thinks. Finally, the agent has to be able to design interactions for itself by considering about a manner of being engaged with other people (i.e., a human operator) and things in the environment (i.e., status of the plant). Thus, it must be a self-referential system as well as a social-referential system. In the following, our ongoing researches for each of the above issues are briefly introduced.

3 Biological Perception Using Evolutional Computing Method

An interface agent must be an apprentice learner with an ability to *learn from observation*, that is, to learn by continuously "looking over the shoulder" of the user as the user interacts with the system to be operated. Therefore, the interface agent has to be able to *form concepts* needed for its reasoning tasks from its own perspectives in a flexible and dynamic fashion. By the way, one of the distinctive characteristic of biological systems is their ability of dynamic recoordination and/or self-production involving a mixture of perceptual recategorization and reconceptualization. From this perspective, most of the conventional concept formation models assumed that a set of features characterizing concepts are provided a priori and does not change during learning. Thus, they rely on the filtered perception/interpretation of the world that the "designer" imposes rather than the "learner"; none of them examines the fundamental question of where features come from and how they are developed. Contrary to that a biological perception should be regarded as a self-producing, internally-inspired activity, rather than as a simple filtering device passively projecting the external reality.

Therefore, we have to deals with, at least initially, all observable features that may contain noisy and irrelevant features, and then, by referring to a user's actual behavior, this initial set must be modified through deletion of irrelevant or noisy features and continuous production of more predictive features, which in turn

alters the representation of the observations and this cycle iterates. Wherein, a predictive feature is such a feature whose value can almost uniquely determine another particular attribute's value. We stress that such a productive process is done only by referencing to the self rather than being guided by the external constraints.

Based on the above considerations, we developed an evolutionary concept formation algorithm [19]. The kernel of our proposing algorithm is Fisher's COBWEB, but the following major two processes are added that make up a broader cycle of self-production; salience-based pruning of attributes and GA-based feature discovery. The former is to select the essential attributes out of a provided set of attributes that may initially contain both relevant and irrelevant ones, and the latter is to construct new attributes using genetic algorithms applied to a set of elementary features logically represented in a disjunctive normal form. The details of the algorithms are presented elsewhere [18].

4 Biological Perception Situated to Time-Criticality

Another important aspect of a biological interface agent is how to organize an "appearances" of the world (i.e., a complex plant), which is different from a mere collection of objective features but is specific to a particular human operator depending upon current time-criticality he/she is forced to work. To be a human-friendly associate for a human operator, an interface agent has to be able to present the current status of the plant to the human operator so that he/she can identify the current status and can remind of the recovering operation to be adopted as uniquely as possible without confusion. In time-critical, high-stakes situations, the time required by people to review information, and confusion arising in attempts to process large amounts of data quickly, can lead to costly delays and errors. Numerous psychological studies have provided evidence that human's decision quality may degrade with increases in the quantity and complexity of data being reviewed, and with diminishment in the time available for a response, while the rate of his/her performing tasks can be increased by filtering or suppressing irrelevant information.

Wherein, an appearance of the environment is closely related with what Gibson called an "affordance" [7]. Affordances, in the ecological view, are invariants in the environment that are simply "picked up"; this information is perceived directly and requires no processing. Affordance exists in the perception of the relation between self and the environment. This is not any objective properties that preexist in the environment isolated from the perceiver, but may emerge out of the interactions between the perceiver and the objects in the environment. That is, the appearances of the world that will be perceived by the actor (i.e., an agent and/or a human operator) is determined by the consequent effectivity of the possible actions that heavily depend upon the physical conditions, available tools, and the internal states of the actor as well as the actor's goals and values. In Gibson's formulation, perception is not a passive affair but an active endeavor to discern relevant information about the particular objective. Given a goal, the observer maneuvers himself/herself so as to better discriminate pertinent information. The information thus gained enables the observer to modulate more precisely further detection and control activities.

Fig.2 shows the situation under which an interface agent is supposed to work. This illustrate a particular type of anomaly "shutdown of ANG-lines" that is generated by a simulator for a gas production plant and is used in our previous publication [18]. This shows that the anomaly has occurred and the observed trend data begin to change from the normal states. An interface agent encountering such an anomalous situation has to be able to do the following, or to assist the human operator to do the following; to identify the occurring anomaly type, to predict the consequences, and to determine and execute the recovery operation. Moreover, these multiple tasks must be finished until the deadline comes, or in more severe cases, they must be done as soon as possible, wherein value of the action effectivity may vary dynamically. Under such a time-critical situation, an interface agent must be able to flexibly organize the appropriate appearance of the plant status discriminating among what is *now* relevant and what is not. In our system, a taxonomy of all possible plant anomalies is organized in a hierarchical fashion using a machine learning technique called concept formation as shown in Fig.3 [18]. Wherein, the root node represents a class of concepts covering all possible anomaly types, and the leaf nodes represent the individual anomalous cases (i.e., a hierarchy via is-a and subset-of relations. In terms of this hierarchy, determination of the appearance of the plant concerns with how to determine the appropriate categorization of the plant anomalies out of the hierarchical taxonomy (Fig.3); to find subtrees whose extensions are regarded as equivalent. The category of the hypothesis on the plausible anomaly types can be defined with a variety of abstraction levels of the hierarchical taxonomy. We call each possible categorization within the taxonomy a *conceptual cover*, that is a categorization of all possible anomalies into mutually exclusive and exhaustive classes, and we have to formulate how to define this category being sensitive to the current situation.

For this purpose, we constructed a general probabilistic reasoning model (i.e., an *influence diagram* [9]) as an agent's decision model that derives an appropriate operation inferring the most plausible plant anomaly type by getting partially observable symptoms from evidential observations obtained so far. The agent's reasoning can be modeled by an influence diagram as shown in Fig.4. Wherein, a chance node H depicts a hypothesis node representing a state of the plant taking the value of anomaly types. Nodes E^i represents a set of symptoms available to an interface agent, each of which corresponds to the features observable in individual instrumentations. As knowledge of plant anomalies, an agent has acquired dependency among the classes of anomaly types and their associating evidences as well as the prior probabilities of occurrence of those hypothetical classes through the concept formation process, all of which are stored in a taxonomy hierarchy [17]. Based on this apriori knowledge, getting new observations from the plant, an agent updates its prior belief on probable anomaly types based on a Bayes theorem. Then, these posterior beliefs are used for calculating the expected utilities of the available options defined in the decision node A based on the knowledge that are defined in a part of the diagram defining the interrelationships among nodes of H, A, T

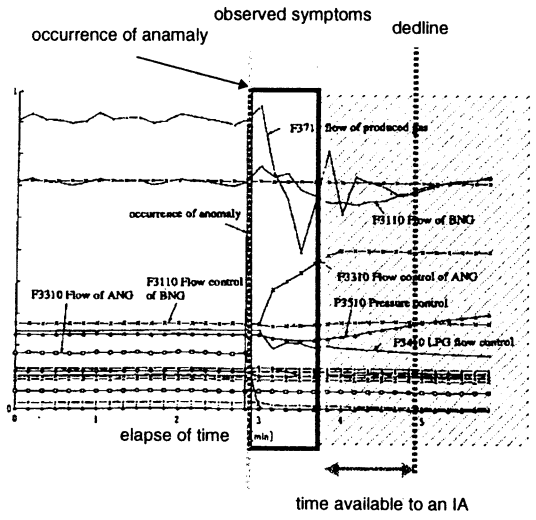


Figure 2: An example of plant behaviors for an anomaly "shutdown of ANG-lines".

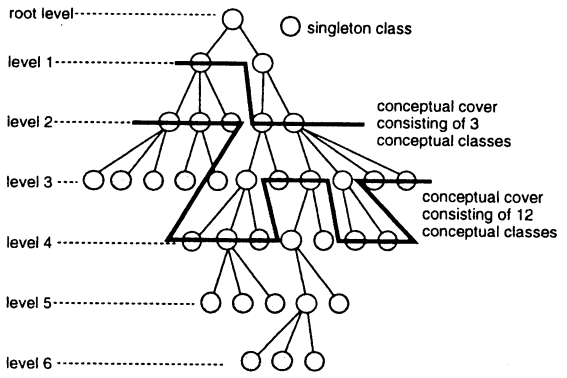


Figure 3: Conceptual cover of anomalous state concepts: a categorization adopted by an interface agent for its decision framing.

and V, and the option having the maximum expected utility would be determined as a recommendation for an interface agent to adopt. A diamond node is a special type of an oval node and is called a value node representing an agent's comprehensive utility. This shows that an agent's utility is determined by the types of anomaly and the adopted operation as well as by the delay. This reflects the fact that some anomaly types require an immediate recovering operations as soon as possible and if it is delayed its utility would drastically decrease (i.e., a time-dependent utility [8]).

In terms of this decision model, a definition of a domain of a hypothesis node showing the repertoire of conflicting anomaly types (i.e., values of a chance node H) corresponds to a conceptual cover showing an appearance of the plant to an interface agent. The qualities of each possible conceptual cover must be evaluated in terms of the effectivity brought about by the adopted operations, which can be formulated by calculating the

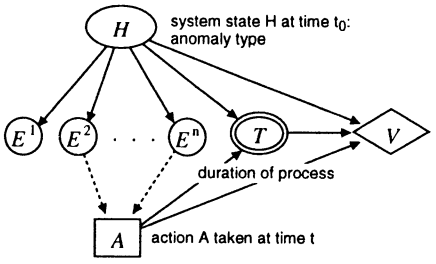


Figure 4: Interface agent's resource-bounded reasoning represented by an influence diagram.

expected utilities of the recommended actions by solving the influence diagram having a definition of a hypothesis node corresponding to that conceptual cover (i.e., an *expected value of categorization* with respect to a conceptual cover). We formulated this procedure for determining the optimal conceptual cover by introducing two types of model modification operators called *conceptual specialization* and *generalization* in our prior publication [23].

In this way, an interface agent can organize the best conceptual cover, i.e., an affordance-rich appearance of the plant to be perceived, consisting of the equivalent classes of anomaly classes in terms both of the uniqueness of the recommended actions and of the effectivity of that action calculated as expected utilities.

5 Agent's Reasoning about Self Living in Multiple Contexts

The selection of the appropriate conceptual cover is important not only to present an affordance-rich information concerning the current plant status to a human operator but also to consider about an interface agent's own problem solving activity. That is, in an extreme emergency, an agent has to behave autonomously without including the human in the loop, when an agent constructs its decision making model and determines the best action solving that model. At this time, an agent has to deal with tractability of decision making inference at the expense of decision quality. That is, an agent has to determine the appropriately manageable model by finding the definition of the hypothesis node as an agent's categorization of the environment (i.e., a plant status). Wherein, a resource-constrained agent has to be able to enhance its decisions by expending some resource to consider the tradeoff between the expected benefit of using more detailed models to increase the expected utility of action and the resource costs entailed by computing decisions with a more detailed model. Note that this activity of finding a better conceptual cover for model modification must be done by the interface agent and this is also a time-consuming activity. Therefore, an agent has to take account of the change of computational costs due to the change of conceptual cover as well as the cost performing the model improvement operation. The theoretical formulation of an agent's managing complexity on its own decision framing and decision making is provided in our previous publications [23].

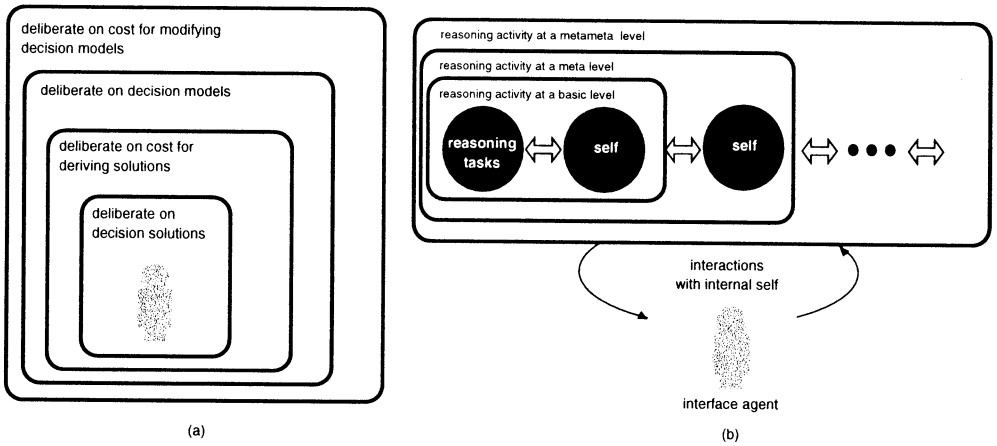


Figure 5: (a) Multiple contexts surrounding an interface agent. (b) Reflection of its own activities.

The agent's deliberation tasks at multiple meta-levels mentioned in the above can be schematically illustrated in Fig.5. An agent has to be able to reason not only about the quality of the solution but also about its own multiple activities needed to derive the solution. In other words, the situated interface agent has to be able to control itself and to reason about self being aware of the multiple contexts or backgrounds surrounding an agent's current task at hand.

6 Agent's Reasoning about Collaborating Others

Working in a "human-in-the-loop" mode as a human-friendly associate, an interface agent has to be able to deliberate on the information that is displayed to the human operator. Consider the case of an interface agent's monitoring complex systems and an agent has access to a set of sensed observations, $\boldsymbol{E}$. Human operators are charged with reviewing data that is a subset of observations, $\mathcal{E} \subset \boldsymbol{E}$. At this time, an agent has to consider about the nature and quantity of evidence $\mathcal{E}$ to be displayed. Assuming that the decision maker is an expert who acts in accordance with an interface agent's diagnostic decision model but requires increasing amounts of time to review larger quantities of information, we can develop an evaluation measure for considering the costs versus benefits of displaying alternate subsets of available information based on the decision model mentioned in the previous section (i.e., an *expected value of revealed information*). In this way, an interface agent can determine the subset of information presented to the human operator that maximizes the expected utility.

Formulation made in the above is based on an assumption that a human operator is an expert and acts in accordance with an interface agent's diagnostic decision model, but it is not usually true. In order to let an interface agent to automate decisions about display to optimize the expected value of an operator's actions, we should incorporate a model of an operator's expected behavior represented as the dependencies between the displayed information $\mathcal{E}$ and his/her taking actions A_i as well as the costs for reviewing the information, both of which depend on the proficient levels and experiences of an assumed operator. By considering the likelihood of alternative operator's actions and likelihoods of delay as functions of displayed data, we can calculate an expected value of displaying additional information to be presented to the human operator (i.e., an *expected value of displayed information*). The details of theoretical formulation of an agent's managing displaying information for a human operator is provided in our previous publications [23].

The agent's deliberation tasks on the display information taking account of the human operator mentioned in the above can be schematically illustrated in Fig.6. That is, an agent has to be able to reason not only about its own reasoning activity, but also about the activity of the human operator by predicting his/her decision making on the basis of what it believes. Thus, an agent has to have a model of interaction between the human operator and itself within the self. Having such multiple perspectives from its own and from *embodied* other people is the essential characteristic for a situated interface agent that has to establish a collaborative relationships while interacting with him. This is sometimes called *social-referential* ability in sociology.

7 Conclusions

In this paper, we at first mentioned about recent paradigmatic shift of system design principles for establishing a human-machine symbiosis. We discussed about the necessary characteristics of an interface agent who intervenes between the interactions and coexists with a human user. Especially, we put an emphasis of its "situatedness" and proposed to establish a new design principle borrowing an idea from a biological, living systems.

References

[1] Bainbridge, L.: Ironies of Automation, *Automatica*, **19**-6, pp.775-779, 1983.

[2] Clancey, W.J.: The Conceptual Nature of Knowledge, Situations, and Activity, in Feltovich, P.J. et al. (Eds.),

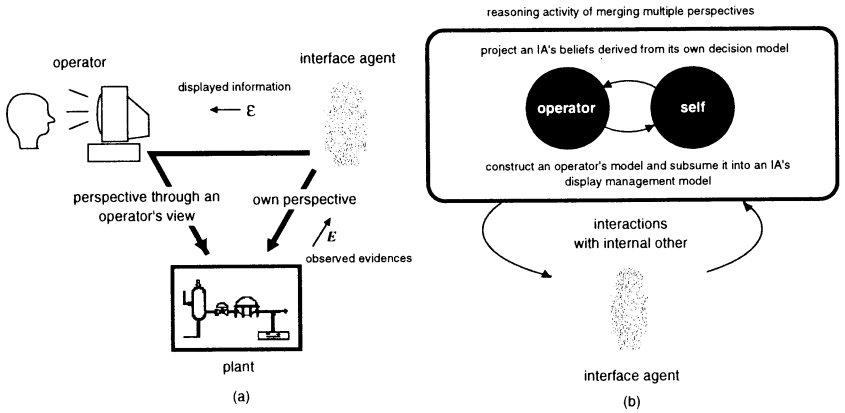


Figure 6: (a) Interaction of multiple perspectives within an interface agent. (b) Reflection on the interaction between the self and the other.

Expertise in Context, AAAI Press, Menlo Park, CA., pp.247-291, 1997.

[3] Fisher, D.: Knowledge Acquisition via Incremental Conceptual Clustering, *Machine Learning*, **2**, pp.139-172, 1987.

[4] Garfinkel, H.: What is ethnomethodology, in Garfinkel, H. (Ed.), *Studies in Ethnomethodology*, Englewood Cliffs, N.J., 1967.

[5] Geyer, F.: The Challenge of Sociocybernetics, *Kybernetics*, **24**-4, pp.6-32, 1995.

[6] Horvitz, E. and Barry, M.: Display of Information for Time-Critical Decision Making, *Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal, 1995.

[7] Gibson, J.J.: The Ecological Approach to Visual Perception, Houghton Mifflin Company, Boston, MA. 1979.

[8] Horvits, E.: Time-Dependent Utility and Action under Uncertainty, *Proc. of the Seventh Conference on Uncertainty in Artificial Intelligence*, Los Angels, pp.151-158, 1991.

[9] Howard, R.A. and Matheson, J.E.: Influence Diagrams, in Howard, R.A. and Matheson, J.E. (Eds.), *The Principles and Applications of Decision Analysis*, Strategic Decision Group, Menlo Park, CA , 1983.

[10] Lave, J.: Cognition in Practice: Mind, Mathematics and Culture in Everyday Life, Cambridge University Press, N.Y., 1988.

[11] Maes, P. and Kozierok, R.: Learning interface agents, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp.459-465, 1993.

[12] Maes, P.: Agents that Reduce Work and Information Overload, *Communications of the ACM*, 37-7, pp.30-40, 1994.

[13] Maturana, H.R.: Biology of Cognition. Reprinted in Maturana, H.R. and Varela, F.J., *Autopoiesis and Cognition: The Realization of the Living*, Reidel Publishing Company, pp.2-62, 1980.

[14] Norman, R. et al., Special Issue: Situated Action, *Cognitive Science*, **17**-1, pp.1-133, 1993.

[15] Poh, K.L., Fehling, M.R. and Horvitz, E.J.: Dynamic Construction and Refinement of Utility-Based Categorization Model, *IEEE Trans. of System, Man, and Cybernetics*, Vol.**24**-11, pp.1653-1663, 1994.

[16] Rouse, W.B.: The Human Role in Advanced Manufacturing Systems, in Compton, D. (Ed.), *Design and Analysis of Integrated Manufacturing Systems*, National Academy Press, Washington, D.C., 1988.

[17] Sawaragi, T., Iwai, S., Katai, O. and Fehling, M.R.: Dynamic Decision-Model Construction by Conceptual Clustering, *Proc. of the Second World Congress on Expert Systems*, pp.376-384, Lisbon, Portugal, 1994.

[18] Sawaragi, T., Takada, Y., Katai, O. and Iwai, S.: Realtime Decision Support System for Plant Operators Using Concept Formation Method, *Preprints of International Federation of Automatic Control (IFAC) 13th World Congress*, Vol.L, pp.373-378, San Francisco, July 1996.

[19] Sawaragi, T., Tani, N. and Katai, O. (1996). Evolutional Concept Formation for Personal Learning Apprentice by Observing Human-Machine Interactions, *Proc. of CSEPC (Cognitive System Engineering for Process Control)*, pp.122-129.

[20] Wickens, C.: Designing for Situation Awareness and Trust in Automation, *Proc. of IFAC Integrated Systems Engineering*, Baden-Baden, Germany, 1994, pp.77-82, Pergamon Press, Oxford, 1994.

[21] Sheridan, T.B.: Supervisory Control. In Salvendy, G. (Ed.), *Handbook of human Factors*, pp.1243-1263, Wiley, NY, 1987.

[22] Wiener, E.L.: Beyond the Sterile Cockpit, *Human Factors*, **27**-1, pp.75-90, 1985.

[23] Sawaragi, T. and Katai, O.: Resource-Bounded Reasoning for Interface Agent Realizing Flexible Human-Machine Collaboration, *Proc. of ROMAN'97*, Sendai, 1997.

[24] Suchman, L.A.: Plans and Situated Actions: The Problem of Human-Machine Communication, Cambridge University Press, N.Y., 1977.

[25] Varela, F.J.: The Re-Enhancement of the Concrete: Some Biological Ingredients for A Nouvelle Cognitive Science, in Steels, L. and Brooks, R. Eds., *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*, pp.11-22, 1995.

[26] Winograd, T. and Flores, F.: Understanding Computers and Cognition: A New Foundation for Design, Ablex Pub. Corp., Norwood, N.J., 1996.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Development of Eye-Sensing HMD and its Application Experiments for Observing Human Ocular Information Processing Characteristics

H. Yoshikawa H. Shimoda S. Fukushima D. Morikawa H. Fujiyama
Graduate School of Energy Science
Kyoto University
Uji-shi, Kyoto. 611

Abstract

A new Head-Mounted Display (HMD), "Eye-Sensing HMD (ES-HMD)" was developed as a VR-based human interface. The basic experiments are conducted to apply ES-HMD for the study of eye movement characteristics of stereoscopic view imposed by VR.

1 Introduction

What the authors call "Eye-Sensing HMD" (ES-HMD) is an extension of HMD device which is added "eye-sensing" function to the conventional HMD. The specific feature of this ES-HMD is that it is equipped with CCD cameras for the both eyes of a user while his wearing HMD and the supporting hardware and software for realtime processing to the monitored images of both eyes from the two CCD cameras to get various eye movement characteristic in situ, such as the change of eye gaze point, size of pupil diameter and eye-blinking[1].

This ES-HMD has the potential of wide application. One is a new-type interface of direct input-output device, and the other is that it can be used as a basic measurement device to diagnose personal visual function from psycho-physiological aspect. Concerning the former type application of ES-HMD, the authors have been conducting on a laboratory experiment to realize a mutual adaptive interface for CAI by utilizing ES-HMD's eye gaze input function[2]. And for the latter area of psycho-physiological measurement device, the authors recently conducted on a laboratory experiment to examine ES-HMD for diagnosing personal stereoscopic view functions in the artificial 3D visual environment configured by VR.

In this paper, the design specification and performance of the developed ES-HMD will be first described, followed by the results of subject experiment on the observed personal characteristics of near reflex of pupil and convergent movement.

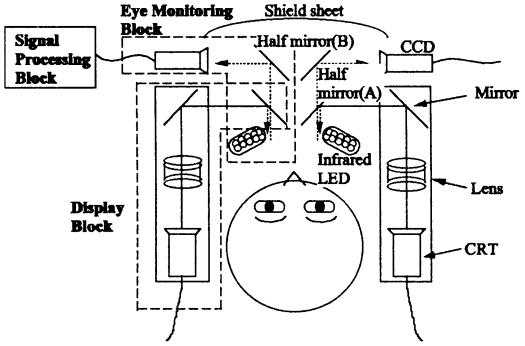


Figure 1: Configuration of the developed ES-HMD.

2 Development of ES-HMD

The developed ES-HMD consists of Display Block, Eye Monitoring Block and Signal Processing Block as shown in Fig.1.

Display Block consists of miniature CRTs, lenses, mirrors and half mirrors. As shown in Fig.1, pictures on CRTs are projected through lenses by way of mirrors and half mirrors to the user's both eyes. A user is supposed to watch picture on about 43 inches screen at 1 meter distance from the screen. User's view through the display will change in accordance with his head motion which will be detected by the position detecting sensor on the top of the HMD.

The purpose of Eye Monitoring Block is to monitor the user eyes and to get pupil images as binary black and white image. In order to obtain a clear pupil image, ten infrared LEDs are placed in front of each eye but outside user's field of view to emit infrared rays without his sensing. The light reflected on his eye surface comes into each CCD camera through focus adjustable lens fixed on camera by way of two half mir-

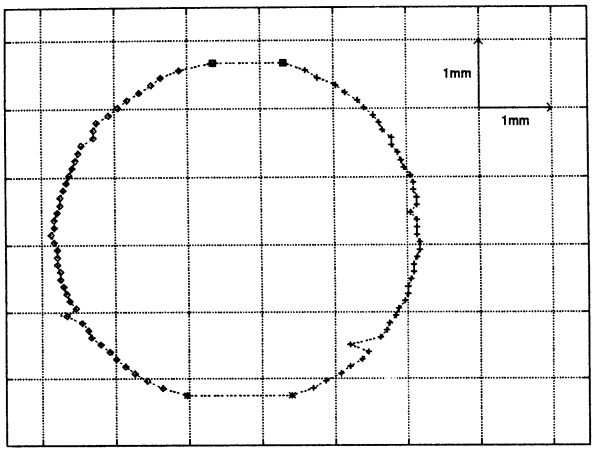


Figure 2: Both ends of pupil reproduced in Signal Processing Block.

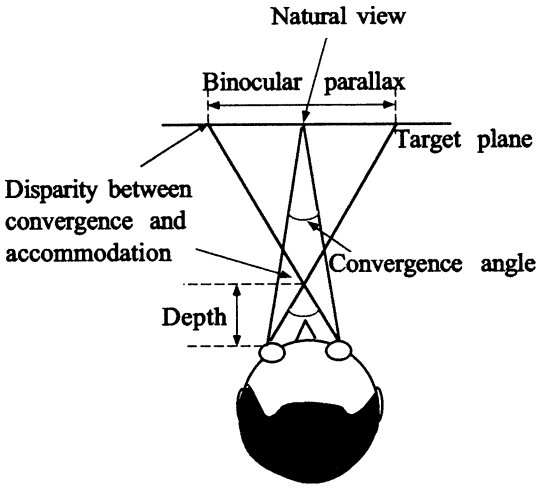


Figure 3: Schematic of binocular parallax.

rors. The resolution characteristics of the developed Eye Monitoring Block is as follows; time resolution of about 16.7[$msec.$], vertical resolution 0.109[mm] and horizontal 0.0467[mm].

By processing the video signal obtained in Eye Monitoring Block, both ends of pupil based on the horizontal axis in the camera coordinate system are then obtained in Signal Processing Block. Fig.2 shows the reproduced contour of pupil by Signal Processing Block. By using a PC for further calculation, various ocular information can be easily obtained in realtime such as pupil diameter, eyeblink, center position of pupil in the camera coordinate system, and gaze point in the display coordinate system.

The developed ES-HMD has an advantage with respect to measurement robustness. The evaluation was made to a subject who wears the ES-HMD, gazes at the small fixed target on the display under the constant brightness condition and moves his head right and left. The fluctuation characteristics of measured pupil diameter during his head motion were compared with those of no head motion condition, and it was found that the measured pupil diameter was very stable with the same order of minor fluctuation natural in constant brightness of display light.

3 Evaluation of ES-HMD in Stereoscopic View Condition

Since ES-HMD has two displays for both eyes, stereoscopic view can be realized by giving binocular parallax for right and left images.

Mechanism of Stereoscopic View

The image of an object in the world is fundamentally seen as a double image on the retina of the both eyes, but this double image is fused into a single image of stereoscopic view by the mechanism of human visual processing. Stereoscopic view by the both eyes is generated by binocular parallax, and as shown in Fig.3, this binocular parallax is represented by the difference of convergence angle. VR uses this mechanism to realize artificial stereoscopic view. But contrary to the human natural view that both convergence and accommodation lie on the same plane in Fig.3, the view imposed by VR gives rise to disparity between convergence and accommodation, and this is considered as one of the reasons why eye fatigue is caused by VR.

Near Reflex

The pupil size of human eye would tend to constrict when human looks at the target which is near to his eyes, and this phenomenon is called near reflex of pupil. It has been generally said that three elements of convergence, accommodation and constriction of pupil give rise simultaneously in near reflex[3], but the behavior of pupil is not yet well clarified in the case of disparity between convergence and accommodation, typical condition of VR.

Vergent Eye Movement

There are two types of movements of both eyes when human tries to see a moving target. One is conjugate eye movement when the target makes two-dimensional movement like from left to right. The other is non-conjugate eye movement when human

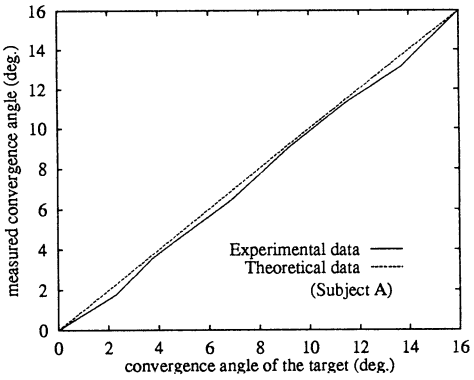


Figure 4: convergence angle under artificial stereoscopic view.

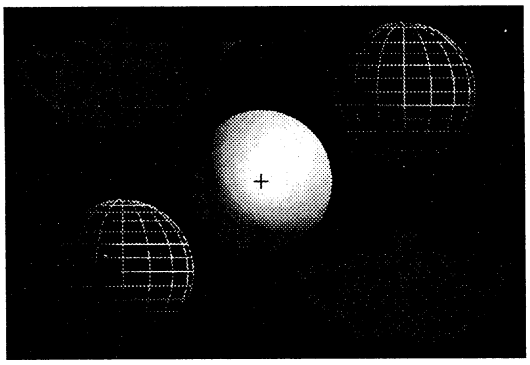


Figure 5: Experimental display image.

tries to follow a target approaching to him. In this case, the converse rotation of the both eyeballs named as vergent eye movement is said to be origin of depth perception.

Experimental Evaluation

The convergence angle in human stereoscopic view situation was estimated theoretically and it was compared with experimental data from a subject wearing ES-HMD. The details of the evaluation method are described in Ref.[4], and the results are shown in Fig.4. The maximum error between theoretical evaluation and the subject data resulted in less than $0.56[deg]$. It shows that convergence angle under artificial stereoscopic view condition can be measured by ES-HMD easily and correctly, although it is said difficult by conventional methods.

4 Basic Experiment on Visual Characteristics in Binocular Stereoscopic View

The use of ES-HMD for psycho-physiological studies of human visual processing, especially for the study of eye movement characteristics in both near reflex and vergent eye movement offers new experimental environment, because the situation of disparity between vergence and accommodation can be easily generated with the accurate measurement on the change of eyeball rotation.

A basic laboratory experiment was conducted on two subjects (A and B, both male students), to observe the changes of pupil size and eyeball rotation of their both eyes. The visual image used in the experiment is as shown in Fig.5. Two identical images of Fig.5 with the controlled horizontal distance between the both were projected on the both eyes, and the subject was asked to try to fuse the both images into a stereoscopic view by watching the cross mark (+) on the white ball. In addition to the static measurement of stereoscopic view characteristics, dynamic tests were conducted to the both subjects, with smooth change of the convergence angle from 0 to $16[deg]$ (this is controlled by horizontal difference between the both images) and then vise-versa, with cyclic periods between 2 and 16 seconds.

The major results from the dynamic tests are summarized as bellow;
(i) *Dynamic Change of Pupil Diameter*: See Fig.6. This is a new experimental proof of influence of vergent eye movement to the change of pupil size in VR. (ii) *Dynamic Change of Eyeball Rotation in Constant Pursuit of Center Ball*: See Fig.7, case (a) for subject A who was good at stereoscopic view in ES-HMD by vergent eye movement, while case (b) for subject B who felt difficulty of stereoscopic view (stereoanomaly). Stereoanomaly can be easily detected by this experiment. (iii) *Dynamic Change of Eyeball Rotation with Intermitted Change of Gaze Points*: See Fig.8 of subject A's left eye. There are two curves in Fig.8, the solid line for the pursuit of only the center ball, while the dotted line for the mix of intermittent changes of gaze point not only for central ball but also for the peripheral figures in turn. It can be said from this result that it is very feasible for online 3-dimensional tracking of eye gaze point by using ES-HMD.

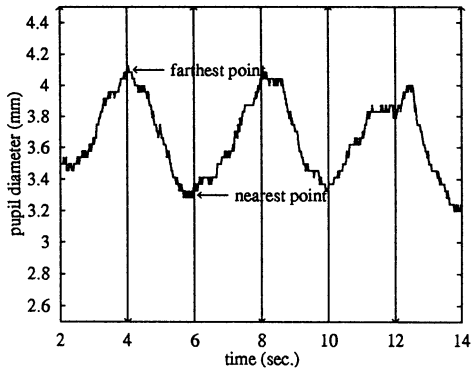


Figure 6: Pupil diameter while gazing at the moving target.

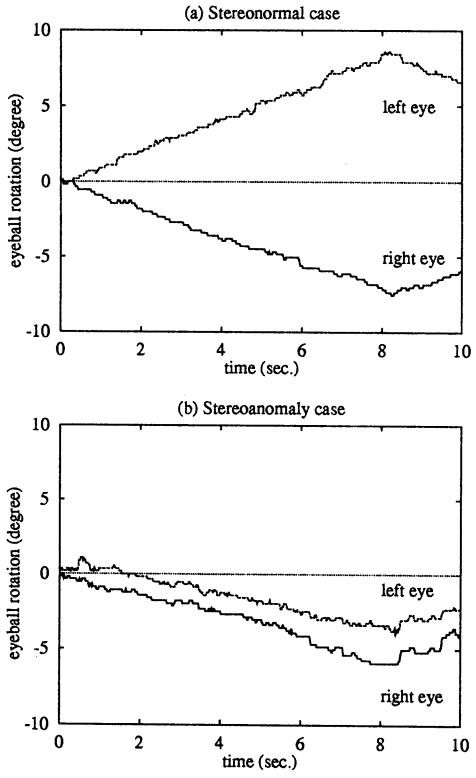


Figure 7: Measured eyeball rotation of two subjects, (a) a stereonormal case and (b) a stereoanomaly case, while gazing at the continuously moving target.

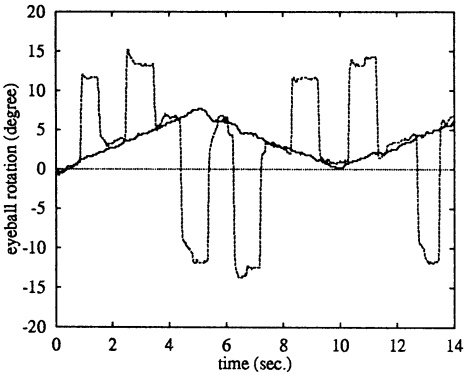


Figure 8: Eyeball rotation of a stereonormal while gazing at the discontinuously moving target.

5 Conclusion

The designing of Eye-Sensing HMD (ES-HMD), a new type HMD device for VR-based human interface and the experimental validation of the developed ES-HMD were first described, and then the results of basic laboratory experiment were discussed with respect to the possibility of the use of ES-HMD for the study of eye movement characteristics of stereo view imposed by VR. It was found that the ES-HMD is a useful device for such kind of basic psycho-physiological research on introducing VR into human interface.

References

[1] S. Fukushima, M. Takahashi and H. Yoshikawa, "A Head Mounted Display with the Function of Measuring Eye Images", *Human Interface News and Reports*, Vol. 11, pp. 197-202, 1996. (in Japanese)

[2] S. Fukushima and H. Yoshikawa, "Application of a Newly Developed Eye Sensing Head-Mounted-Display to a Mutual Adaptive CAI for Plant-Diagnosis", *Elsevier*, Vol. 21B, pp. 225-228, 1997.

[3] R. Osaka, Y. Nakamizo and K. Koga, "Experimental Psychology of Eye Movements", *Nagoya University Press*, 1993. (in Japanese)

[4] H. Fujiyama, S. Fukushima and H. Yoshikawa, "An Experimental Analysis on Gaze Point History of Stereoscopic Vision by Using Eye-Sensing HMD", *Proc. of 13th Symposium on Human Interface*, pp. 69-74, 1997. (in Japanese)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A Study on Optimal Motion of a Biped Locomotion Machine

Katsuyoshi Tsujita, Kazuo Tsuchiya, and Hironori Kawano
Dept. of Aeronautics and Astronautics, Kyoto University, Kyoto 606-01, Japan

Abstract

In this paper, a calculation method of an optimal trajectory of a biped locomotion machine is proposed, which is based on the inverse kinematics and the inverse dynamics. The algorithm proposed is composed of as follows; First, the trajectory of the waist is expressed by Fourier siries, where, the bases are selected appropriately so that the periodic boundary conditions are strictly satisfied.

Moreover, a biped locomotion machine establishes an optimal walking by using kicking forces to the ground at the moment of switching legs. In order to include the effects of the kicking forces, additional terms that indicate impulsive forces are included as discontinuity of the velocity at the moment of switching legs. Then, the angles of each joint are determined by the inverse kinematics. Next, using the inverse dynamics, input torques of each joint are expressed as functions of Fourier coefficients. By defining the performance index as a quadratic form of input torques, the motion planning problem is formulated as an optimization problem of the trajectory of the waist whose parameters are Fourier coefficiens. Fourier coefficients of the trajectory of the waist become its optimization parameters.

Then, using S.Q.P. (Successive Quadratic Programming method) optimal trajectory of a biped locomotion machine is obtained.

1 Introduction

In the proceeding paper([1]), we have studied the motion control of a biped locomotion machine and proposed a hierarchical controller of the system; The upper controller is motion planning system and generates an appropriate walking trajectory. The lower one is motion controlling system and controls each joint using feedback control and also controls gait parameters included in the trajectory.

This paper deals with the optimal motion planning of a biped locomotion machine in the motion planning system. In this paper, the optimal motion planning is formulated and derived as the optimization problem. ([2]) The algorithm to solve this problem becomes iterative one to boundary value problem. On the point of view of kinematics of a biped locomotion machine its trajectory is periodic, and then the trajectory needs to satisfy the boundary conditions at the moment of switching legs strictly. But in order to derive such trajectory, many calculation times are required. So far, some optimization methods of the trajectories of the nonlinear multi-body systems have been proposed([3]). In some of such researches([4],[5]), the input torques are expressed by Fourier series and motion planning problem is reduced to the optimization problem with many variables. But even if we use this method, many calculation times are required in order to satisfy the constraints corresponding to the boundary conditions mentioned above.

In this paper, a calculation method of a trajectory of a biped locomotion machine is proposed, which is based on the inverse kinematics and the inverse dynamics. The algorithm proposed is composed of as follows; First, the trajectory of the waist is expressed by Fourier series, where the bases are selected appropriately so that the periodic boundary conditions are strictly satisfied. Then, the angles of each joint are determined by the the inverse kinematics. Next, using the inverse dynamics, the input torques of each joint are expressed as functions of Fourier coefficients. The performance index is defined as a quadratic form of the input torques, and Fourier coefficients of the trajectory of the waist become its optimization parameters. As a result, the motion planning problem becomes optimization problem of the trajectory of the waist whose parameters are Fourier coefficients.

On the other hand, a biped locomotion machine establishes an optimal walking by using kicking forces to the ground at the moment of switching legs. In order to include the effects of the kicking forces, many terms of Fourier series are required, but it needs many times of calculation. In this paper, it is expressed by the effects of the impulsive forces. The impulsive forces acted to the system causes discontinuities of the velocities of the motion. Considering this fact, additional terms that indicate discontinuities of the velocities at the moment of switching legs are included to Fourier series of the trajectory of the waist.

Then, using S.Q.P. (Successive Quadratic Programming method) optimal trajectory of a biped locomotion machine is obtained.

2 Equations of Motion

We consider a biped locomotion machine composed of the main body and two legs indicated in Fig. 1. Each leg is composed of two links connected to each other through one DOF(degree of freedom) rotational joint, and connected to the main body with one DOF rotational joints. The motion of the biped locomotion machine considered here is constrained in the sagittal plane. Each leg is numbered as No.1 and 2 and leg 1 is considered as supporting leg. Each link of the leg 1 is numbered as No.1 and 2 from the supporting point to the main body and each link of the leg 2 is numbered as No.4 and 5 from the main body to the end of the leg. The main body is numbered as link 3. The actuators are located at joint 2,3,4,5 but there is no actuator at the supporting point(joint 1). We define unit vectors fixed to an inertial space whose origins are corresponded to the supporting point as $\boldsymbol{a}_{01}$, $\boldsymbol{a}_{02}$, $\boldsymbol{a}_{03}$, where the direction of $\boldsymbol{a}_{01}$ is corresponded to the sagittal direction and the direction of $\boldsymbol{a}_{02}$ is vertically up. Similarly, unit vectors whose origins are on the joint i are defined as $\boldsymbol{a}_{i1}$, $\boldsymbol{a}_{i2}$, $\boldsymbol{a}_{i3}$, where $\boldsymbol{a}_{i1}$ is along the axis of link i, and the directions of $\boldsymbol{a}_{i3}$ coincide with the rotational axes of joint i. Using these unit vectors, we define the following unit column-matrices.

$$[\boldsymbol{a}_i]^T = [\boldsymbol{a}_{i1}\ \boldsymbol{a}_{i2}\ \boldsymbol{a}_{i3}] \quad (i = 0, 1, 2, \cdots, 5)$$

We also define the following variables.

θ_i : angle of rotation of link i about joint i.

θ_4' : angle of rotation of link 4 about joint 2. $\theta_4' = \theta_4 + \theta_3$

A_{ij} : Transformation matrix from $[\boldsymbol{a}_j]$ to $[\boldsymbol{a}_i]$.

$$A_{ii-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boldsymbol{\omega}_{ij} = [\boldsymbol{a}_i]^T\omega_{ij}$: Angular velocity vector of $[\boldsymbol{a}_i]$ to $[\boldsymbol{a}_j]$. $\omega_{ii-1}^T = [0, 0, \dot{\theta}_i]$

$\boldsymbol{r}_i = [\boldsymbol{a}_i]^T r_i$: Position vector from the origin of $[\boldsymbol{a}_{i-1}]$ to the origin of $[\boldsymbol{a}_i]$.

$\boldsymbol{R}_i = [\boldsymbol{a}_i]^T R_i$: Position vector from the origin of $[\boldsymbol{a}_i]$ to center of mass of link i.

$\boldsymbol{g} = [\boldsymbol{a}_0]^T g$: Acceleration vector of gravity

We express a cross product of vector $\boldsymbol{x} = [\boldsymbol{a}]^T x$ as $\tilde{x}$ in $[\boldsymbol{a}]$ system as

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \tilde{x} = \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{bmatrix} \qquad (1)$$

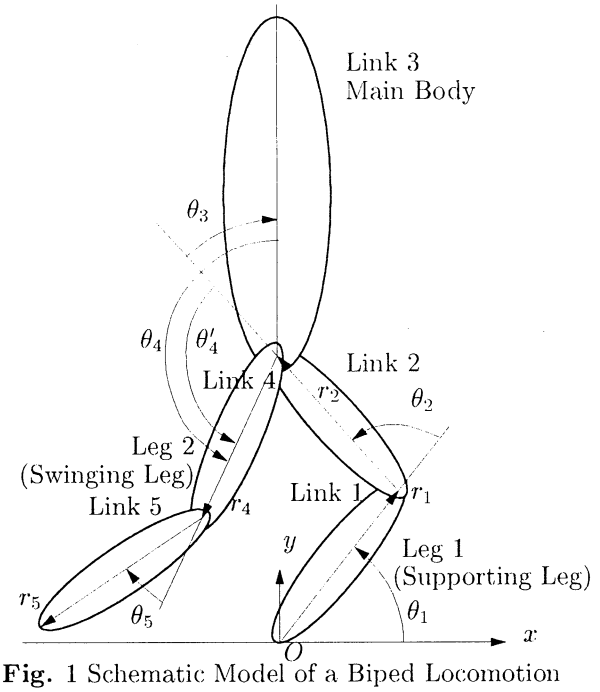


Fig. 1 Schematic Model of a Biped Locomotion

We use the vector $\hat{\omega}$ as the state variable.

$$\hat{\omega} = [\ \bar{\omega}_{10}^T\ \ \bar{\omega}_{21}^T\ \ \bar{\omega}_{32}^T\ \ \bar{\omega}_{43}^T\ \ \bar{\omega}_{54}^T\ ]^T \qquad (2)$$

$$\bar{\omega}_{ij} = A_{i0}^T\omega_{ij} \quad (i = 1, 2, \cdots, 5)$$

Generalized momenta concerning to state variable $\hat{\omega}$ are given as follows;

$$\hat{L} = \begin{bmatrix} \hat{L}_1 \\ \hat{L}_2 \\ \hat{L}_3 \\ \hat{L}_4 \\ \hat{L}_5 \end{bmatrix} = K\hat{\omega}$$

$$K = H^T(\mathcal{L}^T M \mathcal{L} + \mathcal{J})H$$

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_R$$

$$\mathcal{L}_r = \begin{bmatrix} 0 & & & & \\ \widetilde{A_{10}^T r_1} & & & O & \\ \widetilde{A_{10}^T r_1} & \widetilde{A_{20}^T r_2} & & & \\ \widetilde{A_{10}^T r_1} & \widetilde{A_{20}^T r_2} & & & \\ \widetilde{A_{10}^T r_1} & \widetilde{A_{20}^T r_2} & 0 & \widetilde{A_{40}^T r_4} & 0 \end{bmatrix}$$

$$\mathcal{L}_R = diag\left[\ \widetilde{A_{10}^T R_1}\ \ \widetilde{A_{20}^T R_2}\ \ \widetilde{A_{30}^T R_3}\ \ \widetilde{A_{40}^T R_4}\ \ \widetilde{A_{50}^T R_5}\ \right]$$

$$H = \begin{bmatrix} I_3 & & & & \\ I_3 & I_3 & & O & \\ I_3 & I_3 & I_3 & & \\ I_3 & I_3 & I_3 & I_3 & \\ I_3 & I_3 & I_3 & I_3 & I_3 \end{bmatrix}$$

$$M = diag[\ m_1 \quad m_2 \quad m_3 \quad m_4 \quad m_5\ ]$$
$$\mathcal{J} = diag[\ J_1 \quad J_2 \quad J_3 \quad J_4 \quad J_5\ ]$$

where, J_i $(i = 1, \cdots, 5)$ are inertia matrices of each link at the center of mass. m_i $(i = 1, \cdots, 5)$ are the mass matrices of each link.

Equations of motion concerning to generalized momenta $\hat{L}$ are derived as follows;

$$\dot{\hat{L}}_1 = \hat{G}_1 \tag{3}$$
$$\dot{\hat{L}}_2 + \tilde{V}_2^T \hat{P}_2 = \hat{G}_2 + \hat{u}_2 \tag{4}$$
$$\dot{\hat{L}}_3 + \tilde{V}_3^T \hat{P}_3 = \hat{G}_3 + \hat{u}_3 \tag{5}$$
$$\dot{\hat{L}}_4 + \tilde{V}_4^T \hat{P}_4 = \hat{G}_4 + \hat{u}_4 \tag{6}$$
$$\dot{\hat{L}}_5 + \tilde{V}_5^T \hat{P}_5 = \hat{G}_5 + \hat{u}_5 \tag{7}$$

where,

$$\hat{P} = \begin{bmatrix} \sum_{i=1}^{5} P_i \\ \sum_{i=2}^{5} P_i \\ P_3 + P_4 + P_5 \\ P_4 + P_5 \\ P_5 \end{bmatrix}$$
$$P = M\mathcal{L}H\hat{\omega}$$
$$\tilde{V} = diag[\ \tilde{V}_1 \quad \tilde{V}_2 \quad \tilde{V}_3 \quad \tilde{V}_4 \quad \tilde{V}_5\ ]$$
$$V = \mathcal{L}_r H\hat{\omega}$$
$$\hat{G} = H^T \mathcal{L}^T M\hat{g}$$
$$\hat{g} = [g^T g^T g^T g^T g^T]^T$$
u_i : a control torque acting at joint i.

As mentioned above, a biped locomotion machine establishes an optimal walking by using kicking forces to the ground at the moment of switching legs. Here, the kicking forces are modeled as the impulsive forces. The variance of angular momentum $\hat{L}_i$ by the impulsive forces is denoted by $\Delta\hat{L}_i$. The variances $\Delta\omega_{ij}$ of angular velocities ω_{ij} are, then, expressed in terms of $\Delta\hat{L}_i$ as

$$\sum_i K_{1i}\Delta\omega_{ii-1} = \Delta\hat{L}_1 \tag{8}$$
$$\sum_i K_{2i}\Delta\omega_{ii-1} = \Delta\hat{L}_2 \tag{9}$$
$$\sum_i K_{3i}\Delta\omega_{ii-1} = \Delta\hat{L}_3 \tag{10}$$
$$\sum_i K_{4i}\Delta\omega_{ii-1} = \Delta\hat{L}_4 \tag{11}$$
$$\sum_i K_{5i}\Delta\omega_{ii-1} = \Delta\hat{L}_5 \tag{12}$$

The variance $\Delta\omega_{32}$ of the angular velocity of the main body is expressed by using Eqs.(8) $\sim$ (12) as follows;

$$\Delta\omega_{32} = K_{13}^{-1}(K_{11}\Delta\omega_{10} + K_{21}\Delta\omega_{21} + K_{43}\Delta\omega_{43} + K_{54}\Delta\omega_{54}) \tag{13}$$

3 Formulation of Optimization

In this section, we consider an optimization of the trajectory of the biped locomotion machine.

Figure 2 shows a trajectory of a biped locomotion machine in the sagittal plane, where, S is stride.

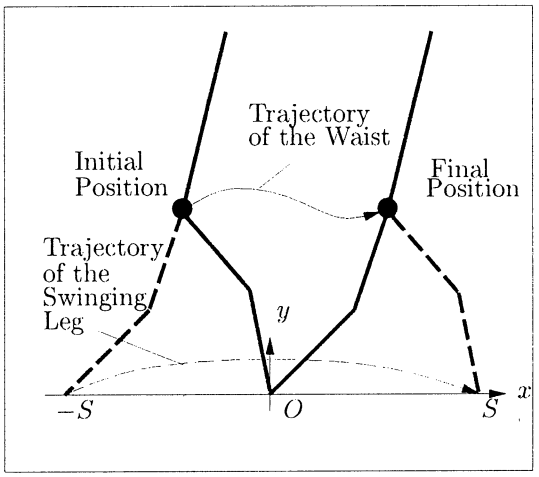


Fig. 2 Trajectory of a biped locomotion machine in the sagittal plane

3.1 Trajectories of each link

The desired stride S and desired walking-period t_f are given. And the trajectory of the end of the swinging leg $\boldsymbol{z}_c = (x_c, y_c)$ is given as a function of time as follows;

$$x_c = x_c(t, S, t_f) \tag{14}$$
$$y_c = y_c(t, S, t_f) \tag{15}$$

On the other hand, the trajectory of the waist in the sagittal plane $\boldsymbol{z}_b = (x_b(t), y_b(t))$ is expressed using Fourier series as follows;

$$x_b(t) = vt + \alpha_0 + \sum_{n=1}^{N}\left\{\alpha_{3n-2}\sin(\frac{(2n-1)\pi}{t_f}t) + \alpha_{3n-1}\cos(\frac{2n\pi}{t_f}t) + \alpha_{3n}\sin(\frac{2n\pi}{t_f}t)\right\} \tag{16}$$

$$y_b(t) = \beta_0 + \sum_{n=1}^{N} \left\{ \beta_{3n-2} \sin(\frac{(2n-1)\pi}{t_f} t) + \beta_{3n-1} \cos(\frac{2n\pi}{t_f} t) + \beta_{3n} \sin(\frac{2n\pi}{t_f} t) \right\} \quad (17)$$

$$v = \frac{S}{t_f} \quad (18)$$

Note that using this expression of the trajectory, continuity of the trajectory is satisfied strictly. On the other hand, the effect of the impulsive force acted on the system is expressed as the discontinuity of the velocity at the moment of switching legs which is expressed by the terms $\sin(\frac{(2n-1)\pi}{t_f} t)$.

The trajectories of each link of the legs are calculated as functions of the trajectory of the waist and that of the end of the swinging leg as follows;

$$\left.\begin{aligned} \theta_i &= \theta_i(t, x_b, y_b) \\ \omega_{ii-1} &= \omega_{ii-1}(t, x_b, y_b, \dot{x}_b, \dot{y}_b) \\ \dot{\omega}_{ii-1} &= \dot{\omega}_{ii-1}(t, x_b, y_b, \dot{x}_b, \dot{y}_b, \ddot{x}_b, \ddot{y}_b) \\ & \qquad (i = 1, 2) \end{aligned}\right\} \quad (19)$$

$$\left.\begin{aligned} \theta'_4 &= \theta'_4(t, x_c, y_c) \\ \theta_5 &= \theta_5(t, x_c, y_c) \\ \dot{\theta}'_4 &= \dot{\theta}'_4(t, x_c, y_c, \dot{x}_c, \dot{y}_c) \\ \omega_{54} &= \omega_{54}(t, x_c, y_c, \dot{x}_c, \dot{y}_c) \\ \ddot{\theta}'_4 &= \ddot{\theta}'_4(t, x_c, y_c, \dot{x}_c, \dot{y}_c, \ddot{x}_c, \ddot{y}_c) \\ \dot{\omega}_{54} &= \dot{\omega}_{54}(t, x_c, y_c, \dot{x}_c, \dot{y}_c, \ddot{x}_c, \ddot{y}_c) \end{aligned}\right\} \quad (20)$$

where, $\theta'_4 = \theta_4 + \theta_3$ is the angle of rotation of link4, which is temporarily introduced in order to split the motion of the main body and swinging leg on the inverse kinematics.

On the other hand, the trajectories of the main body θ_3, ω_{32} are determined as follows; It is assumed that the inclination angle of the main body to the inertial space is small.

$$\theta_1 + \theta_2 + \theta_3 - \pi/2 \simeq 0 \quad (21)$$

Then, Eq. (3) is linearized as

$$\dot{\omega}_{32} = F_1(\theta_i, \omega_{ii-1}, \dot{\omega}_{ii-1}\theta'_4, \dot{\theta}'_4, \ddot{\theta}'_4)\theta_3 + F_2(\theta_i, \omega_{ii-1}, \dot{\omega}_{ii-1}, \theta'_4, \dot{\theta}'_4, \ddot{\theta}'_4) \quad (22)$$
$$(i = 1, 2, 5)$$

In Eq. (22) θ'_4, $\dot{\theta}'_4$, $\ddot{\theta}'_4$, θ_i, ω_{ii-1}, $\dot{\omega}_{ii-1}$ $(i = 1, 2, 5)$ are already calculated at Eqs. (19)(20), so Eq.(22) is a linearized differential equation in terms of only variable θ_{32} left unknown that determines the motion of the main body. In order to determine the trajectory of the main body, we have to solve Eq.(22) as a boundary condition problem.

A boundary condition in terms of θ_{32} is given as follows;

$$\theta_3(0) = \theta_3(t_f) + \{\theta_2(t_f) + \theta_1(t_f) - \theta_2(0) - \theta_1(0)\} \quad (23)$$

Eq.(23) indicates that the inclination angles of the main body to the inertial space at the initial and final states have to coincide with each other.

The trajectory of the main body is given as the solution of Eq.(22) under the boundary conditions Eqs. (13) , (23).

3.2 Input Torque

Using the inverse dynamics, the input torque of each joint during single-supporting-phase is given as follows;

$$\hat{u}_2(\theta_i, \omega_{ii-1}, \dot{\omega}_{ii-1}) = \dot{\hat{L}}_2 + \tilde{V}_2^T \hat{P}_2 - \hat{G}_2 \quad (24)$$

$$\hat{u}_3(\theta_i, \omega_{ii-1}, \dot{\omega}_{ii-1}) = \dot{\hat{L}}_3 + \tilde{V}_3^T \hat{P}_3 - \hat{G}_3 \quad (25)$$

$$\hat{u}_4(\theta_i, \omega_{ii-1}, \dot{\omega}_{ii-1}) = \dot{\hat{L}}_4 + \tilde{V}_4^T \hat{P}_4 - \hat{G}_4 \quad (26)$$

$$\hat{u}_5(\theta_i, \omega_{ii-1}, \dot{\omega}_{ii-1}) = \dot{\hat{L}}_5 + \tilde{V}_5^T \hat{P}_5 - \hat{G}_5 \quad (27)$$
$$(i = 1, \cdots, 5)$$

On the other hand, the impulsive torques acted at each joint are expressed as follows;

$$\Delta\hat{u}_2 = \Delta\hat{L}_2 \quad (28)$$

$$\Delta\hat{u}_3 = \Delta\hat{L}_3 \quad (29)$$

$$\Delta\hat{u}_4 = \Delta\hat{L}_4 \quad (30)$$

$$\Delta\hat{u}_5 = \Delta\hat{L}_5 \quad (31)$$

3.3 Performance Index

The performance index is composed of two parts. The first part evaluates the cost of the input torque of each joint in the single-supporting-phase. The other one estimates the effects of impulsive forces at the moment of switching legs. The performance index is defined as a quadratic form as follows;

$$J = J_L + J_H \quad (32)$$

$$J_L = \int_0^{t_f} \sum_{i=2}^{5} \frac{R_i}{K_{Ti}^2} \hat{u}_i^2 dt \quad (33)$$

$$J_H = W \sum_{i=2}^{5} \hat{u}_i^2 \quad (34)$$

where, W is a coefficient and R_i,K_{Ti} are values of electrical resistor and torque coefficient of each actuator, respectively.

The first term of the performance index evaluates the energy consumption at the coils of electric DC actuators in the single-supporting-phase. The second term evaluates the impulsive one.

3.4 Algorithm

A motion planning algorithm of a biped locomotion machine proposed is shown in **Fig.** 3.

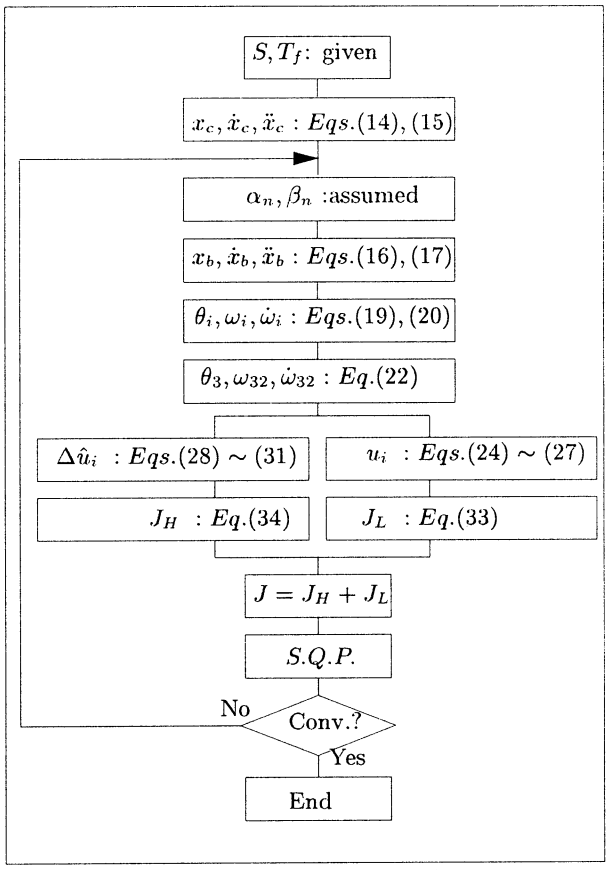


Fig. 3 Flow chart of a motion planning (S.Q.P. : Successive Quadratic Programming Method)

4 Numerical Examples

Using Motion Planning algorithm proposed, dynamic performances a biped locomotion machine are investigated. Table 1. shows physical parameters of a biped locomotion machine, which are those of an experimental setup.

Table 1 Parameters

	Length [m]	Mass [kg]	Inertia [kgm^2]
Link 1,5	0.20	1.5	5.0×10^{-3}
Link 2,4	0.20	2.0	7.0×10^{-3}
Link 3	0.30	5.0	1.0×10^{-2}

The trajectory of the waist is expressed using Fourier series up to the 2nd mode.

Figures 4 and 5 show stick figures of a biped locomotion machine corresponding to the optimal solutions, **Fig.** 4 shows the case without the impulsive forces and **Fig.** 5 shows that with the impulsive forces. **Fig.** 6 shows the trajectory of the waist with weight value of the performance index W as a parameter. From these figures, an optimal trajectory of a biped locomotion machine is planned with the algorithm proposed in this paper, and we may note that a biped locomotion machine has some walking patterns according to whether it use the impulsive forces actively or not. When W is small, the optimal motion derived has a feature that center of mass of the system is almost conserved in height. On the other hand, when W is large, the optimal motion of the system derived becomes like inverted pendulum.

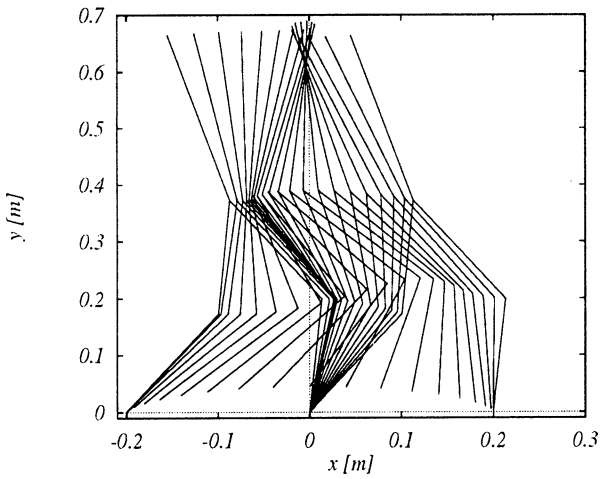


Fig. 4 Stick figure of a biped locomotion machine (without impulsive force)

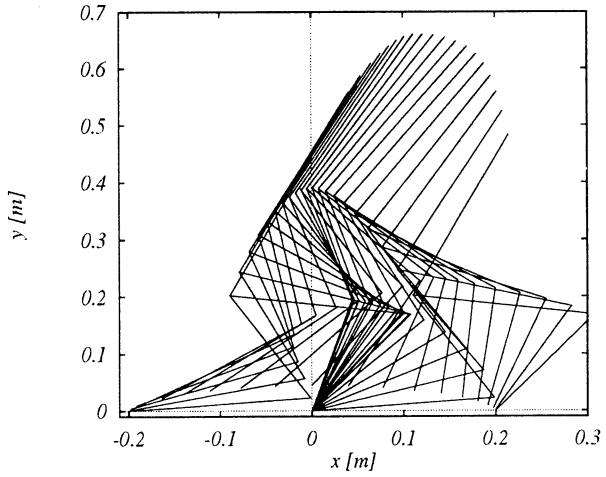


Fig. 5 Stick figure of a biped locomotion machine (with impulsive force)

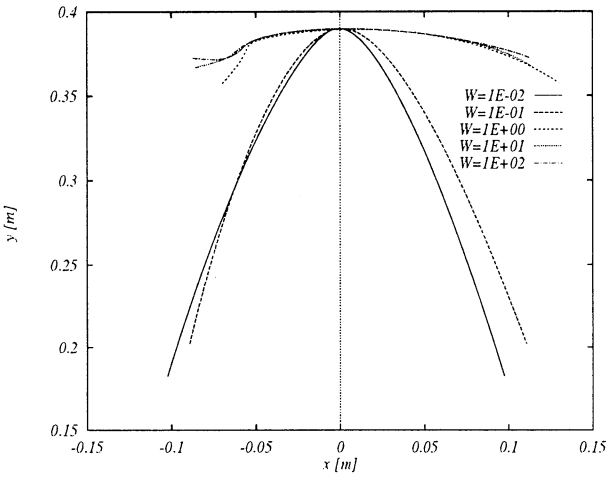


Fig. 6 Trajectory of the waist (W as parameter)

5 Conclusion

A motion planning method of a biped locomotion machine is proposed. The trajectory of the waist is expressed by Fourier Series satisfying continuity conditions of the trajectory at the moment of switching legs and it also includes the terms that express discontinuities of the velocities at the moment of switching legs as the effects of the impulsive forces. Based on the inverse kinematics, the trajectories of each joint are derived. Then, using inverse dynamics, the input torques are calculated, and the performance index is defined by two parts, that is, quadratic form of the input torques and the effects of impulsive forces at the moment of switching legs. For the optimization algorithm, S.Q.P. method is used. Using these formulation, an optimal trajectory of a biped locomotion machine is obtained.

References

[1] K.Tsujita and K.Tsuchiya : An Experimental Study of a Biped Locomotion Machine using Reaction Wheels, Experimental Robotics IV the 4th International Symposium: Lecture Notes in Control and Information Sciences 223, Springer-Verlag, 558–565 (1997)

[2] A.E.Bryson, Jr. and Yu-Chi Ho : Applied Optimal Control, Hemisphere Pub.Corp. (1975)

[3] Y.Wada and M.Kawato : A neural network model for arm trajectory formulation using forward and inverse dynamics models, Neural Networks, Vol. 6, No. 7, 919–932 (1993)

[4] C.Fernandes, L.Gurvit and Z.X.Li : Optimal Nonholonomic Motion Planning for a Falling Cat, Nonholonomic Motion Planning, KAP, 379–421 (1992)

[5] T.Akiyama and Y.Sakawa : A motion planning of space robot using nonlinear planning method, SICE, Vol.31, No.2, 193-197 (1995) (in Japanese)

[6] T.McGeer : Passive Dynamic Walking, International Journal of Robotics Research, Vol.9, No.2, 62-81, April 1990

[7] Ming-Shaung Ju and J.M. Mansour : Simulation of the Double Limb Support Phase of Human Gait, ASME Journal of Biomechanical Engineering, Vol.110,223-229 August 1988

[8] M.L.Audu and D.T.Davy : The Influence of Muscle Model Complexity in Musculoskeletal Motion Modeling, ASME Journal of Biomechanical Engineering, Vol.107,147-157 May 1985

[9] C.-S.Tsai and J.M.Mansour : Swing Phase Simulation and Design of Above Knee Prostheses, ASME Journal of Biomechanical Engineering, Vol.108,65-72 February 1986

[10] G.Taga, Y.Yamaguchi and H.Shimizu : Self-organized Control of Bipedal Locomotion, HOLONICS Vol.2 No.2, 131-151 (1991) (in Japanese)

[11] Y.Hurmuzlu : Dynamics of Bipedal Gait: Part I – Objective Functions and the Contact Event of a Planar Five-Link Biped, ASME Journal of Applied Mechanics, Vol.60 331–336 June 1993

[12] Y.Hurmuzlu : Dynamics of Bipedal Gait: Part II – Stability Analysis of a Planar Five-Link Biped, ASME Journal of Applied Mechanics, Vol.60 337–343 June 1993

[13] Y. Hurmuzlu : On the Measurement of Dynamic Stability of Human Locomotion, ASME Journal of Biomechanical Engineering, Vol.116, 30–36 February 1994

[14] H. Takeuchi and T. Ohtsuka : Optimal Walking Problem for Biped Robot, Proceedings of the 15th Annual Conference of the Robotics Society of Japan, Vol.1, 95-96 (1997) (in Japanese)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

A Simulation Study on Cooperative Behavior in a Fish School

N. Sannomiya
Dept. of Electronics and
Information Science
Kyoto Institute of Technology
Kyoto 606, Japan

H. Nakamine
Dept. of Industrial
Arts Education
Kyoto University of Education
Kyoto 612, Japan

Abstract

Various patterns of fish behavior are obtained according to different environments around the school. In this paper a fish school is considered as an autonomous decentralized system. Each individual behaves autonomously, and a specific order of the school is established based on the environmental effect and the information exchange among other individuals. The relationship between the order of the school and the quantity of the information exchange is investigated by using the physical model, which consists of a propulsive mechanism, an environmental effect and a schooling mechanism. As an example of cooperative behavior in a fish school a simulation is made for the case where the school is affected by a box-shaped trap.

1 Introduction

The schooling behavior of fish is a complicated phenomenon which shows different patterns adapted to various environments. The fish behavior has been investigated from the viewpoints of ethology and fisheries science[1-5]. Of these Shaw[2] summarized that the adaptive change of behavior patterns comes from many reasons ; shelter from enemy attack, economy in energy, making a pretence of big fish and so on. It is observed that the characteristics of the fish behavior includes diversity and redundancy. Therefore, by making reference to such an ethological knowledge, we can aim at establishing a new process design tool. This design tool may be useful to solve complicated engineering problems such as a multi-agent scheduling system[6].

It is difficult to clarify the principle of fish behavior using only the traditional methods of fisheries science. Some of the approaches developed in systems engineering may be applicable to consider this problem. In our earlier papers[7-9], a mathematical model was proposed to consider the nature of fish behavior.

In this paper, we consider a fish school as an autonomous decentralized system. The autonomous decentralized system is a system whose functional order is dependent upon cooperative interactions among its subsystems. In other words the system does not have any supervisor for the entire system. Instead, each subsystem has the autonomy to control each part of the system.

Computer simulations are executed for the case where the school is affected by a box-shaped trap. The trap gives the fish school a strong constraint. Consequently an escape from the trap becomes a difficult work for the school. The relationship between the order of the school and the state of each individual is investigated under this environment. Furthermore the condition for escape from the trap is obtained based on the simulation results.

2 Model Description

In general, fish swim in a three-dimensional space. However, traveling in two dimensions is more important than motion in the direction of depth. Then, for simplicity the motion of fish is assumed to be restricted within a two-dimensional space. Let the position and the velocity of individual fish i be x_i and v_i, respectively, where x_i, $v_i \in R^2$. Then the motions of N_f fish in a school can be described by[9]

$$\left.\begin{aligned} \dot{x}_i &= v_i \\ m_i\dot{v}_i &= F_i(t, x_1, v_1, x_2, v_2 \cdots, x_{N_f}, v_{N_f}) \\ i &= 1, 2, \cdots, N_f \end{aligned}\right\} \quad (1)$$

where m is the mean mass of the individuals. F_i is the resultant of all forces which cause the motion of individual i. In our model F_i consists of three components, i.e. $F_i = F_{i1} + F_{i2} + F_{i3}$, which are given as follows.

2.1 Force due to his own swimming ability

An individual fish has the character of swimming forwards at his own favorite speed (called the characteristic velocity) when other causes do not exist for the motion of fish. The character is expressed by

$$F_{i1} = -a_i^1(||v_i|| - a_i^2)(||v_i|| - a_i^3)v_i \qquad (2)$$

where $0 < a_i^2 < a_i^3$. As a result of phase plane analysis, in the case of $a_i^1 > 0$, the characteristic velocity of individual i is given by a_i^3, and in the case of $a_i^1 < 0$, the characteristic velocity is a_i^2. We call this F_{i1} a propulsive force.

2.2 Interaction among other fish

An individual fish regulates his position and velocity with the positions and velocities of other individuals in school. The character is given by

$$F_{i2} = \sum_{j\in N(i)} b_i(r_{ij})\frac{x_j - x_i}{r_{ij}} + \sum_{j\in N(i)} c_i(r_{ij})\frac{v_j - v_i}{M} \qquad (3)$$

$$b_i(r_{ij}) = \begin{cases} \dfrac{(k_{bi}^2 - k_{bi}^1)r_{ij}}{\alpha_{i1}} + k_{bi}^1 & \text{for } 0 < r_{ij} \le \alpha_{i1} \\ k_{bi}^2 & \text{for } \alpha_{i1} < r_{ij} \le \alpha_2 \\ 0 & \text{for } r_{ij} > \alpha_2 \end{cases} \qquad (4)$$

$$c_i(r_{ij}) = \begin{cases} k_{ci} & \text{for } 0 < r_{ij} \le \delta \\ 0 & \text{for } r_{ij} > \delta \end{cases} \qquad (5)$$

where $r_{ij} = ||x_j - x_i||$. The first term of (3) is an interactive force to keep a proper distance between neighboring individuals. The second term is a schooling force to make the velocity of each fish uniform. In (3), $N(i)$ is the subset of $\{1, 2, \cdots, i-1, i+1, \cdots, N_f\}$ whose element consists of the individual number existing near individual i. Let M be the number of the elements in $N(i)$.

The functions $b_i(r_{ij})$ and $c_i(r_{ij})$ represent the dependence of the respective forces on the distance r_{ij}.

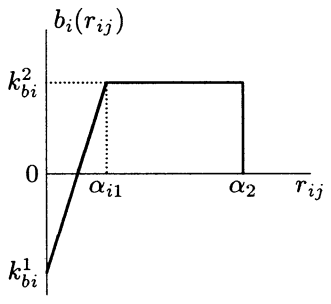


Fig.1 The graph of b_i

Figure 1 shows the graph of $b_i(r_{ij})$. The form of $b_i(r_{ij})$ should be chosen so that the repulsion ($b_i < 0$) acts on individual i for small r_{ij} and the attraction ($b_i > 0$) acts for large r_{ij}. Alternatively, the form of $c_i(r_{ij})$ is chosen so that the schooling force acts at a constant magnitude between the individuals existing within a critical distance.

2.3 Environmental effect

An individual fish receives a certain stimulus from the environment. In the case of our model, the wall of a water tank and an obstacle are environmental elements that give any effect for fish motion. When a fish school swims in a water tank, it swims along the wall in many cases. However, even if a fish approaches a wall very closely, the fish never strikes against it. Consequently, there are both an attractive element and a repulsive element for fish school. Then, we have as the third component:

$$F_{i3} = k_{wi}^+ \sum_{l=1}^{L} f_{wil}^+ + k_{wi}^- \sum_{l=1}^{L} f_{wil}^- \qquad (6)$$

$$f_{wil}^+ = \begin{cases} v_{il}e_l\dfrac{d^+ - d_{il}}{d^+} & \text{for } v_{il} > 0 \text{ and } d_{il} < d^+ \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

$$f_{wil}^- = \begin{cases} v_{il}e_l\dfrac{d^- - d_{il}}{d^-} & \text{for } v_{il} < 0 \text{ and } d_{il} < d^- \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

where L is the number of the wall sides of the environmental elements. The unit vector e_l is normal to the wall l, and v_{il} is the velocity component normal to the wall l, being given by $v_{il} = -e_l^{\mathrm{T}}v_i$. The quantity d_{il} means the distance from the individual i to the wall l.

In (6), the first and the second term are the repulsive and the attractive force from the element, respectively. The repulsive force acts when the individual approaches the wall ($v_{il} > 0$), and the attractive force acts when it goes away from the wall ($v_{il} < 0$). Both forces act when the individual is located near the wall ($d_{il} < d^+$ or $d_{il} < d^-$).

2.4 Model parameters

The parameters included in the model should be determined based on the observation data. In order to construct the model for fish behavior, we examined the observation data of a water tank experiment. The experiment was carried out at Tokyo Univerisity of Fisheries[8]. The water tank is rectangular. It is 150cm long and 100cm wide. The water in the tank

Table 1 Parameter estimates ($N_f = 5$)

i	a_i^1 (g·sec/cm^2)	a_i^2 (cm/sec)	a_i^3 (cm/sec)	k_{bi}^1 (g·cm/sec^2)	k_{bi}^2 (g·cm/sec^2)	k_{ci} (g/sec)	α_{i1} (cm)	k_{wi}^+ (g/sec)	k_{wi}^- (g/sec)
1	-0.00366	11.9	35.0	-6.20	4.20	1.57	8.47	10.6	3.19
2	-0.00275	11.0	38.4	-6.40	4.38	1.31	8.44	13.1	2.90
3	-0.00330	10.9	37.9	-4.40	3.11	1.49	8.35	7.3	3.58
4	-0.00180	10.6	46.4	-5.60	2.57	1.30	9.64	19.8	3.91
5	-0.00122	10.4	52.9	-6.90	4.02	1.46	8.88	13.8	3.17

$m_i = 2.58$g, $\delta = \alpha_2 = 50$cm, $d^+ = 5$cm, $d^- = 20$cm

is 7cm deep. The fish used in the experiment are bitterling (Rhodeus ocellatus ocellatus), range in length from 3 to 5cm.

The unknown parameters included in the model were estimated by using the time series data obtained by the experiment. The time series data is sampled at the intervals of $\Delta t = 0.5$sec. The number of data is chosen as 1200 for each individual. The model parameters for the school of N_f individuals are determined by applying the least squares algorithm to the time series data. Table 1 shows the parameter estimates thus obtained, where the number of individuals in the school is set to $N_f = 5$.

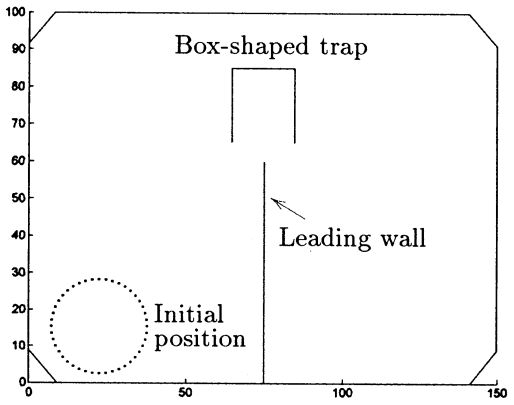


Fig.2 Configuration in the water tank

3 Simulation Results

Simulation was carried out by using a model with the parameter values shown in Table 1. The number of fish for simulation is twenty i.e. $N_f = 20$. We assume that twenty fish consist of five groups, and each group has four fish with the same parameters. In other words, the individual number i shown in Table 1 corresponds to the group number. Because all the parameters a_i^1 are negative, a_i^2 means the characteristic velocity for the individuals of group i. Therefore, it is found out that though a characteristic velocity is fast in order of the group number, the difference is not so large among the groups.

As a case study we investigate a fish school swimming in the water tank shown in Fig.2. There are a box-shaped trap and a leading wall in the tank. At one side of the box the trap has an entrance from which fish can enter and go out freely.

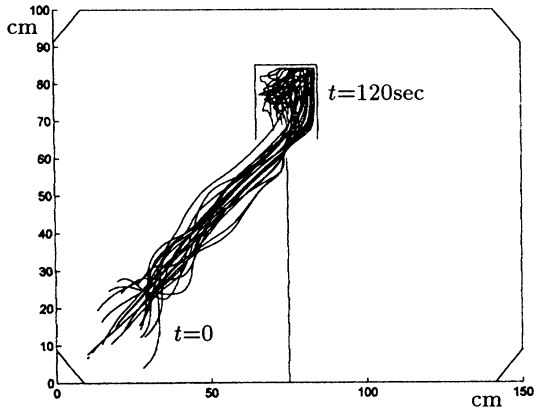
Figure 3 shows the trajectories of twenty fish during a 120s time interval. Three figures (a), (b) and (c) show the results for the respective values of M. Although the initial condition of the school is same for three cases(see Fig.2), the succeeding trajectories are quite different.

Figure 3(a) shows the simulation result for the case where $M = 19$. It is found that a few fish enter the trap, and then the other fish also enter the trap. The school does not go out of the trap, because the individuals located near the entrance receive the influence of many individuals located at the inside of the trap.

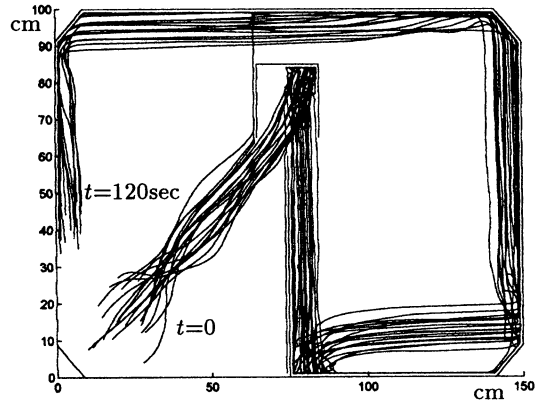
Figure 3(c) shows the result for the case where $M = 4$. In this case, from the first the fish school is decomposed into a subschool staying in the trap and a subschool swimming outside the trap. After that, the subschool outside the trap enters into the trap. However, the school in the trap is soon divided into two subschools and one subschool goes out of the trap. The reason is that, because M is small, the information is exchanged only locally among other individuals. Consequently an order can not be established as the whole school.

Figure 3(b) shows the result for the case where $M = 13$ as an example of the moderate value of M. It is found out that the fish school enters the trap from the first. Afterwards a few fish go out of the trap, and then the other fish also go out. Consequently the whole school goes out of the trap.

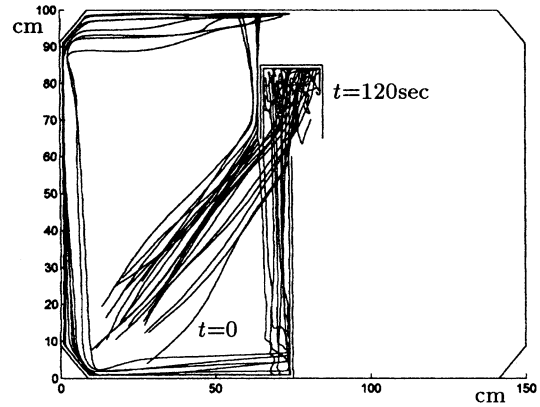
We consider two individuals, i.e. an individual i and the individual j located at the nearest position from i. When the angle between two velocity vectors v_i and v_j is less than 0.1 rad, the individual i is prescribed to keep an order.



(a) $M = 19$



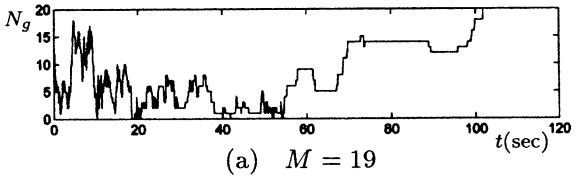
(b) $M = 13$



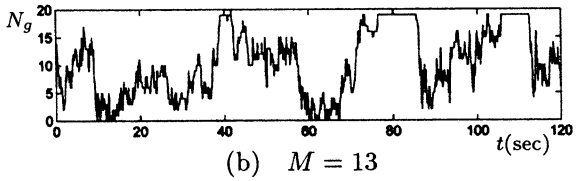
(c) $M = 4$

Fig.3 Fish behavior affected by a box-shaped trap (t=0~120sec)

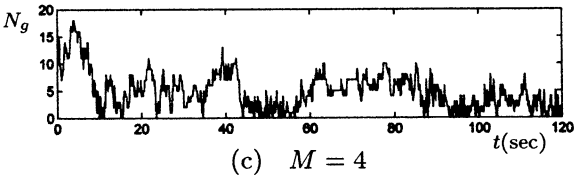
We define N_g as the number of individuals keeping an order. Figure 4 shows the variation of N_g with time. As shown in this figure, for the case where $M = 19$, an order is established locally at the first step, and the order spreads in the whole school gradually at the next step. In addition the order established



(a) $M = 19$



(b) $M = 13$



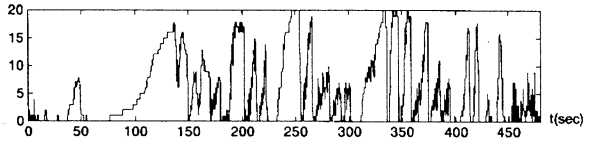
(c) $M = 4$

Fig.4 Time variation of the number of individuals keeping an order

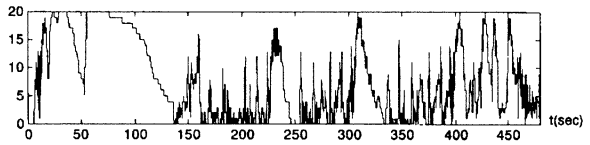
once is not broken easily. On the other hand, the order is not established on the whole school for the case where $M = 4$. As shown in Fig.3, these two results correspond to the cases where a part of individuals in the school are captured in the trap.

In the case where $M = 13$, it is found out that the establishment and the collapse of the order are realized easily and are repeated periodically. By the correspondence with the trajectory of the fish school, we find that the order of the school collapses when the fish school comes across the box-shaped trap or the corner of the water tank. Subsequently, the order is established gradually in the inside of the trap or at the corner. When the establishment of the order is completed, the fish school begins to move again. In other words, when M takes a moderate value, a transition between the order establishment and the order collapse can be realized easily under the influence of the environment. Consequently the fish school can escape from the trap promptly and flexibly.

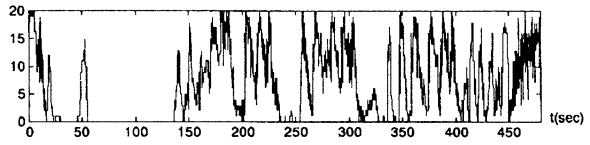
In the case where $M = 19$, it is found out that the escape from the trap is difficult. However, if time passes sufficiently ($t >$ 130sec), the whole school can escape from the trap. In order to investigate the process of escape, we calculate the magnitudes of the three forces contained in (1) at each time instant and for each individual. Let $N_j (j = 1, 2, 3)$ be the number of individuals whose $|F_{ij}|$ has the largest value among three forces.



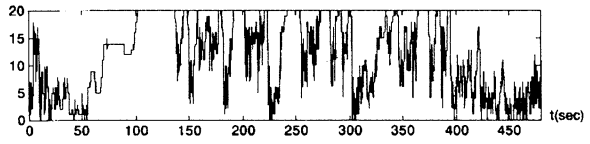
(a) N_1



(b) N_3



(c) N_2



(d) N_g

Fig.5 Process of escape from a box-shaped trap (M = 19, t=0~480sec)

Figure 5 shows the simulation result during a long time interval, i.e. t=0~480sec. It shows the time variations of N_1, N_3, N_2 and N_g. As shown in the figure, N_1 increases gradually during t=100~130sec, while N_3 decreases. N_2 increases at $t \approx$ 130sec abruptly. This result is explained in the following way. That is, at the first step the individuals are captured in the trap by the attractive force in F_{i3}. At the next step, some individuals with larger value of F_{i1} begin to move towards the entrance of the trap. As such individuals increase, the order of the fish school is establised gradually as shown in Fig.5(d). Finally the individuals in order lead the individuals in disorder to the entrance by the effect of F_{i2}. Consequently, the whole school can escape from the box-shaped trap.

The trajectory of the fish school during 240s time interval is shown in Fig.6. It is observed from the figure that the school can escape from the trap at $t = 130$ sec. Just then the school comes across the leading wall and is decomposed into two subschools. At $t = 180$ sec the subschools meet again at the upper left corner of the tank. Afterwards they continue to behave as one school.

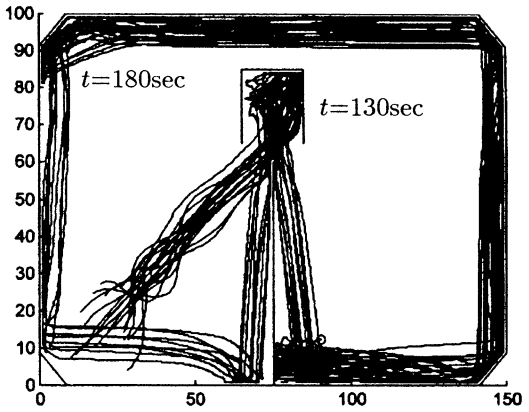


Fig.6 Fish behavior after escape from the trap ($M = 19$, $t = 0 \sim 240$ sec)

4 Summary

A mathematical modeling and simulation has been carried out to show various patterns of fish behavior adapted to different environments. When the school comes across a box-shaped trap, three behavior patterns are obtained, i.e, (i) the school continues to stay in the trap, (ii) the school is divided into subschools and (iii) the school escapes from the trap. The case of (iii) corresponds to the moderate value of M. In this case the decision as the whole school can be made easily adapted to variation of the environment. In this sense such a system is considered to be flexible.

On the other hands, in the case of (i) the school with M = 19 is considered as a rigid decision-making system because each individual has to behave unanimously. Consequently the adaptation to environmental variations is made at a slow pace. In addition it is noted that, in the case where M = 19, the deadlock state such as keeping the school in a trap may be solved both by making a selfish effort of each member and by satisfying the unanimous condition for such efforts.

Finally we point out that the present fish behavior model is a redundant system and does not have any specific objective. Its function consists only of swimming at a favorite speed of each member and making a school cooperatively. Nevertheless the individuals can get out of the deadlock state by themselves even if the behavior is slow and is not optimal. Such an ethological knowledge may be useful to construct a tough process design tool for practical optimization problems.

Acknowledgements

This work is partly supported by the "Research for the Future" Program JSPS-RFTF97I00102 of the Japan Society for the Promotion of Science and also the Grant-In-Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant : Priority-Area Research No.07243102.

References

[1] C. M. Breder, "Studies on social grouping in fishes" *Bull. Amer. Mus. Natural Hist.*, Vol.117, pp.399-468, 1959.

[2] E. Shaw, "Schooling in fishes" in *Development and Evolution of Behavior*, L. S. Aronson et al. Eds., Freeman, pp.452-480, 1970.

[3] D. V. Radakov, *Schooling in the Ecology of Fish*, Wiley, 1973.

[4] M. Inoue, *Fish Behavior and Fishing*, Koseisha Koseikaku, pp.188-204, 1978 (in Japanese).

[5] B. L. Partridge, "The structure and function of fish schools" *Scientific American*, pp.90-99, 1982.

[6] H. Iima, R. Kudo, N. Sannomiya and Y. Kobayashi, "An autonomous decentralized scheduling algorithm for a job shop process with multi-function machine in parallel" Proc. of 3rd Intern. Symp. on Autonomous Decentralized Systems, pp.69-74, 1997.

[7] N. Sannomiya and K. Matuda, "A mathematical model of fish behavior in a water tank" *IEEE Trans. on Systems Man Cybernetics*, Vol.SMC-14, No.1, pp.157-162, 1984.

[8] N. Sannomiya, H. Nakamine and K. Matuda, "Application of system theory to modeling of fish behavior" Proc. of 29th IEEE Conf. on Decision and Control, pp.2794-2799, 1990.

[9] N. Sannomiya and M.A. Doustari, "A simulation study on autonomous decentralized mechanism in fish behavior model" *Intern. Jour. Systems Science*, Vol.27, No.10, pp.1001-1007, 1996.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998, begining page-ending page

Developing Artificial Retina by Evolutionary Cellular Automata and Self-organizing Neural Networks

Xiaoqiang Wu
Institute of System Science
Academia Sinica
Beijing, P.R.China,100080

Yongguang Zhang
Institute of System Science
Academia Sinica
Beijing, P.R.China, 100080

M.Sugisaka
Dept. of E.&E Engineering
Oita university
Oita, Japan, 870-11

Abstract

Retinas are very important for human-beings to get information from environments. In this paper, we propose a new method to build artificial retinas which have many features similar to real ones. We use evolutionary cellular automata to extract some basic characteristics of objects and use self-organizing neural networks to distinguish different objects.

Keywords: artificial retina, cellular automata, evolutionary computation, self-organizing neural networks.

1 Introduction

The ability of observing is very important for autonomous robots to survive in the complex and fast changing world. Great efforts have been done to make robots have their own eyes. Many devices are invented. Most of them depend on a camera and a image processor, which are called artificial retinas[1]. Various methods are used to do feature extraction in artificial retinas, such as mathematical transforms and neural networks[2][3]. These methods really work, they can do edge detection, recognize alphabets and so on. But they have a common weakness, that is the algorithms they used are not parallel and take a lot of time on computation and training.

By comparing and contrasting principles that underlie the human visual experience and studies of mammalian neurophysiology, we advance a new methodology to build artificial retinas which make use of cellular automata to do feature extraction. This idea comes from the following aspects. Cellular automata are famous for their parallelism and the structure[4] of 2-D cellular automata fits well with CCD cameras. Detail explanations can be found in the following sections. On the foundation of the preprocessing of cellular automata, self-organizing neural networks are used to do vision integration.

This paper is organized as following: in section 2, the system model is illustrated; in section 3, evolutionary cellular automata are discussed; in section 4, the framework of vision integration by using self-organizing neural networks is given; in section 5, we summarize the advances and weaknesses of this methodology, potential applications of this methodology are also discussed.

2 The Model of Artificial Retina

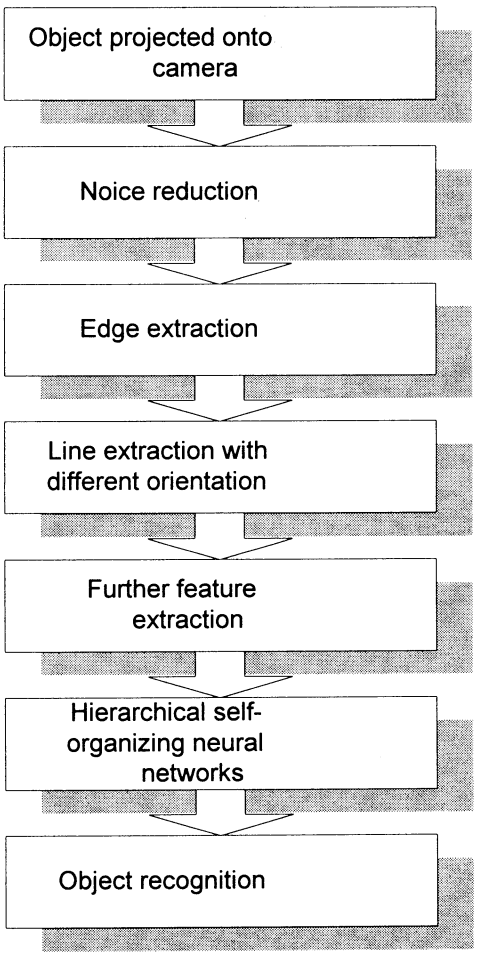


Fig.1

Fig.1 introduces the basic modules and processing steps of our artificial retina. When the

Object is projected onto the camera, it first passes through a noise reduction module, then edge extraction module is used to do edge detection on this image. After edge extraction, the image is inputted into several feature extraction modules , these modules can detect lines, angles and so on. At last we use self-organizing neural networks to integrate all this features and distinguish different objects.

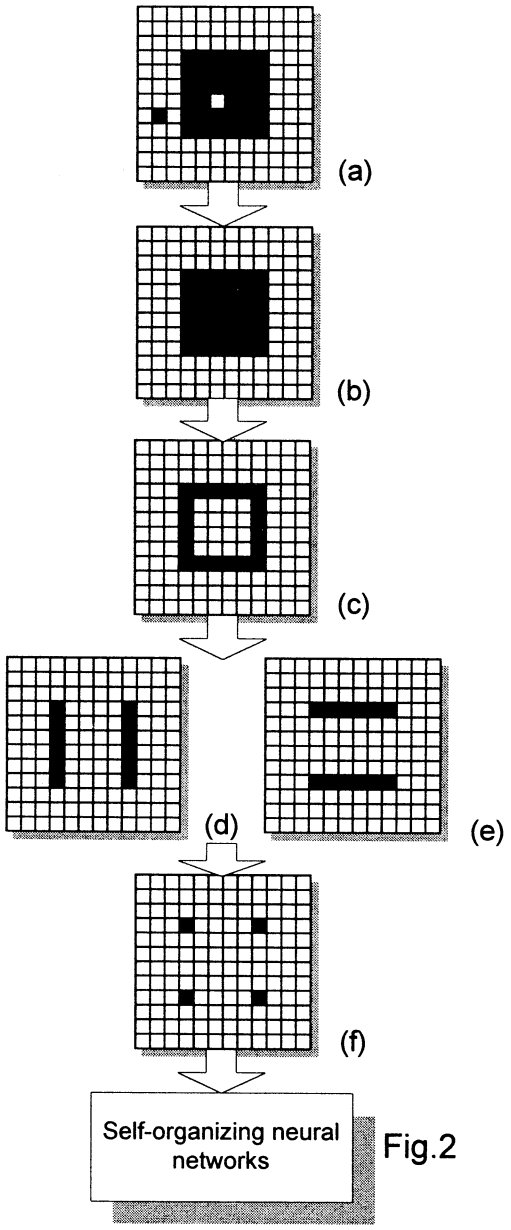


Fig.2

Here, we give an example of how this artificial retina recognize a square.(Fig.2) For simple, we suppose that the image shot by camera(Fig.2a) is a binary image and each cell in automata changes its color(black or white) according to its eight direct neighbors' conditions. The cellular automata used in this example are listed below.

cellular automaton one: if the neighbors of a cell are all different from itself, then the cell reverse its color. (used in Fig.2a to get Fig.2b)

cellular automaton two: if the neighbors of a cell are all the same with itself, then the cell turn to white. (used in Fig.2b to get Fig.2c)

cellular automaton three: if a cell is black, and either its upstairs neighbor or its downstairs neighbor is black then the cell remains unchanged, else it turn from black to white. (used in Fig.2c to get Fig.2d)

cellular automaton four: if a cell is black, and either its left neighbor or its right neighbor is black then the cell remains unchanged, else it turn from black to white. (used in Fig.2c to get Fig.2e)

Further feature extraction can be done by cellular automata, neural networks, digital logic and so on. For example, by using a simple AND logic operator between Fig.2d and Fig.2e, bottom detection Fig.2f is derived.

Similar structures can be found in human-being's primary visual cortex, where exist parallel situated line detecting columns and further feature extraction is done depending on the outputs of these line detection neurons[5]. Self-organizing neural networks are used to recognize objects, we use hierarchical structures to do this, it will be discussed in section 4.

3 Evolutionary Cellular Automata

In the previous section, we give a simple example on how cellular automata can be used to do edge and line extraction. But this example is too simple to work in real world. In order to make this artificial retina work practically, complex cellular automata should be designed. In fact, it is almost impossible to do this work manually, evolution computation and artificial life are introduced.

Let us imagine a virtual world where a lot of 2-D binary cellular automata live. There also live a lot of lines, dots, and angles which are "plants" of the virtual world. These automata can "eat" lots of thing, but only lines with certain orientation are their foods, others are poisonous plants. When an automaton meets a plant, the image of this plant is projected onto the automaton, then the automaton uses its rule to eat it. If the image of the plant remains unchanged after several steps, we say it has swallowed the plant, else the automaton will leave and try to find another plant. Each automaton starts with some energy and has a restriction of age. The

automata will consume some energy when they are eating plants. when they eat the poisonous plants, their energy will also reduce; only when they eat foods, will their energy level arise.

These automata have the abilities of reproduction. When two automata come across, they can mate with each other, of course this will consume them some energy. If in a binary automaton, the next status of the cell is determined by the current status of itself and its eight neighbors, there will be 512 possible conditions the cell will meet at each step. So we need 512 bits to set a rule

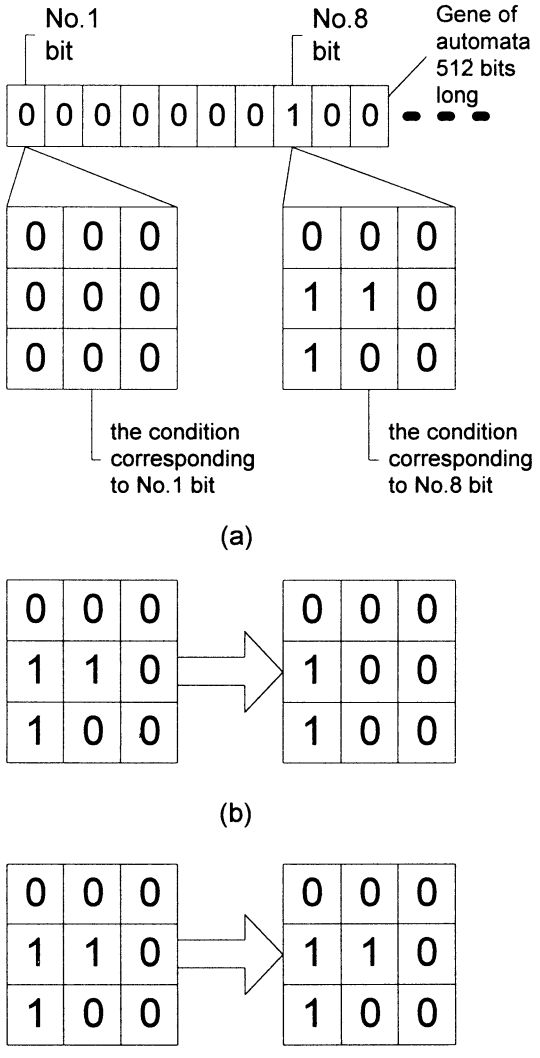


Fig.3

which actually is the gene of the automaton. Each bit represents a certain condition the cell may meet, and the value of this bit determines whether the cell will remain its status or reverse its status. For example, if the 8th bit of the gene is 1, it means under the condition Fig.3a the cell will reverse its status (Fig.3b), else it will remain its status (Fig.3c). Genetic operators, such as mutation and crossover, are used in automata's reproductions.

After several generations, Only the automata who eat foods properly can survive. These automata are good at recognizing lines with the certain orientation, they can be used in the artificial retina.

The automata which do edge extraction or further feature extraction can be obtained similarly.

4 Vision Integration by Hierarchical SOM

Self-organizing phenomena happen everywhere in human-beings' brains. To some extent, we can say they are the foundations of brain functions. Here, we use hierarchical self-organizing map(SOM) to integrate the features extracted by previous modules of the artificial retina. Compared with traditional SOM[6], we introduce a fairly novel structure.

Because in the previous modules, features of objects are extracted hierarchically—complex features are extracted from simple features, we use SOM hierarchically to do vision integration. See Fig.4. The features are not considered identical.

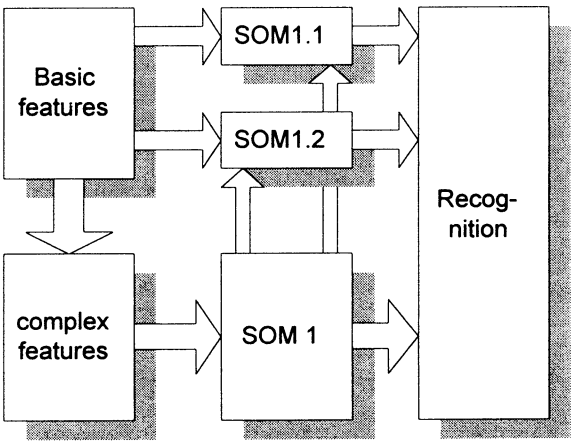


Fig.4

The more complex the features is, the more important it is, the earlier it is used in SOM.

Compared with traditional SOM, see Fig.5, hierarchical SOM saves a lot of time in training and lots of unnecessary neurons. Because if we use basic features and complex features identically in building SOM, it means the SOM have to receive too many inputs to work quickly. But in hierarchical SOM, each SOM receives much less

inputs than traditional SOM does. When complex features are sufficient to distinguish an object, the recognition result will be reported directly, otherwise outputs will be combined with basic features to do further recognition by SOM.

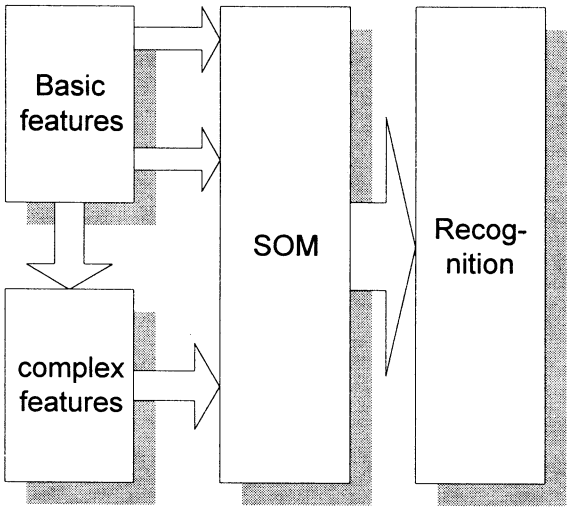


Fig.5

5 Summary

In this paper, we propose a fairly novel methodology to build artificial retinas, in which evolutionary cellular automata and self-organizing neural networks are used. For the reason of time, our paper is only a schematic description on recognizing static objects, simulations of the artificial retinas are going on. Further works on moving objects will be done to make it more practical. Cellular automata machines become popular now, we hope in the near future we can see this kind of artificial retina working on autonomous robots.

REFERENCES

[1] Kuma, K.;Lange, E. ..., "Artificial retinas—fast, versatile image processors", Nature, Vol: 372, Iss: 6502, p.197-8

[2] Funatsu, E.; Hara, K.; Toyoda, T...., "An artificial retina chip made of a 128*128 pn-pn variable-sensitivity photodetector array", IEEE Photonics Technology Letters, Vol: 7, Iss: 2, p. 188-90

[3] Kim, K.I.; Shin, C.W.; Inoguchi, S., "Collision avoidance using artificial retina sensor in ALV", Proceedings of the Intelligent Vehicles '95. Symposium

[4] J.Casti. "Reality Rules: Picturing the World in Mathematics", JOHN WILEY & SONS INC. 1992

[5] Stephen W. Kuffler, John G. Nicholls, A. Robert Martin. "From Neuron to Brain", second edition, Sinauer Associates Inc. 1984

[6] T. Kohonen. "Self-Organizing Maps", Springer-Verlag. 1995

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

Effectiveness of Matrix-based Encoding for Spanning Tree Problem on Sparse Graph

Gengui Zhou Mitsuo Gen Masato Takayama
Department of Industrial and Systems Engineering
Ashikaga Institute of Technology
Ashikaga 326, Japan

Abstract

In this paper, we propose a matrix-based tree encoding to cope with the MST problem on a sparse graph. Each chromosome consists of a matrix whose gene (element) is an integer. This encoding occupies more memory, but it is more efficient to deal with the MST problem on a sparse graph. Numerical examples show that this new encoding is effective to get the optimal or near-optimal solution of the MST problem on a sparse graph.

Keywords: Genetic Algorithm, Spanning Tree.

1 Introduction

Spanning tree is one of classical combinatorial optimization problem. In one form or another, explicitly or implicitly, spanning trees play a pivotal role in a majority of network design and analysis problems. A minimum spanning tree (MST) is usually the optimal solution for a given problem in those practical cases. It has been considered the most fundamental topology in telecommunication network in the last decades.

Consider a connected undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of *vertices* representing terminals or telecommunication stations *etc.*, and $E = \{e_1, e_2, \ldots, e_m\}$ is a finite set of *edges* representing connections between these terminals or stations. Each edge has an associated positive real number denoted with $W = \{w_1, w_2, \ldots, w_m\}$ representing distance, cost and so on. Let the $\boldsymbol{x} = (x_1 x_2 \ldots x_m)$ be defined as follows:

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

Then a spanning tree of graph G can be expressed by the vector $\boldsymbol{x}$. Let T be the set of all such vectors corresponding to spanning tree in graph G, the well-known MST problem can be formulated as:

$$\text{Min } \{ z(\boldsymbol{x}) = \sum_{i=1}^{m} w_i x_i \mid \boldsymbol{x} \in T \} \quad (1)$$

Since the problem was first formulated by Boruvka [3], the MST algorithms have been implemented and studied extensively, such as Kruskal [5], Prim [6] and so on. Although the MST problem is computationally easy, an additional constraint often makes the problem hard; for example, a bound on the degree, or a bound on the diameter of the tree. All these problems are denoted as constrained spanning tree problem, and those above polynomial-time algorithms are no longer effective to them.

Recently, more attention has been paid by using genetic algorithms to deal with those constrained spanning tree problems [7]- [9]. Several tree encodings have been suggested to code the solution of the problem, such as Prüfer number and so on [2]. But all of them are focused on dealing with the MST problems on a complete graph. Actually, it is not always the case in practice because most practical network structures are not based on a complete graph. It means that there is not always a connection or edge for any pair of vertices.

In this paper, we propose a matrix-based tree encoding to cope with the MST problem on a sparse graph. Each chromosome consists of a matrix whose gene (element) is an integer. If there is no connection between a pair of vertices, its corresponding gene is a larger positive number M ($M >> 0$), otherwise, an integer arranging from 1 to 256. Actually, this encoding occupies more memory, but it is more efficient to deal with the MST problem on a sparse graph. Numerical examples show that this new encoding is both effective and efficient to get the optimal solution of the MST problem, compared with the GA approach by using other tree encoding such as Prüfer number.

2 Genetic Algorithms Approach

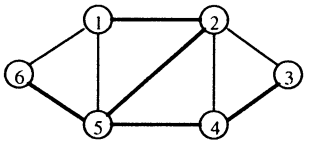
A GA simulates the evolutionary processes by taking an initial population of individuals and applying genetic operators in each reproduction [2] [4]. In optimization terms, each individual in the population is encoded into a string or *chromosome* which represents a possible solution to a given problem. The fitness of

an individual is evaluated with respect to a given objective function. Highly fit individuals or solutions are given opportunities to reproduce by exchanging pieces of their genetic information, in a crossover procedure, with other highly fit individuals. This produces new "offspring" solution(i.e. children), which share some characteristics taken from both parents. Mutation is often applied after crossover by altering some genes in the strings. The offspring can either replace the whole population (generational approach) or replace less fit individuals (steady-state approach). This evaluation-selection-reproduction cycle is repeated until a satisfactory solution is found. The basic steps of a simple GA are shown as follows:

2.1 Genetic representation

The first step in designing a genetic algorithm for a particular problem is to devise a suitable representation scheme.

In this paper, we develop a matrix-based tree encoding to cope with the MST problem on a sparse graph. Each chromosome consists of a vertex-adjacent matrix whose gene (element) is an integer. If there is no connection between a pair of vertices, its corresponding gene is a larger positive number M ($M >> 0$), otherwise, an integer arranging from 1 to 256. In this way the connectivity among vertices is well kept when the Prim's MST algorithm [6] is applied to obtain a solution, i.e., the tree solution with no-exiting edges will not be generated in the case of sparse graph. The encoding can be illustrated as Figure 1. Although



A spanning tree on a sparse graph

v_1	—	12	M	M	67	89
v_2	12	—	72	128	6	M
v_3	M	72	—	21	M	M
v_4	M	128	21	—	35	M
v_5	67	6	M	35	—	52
v_6	89	M	M	M	52	—

Its matrix-based chromosome

Figure 1: A tree and its matrix-based chromosome

this encoding occupies more memory as each chromosome is a matrix. Actually the length of each chromosome is only $(n(n-1)/2)$ for an n-vertex graph because it is symmetrical matrix. Besides, the search space size is $256^{(n(n-1)/2)}$ and the probability to randomly produce a spanning tree is 1. Therefore, it is surely more efficient to deal with the MST problem on a sparse graph.

2.2 Initial population

The initial population in a GA optimization is usually generated by a random sampling of the search space. However, it is a good rule to avoid introducing unacceptable bad individuals beside individuals of reasonable quality. Indeed, while the former individuals might contain valuable genetic material, their poor performance would make them quickly lose in competition with the latter. Not only would the information in the bad individuals be precluded from entering the rest of the population, but the (much) stronger individuals would quickly overtake the whole population, heightening the danger of premature convergence.

In dealing with the MST problem on a sparse graph, the Prüfer number will generate many such infeasible individuals because some edges encoded does not exist actually. However, by using the matrix-based tree encoding, no such infeasible individuals will be generated since a larger integer M ($M >> 0$) is assigned to its corresponding gene in the chromosome when no edge exists between a pair of vertices. The whole procedure of generating the initial population for a n-vertex sparse graph is as follows:

procedure: initial population
begin
  $pop\_size \leftarrow 1$;
  for ($i = 1$ to n)
    assign a larger integer M to a_{ii};
    for ($j = i$ to n)
    if edge from i to j exists **then**
      assign an integer randomly within 256 to a_{ij};
    else
      assign a larger integer M to a_{ij};
    end
  $pop\_size \leftarrow pop\_size + 1$;
end

where $a_{ij}, i, j = 1, 2, \ldots, n$ is the allele of a chromosome.

2.3 Crossover

Crossover operates on two chromosomes at a time and generates offspring by combining both individuals' features. The operation on the matrix-based tree encoding can be illustrated as Figure 2.

2.4 Mutation

Mutation is a background operator, which produces spontaneous random changes in various chromosomes. The operation on the matrix-based tree encoding can be illustrated as Figure 3.

2.5 Evaluation and selection

Evaluation is to associate each individual with a fitness value which reflects how good it is, compared

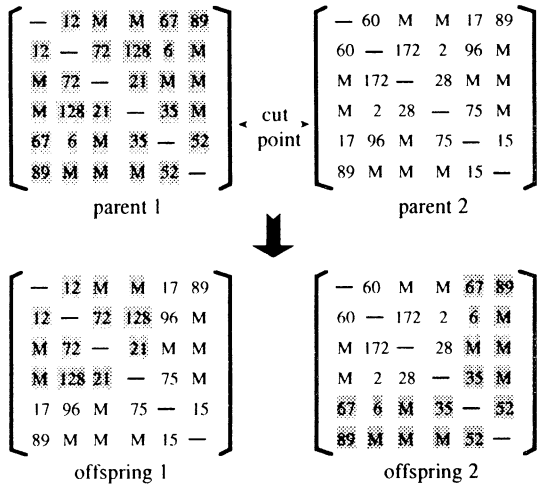


Figure 2: Illustration of crossover operation

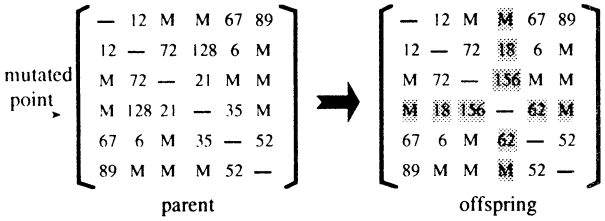


Figure 3: Illustration of mutation operation

with other individuals in the population. In the proposed GA approach, the evaluation procedure consists of two steps:

(1) convert each chromosome into a tree in the form of vector $\boldsymbol{x}$ by using the Prim's algorithm [6];
(2) calculate total cost of the tree according to the objective function (1) and take the reciprocal value of it as the fitness value of that chromosome.

As to selection operation, $(\mu + \lambda)$ - selection strategy is adopted because it can enforce the best chromosomes into the next generation [1]. In the proposed GA approach the selection operation selects μ best chromosomes from μ parents and λ offspring. If there are no μ different chromosomes available, the vacant pool of population is filled up with renewal chromosomes generated randomly.

3 Computational Experience

The numerical examples were generated randomly. The complete graphs with n vertices have $m = n(n-1)/2$ edges and the sparse graph with n vertices have the edges with $m = (n(n-1)/2) \times D$ where D is the density of a graph. $1 - D$ reflects the sparse degree of the graph. Here, the sparse percentage is 10%, 50% and 80% respectively. If there are connection between pairs of vertices, the weights are integers which are randomly generated and uniformly distributed over $[1, 50]$. If there are no connection between pairs of vertices, the weights are an larger integer $M = 1000$.

The parameters for the GA set as follows: population size $pop\_size = 50$; crossover probability $p_c = 0.5$; mutation probability $p_m = 0.4$; maximum generation $max\_gen = 200$; and run by 20 times.

In all 20 runs, the results are given out in Table 1 and Table 2. Table 1 shows the MST results on 10 vertex graph by different approaches. Although the GA approaches with both encoding schemes can obtain the optimal solutions, their effectiveness is different on different graph in the sense of sparse degree. When the graph is a complete graph or less sparse, The GA with Prüfer number is more effective to evolve to the optimal solution. When the graph is a severe sparse graph or more, he GA with matrix-based tree encoding has much chance to evolve to the optimal solution than one with Prüfer number.

Figure 4 clearly illustrates the ability of convergency of the two different encodings on the MST with sparse graph. On sparse graph the matrix-based tree encoding is able to get the optimal solution of the MST more quickly than the Prüfer number does. Obviously it illustrates the high efficiency of the matrix-based tree encoding on sparse graph in the evolutionary process by the GA approach on the MST problem.

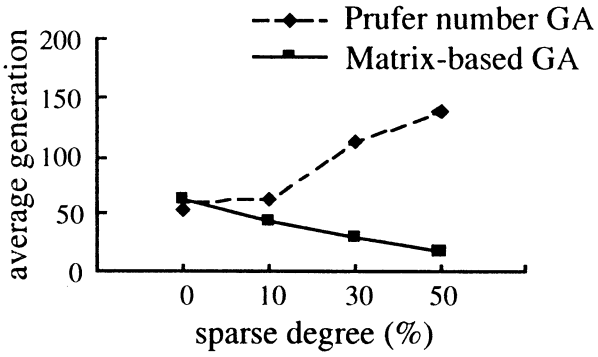


Figure 4: Illustration of the ability of convergency

Table 2 shows the MST results on a larger problem with 30-vertex graph. Both the GA approaches can find the near-optimal solutions, but their effectiveness is different on different graph in the same way. With the increase of the problem scale, the Prüfer number is even not so effective as the matrix-based tree encoding on evolving to the near-optimal solutions. Both the best results and average results are all not so close to the optimal solutions whether it is on a sparse graph or not. In the cases of more sparse graph, the GA with Prüfer number even can not find a feasible solution. It is because too much sparse of a graph makes all the chromosomes be infeasible. But the GA

Table 1: Computational results on 10-vertex graph by GA approach

Sparse Degree	Prim Algorithm	Prüfer Number GA		Matrix-based GA	
		BR	%	BR	%
0%	134	134	50	134	40
10%	146	146	75	146	65
30%	155	155	25	150	100
50%	150	150	15	245	100

BR: best result; %: the frequency to obtain the best result.

Table 2: Computational results on 30-vertex graph by GA approach

Sparse Degree	Prim Algorithm	Prüfer Number GA		Matrix-based GA	
		BR	AR	BR	AR
0%	350	430	499.15	397	439.55
10%	330	470	527.65	361	414.90
30%	339	514	601.20	378	411.15
50%	330	601	704.18	350	385.15

BR: best result; AR: average result

with matrix-based tree encoding can obtain not only the feasible solutions but also the much near-optimal solutions. Both tables shows the effectiveness of the matrix-based tree encoding on MST problem by the GA approach and even the high effectiveness on MST problem with sparse graph.

4 Conclusion

Finding a MST on a sparse graph is a very practical problem in variety of network optimization design. In this paper we developed a new tree encoding, denoted as matrix-based tree encoding, for the GA approach on this problem. In order to verify the effectiveness of the proposed tree encoding scheme, we tested the new encoding on the graphs with different sparse degree compared with the Prüfer number. The numerical results show that this new encoding is effective to get the optimal or near-optimal solutions of the MST problem on sparse graph.

Acknowledgment

This research work was partially supported by the International Scientific Research Program (No. 07045032: 1995.4 - 1998.3) Grant-in-Aid for Scientific Research by the Ministry of Education, Science and Culture of the Japanese Government.

References

[1] T. Bäck and F. Hoffmeister, "Extended selection mechanisms in genetic algorithms," in R. Belew, and L. Booker, eds., *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Los Altos, CA, pp. 92-99, 1991.

[2] M. Gen, and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley, NY, 1997.

[3] R.L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, vol. 7, pp. 43-57, 1985.

[4] J.H. Holland, *Adaption in natural and Artificial Systems*, MIT Press, Cambridge, MA, 1975.

[5] J.B. Kruskal Jr, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. ACM*, vol. 7, pp. 48-50, 1956.

[6] R.C. Prim, "Shortest connection networks and some generalizations," *Bell Sys. Tech. J.*, vol. 36, pp. 1389-1401, 1957.

[7] G. Zhou and M. Gen, "Approach to the degree-constrained minimum spanning tree problem using genetic algorithms," *Engineering Design and Automation*, vol. 3, no. 2, pp. 156-165, 1997.

[8] G. Zhou and M. Gen, "A note on genetic algorithm approach to the degree-constrained spanning tree problems," *Networks*, vol. 30, pp. 105-109, 1997.

[9] G. Zhou and M. Gen, "An effective genetic algorithm approach to the quadratic minimum spanning tree problem," *Computer &. Operations Research*, forthcoming issue.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Spanning Tree-based Genetic Algorithm for Bicriteria Transportation Problem with Fuzzy Coefficients

Yin-Zhen Li, Mitsuo Gen and Kenichi Ida
Dept. of Industrial and Systems Engg.
Ashikaga Institute of Technology
Ashikaga, Tochigi, 326

Abstract

In this paper, we present a new approach of the genetic algorithm for solving bicriteria transportation problem with fuzzy coefficients. The transportation cost and transportation time were considered and described as fuzzy data. The Prüfer number encoding based on a spanning tree is used to representing the transportation tree. The criterion of feasibility for a chromosome was designed. The ranking fuzzy numbers with intergal value was used to evaluate chromosomes. Finally, numerical experiments will be shown the efficiency of the proposed algorithm.

1 Introduction

The real-world situations are often not so deterministic and a common problem is the difficulty for determining the proper values of model parameters. On the consideration of practical models, it is necessary to use fuzzy numbers to represent the imprecise conditions. One way of handing such uncertainty in decision making is fuzzy mathematical programming. Kaufmann and Gupta [3] firstly examined fuzzy transportation problem considering fuzzy coefficients.

Recently, there are many investigation using evolutionary approachs to solve the variety transportation problems. Michalewicz *et al.* [6, 7] firstly discussed the use of genetic algorithm (GA) for solving linear and nonlinear transportation problems. They used these problems as an example of constrained optimization problems, and investigated how to handle such constraints with GA. The matrix representation was used to construct a chromosome and designed the matrix-based crossover and mutation in their investigation.

In this paper, we present a new approach of the genetic algorithm for solving bicriteria transportation problem with fuzzy coefficients. Transportation problem as a special type of the network problems have a special data structure in solution characterized as a transportation graph. We focuse on this special graph and utilize the Prüfer number [1] encoding based on a spanning tree which is adopted as it is capable of equally and uniquely representing all possible trees. Using the Prüfer number representation the memory only requires $m+n-2$ on the problem. Transportation problem have separable set of nodes that from origins and destinations. From this point, we design the criterion for feasibility of the chromosome. For ranking the fuzzy numbers, we use a ranking method of fuzzy numbers with integral value [5]. The mixed strategy with $(\mu + \lambda)$-selection and roulette wheel selection is used. The proposed algorithm can find Pareto solutions for the bicriteria transportation problem in the criterion space. Finally, numerical experiments will be shown the efficiency of the proposed algorithm.

2 Problem Description

In some transportation systems, the influence of the traffic complication on the transportation system induce that some or all of the coefficients of objectives are not exactly known with certainty frequently occur in practice. For example as transportation cost, delivery time, and so on.

Consider the following two objectives: minimizing total transportation cost and minimizing total delivery time. Let $\tilde{c}_{ij}^1$ be fuzzy data representing the transportation cost of shipping one unit that from origin i to destination j, $\tilde{c}_{ij}^2$ be fuzzy data representing the delivery time of shipping one unit of the product that from origin i to destination j, a_i be the number of units available at origin i, and b_j be the number of units demanded at destination j.

This problem with m origins and n destinations can be formulated as follows:

$$\begin{aligned}
\text{min} \quad & \tilde{z}_1 = \sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{c}_{ij}^1 x_{ij} \\
\text{min} \quad & \tilde{z}_2 = \sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{c}_{ij}^2 x_{ij} \\
\text{s. t.} \quad & \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1, 2, \cdots, m \\
& \sum_{i=1}^{m} x_{ij} \ge b_j, \quad j = 1, 2, \cdots, n \\
& x_{ij} \ge 0, \quad \forall\, i, j
\end{aligned}$$

where x_{ij} is the unknown quantity to be transported from origin i to destination j,

Transportation problem have some characteristics in solution: (1) it has $m+n-1$ basic variables and they correspond to some cell in the transportation tableau, (2) the basis must be transportation tree, that is, there must be at least one basic cell in each row and in each column of the transportation tableau and (3) the basis should not contain a cycle.

Denote that origins $1, 2, ..., m$ as the component of the set $O = \{1, 2, ..., m\}$ and denote that destinations $m+1, ...m+n$ as the component of the set $D = \{m+1, ..., m+n\}$. Obviously, the transportation problem has $m+n$ nodes and $m \times n$ edges.

In this paper, we assume a balanced transportation problem that the total supply $\sum_{i=1}^{m} a_i$ is equal to total demand $\sum_{j=1}^{n} b_j$. Because the unbalanced transportation problem with a dummy origin or a dummy destination can be converted to a balanced transportation problem.

3 Ranking Fuzzy Numbers

Here, we use the Liou and Wang's method to rank the fuzzy numbers. This method ranks fuzzy numbers which can be triangular, trapezoidal, with integral value instead of a relative value. The left integral value is used to reflect the pessimistic viewpoint and the right integral value is used to reflect the optimistic viewpoint of the decision maker. A convex combination of right and left integral values through an index of optimism is called the total integral value. Denote $I(\tilde{A})^L$ and $I(\tilde{A})^R$ are left and right integral value. The total integral value of the TFN $\tilde{A}$ is

$$\begin{aligned} I_{\mathrm{T}}^{\alpha}(\tilde{A}) &= \alpha I(\tilde{A})^R + (1-\alpha) I(\tilde{A})^L \\ &= \frac{1}{2}[\alpha a_3 + a_2 + (1-\alpha) a_1] \end{aligned} \quad (1)$$

when a degree of optimism $\alpha \in [0, 1]$ of a decision maker is given. A large α indicates a higher degree of optimism [5]. Especially, when $\alpha = 0$, the total integral value $I_{\mathrm{T}}^{0}(\tilde{A})$ which represents a pessimistic decision maker's viewpoint is equal to the left integral value of $\tilde{A}$, *i.e.* $I_{\mathrm{L}}(\tilde{A})$. For an optimistic decision maker, *i.e.* $\alpha = 1$, the total integral value $I_{\mathrm{T}}^{1}(\tilde{A})$ is equal to $I_{\mathrm{R}}(\tilde{A})$. And for a moderate decision maker, $\alpha = 0.5$, the total integral value becomes $I_{\mathrm{T}}^{0.5}(\tilde{A}) = \frac{1}{2}[I_{\mathrm{R}}(\tilde{A}) + I_{\mathrm{L}}(\tilde{A})]$ which the integral value is same as ordinary representives [3].

This total integral value of fuzzy numbers is used as the ranking function. For any fuzzy numbers $\tilde{A}_i$ and $\tilde{A}_j$, we have the following criteria for ranking fuzzy numbers: if $I_{\mathrm{T}}^{\alpha}(\tilde{A}_i) < I_{\mathrm{T}}^{\alpha}(\tilde{A}_j)$, then $\tilde{A}_i < \tilde{A}_j$; if $I_{\mathrm{T}}^{\alpha}(\tilde{A}_i) = I_{\mathrm{T}}^{\alpha}(\tilde{A}_j)$, then $\tilde{A}_i = \tilde{A}_j$; if $I_{\mathrm{T}}^{\alpha}(\tilde{A}_i) > I_{\mathrm{T}}^{\alpha}(\tilde{A}_j)$, then $\tilde{A}_i > \tilde{A}_j$. The more full details of the method refer to reference [5].

4 Spanning Tree-based Genetic Algorithm

4.1 Representation

Prüfer number is one of the node encoding for the tree. It can be used to encode a transportation tree. The constructive procedure for a Prüfer number according to a tree is as follows:

Procedure: Convert Tree to Prüfer Number

Step 1. Let i be the lowest numbered leaf node in tree T. Let j be the node which is the predecessor of i. Then j becomes the rightmost digit of Prüfer number $P(T)$, $P(T)$ is built up by appending digits to the right; thus, $P(T)$ is built and read from left to right.

Step 2. Remove i and the edge (i, j) from further consideration. Thus, i is no longer considered at all and if i is the only successor of j, then j becomes a leaf node.

Step 3. If only two nodes remain to be considered, then $P(T)$ has been formed, stop; else return to Step 1.

To construct a transportation tree, the constraint of nodes must be considered. From a Prüfer number, an unique transportation tree also is possible to be generated by the following procedure:

Procedure: Convert Prüfer Number to Transportation Tree

Step 1. Let $P(T)$ be the original Prüfer number and let $\bar{P}(T)$ be the set of all nodes that are not part of $P(T)$ and designed as eligible for consideration.

Step 2. Repeat the process (2.1)–(2.5) until no digits left in $P(T)$.

(2.1) Let i be the lowest numbered eligible node in $\bar{P}(T)$. Let j be the leftmost digit of $P(T)$.

(2.2) If i and j not in the same set O or D, add the edge (i, j) to tree T. Otherwise, select the next digit k from $P(T)$ that not included in the same set with i, exchange j with k, add the edge (i, k) to the tree T.

(2.3) Remove j(or k) from $P(T)$ and i from $\bar{P}(T)$. If j(or k) does not occur anywhere in the remaining part of $P(T)$, put it into $\bar{P}(T)$. Designate i as no longer eligible.

(2.4) Assign the available amount of units to $x_{ij} = \min\{a_i, b_j\}$ (or $x_{ik} = \min\{a_i, b_k\}$), where $i \in O$ and $j, k \in D$ to the edge (i, j) (or (i, k)).

(2.5) Update availability $a_i = a_i - x_{ij}$ and $b_j = b_j - x_{ij}$ (or $b_k = b_k - x_{ik}$).

Step 3. If no digits remain in $P(T)$ then there are exactly two nodes , i and j still eligible in $\bar{P}(T)$ for consideration. Add edge (i,j) to tree T and form a tree with $m+n-1$ edges.

Step 4: If no available amount of units to assign, then stop. Otherwise, there are remaining supply r and demand s, add edge (r,s) to tree and assign the available amount of units $x_{rs} = a_r = b_s$ to edge. If there exists a cycle, remove the edge that assigned zero flow. A new spannning tree is formed with $m+n-1$ edges.

4.2 Initialization

The initialization of a chromosome(a Prüfer number) is performed from that randomly generated $m+n-2$ digits in range $[1, m+n]$. However, it is possible to generate an infeasible chromosome which is not adapted to generate a transportation tree.

We design the handling for feasibility of the chromosome with the following criterion: Denote that s_O and s_D are the sum of connections of nodes which are included in set O and D respectively from $P(T)$. Also we denote that $\bar{s}_O$ and $\bar{s}_D$ are the appearing times of those nodes in $\bar{P}(T)$ and included in set O and D, respectively. If $s_O + \bar{s}_O = s_D + \bar{s}_D$, then $P(T)$ is feasibility. Otherwise $P(T)$ is infeasible.

4.3 Genentic Operators

Here, the one-cut-point crossover operator is used. For avoiding unnecessary decoding from which an infeasible chromosome(a Prüfer number) may be generated after crossover operator, we add the criterion for feasibility of the chromosome. A Prüfer number via this criterion is always feasible and can be decoded into a corresponding tranportation tree.

The inversion mutation and displacement mutation are used. This two mutation operators always generate feasible chromosomes if the parents are feasible, because the criterion $s_O + \bar{s}_O = s_D + \bar{s}_D$ is unchanged after these operations.

4.4 Evaluation and Selection

In this approach, the evaluation procedure consists of two steps: (1) convert a chromosome into a tree, and (2) calculate each objective function. It is given as follows:

procedure: evaluation
begin
  $T \leftarrow \{\emptyset\}$;
  $p \leftarrow 0$;
  define $\bar{P}(T)$ according to the $P(T)$;
  repeat
    select the leftmost digit from $P(T)$, say i;
    select the eligible node with the lowest numbered from $\bar{P}(T)$, say j;
    if $i,j \in O$ or $i,j \in D$ **then**
      select next digit from $P(T)$ not in the set with j, say k;
      exchange k with i and take $i \leftarrow k$;
    end
    assign the flow:
      $x_{ij} \leftarrow \min\{a_i, b_j\}$, $i \in O, j \in D$;
    $T \leftarrow T \cup \{x_{ij}\}$;
    $\tilde{z}_1(T) \leftarrow \tilde{z}_1(T) + \tilde{c}^1_{ij} x_{ij}$; $\tilde{z}_2(T) \leftarrow \tilde{z}_2(T) + \tilde{c}^2_{ij} x_{ij}$;
    update available amount:
      $a_i \leftarrow a_i - x_{ij}$; $b_j \leftarrow b_j - x_{ij}$;
    remove i from $P(T)$;
    remove j from $\bar{P}(T)$;
    if i does not occur anywhere in remaining $P(T)$
      then put i into $\bar{P}(T)$;
    end
    $p \leftarrow p+1$;
  until($p \leq m+n-2$)
  assign the flow for $r,s \in \bar{P}(T)$:
    $x_{rs} \leftarrow \min\{a_r, b_s\}$, $r \in O, s \in D$;
  $T \leftarrow T \cup \{x_{rs}\}$;
  $\tilde{z}_1(T) \leftarrow \tilde{z}_1(T) + \tilde{c}^1_{rs} x_{rs}$; $\tilde{z}_2(T) \leftarrow \tilde{z}_2(T) + \tilde{c}^2_{rs} x_{rs}$;
  repeat
    if $a_i > 0, \forall\, i$ and $b_j > 0, \forall\, j$ **then**
      assign the flow:
        $x_{ij} \leftarrow \min\{a_i, b_j\}$, $i \in O, j \in D$;
      $T \leftarrow T \cup \{x_{ij}\}$;
      update available amount:
        $a_i \leftarrow a_i - x_{ij}$; $b_j \leftarrow b_j - x_{ij}$;
    end
    $\tilde{z}_1(T) \leftarrow \tilde{z}_1(T) + \tilde{c}^1_{ij} x_{ij}$; $\tilde{z}_2(T) \leftarrow \tilde{z}_2(T) + \tilde{c}^2_{ij} x_{ij}$;
  until(no available amount)
  if there exists a cycle **then**
    find edge with zero flow and remove it;
  end
end

Fitness function: the fitness function is derived with following steps.

(1) Choose the solution points which contain the minimum $I^{\alpha}_{\mathrm{T}}(\tilde{z}^{min}_q)$ (or the maximum $I^{\alpha}_{\mathrm{T}}(\tilde{z}^{max}_q)$) that corresponding to each objective function value, and then compare with the stored solution points at the previous generation and select the best points to save again.

(2) Solve the weights for each objective function and construct the fitness function.

$$\delta_q = I^{\alpha}_{\mathrm{T}}(\tilde{z}^{max(t)}_q) - I^{\alpha}_{\mathrm{T}}(\tilde{z}^{min(t)}_q), \quad q = 1,2$$

$$\beta_q = \frac{\delta_q}{\sum_{q=1}^{2} \delta_q}, \quad q = 1,2$$

$$eval(T_k) = \sum_{q=1}^{2} \beta_q I^{\alpha}_{\mathrm{T}}(\tilde{z}_q(T_k)), \ \forall\, k$$

In the selection procedure, we use the mixed strategy with $(\mu+\lambda)$-selection and roulette wheel selection can enforce the best chromosomes into the next generation.

5 Numerical Experiment

The computer simulations were experimented with the following problem where the coefficients of the problem were described as triangular fuzzy numbers(TFN) given Table 1.

Table 1: TFN of Coefficients in the BTP

k	$\tilde{c}_{i1}^k$			$\tilde{c}_{i2}^k$			$\tilde{c}_{i3}^k$			$\tilde{c}_{i4}^k$			O_i
	1	2	3	1	2	3	5	7	9	6	7	9	8
1	1	1	2	6	9	12	2	3	5	3	4	5	19
	6	8	10	7	9	10	2	4	6	5	6	8	17
	3	4	5	6	4	6	1	3	5	2	4	6	8
2	3	5	7	6	8	9	7	9	9	8	9	12	9
	4	6	8	1	2	3	4	5	6	1	3	3	17
D_j		11			3			14			16		44

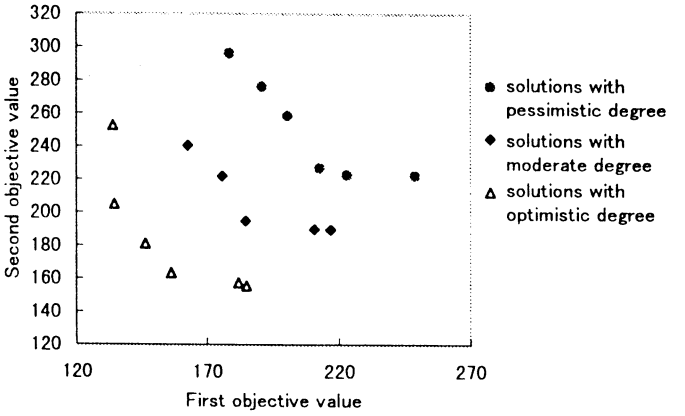


Figure 1: Obtained solutions with three case of degree of optimism (α)

The coded algorithm was run by 10 times with best mechanism(crossover rate p_c=0.2 and mutation rate p_m=0.6) with three different degree of optimism, i.e., α=0, 0.5, 1. The results with population size 30 and maximum generation 1000 are shown in Figure 1 which are found in the Pareto frontier. The obtained Pareto solutions of the optimistic case are given as follows:

```
(1) objective value: (134.0, 252.5)
    solution matrix:
              5  3  0  0
              3  0  0 16
              3  0 14  0
(2) objective value: (134.5, 205.0)
    solution matrix:
              0  3  0  5
             11  0  0  8
              0  0 14  3
(3) objective value: (146.5, 181.0)
    solution matrix:
              0  3  0  5
             11  0  8  0
              0  0  6 11
(4) objective value: (156.5, 163.5)
    solution matrix:
              0  3  5  0
             11  0  8  0
              0  0  1 16
(5) objective value: (182.0, 157.5)
    solution matrix:
              0  0  8  0
             11  0  6  2
              0  3  0 14
(6) objective value: (185.0, 155.5)
    solution matrix:
              0  0  8  0
             11  2  6  0
              0  1  0 16
```

where objective values are given with integral value of fuzzy numbers.

6 Conclusion

In this paper, we presented a spanning tree-based genetic algorithm for solving bicriteria transportation problem with fuzzy coefficients. In the bicriteria space, the Pareto solutions were determined based on the ranked values of fuzzy objectives, and spanning tree-based genetic algorithm was used to search this kind of Pareto solutions. The efficiency of the proposed genetic algorithm was shown with computer simulations.

References

[1] Dossey, J., A. Otto, L.spence and C. Eynden, *Discrete Mathematics*, Harper Collins, 1993.

[2] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.

[3] Kaufmann, A. and Gupta, M.M., *Fuzzy Mathematical Models in Engineering and Management Science*, Van Nostrand Reinhold, 1988.

[4] Li, Y.Z. and M. Gen, Spanning Tree-Based Genetic Algorithm for Bicriteria Transportation Problem, *Proc. of The Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, Canberra, 1997.

[5] Liou, T. and M. J. Wang, "Ranking Fuzzy Numbers with Integral Value", *Fuzzy Sets and Systems*, Vol.50, pp.247-255, 1992.

[6] Michalewicz, Z., G. A. Vignaux and M. Hobbs, A Non-Standard Genetic Algorithm for the Nonlinear Transportation Problem, *ORSA Journal on Computing*, Vol.3, No.4, pp.307-316, 1991.

[7] Vignaux, G.A. and Z. Michalewicz, A Genetic Algorithm for the Linear Transportation Problem, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.21, No.3, pp.445-452, 1991.

[8] Zhou, G. and M. Gen, Approach to Degree-Constrained Minimum Spanning Tree Problem Using Genetic Algorithm, *Engineering Design & Automation*, Vol.3, No.2, pp.157-165, 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Bicriteria Machine Scheduling Problem Using Genetic Algorithms

Dawei Zheng
Department of Industrial
and Systems Engineering
Ashikaga Institute of Technology
Ashikaga, 326 Japan

Mitsuo Gen
Department of Industrial
and Systems Engineering
Ashikaga Institute of Technology
Ashikaga, 326 Japan

Abstract

In this paper, a new implementation of Genetic Algorithms (GAs) is developed for bicriteria machine scheduling problem. TIMETABLER method is combineded to decode each chromosome. We proposed adaptive hyperplane technique to construct fitness function to deal with the bicriteria case. The adaptive hyperplane is used to guide genetic search to approach the Pareto frontier and exploit the set of Pareto optimal solutions of the problem. Numerical experiment shows the efficiency of the proposed GAs approach on this problem.

Key words: Machine scheduling problem, Genetic algorithms, Bicriteria decision making.

1 Introduction

Machine scheduling problem is the study of constructing schedules of machine processing for a set of jobs in order to ensure the execution of all jobs in a reasonable amount of time or cost. We will deal with what machines to be allocated to which jobs and how to order the jobs in an appropriate processing sequence. As the problem is *NP-hard* and no effective algorithms exist [7], we proposed a genetic algorithms (GAs) approach to deal with it.

In recent years, Genetic Algorithms (GAs) have emerged as one of the most effective and robust optimization tools [3] [5] [8]. GAs can handle any kind of objectives and any kind of constraints, which make GAs highly attractive algorithms for multicriteria optimization problems.

In this paper, a new implementation of GAs is developed for machine scheudling problem. As the performance measure of early and tardy completion of jobs is very natural because one's aim is usually to minimize simultaneously both earliness and tardiness of all jobs. The first objective is minimize simultaneously both earliness and tardiness of all jobs. The maximam completion time all jobs (makespan) is also the important aim at machine scheduling problem, so the another objective is to minimize the makespan. Using GAs method, we can find the Pareto set and Pareto frontier of the bicriteria machine scheduling problem.

2 Bicriteria Machine Scheduling Problem

We address a single machine scheduling problem in which there is a single machine with n jobs available for processing time at time zero. Each job is identified by an integer from 1 through n, and each job i is described by these attributes:

$p(i)$: processing time

$d(i)$: due date

$e(i)$: early penalty

$t(i)$: tardy penalty

A schedule S is defined as a vector of job completion times $[c(1), c(2), \cdots, c(n)]$. The first objective is to find a schedule as follows:

$$\min \quad Z_1(S) = \sum_{i=1}^{n} [e(i)\max\{d(i) - c(i), 0\} + t(i)\max\{c(i) - d(i), 0\}] \tag{1}$$

where the first term on the right hand side of function 1 represents earliness cost and the second term represents tardiness cost. In the notation of Rinnooy Kan [9], the problem may be described as $n/1//ET$, or may be called ET for short, where ET indicates that the performance measure accounts for earliness and tardiness.

As the makespan also is very important aim at machine scheduling problem, the second objective is to find a schedule as follows:

$$\min \quad Z_2(S) = \max\{c(i)|i = 1, 2 \cdots, n\} \qquad (2)$$

Unlike standard mathematical programming with a single objective function, an optimal solution in the sense that two objective functions simultaneously minimized does not necessarily exist. We can only get Pareto optimal solutions.

3 TIMETABLER method

Given a job permutation, we can find a schedule which total penalty is minimize by using a heuristic algorithm, denoted as TIMETABLER. The procedure of TIMETABLER is show as follows.

Procedure: TIMETABLER:

t: the stage number

$c(i)$: is the completion time for job i

$S(t, P)$: a partial schedule of $(t-1)$ jobs, where P is a permutation of jobs. We treat P as an ordered set and let iP be the permutation in which P is immediately preceded by job i.

P': an ordered set of unscheduled jobs; jobs that will follow those in P' are in P.

Step 1 Let $t = 1$ with $S(t, P)$ being null and P being empty.

Step 2 Select the last job i in P', and delete it from P'.

Step 3 Move to the next stage as follows:

3.1 Adding j to $S(t, P)$ to create $S(t+1, iP)$ such that increasing $c(i)$would not reduce cost of $S(t+1, iP)$, using the following steps:

3.1.1 Initially job i is started at time 0, shifting jobs in $S(t, P)$ to the right if necessary

3.1.2 Shift job i to the right until the marginal cost of doing is not negative

3.2 let $t = t + 1$;

3.3 Setting $P = iP$

Step 4 If P' is not empty, go to Step 2. Otherwise, stop.

Consider an example with four jobs problem in Table 1, the following numerical example illustrates the TIMETABLER procedure.

Figure 1 shows how the TIMETABLER would develop schedules for the job permutation is 4, 3, 2, 1.

In Stage 1, job 1 are added to the null schedule so that increasing $c(1)$ does not reduce cost. This occurs by setting $c(1) = d(1)$. At stage 2, job 2 is added before job 1 and $c(2)$ is set to $d(2)$ because $c(2) > t(1)$, causing $c(1)$ to shift 18 to 29. In stage 3, job 3 is added with $c(3) = d(3)$, which does not affect job 1 or job 2. At stage 4, because the left time after job 3 is not enough for job 4 to operate, job 4 is added with $c(4) = d(4)$. So we can calculate the total cost according to the starting time.

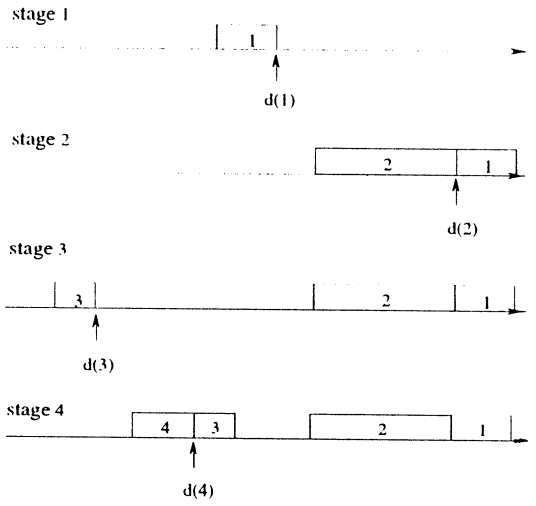


Figure 1: Application of the procedure TIMETABLER to a four job sample problem

4 Genetic Algorithms

In the last few years, GAs have been successfully applied to optimization problems in diverse fields. Representation, crossover, mutation, evaluation and selection will be explain at this section.

Table 1: Scheduling Problem for 4 jobs

Job No.	Processing time	Due date	Early penalty	Tardy penalty
1	3	18	9	12
2	7	26	18	38
3	2	4	20	25
4	3	12	10	12

Representation: A chromosome is defined as $V = (j_1\ j_2\ \cdots\ j_n)$, where n is the number of job.

One-point Crossover: Crossover is implemented with One-point crossover operator [5]. One-point crossover picks one cut point randomly, and then takes the pre-cut section of the first parent and fills up the offspring by taking in order each legitimate gene from the second parent as shown in Figure 2.

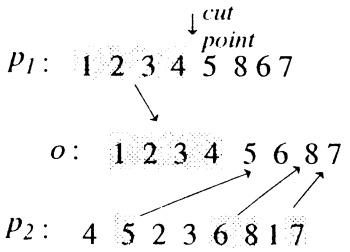


Figure 2: One-point crossover

Reciprocal Mutation: Reciprocal exchange mutation selects two genes at random and then swaps the genes on these positions as illustrated in Figure 3.

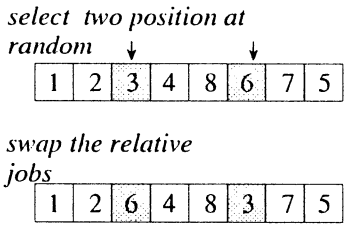


Figure 3: Reciprocal mutation

Evaluation and selection: The adaptive hyperplane approach [2] in criteria space is proposed to construct fitness function so as to force genetic search to exploite the set of nondominated points and to approach the Pareto frontiers.

For a bicriteria optimization problem, we have two objectives, $\mathbf{z} = (z_1\ z_2)$. Firstly, from the current chromosomes we can find a solution set $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots\} \in R^n$, where $\boldsymbol{x}_i$ is a nondominated point. Then we can determine two special points from P as follows:

$$z_1^{\min}(\boldsymbol{x}_{i_1}) = \min\{z_1(\boldsymbol{x}_i) | \boldsymbol{x}_i \in P\}$$
$$z_2^{\min}(\boldsymbol{x}_{i_2}) = \min\{z_2(\boldsymbol{x}_i) | \boldsymbol{x}_i \in P\}$$

The evaluation function is given as follows:

$$eval(V) = w_1 z_1 + w_2 z_2 \qquad (3)$$

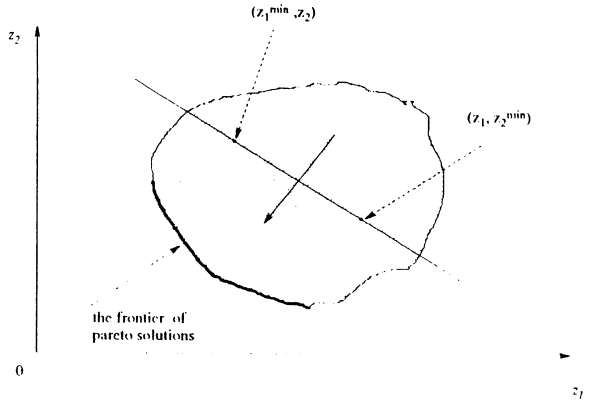


Figure 4: illustrative explanation of adaptive hyperplane method

where

$$w_i = (-1)^{i+1} \frac{\begin{vmatrix} z_1^{\min}(\boldsymbol{x}_{i_1}) & z_2(\boldsymbol{x}_{i_1}) \\ z_2(\boldsymbol{x}_{i_2}) & z_2^{\min}(\boldsymbol{x}_{i_2}) \end{vmatrix}}{\begin{vmatrix} z_1^{\min}(\boldsymbol{x}_{i_1}) & 1 \\ z_1(\boldsymbol{x}_{i_2}) & 1 \end{vmatrix}}$$
$$i = 1, 2$$

This evaluation function just forms a line in bicriteria space which divides the bicriteria space into two half spaces: one containing positive ideal solution denoted with Z^+ and another containing negative ideal solution denoted with Z^-. The feasible solution space F is correspondingly divided into two parts: one is $F^- = F \cap Z^-$ and the other is $F^+ = F \cap Z^+$. For the minimizing problem, it is easy to verify that a solution in F^- has lower value of objective function (3) than those in F^+. So chromosomes in the half space F^- have relatively larger chance to enter next generation.

With the process of GAs searching in solution space, the corresponding objective in criteria space gets close to the frontier of Pareto solution under the action of the adaptive hyperplane. Figure 4 clearly demonstrates the mechanism of the searching process. In this process, the adaptive hyperplane is continuelly reformed in each generation iteration and get closer and closer to the frontier of Pareto optimal solution in the criteria space.

Elitist selection [10] was used to ensure best chromosome passed on next generation.

5 Numerical Experiment

There are one machine and 8 jobs, due time, processing time early penalty and tardy penalty for each job are shown as follows:

Table 2: Scheduling Problem for 8 jobs

Job No.	Processing time	Due date	Early penalty	Tardy penalty
1	121	260	12	8
2	147	269	9	15
3	102	400	11	10
4	79	266	1	21
5	130	337	18	7
6	83	336	15	17
7	96	683	14	1
8	88	719	21	12

By using the proposed approach, the result is shown in Table 3. Table 3 shows that all schedules are Pareto solutions in the sense of bicriteria case. z_1 is the total penalty and the z_2 is the makespan. Figure 5 clearly illustrates all these Pareto solutions and the Pareto frontier.

Table 3: Pareto Solutions

the sequence of job	z_1	z_2
7 8 5 1 3 6 2 4	6252	879
5 8 7 1 3 6 2 4	7202	859
7 8 2 3 6 5 1 4	8867	840
3 8 2 7 6 5 1 4	14378	834
8 6 2 7 3 5 1 4	16574	814
5 3 4 8 6 7 2 1	26036	813

The frontier of pareto solutions was shown in Figure 5. According to those Pareto solutions, the decision maker can arrange the job scheduling to earn maximam benefit.

6 Conclusion

In this paper, GA is used to solve the bicriteria machine schedule problem. TIMETABLER method was combined to decode chromosome and calculate objective value. We use adaptive hyperplane technique to guide genetic search to approach the Pareto frontier. Numerical example show that the adaptive hyperplane technique can efficiently search for all Pareto optimal solutions for the machine scheduling problems.

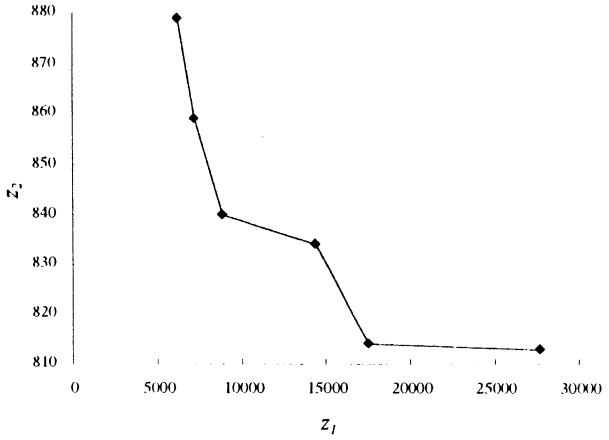


Figure 5: The frontier of Pareto solutions by GA

References

[1] Baker, K. and G. Scudder: Sequencing with earliness and tardiness penalties: a review, *Operations Research*, vol.38, pp.22-36, 1990.

[2] Zheng, D. and M. Gen: Multiobjective Optimization using Genetic Algorithm, *Journal of Engineering Valuation and Cost Analysis*. forthcoming special issue.

[3] Fogel, D.: *Evolution Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, 1995.

[4] Gen, M. and R. Cheng: Interval Programming using Genetic Algorithms. *Proceeding of Firstly International Symposium on Soft Computing for Industrial*, 1996.

[5] Gen, M. and R. Cheng: *Genetic Algorithms and Engineering Design*, John Wiley & Sons, 1997.

[6] Blazewicz, J., K. Ecker, G. Schmidt and J. Weglarz: *Scheduling in Computer and Manufacturing Systems*, Springer-Verlag, 1993.

[7] Lawler, L.E.: A Pseudopolynomial Algorithm for Sequence Jobs to Minimize Total Tardiness, *Annals of Discrete Mathematics*, Vol.1, pp.331-342, 1977.

[8] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd ed., Springer-Verlag, 1994.

[9] Rinnooy Kan, A.H.G.: *Machine Scheduling Problems: Classification, Complexity, and Computations*, Martinus Nijhoff, The Hague, 1976.

[10] Tamaki, H., H. Kita and S. Kobayashi: Multi-Objective Optimization by Genetic Algorithms: A Review, *Proceeding of 1996 IEEE International Conference on Evolutionary Computation*, pp.517-522, 1985.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998, begining page-ending page

Evolutionary Feature Recognition of CAD Data Employing Boundary Representation Method

Yasuhiro TSUJIMURA
Department of Industrial and Systems Engineering
Ashikaga Institute of Technology
268 Ohmae-cho, Ashikaga 326, Japan
tujimr@ashitech.ac.jp

Mitsuo GEN
Department of Industrial and Systems Engineering
Ashikaga Institute of Technology
268 Ohmae-cho, Ashikaga 326, Japan
gen@genlab.ashitech.ac.jp

Masaki HIJI
Graduate School of Engineering
Ashikaga Institute of Technology
268 Ohmae-cho, Ashikaga 326, Japan
mhiji@genlab.ashitech.ac.jp

Abstract

This paper proposes an evolutionary computation-based method for recognizing features of CAD data. Feature-based chromosome scheme is developed in which its locus corresponds to two features of CAD data provided by using the Boundary Representation method. The efficiency of the proposed method is shown through experimental results.

Keywords : Computer-aided Design(CAD), Feature Recognition, Evolutionary Computation, Boundary Representation

1 Introduction

Recently, a lot of companies integrate CAD/CAM system to enhance the product quality and productivity, and to reduce overall product life cycle cost. However, practical CAD data are usually very large and complex, so integration of a CAD/CAM system to a practical production system is very difficult due to a difficulty to store such large actual CAD data in a computer system.

To eliminate the difficulty, possible common data in all the CAD data are shared. Sharing common data among the large number of CAD data can reduce amount of whole the CAD data. The common data to be shared can be obtained by recognizing features of the CAD data, *i.e.*, the common data are the features themselves.

In this paper, we propose a method to recognize features from CAD data using evolutionary computation for sharing CAD information. Feature-based representation scheme is proposed as a suitable one for feature recognition of CAD data. Furthermore, the efficiency of the proposed method is demonstrated using some simplified CAD data.

2 Boundary Representation of Solid Model

In this section, the method for constructing solid model, which can be processed on computer system, is discussed. We employ the *Boundary Representation* method [1] - [5] as the construction method of solid models.

The boundary of a solid is the primary interface between the solid and surrounding environment. Reflection of light from a solid and, therefore, the solid's appearance is determined by the surface properties: color and texture. Even the bounding surface of transparent objects influences light reflection. The bounding surface of a solid is where is contacts other solids. Manufacturing (machining) processes are concerned with the interaction of the surface envelope of a tool path with the surface of a part's raw-stock shape. An object can be distinguished using the

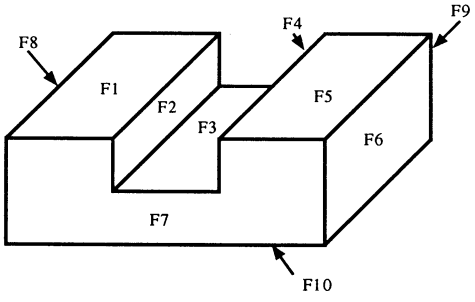


Figure 1: Solid model

boundary. Thus, most of the ingredients are obtained for Boundary Representations. The solid model (Figure 1) using Boundary Representation can be illus-

Table 1: Illustration of solid model using Boundary Representation

Face Name	Face Type	The Number of Edge	Neighboring Face Loop (Edge Type)	Face Score
F1	1	4	F7(1) - F8(1) - F9(1) - F2(1)	1.0
F2	1	4	F7(1) - F1(1) - F9(1) - F3(1)	1.5
F3	1	4	F7(1) - F2(1) - F9(1) - F4(1)	2.0
F4	1	4	F7(1) - F4(1) - F9(1) - F5(1)	1.5
F5	1	4	F7(1) - F5(1) - F9(1) - F6(1)	1.0
F6	1	4	F7(1) - F5(1) - F9(1) - F10(1)	1.0
F7	1	8	F1(1) - F8(1) - F10(1) - F6(1) - F5(1) - F4(1) - F3(1) -F2(1)	1.0
F8	1	4	F7(1) - F1(1) - F9(1) - F10(1)	1.0
F9	1	8	F1(1) - F2(1) - F3(1) - F4(1) - F5(1) - F6(1) - F10(1) - F8(1)	1.0
F10	1	4	F7(1) - F5(1) - F9(1) - F5(1)	1.0

trated in Table 1. The values of *Face Type* and *Edge Type* in Table 1 can be given by Table 2. and *Neighboring Face Loop* represents the relationship between the face and neighboring faces.

Face Score is used to measure the face complexity based upon the convexity or concavity of the solid model. Face Score can be calculated by equation (1), where *Average Edge Score* is given by equation (2), and Angle Score is the score of edge for its angle (see Figures 2 and 3).

$$\text{Face Score} = 12 * (\text{Face Type} - 1) + \text{Average Edge Score} \quad (1)$$

$$\text{Average Edge Score} = \frac{\sum(\text{Edge Type} * \text{Angle Score})}{\text{The Number of Edges}} \quad (2)$$

Table 2: Types of Face and Edge

Face	Type	Edge	Type
Plane	1	Straight	1
Cylinder	2	Ellipse	2
Cone	3	Circle	3
Torus	4		
Sphere	5		

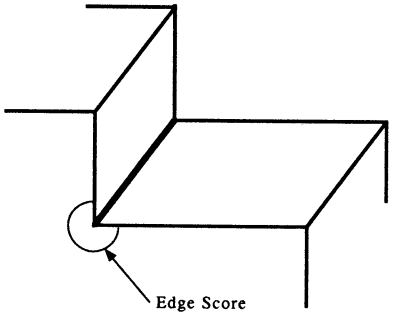


Figure 2: Measurement of Edge Score

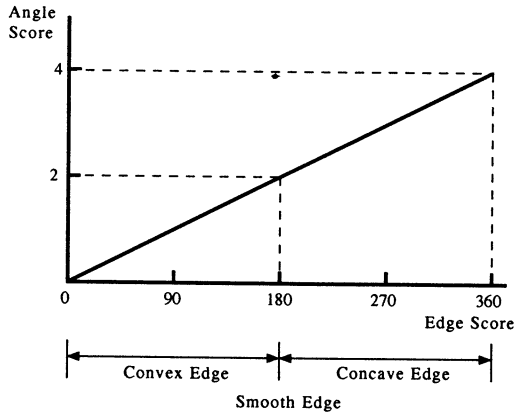


Figure 3: Illustration of Angle Score

3 Evolutionary Computation Approach

3.1 Chromosome Representation

In this paper, we use the *random key representation* for encoding[6]. Random key representation encodes

a solution with *random numbers.*

The allele talces a random decimal number from an open interal (0,1). Here, the allele is only used as keys for sorting. The locus corresponds to two features: the Face Score of each face and the *Average Face Score of Neigboring Face Loop* (see equation (3)).

$$\text{(Average Face Score of Neighboring Face Loop}(AFS)) = \frac{\sum(\text{Face Score of Neighboring Face})}{(\text{The Number of Neighboring Face})} \quad (3)$$

For example, the solid model of Figure 1 can be coded into the feature-based chromosome shown in Figure 4. In Figure 4, the refference list can be obtained by sorting the keys of these features in descending order. The refference list is compared with every refference model stored in database on both Face score and Average Face Score of Neighboring Face Loop, and then one of the refference models which matches the nost can be obtained.

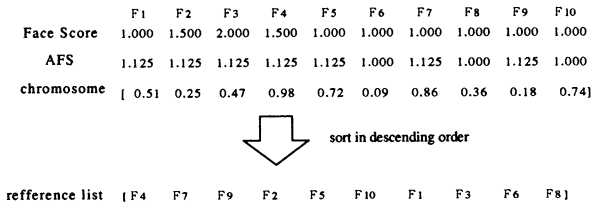


Figure 4: Chromosome representation

3.2 Crossover

The arithmetical crossover [6] is employed as crossover operator. The arithmetical crossover is defined as a convex combination of two vector. If the constraint set is convex, this operation ensures that children are feasible if both parents are feasible. For a pair of parents p_1 and p_2, the crossover operator can produce two offspring o_1 and o_2 as follows:

$$o_1 = \lambda \times p_1 + (1-\lambda) \times p_2$$
$$o_2 = \lambda \times p_2 + (1-\lambda) \times p_1$$

where λ is randomly generated from (0,1). An example is shown in Figure 5

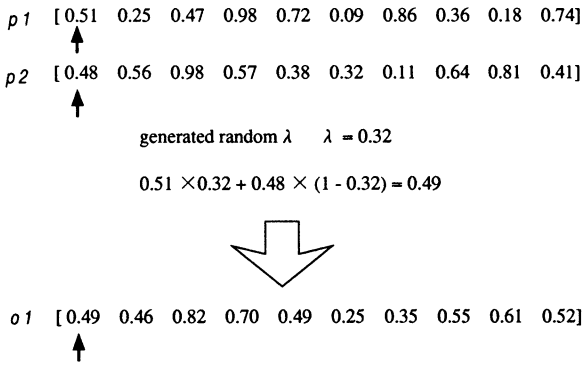


Figure 5: An example of a crossover operator

3.3 Mutation

Mutation is designed to perform swapping mutation [6], *i.e.*, it selects two genes randomly in a chromosome and exchanges their positions. An example is shown in Figure 6

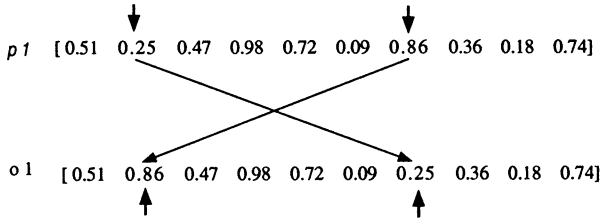


Figure 6: Illustration of a mutation operator

3.4 Evaluation and Selection

The evolution is controlled by the following evalutional function, and $i = 1,2,\ldots,n$ is denotes the numbers of face.

$$Sfs_i = |Bfs_i - Ifs_i| \quad (4)$$
$$Sns_i = |Bns_i - Ins_i| \quad (5)$$
$$T = \sum_{i=1}^{n} S_i \quad (6)$$
$$S_i = \begin{cases} 1 & ;\delta > Sfs_i \text{ and } Sns_i \\ 0 & ;\text{otherwise} \end{cases}$$
$$i = 1,2,\ldots,n$$

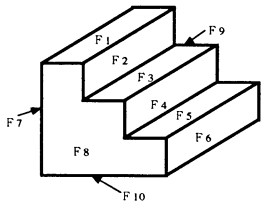
where

Bfs_i	the Face Score of face i for refference solid models
Ifs_i	the Face Score of face i for target solid models
Bns_i	the Average Face Score of face i in Neighboring Face Loop for refference solid models
Ins_i	the Average Face Score of face i in Neighboring Face Loop for traget solid model
δ	a threshold value
T	the agreement value

The agreement value T is used as the fitness for each chromosome, and the roulette wheel approach[6] is selection.

4 Experimental Result

In order to verify the effectiveness of the proposed method, we tested our method by recognizing a target solid model given in Figure 7 among the reference models shown in Figure 8.



Face Name	Face Type	The number of Edge	Face Score	Neighboring Face Loop (Edge Type)	AFS
F01	1	4	1.00	F07(1) - F08(1) - F02(1) - F09(1)	1.125
F02	1	4	1.50	F01(1) - F08(1) - F03(1) - F09(1)	1.125
F03	1	4	1.50	F02(1) - F08(1) - F04(1) - F09(1)	1.250
F04	1	4	1.50	F03(1) - F08(1) - F05(1) - F09(1)	1.250
F05	1	4	1.50	F04(1) - F08(1) - F06(1) - F09(1)	1.125
F06	1	4	1.00	F05(1) - F08(1) - F10(1) - F09(1)	1.125
F07	1	4	1.00	F01(1) - F09(1) - F10(1) - F08(1)	1.000
F08	1	8	1.00	F07(1) - F10(1) - F06(1) - F05(1) - F04(1) - F03(1) - F02(1) - F01(1)	1.250
F09	1	8	1.00	F07(1) - F01(1) - F02(1) - F03(1) - F04(1) - F05(1) - F06(1) - F10(1)	1.250
F10	1	4	1.00	F07(1) - F09(1) - F06(1) - F08(1)	1.000

Figure 7: Target solid model

The parameters for genetic algorithm are set as follows: *population size* = 4, *crossover rate* = 0.2, *mutation rate* = 0.7, *maximum generation* = 1000. The result is summarized in Table 3, where the result is obtained by averaging results among 100 runs. In Table 3, a reference model D was recognized as the fittest one with high frequency in 100 runs.

5 Conclusion

In this paper, we proposed the new feature recognition method of CAD data using evolutionary computation. And we demonstrated the effectiveness of the proposed method through an experimental result.

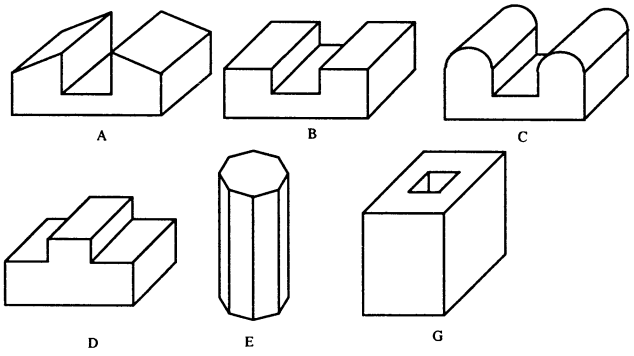


Figure 8: Reference models

Table 3: Recognition result

recognized model	A	B	D
frequency	1/100	12/100	87/100
average fitness	6.00	6.00	6.96
average generation	21.00	43.17	172.72

The rsult showed a good ability of our method for feature recognition. For feature study, we will improve the proposed method to deal with more complex solid models.

References

[1] Yasuhiro Tsujimura, Mitsuo Gen and Masaki Hiji: Feature Recognition of CAD Data Using Evolutionary Computation, *Proceedings of 1997 Spring Meeting of Japan Industrial Management Association*, pp.164 - 165, 1997. (in Japanese)

[2] Nikkei CG ed.: *New Foundation of CAD*, Nikkei BP, 1996.(in Japanese)

[3] Mamoru Hosaka and Toshio Sata: *Integrated CAD/CAM System*, Ohm Pub., 1994.(in Japanese)

[4] Yukinori Kakazu and Masasi Furukawa: *Shape Disposal Engineering for CAD/CAM/CG*, Morikita Pub., 1995. (in Japanese)

[5] Chu-Chai Henry Chan: *Artificial Neural-Network-Based Feature Recognition and Grammar-Based Feature Extraction to Integrate Design and Manufacturing*, Ph.D Disser., Univ. of Iowa, 1994.

[6] Mitsuo Gen and Runwei Cheng: *Genetic Algorithms and Engineering Design*, Jhon Wiley & Sons, New York, 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Bicriteria Network Design Using Spanning Tree-Based Genetic Algorithm

Mitsuo Gen, Kenichi Ida, and JongRyul Kim
Dept. of Indust. and Inform. Engg.
Ashikaga Institute of Technology
Ashikaga, Tochigi, 326, Japan

JaeUk Lee
Dept. of Computer Engg.
Dongseo University
Pusan, Korea

Abstract

Network optimization problems, especially, in the computer networking, LANs typically consist of several LAN segments connected together via bridges. The use of these transparent bridges requires *loop-free* paths between LAN segments. Therefore, only spanning tree topologies can be used as active LAN configurations.

In this paper, we attempt to the LAN topology design problem with bicriteria which is minimizing the cost and message delay using spanning tree-based genetic algorithm. Finally, we get some experiments in order to certify the quality of the designs obtained by using spanning tree-based genetic algorithm.

Keywords: Network Design, Message Delay, Spanning Tree-based Genetic Algorithm, Local Area Network(LAN)s

1 Introduction

The optimization problems for the communication network have been taken attentions by many related researchers, such as network designers, network analysts, network administrators according to the scale of communication network. Especially, in the computer networking, LANs are commonly used as the communication infrastructure that meets the demands of the users in local environment. These computer networks typically consist of several LAN segments connected together via bridges. The use of these transparent bridges requires *loop-free* paths between LAN segments[9]. Therefore, only spanning tree topologies can be used as active LANs configurations. The performance criteria of these systems, such as cost, message delay, traffic, reliability, and so on, are important and are largely determined by network topology.

Genetic algorithm(GA) has received a great deal of attention regarding their potential as optimization techniques for network optimization problems and is often used to solve many real-world problems[1][2]. And there have been many investigations of GA for multicriteria problems. GA approach has been reported in many applications with multiobjective, such as the transportation problems[8], the minimum spanning tree problems[7], the production process planning problems[5], and so on. The determination of the optimal LAN topology is a very complicated combinatorial optimization problem and can be classified as *NP*-hard optimization problem. Also, when the topology design problem has two objectives, the topology design of LAN is more difficult. Therefore, a heuristic algorithm that is based on a genetic algorithm gets attentions.

In this paper, we attempt to the network topology design problem with bicriteria which is minimizing the cost and the message delay using spanning tree-based genetic algorithm. Because of the feature that only spanning tree topology can be used as active LAN configurations, we employ the spanning tree-based genetic algorithm. We also employ the Prüfer number encoding to represent the spanning tree, which is capable of equally and uniquely representing all possible trees[6]. Using Prüfer number encoding, the memory only requires $n+m-2$ on the problem with n service centers and m users. The objectives of this paper is to minimize the cost needed to design the networks system and also to minimize the average networks message delay.

2 The Bicriteria LAN Topology Design

2.1 Formulation of the problem

We define the following notations in order to formulate the bicriteria LAN topology design considering the message delay and connected cost as the criteria: Let n be the number of service center and m be the number of user to be clustered. Suppose that $\boldsymbol{U} = [u_{jj'}]$ is $m \times m$ user traffic matrix where $u_{jj'}$ represents the traffic from user j to user j', $\boldsymbol{T} = [t_{ik}]$ is $n \times n$ service center traffic matrix where t_{ik} means the traffic forwarded from users in center i to users in center k. Also, we express $\boldsymbol{X}_1$ as $n \times n$ service center topology matrix where if the centers i and k are connected, $x_{1ik} = 1$, otherwise, $x_{1ik} = 0$, and $\boldsymbol{X}_2$ as $n \times m$ clustering matrix where if user j belongs to center i, $x_{2ij} = 1$, otherwise, $x_{2ij} = 0$. So we can suppose that $\boldsymbol{X}$ is $n \times (n+m)$ spanning tree matrix($[\boldsymbol{X}_1 \ \boldsymbol{X}_2]$). Let C_i be the traffic capacity of center i, and let β_{ik} be the delay per bit due to the link between centers i and k. We also define g_i as the maximum number which is capable of connecting to center i, w_{1ik} as the weight of the link between centers i and k, and w_{2ij} as the weight of the link between center i and user j. We also define the two kinds of traffic as follows:

$$a_{ik}^{p}(\boldsymbol{X}) = \begin{cases} 1, & \text{if traffic from center } i \text{ to center } k \\ & \text{through center } p \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

$$b_{ik}^{(p,q)}(\boldsymbol{X}) = \begin{cases} 1, & \text{if traffic from center } i \text{ to center } k \\ & \text{passes through existing link} \\ & \text{connecting centers } p \text{ and } q \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

And M/M/1 model[3] is used in this paper to describe a single cluster(LAN segment) behavior. Then we can formulate the bicriteria LAN topology design

problem as the following nonlinear 0-1 programming model:

$$\min \quad c_1(\boldsymbol{X}) = \frac{1}{\Gamma}\left[\sum_{i=1}^{n}\frac{l_i(\boldsymbol{X})}{C_i - l_i(\boldsymbol{X})} + \sum_{i=1}^{n}\sum_{k=1}^{n}\beta_{ik}\cdot f_{ik}(\boldsymbol{X})\right] \quad (1)$$

$$\min \quad c_2(\boldsymbol{X}) = \sum_{i=1}^{n-1}\sum_{k=i+1}^{n} w_{1ik}\cdot x_{1ik} + \sum_{i=1}^{n}\sum_{j=1}^{m} w_{2ij}\cdot x_{2ij} \quad (2)$$

$$\text{s. t.} \quad \sum_{j=1}^{m} x_{2ij} < g_i,\ i = 1,2,\cdots,n \quad (3)$$

$$\sum_{i=1}^{n} x_{2ij} = 1,\ j = 1,2,\cdots,m \quad (4)$$

$$l_i(\boldsymbol{X}) < C_i,\ i = 1,2,\cdots,n \quad (5)$$

where the total offered traffic Γ is represented as follows:

$$\Gamma = \sum_{i=1}^{n}\sum_{k=1}^{n} t_{ik}.$$

The total traffic at center k, $l_k(\boldsymbol{X})$, is represented as follows:

$$l_p(\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{k=1}^{n} t_{ik}\cdot a_{ik}^{p}(\boldsymbol{X}),\quad p = 1,\cdots,n.$$

The total traffic through link (p,q), $f_{pq}(\boldsymbol{X})$, is represented as follows:

$$f_{pq}(\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{k=1}^{n} t_{ik}\cdot b_{ik}^{(p,q)}(\boldsymbol{X}),$$

$$p = 1,\cdots,n \quad q = 1,\cdots,n.$$

We can also represent the service center traffic matrix as follows: $\boldsymbol{T} = \boldsymbol{X}_2^T\boldsymbol{U}\boldsymbol{X}_2$.

3 Spanning Tree-based GA

3.1 Representations and Initialization

Here we employ the spanning tree representation using Prüfer number in order to represent active LAN configurations.

We can use only $k-2$ digits permutation to uniquely represent a tree where each digit is an integer between 1 and k inclusive. This permutation is usually known as the *Prüfer number*. For any tree there are always at least two leaf nodes[10]. Based on this observation we can easily construct an encoding as follows:

Procedure: Prüfer Number Encoding

Step 1: Let node i be the smallest labeled leaf node in a labeled tree T.

Step 2: Let j be the first digit in the encoding as the node j incident to node i is uniquely determined. Here we build the encoding by appending digits to the right, and thus the encoding is built and read from left to right.

Step 3: Remove node i and the link from i to j; thus we have a tree with $k-1$ nodes.

Step 4: Repeat the above steps until one link is left. We produce a Prüfer number or an encoding with $k-2$ digits between 1 and k inclusive.

An example is given to illustrate this kind of encoding. The Prüer number(1 1 2 2) correspondings to a spanning tree on a five node complete graph represented in Figure 1.

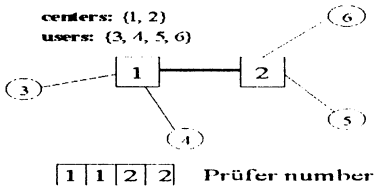


Figure 1: A tree and its Prüfer number

Prüer number is more suitable for encoding a spanning tree, especially in some research fields, such as transportation problems, minimum spanning problems, and so on. Also, the verification for the excellence of Prüer number is addressed by Zhou and Gen[6].

From Figure 1, we can see that all digits of the Prüfer number are the figures of the centers displayed by box. So, the initialization of a chromosome(*i.e.*, a Prüfer number) is performed from that randomly generated $n+m-2$ digits in range $[1,n]$. Therefore in a chromosome, the center is represented but the user does not appear in a chromosome. used as the elements of the set $\bar{P}$. Here suppose the centers are represented by the number from 1 to n and the users are represented by the number from $n+1$ to $n+m$.

3.2 Evaluation

With bicriteria LAN topology design problem, we can only calculate each objective value and can not simply evaluate its fitness value when the objective functions conflict with each other in practice. In other words, we can not obtain the absolute optimal solution but can only get the Pareto optimal solutions. Here, the Pareto optimal solutions concept is defined as follows: a feasible solution ($\boldsymbol{X}'$) is called Pareto optimal if there no exist no feasible solution $\boldsymbol{X}$ which would decrease some objective function without causing a simultaneous increase in at least one other objective function.

As to the fitness function for survival, we employ the weighted sums method to construct the fitness function. Then we use the following evaluation for combining the bicriteria objective functions into one overall fitness function and evaluating each chromosome.

Step 1: Convert chromosomes represented by Prüfer number to spanning matrix $\boldsymbol{X}$.

Step 2: Calculate the objective values($c_r(\boldsymbol{X})$, $r = 1, 2$).

Step 3: Choose the solution points which contain the minimum $c_r^{\min}$(or the maximum $c_r^{\max}$) corresponding to each objective function value, and then compare with the stored solution points at the previous generation and select the best points to save again.

$$c_r^{\min(t)} = \min_k \{c_r^{\min(t-1)}, c_r^{(t)}(\boldsymbol{X}_k)\}$$

$$c_r^{\max(t)} = \max_k \{c_r^{\max(t-1)}, c_r^{(t)}(\boldsymbol{X}_k)\}$$

$r = 1, 2 \qquad k = 1, 2, \cdots, chr\_size$ where $c_r^{\min(t)}$, $c_r^{\max(t)}$ is the minimum value and the maximum value of r-th objective function at generation t respectively, $c_r^{(t)}$ is the r-th objective function value of k-th chromosome at generation t, and chr_size is equal to the pop_size plus the offsprings generated after genetic operations.

Step 4: Solve the following equation to get weights for the fitness function:

$$\alpha_r = c_r^{\max(t)} - c_r^{\min(t)}, \quad r = 1, 2$$

$$\lambda_r = \frac{\alpha_r}{\sum_{r=1}^{2} \alpha_r}, \qquad r = 1, 2$$

Step 5: Calculate the fitness function value for each chromosome as follows:

$$eval(\boldsymbol{X}_k) = \sum_{r=1}^{2} \lambda_r c_r(\boldsymbol{X}_k), \; k = 1, 2, \cdots, chr\_size$$

Step 6: Convert the evaluation value to fitness as following equation:

$$F(\boldsymbol{X}_k) = \frac{1}{eval(\boldsymbol{X}_k)}, \; k = 1, 2, \cdots, chr\_size$$

where $F(\boldsymbol{X}_k)$ means a fitness value for the k-th chromosome.

3.3 Selection

The selection used here is the method combined with the *roulette wheel* and *elitist* approach. , in order to enforce the GA proposed to freely search solution space. The roulette wheel selection which is one of the fitness-proportional method is used to randomly reproduce new generation and the elitist method is employed to preserve the best chromosome for the next generation and overcome the stochastic errors of sampling.

Using this selection process, we can keep the best chromosome from the current generation to the next generation.

3.4 Crossover

We employed the multi-point crossover(or called uniform crossover). This type of crossover is accomplished by selecting two parent solutions and randomly taking a component from one parent to form the corresponding component of the offspring(see Figure 2).

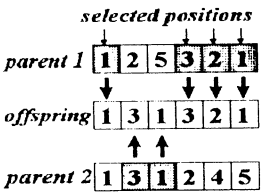


Figure 2: The crossover

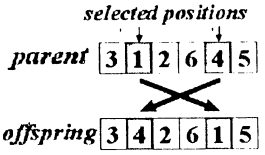


Figure 3: The mutation

3.5 Mutation

We used here swap mutation which simply select two positions at random and swap their contents(see Figure 3).

3.6 Modification of the Chromosomes

We adopt the repairing strategy to modify the connection number for center in an illegal chromosome. Let $\bar{G}$ be the set of centers whose the maximum number of connection has not been checked and modified in a chromosome. If a center i violates the constraint with the maximum number g_i of connection for center i, this means that the number of this center in the chromosome is more than $g_i - 1$. Then decrease the number of the center by checking the extra center and randomly replace it with another center from $\bar{G}$.

Obviously it is easy to go on with this modification method although there are different connection constraints on different centers.

4 Numerical Examples

The spanning tree-based genetic algorithm proposed in this paper for solving bicriteria LAN topology design problem is implemented in C language and run on a Pentium 133 PC. The performance of this approach is tested with the following problem: we employ the problem with 6 service centers($n = 6$), 30 users($m = 30$), $g_i = 10$, $C_i = 300$, and $\beta_{ij} = 0.1$. And it can be seen that the user matrix $\boldsymbol{U}$ is as follows[9]:

$$\boldsymbol{U} = \begin{pmatrix} U_3 & U_1 & O & U_1 & O & U_1 \\ U_1 & U_3 & U_1 & O & U_1 & O \\ O & U_1 & U_3 & U_1 & O & U_1 \\ U_1 & O & U_1 & U_3 & U_1 & O \\ O & U_1 & O & U_1 & U_3 & U_1 \\ U_1 & O & U_1 & O & U_1 & U_3 \end{pmatrix}$$

where O is the zero matrix, U_1 is 6×6 matrix where if $i + j = 7$, $u_{1ij} = 1$, otherwise $u_{1ij} = 0$, and U_3 is 6×6 matrix where diagonal components are 0 and the others are 3.

And the connection costs, *i.e.*, $w_{1ij} \in [100, 250]$, $w_{2ij} \in [1, 100]$, are randomly generated.

The parameters for genetic algorithm are set as follows: $pop\_size = 10$, $max\_gen = 500$, $p_C = 0.4$, $p_M = 0.1$ and experimented by 50 times. The Pareto optimal solutions can be found at most time which are illustrated in Figure 4. The obtained solutions is formed in Pareto frontier. Form the result, we can see the proposed algorithm has a good performance on the bicriteria LAN topology design problem.

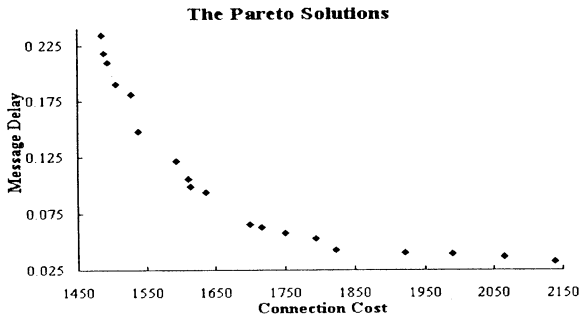


Figure 4: The Pareto Solutions for Example

We used the TOPSIS method proposed by Yoon and Hwang[11], to determine the best compromise solution among Pareto solutions. TOPSIS stands for technique for order preference by similarity to ideal solution, which is based upon the concept that the chosen alternative should have the shortest distance from the positive ideal solution and the farthest from the negative ideal solution. The detail information about TOPSIS can be referred in references [2] and [11]. So, in Figure 5, we represent the best compromise solution for the Pareto solutions of Figure4, using TOPSIS method. And the chromosome of the best compromise solution in Figure 5 is [6355664152334152532242112646556124] with 1614 connection cost and 0.09935 message delay.

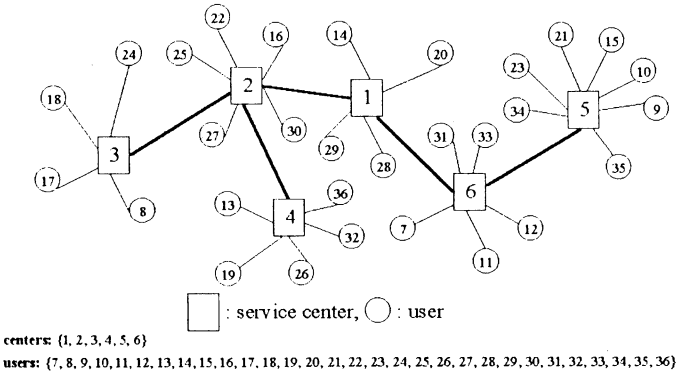


Figure 5: The Best Compromise Solution for Pareto Solutions

5 Conclusion

In this paper, we proposed a evolutionary algorithm by employing spanning tree-based genetic algorithm to solve bicriteria LAN topology design problems. We also developed the method to search for the Pareto solutions. And our proposed spanning tree-based method is able to find the LAN configurations, *i.e.*, partition of the users into service centers and routing between the service centers, with minimizing the connection cost and the average message delay. Finally, we try on some experiments to demonstrate the effectiveness of our proposed method. Also, the registered Pareto optimal solutions technique was used to give out the compromise solution and the TOPSIS method is used to determine the best compromise solution among Pareto solutions. The numerical analysis shows that the set of Pareto solutions approximately in nondominated hyperplane of the criteria space.

Acknowledgement

This research was supported by the International Scientific Research Program, the Grant-in-Aid for Scientific Research(No. 07045032:1995.4-1998.3) between Ashikaga Institute of Technology, Japan and Institute of Policy and Management, Chinese Academy of Sciences by the Ministry of Education, Science and Culture, the Japanese Government.

References

[1] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd ed., Springer-Verlag, New York, 1994.

[2] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.

[3] Bertsekas, D. and R. Gallager, *Data Networks*, 2nd ed., Prentice-Hall, New Jersey, 1992.

[4] Tanenbaum, A. S., *Computer Networks*, 3rd ed., Prentice-Hall, New Jersey, 1996.

[5] Zhou, G., and M. Gen, Evolutionary computation on multicriteria production process planning problem, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.419-424, 1997.

[6] Zhou, G., and M. Gen, Approach to degree-constrained minimum spanning tree problem using genetic algorithm, *Engineering Design & Automation*, Vol.3, No.2, pp.157-165, 1997.

[7] Zhou, G., and M. Gen, "The Genetic Algorithms Approach to the Multicriteria Minimum Spanning Tree Problem", *Proceedings of the First Asia-Pacific Conference on Simulated Evolution and Learning*, pp.387-394, 1996.

[8] Gen, M. and Y. Z. Li, Spanning tree-based genetic algorithm for bicriteria transportation problem, *Proceedings of Japan-China Joint International Workshops on Information Systems*, pp.123-134, 1997.

[9] Elbaum, R. and M. Sidi, Topological design of local-area networks using genetic algorithms, *IEEE/ACM Transactions on Networking*, Vol.4, No.5, 1996.

[10] Skiena, S., *Implementing Discrete Mathematics Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Reading, MA, 1990.

[11] Hwang, C. and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin, 1981.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB Ⅲ'98),
Oita, Japan, 19-21, January, 1998,

An Evolutionary Computation-based Model for Process Plan Selection Considering AGV Guidepath and Operation Flexibility

Chiung MOON, Janghyung Kim and Kijung An
Dept. of Information Engineering
Cheju National University
Cheju 690-756, Korea
E-mail:moon@genlab.ashitech.ac.jp

Mitsuo GEN
Dept. of Industrial & Systems Eng.
Graduate School of Engineering
Ashikaga Institute of Technology
Ashikaga 326, Japan
E-mail:gen@genlab.ashitech.ac.jp

Abstract

A new process plan selection problem considering operation flexibility and AGV guidepath is proposed. The process plan for each part should be selected based on the multiple criterions: (1) minimizing the sum of workstation processing time and the material handling time of all parts. (2) balancing workload between workstations. An evolutionary computation-based model is developed to solve this problem. The efficiency of the proposed approach is verified with a numerical example.

Keywords — Process Plan Selection, Operation Flexibility, AGV Guidepath, Genetic Algorithm

1. Introduction

Computer-aided process planning(CAPP) is the critical link between CAD and CAM, both of which need this indispensable interface. The development of a high-level CAPP system must be based on not only the thorough understanding of process planning principles and methodologies, but also the dynamic information of shop floor and CAD department[4,7].

One of the problems linking the process planning with the production planning and control problem are the problem of selecting process plans. Typically, for a part to be manufactured in a modern manufacturing system, multiple process plans are generated[7]. The process planners face the problem of selecting from the set of process plans a subset of process plans with the minimum corresponding costs.

Kusiak and Finke[6], Bhaskaran[2], Zhang and Huang[8], Awadh et al.[1] and Seo and Egbelu[7] developed models for the process plan selection problem.

In this paper, a new problem is proposed for the selection of a set of process plans. The problem in this paper differs from those of the traditional researches as follows: (1) the problem is considering the operation flexibility which is concerned with the possibility of performing an operation on more than one workstation. (2) the transportation and handling time is expresed as a function of the process plan and automated guided vehicle(AGV) guidepath. The problem is formulated as a bicriteria problem and an evolutionary algorithm based on a genetic algorithm(GA) is presented.

2. Model Development

In order to formulate a process plan selection problem, the following notations are introduced:

pt_{ijk} : unit processing time to perform operation j of part type i using workstation k

pv_i : production volume for part type i

cm_k : maximum available time of workstation k

tm_{kl} : transportation time from workstation k to l

abl : available capacity of AGV per trip

tw_k : workload in workstation k

$$tw_k = \frac{pt_{ijk} \times pv_i}{cm_k}$$

$\lceil w \rceil$: minimum integer value more than or equal to w.

ew : average workload of workstations

$$s_{ikl} = \begin{cases} 1 & \text{if two workstations } \boldsymbol{k}, \boldsymbol{l} \text{ for part type } \boldsymbol{i} \text{ are used successively} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{if workstation } \boldsymbol{k} \text{ selected to perform operation } \boldsymbol{j} \text{ of part type } \boldsymbol{i} \\ 0 & \text{otherwise} \end{cases}$$

2.1 Objective Functions

The total machining time for all part is determined by the production volume of the parts to be processed and the machining time associated with operations. The time function is defined as follows:

$$f_1 = \sum_{i}^{np} \sum_{j}^{no_i} \sum_{k}^{nt} pv_i \cdot pt_{ijk} \cdot x_{ijk} \qquad (1)$$

For the definition of transportation time, n_{ikl}, the number of trips between workstations k and l for part type i, can be calculated as follows:

$$n_{ikl} = s_{ikl} \times \left\lceil \frac{pv_i}{abl} \right\rceil \qquad (2)$$

The transportation time between workstations k and l for part type i, can be calculated as follows:

$$t_{ikl} = n_{ikl} \times tm_{kl} \qquad (3)$$

Therefore, the total transportation time associated with process plans for all parts is defined as follows:

$$f_2 = \sum_{i}^{np} \sum_{j}^{no_i - 1} \sum_{k}^{nt} \sum_{l}^{nt} t_{ikl} \cdot x_{ijk} \cdot x_{i(j+1)l} \qquad (4)$$

Consequently, in order to determine the process plans for all the parts, the objective function(F_1) is as follows:

$$F_1 = f_1 + f_2 \qquad (5)$$

The second objective function(F_2) for balancing workloads between workstations can be formulated as follows:

$$F_2 = \sum_{k}^{nt} \frac{(tw_k - ew)^2}{nt} \qquad (6)$$

2.2 Mathematical Model

The overall model can be formulated as follows:

minimize $F1$
minimize $F2$
subject to

$$\sum_{k}^{nt} x_{ijk} = 1, \qquad \forall (i, j) \qquad (7)$$

$$\sum_{i}^{np} \sum_{j}^{no_i} pv_i \cdot pt_{ijk} \cdot x_{ijk} \le cm_k, \ \forall k \qquad (8)$$

$$\sum_{i}^{np} \sum_{j}^{no_i} x_{ijk} \ge 1, \ \forall k \qquad (9)$$

$$x_{ijk} \in \{0, 1\}, \ \forall (i, j, k) \qquad (10)$$

The operation flexibility concerned with an operation can be performed on alternative workstations with the difference machining time. The first constraint ensures that only one workstation is selected for each operation of a part type. For the machining of part types, several workstations are used. The total machining time for all the part types using a workstation is less than or equal to one's available time. This constraint can be expressed as Equation (8).

Sometimes, several machinings can be concentrated on a special workstation. This is a cause of overload at any workstation. Thus, we can include a constraint to prevent the overload at each workstation and for the load balancing of all workstations. The constraint is given as Equation (9).

3. Genetic Algorithm Approach

GA is one of the intelligent search algorithms based on the principles of natural genetics and genetic selection. It is one of the suitable methodologies to solve the large scale and complex engineering design problems[3]. So, the GA that has been used to solve many NP-complete problems, can also be applied to process plan selection problem.

3.1 Representation and Initialization

In the process plan selection problem, each part type has a series of operations. An alternative workstation can be selected for one operation. Therefore, a gene of chromosome should contain the information of selected workstation standing for each operation. A chromosome representation defined as a series of the operations for all the parts is shown as follows:

Part1 Part 2 ··· Part *np*

[4 2 ··· *nt* 3 4 *nt* 2 3 ··· 1 ··· 4 5 6 1 ···]

In this chromosome, the cardinality of elements for each part means the number of operations and the assigned values to the elements mean workstation number for the machining of operations. The assignment of workstation to operations is made by generating random numbers and determine the corresponding workstation numbers.

The initial population of chromosomes is generated randomly within the range [1, *nt*].

3.2 Evaluation and selection

As the evaluation function for survival, the weighted sums method is used to construct the fitness function which multiple objective functions $F_1(C_k)$ and $F_2(C_k)$ are combined into one overall objective function at hand. The fitness function is handled in the following way.

(1) Choose the solution points which contain the minimum $F_1^{\min}$ (or $F_1^{\max}$) and $F_2^{\max}$ (or $F_2^{\min}$) corresponding to each objective function, then compare with the stored solution points at the previous generation and select the best points to store again.

$$F_q^{\min(t)} = \min_k\{F_q^{\min(t-1)}, F_q(C_k) | k = 1,2,\ldots, i\_size\},\ q = 1,2$$

$$F_q^{\max(t)} = \max_k\{F_q^{\max(t-1)}, F_q(C_k) | k = 1,2,\ldots, i\_size\},\ q = 1,2$$

where $F_q^{\max(t)}(F_q^{\min(t)})$ is the maximum (minimum) value of objective function q at generation t. And *i_size* is the number of individuals on current generation, i.e. the sum of population and generated offspring.

(2) Solve the following equations to get weights for evaluation function:

$$\delta_1 = F_1^{\max(t)} - F_1^{\min(t)},$$

$$\delta_2 = F_2^{\max(t)} - F_2^{\min(t)}$$

$$w_1 = \frac{\delta_1}{\delta_1 + \delta_2}, \qquad w_2 = \frac{\delta_2}{\delta_1 + \delta_2}$$

(3) Calculate the fitness value for each chromosome as follows:

$$eval(C_k) = w_1 F_1(C_k) + w_2 F_2(C_k) \quad (11)$$

In multicriteria context, usually the pareto optimal solutions are charac rized as the solutions of the multiobjective decision making problem. The module for pareto optimal solutions is illustrated as follows:

Module for pareto optimal solutions:

```
begin
  for generation index t = 0 to max_gen;
      count the number of chromosomes
      from generated offspring off_size;
      chr_size ← pop_size + off_size;
      for k = 1 to chr_size;
          evaluate a chromosome C_k;
          obtain the solution vector
          F_k = [F_1(C_k), F_2(C_k)];
          register Pareto optimal solutions
          and delete non-Pareto optimal
          solutions;
      endfor
  endfor
end
```

3.3 Crossover

A *part type exchange* crossover is proposed for the process plan selection problem. The procedure of crossover is as follows:

Step 1: Select a pair of chromosomes for crossover by generating random numbers.

Step 2: Generate a random number from the range [1, *np*] as the position of the crossover, that is, determine an arbitrary part number for crossover.

Step 3: Produce two offsprings by exchanging a set of genes of the selected position.

3.4 Mutation

In this problem, mutation is performed as random perturbation within the permitive integer range of the number of operations for a part type. The procedure of mutation operator is as follows:

Step 1: For selection of an arbitrary part type, a random integer number *no_part* is generated from the range [1, *np*].

Step 2: If a selected part type is *no_part* = *i*, randomly select a position *pos_m* for mutation from the range [1,*noi*].

Step 3: If a selected position is *pos* = *j*, replace the value of the position *j* with a generated random integer value from the range [1, *nt*].

4. Numerical Example

A system with 4 workstations and 4 part types is considered. The part types 1, 2, 3 and 4 have 6, 4, 5 and 4 operations respectively. The machining data is given in Table 1.

Table 1. Machining data for example

	Part 1						Part 2				Part 3					Part 4				MAT
OP / MA	1	2	3	4	5	6	1	2	3	4	1	2	3	4	5	1	2	3	4	
1	7	--	3	--	8	5	4	7	6	5	--	5	2	7	8	7	--	7	3	1500
2	8	3	--	6	2	8	7	5	--	4	4	7	2	--	3	3	7	4	--	1700
3	--	--	5	5	6	4	6	4	7	7	7	4	5	3	2	4	3	5	8	1800
4	5	2	7	9	5	3	3	8	6	8	4	5	3	7	3	7	--	7	3	1400
PV	44						52				38					45				

OP: a set of operations for each part MA:workstation
PV: production volume MAT: workstation capacity

In Table 2, the transportation time of AGV between workstations is given.

Table 2. Transportation time of AGV

To / From	1	2	3	4
1	12	9	7	12
2	4	6	8	-
3	-	12	8	12
4	8	7	12	7

The GA is implemented on IBM/PC compatible with Pentium 133. The GA is applied with the parameters as follows: max_gen=1000, pop_size=100, p_c=0.5 and p_m=0.3. After running the GA, we can obtain 9 pareto solutions as shown in Fig. 1.

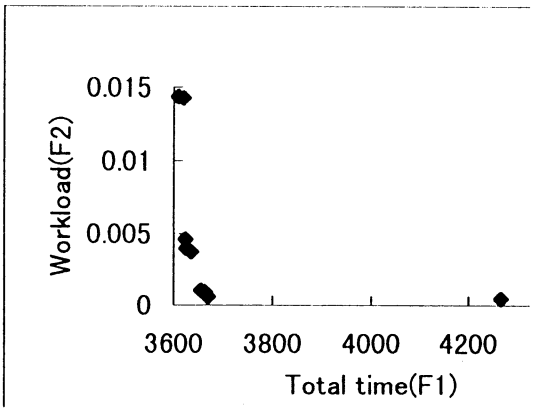


Fig. 1 Pareto solutions

The CPU time taken to solve the problem is 2.03 min. A solution of the corresponding process plan in the case of F1=4075 and F2=0.020855 is as follows:

4 4 1 3 2 3 4 2 1 2 2 3 2 3 3 2 3 2 1

The approach is found to be effective in offering a set of pareto solutions. The result indicate that this approach can be applied to real-life process plan selection problem with multiple objectives.

5. Conclusion

A problem of flexible process plan selection for CAPP system is studied by considering the AGV guidepath and operation flexibility. The problem is formulated as a bicriteria mathematical model minimizing the total processing and transportation time and minimizing the load variation between workstations. To solve this problem, also a GA approach is developed.

From the numerical example, the proposed approach is found to be effective in offering a set of satisfactory pareto solutions within a satisfactory CPU time, which is essential in a multiple objective environment, to enable to decision maker to determine the best solution. This approach can be effectively used to solve the complex and large size process plan selection problem whthin CAPP system.

Reference

[1] Awadh, B., N. Sepehri and O. Hawaleshka, A Computer-Aided Process Planning Model Based on Genetic Algorithms, *Computers & Ops. Res,* 22, 841-856, 1995.

[2] Bhaskaran, K., Process Plan Selection, *Int. J. Prod. Res,* 18, 1527-1539, 1990.

[3] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design,* John Wiley & Sons, New york, 1997.

[4] Ham, I and C. Y. Lu, Computer-Aided Process Planning: The Present and the Future. *Annals of the CIRP,* 37(2), pp.1-11, 1988.

[5] Kusiak, A., *Intelligent Manufacturing Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1990.

[6] Kusiak, A. and G. Finke, Selection of Process Plans in Automated Manufacturing Systems, *IEEE J. of Robotics and Automation,* 4, 397-402, 1988.

[7] Seo, Y. and P. G. Egbleu, Process Plan Selection Based on Product Mix and Production Volume, *Int. J. Prod. Res,* 34, 2639-2655, 1996.

[8] Zhang, H. C. and S. H. Huang, A Fuzzy Approach to Process Plan Selection, *Int. J. Prod. Res.* 32, 1265-1279, 1994.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Hitting Robot with a Flexible Link Hammer

T. Izumi
Dept. Electronic and Control Systems Eng.
Shimane University
Matsue, Shimane, Japan 690

H.Zhou
Dept. Electronic and Control Systems Eng.
Shimane University
Matsue, Shimane, Japan 690

Y. Hitaka
Dept. Electric Eng.
Ube National College of Technology
Ube, Yamaguchi, Japan 755

Abstract

This paper describes a hitting robot with a flexible link hammer which can hit an object from an arbitrary direction in a 3-dimensional space. The conditions that the hammer can strike an object flatwise with only a normal velocity to its face are investigated. And a manipulating method to satisfy the hitting conditions is obtained. This system can generate the precise impulsive force repeatedly.

1 Introduction

A flexible manipulator is applicable to various operations with a contact [1]. For these operations, the last link of the manipulator is chosen to be flexible while the other links are assumed rigid [2][3]. If a heavy payload is attached to the tip of the last flexible link, the manipulator can be used as a hitting robot [4].

This paper describes a hitting robot with a flexible link which can hit an object from any direction in a 3-dimensional space. The aim of this paper is to find out the hitting conditions, so that the robot can hit a specified point on an object from a normal direction without a tangential velocity to its face. First, a dynamical equation of a flexible link hammer is obtained with the consideration of gravity. The hitting time and swing-up angle of the hammer which satisfy the necessary conditions are derived from the dynamical equation and the geometrical relation between the hammer and the object. The base of the hammer is manipulated so that the hitting conditions are realized. Last, the theoretical results are verified by experiments of driving nails into a board in a gravitational field.

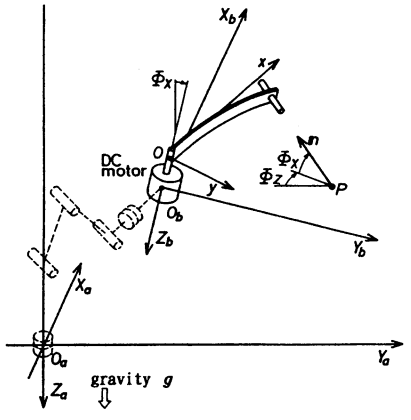


Figure 1: Concept of a flexible link hammer robot

2 Dynamical Equation

This hitting robot system is composed of a multi-degree of freedom manipulator and a flexible link hammer at the end-effector, as shown in Fig. 1. Let M be the mass of the hammer head, and ρA be the line density of the flexible link. The hammer head is subjected to a gravity force $Mg\sin\Phi_X$ as shown in Fig. 2. Ignoring the distributed gravity force on the link as $\rho A\ell < M$, and letting $\theta(t)$ be an angle of the hub rotated by the DC motor, the component of the gravity projected along the direction of the link deflection is

$$P(t) = Mg\sin\Phi_X\cos\theta(t) \tag{1}$$

The deflection $y(x,t)$ of the link in Σ_r is governed by

$$EI\,y''''(x,t) + \rho A\ddot{y}(x,t) = -\rho Ax\ddot{\theta}(t) + Mg\sin\Phi_X\cos\theta(t)\delta(x-\ell) \tag{2}$$

Here the prime and the dot indicate partial differentiation with respect to x and t, respectively, and EI is the uniform flexural stiffness. The second term of the right hand side of (2) is a exciting force due to gravity. It becomes zero when the link is driven within a horizontal plane, and even under gravity, is deleted from (2) in a steady state problem of vibration when the angle $\theta(t)$ is independent of t . However, the transient

deflection of the link due to the variable $\theta(t)$ must be taken into consideration in a hitting operation. Therefore, the second term in the right hand side of (2) is very important in this paper. Assumed to be concentrated at the end $x = \ell$, the force is represented using a delta function. The link at $x = 0$ is clamped to the hub and its inertia is assumed to be infinity because the DC motor is driven by a speed control method with high gain feedback. The hammer head has not only mass M but also rotary inertia J. These boundary conditions lead to the frequency equation . The i-th root β_i of this equation gives a natural frequency of the i-th mode of vibration,

$$\Omega_i = \sqrt{\frac{EI}{\rho A}} \left(\frac{\beta_i}{\ell}\right)^2, \quad i = 1, 2, 3, \cdots, N \tag{3}$$

and the i-th eigen-function $\phi_i(x)$. The solution of (2) is represented by a linear combination of $\{\phi_i(x) : i = 1, 2, 3, \cdots, N\}$

$$y(x,t) = \sum_{i=1}^{N} a_i(t)\phi_i(x) \tag{4}$$

where the i-th coefficient $a_i(t)$ has to satisfy the following equation.

$$\ddot{a}_i(t) + \Omega_i^2 a_i(t) = -b_i\ddot{\theta}(t) + \phi_i(\ell)\frac{Mg\sin\Phi_X}{\rho A}\cos\theta(t) \tag{5}$$

where b_i is an inner product $(x, \phi_i(x))$. The values b_i and $\phi_i(\ell)$ converge to zero as $i \to \infty$. The right hand side of (5) has not only an acceleration of the hub but also a exciting force due to gravity. If $\Phi_X = 0$, equation (5) coincides with the formula where the hammer is driven in a horizontal plane .

3 Hitting Condition

3.1 *Hitting from the Normal Direction*

In a hitting operation, it is desired that an impact force of a hammer works effectively and that an object face is not damaged by the edge of the hammer head. Therefore, the face of the hammer head must contact with an object face flatwise. However, it is difficult to strike an object flatwise because the relative deflection of the link has many vibration modes as known from (4). In this paper, the strikes are achieved by utilizing only the first mode and suppressing the higher modes of the link vibration. Since it is easy to suppress the higher modes , only the first mode of the link vibration is discussed so as to strike an object flatwise. In order to solve (5) for $i = 1$, the initial values of $a_1(t)$ are given from a static deflection $y(0)$ and an initial velocity $\dot{y}(0)$ as follows.

$$\begin{aligned} a_1(0) &= G(\phi_1(\ell)/\Omega_1^2)\cos\theta_s \\ \dot{a}_1(0) &= 0 \end{aligned} \tag{6}$$

where G denotes the coefficient related to gravity as

$$G = \frac{Mg\sin\Phi_x}{\rho A} \tag{7}$$

and θ_s is the angle of the link at $x = 0$ and $t = 0$. Even if the hub is accelerated by a function $\ddot{\theta}(t) = (\omega_a/T_a)$ for $0 < t < T_a$, the solution of (5) is expressed by a very complicated form. Therefore $\omega_a < \Omega_1$ is assumed for simplicity. Then the relative deflection of the first mode can be represented for $t \geqq T_a$ as

$$\begin{aligned} y_1(x,t) &= -\omega_a(b_1\phi_1(x)/\Omega_1)\sin\Omega_1(t - T_a/2) \\ &\quad + G(\phi_1(x)\phi_1(\ell)/\Omega_1^2)\cos\{\theta_s + \omega_a(t - T_a/2)\} \end{aligned} \tag{8}$$

The tangent angle $\gamma(t)$ of the link at $x = \ell$ is given by differentiating (8) with respect to x as

$$\begin{aligned} \gamma(t) &= \tan^{-1}\left[-\omega_a(b_1\phi_1'(\ell)/\Omega_1)\sin\Omega_1(t - T_a/2)\right. \\ &+ \left. G(\phi_1'(\ell)\phi_1(\ell)/\Omega_1^2)\cos\{\theta_s + \omega_a(t - T_a/2)\}\right] \end{aligned} \tag{9}$$

On the other hand, the angular acceleration $\ddot{\theta}(t) = (\omega_a/T_a)$ for $0 < t < T_a$ makes the angular velocity $\omega(t)$. In this case the angle $\theta(t)$ of the hub is represented for $t \geq T_a$ by

$$\theta(t) = \omega_a(t - T_a/2) + \theta_s \tag{10}$$

In general, the hitting face of the hammer head is parallel to the tangent of the link at $x = \ell$. Therefore an

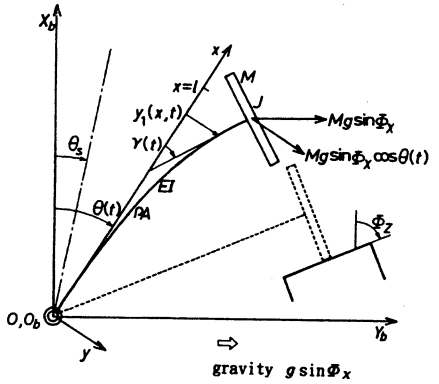


Figure 2: Motion of the flexible link hammer in the x-y plane which is rotated by Φ_X

object can be stricken flatwise from a normal direction when the hammer collides with it in the manner as shown by the dotted line in Fig. 2. This condition is realized when

$$\gamma(t) = 0 \tag{11}$$

$$\theta_s + \omega_a(t - T_a/2) = \Phi_Z \tag{12}$$

The solutions (t_1, θ_{s1}) of simultaneous equations of (12) and (11) based on (9) are obtained as

$$t_1 = (\pi - \alpha)/\Omega_1 + T_a/2 \tag{13}$$

$$\theta_{s1} = \Phi_Z - \omega_a(\pi - \alpha)/\Omega_1 \tag{14}$$

where

$$\alpha = \sin^{-1}\{G\phi_1(\ell)/(b_1\Omega_1\omega_a) \cdot \cos\Phi_Z\} \tag{15}$$

The positive and smallest value is selected for t_1 out of many solutions in order to get a useful hitting condition. In this case $\theta(t)$ starts from θ_{s1} and the corresponding $\gamma(t)$ becomes zero at t_1.

Consequently the flexible link hammer can hit the object face flatwise when the hub is angled at θ_{s1} before hitting.

3.2 *Velocity Components of the Hammer Head*

Let us show the velocity of the tip of the hammer head when the hub is driven by an angular velocity vector ${}^b\boldsymbol{\omega} = [0, 0, \omega_a]^T$ in $\boldsymbol{\Sigma}_b$. The velocity of the tip is the sum of two vectors, namely, a velocity vector ${}^b\boldsymbol{v}_r$ due to the rotation of $\boldsymbol{\Sigma}_r$ and a velocity vector ${}^r\boldsymbol{v}_f$ due to the deflection of the flexible link. The former is represented by a cross product

$${}^b\boldsymbol{v}_r = {}^b\boldsymbol{\omega} \times {}^bT_r(\theta)[\ell, h, 0]^T \tag{16}$$

where h is the height of the hammer head, and ${}^bT_r(\theta)$ is a rotation transformation about the Z_b axis as

$${}^bT_r(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

The x and y components of the latter are given by using (8) as

$${}^r\boldsymbol{v}_{fx} = -h\dot{y}_1'(\ell, t), \quad {}^r\boldsymbol{v}_{fy} = \dot{y}_1(\ell, t) \tag{18}$$

Assuming $\omega_a/\Omega_1^2 \ll 1$, the vector ${}^r\boldsymbol{v}_f$ is expressed by

$${}^r\boldsymbol{v}_f = \begin{bmatrix} hb_1\omega_a\phi_1'(\ell)\cos\Omega_1(t - T_a/2) \\ -b_1\omega_a\phi_1(\ell)\cos\Omega_1(t - T_a/2) \\ 0 \end{bmatrix} \tag{19}$$

The resultant velocity vector is represented by

$${}^b\boldsymbol{v} \triangleq [v_{Xb}, v_{Yb}, 0]^T = {}^b\boldsymbol{v}_r + {}^bT_r(\theta){}^r\boldsymbol{v}_f \tag{20}$$

By using these expressions, the vector ${}^b\boldsymbol{v}$ can be analyzed into two components of a normal velocity $v_n(t)$ and a tangential velocity $v_t(t)$ to the hammer head face as follows:

$$[v_n(t), v_t(t), 0]^T = {}^rT_b(\pi/2 + \theta(t) + \gamma(t)) \; {}^b\boldsymbol{v} \tag{21}$$

If the hammer is swung-up by θ_{s1} of (14) and hits an object at $t = t_1$, $\gamma(t_1)$ becomes zero. Then $v_n(t_1)$ and $v_t(t_1)$ are represented by

$$v_n(t_1) = \ell\{1 + b_1(\phi_1(\ell)/\ell)\cos\alpha\}\omega_a \tag{22}$$

$$v_t(t_1) = h(1 + b_1\phi_1'(\ell)\cos\alpha)\omega_a \tag{23}$$

It follows from (23) that a hitting at $t = t_1$ brings a tangential velocity to the object face when h is not ignored. This velocity component causes various troubles such as the inability to drive a nail perpendicularly into a board. Therefore the tangential velocity must be made zero at the hitting instant.

3.3 *Hitting from the Normal Direction without a Tangential Velocity*

The tangential velocity in (21) becomes zero when

$$\tan(\theta(t) + \gamma(t)) = -v_{Xb}(t)/v_{Yb}(t) \tag{24}$$

where $v_{Xb}(t)$ and $v_{Yb}(t)$ are the components of ${}^b\boldsymbol{v}$ as defined in (20). The right hand side of (24) is reduced into $\tan(\theta(t) - \kappa(t))$ by substituting (20) for $v_{Xb}(t)$ and $v_{Yb}(t)$, respectively, and by defining

$$\tan\kappa(t) \triangleq (-h\omega_a + {}^r\boldsymbol{v}_{fx}(t))/(\ell\omega_a + {}^r\boldsymbol{v}_{fy}(t))$$

which can be rewritten by the substitution of (19) as

$$\tan\kappa(t) = -\frac{h\{1 - b_1\phi_1'(\ell)\cos\Omega_1(t - T_a/2)\}}{\ell\{1 - b_1(\phi_1(\ell)/\ell)\cos\Omega_1(t - T_a/2)\}} \tag{25}$$

Consequently a tangential velocity is equal to zero when

$$\gamma(t) = -\kappa(t) \tag{26}$$

On the other hand, the hammer can strike an object face with Φ_Z when

$$\theta_s + \omega_a(t - T_a/2) + \gamma(t) = \Phi_Z \tag{27}$$

as seen from Fig. 3. It is impossible to obtain an analytical solution for (t_{1d}, θ_{s1d}) from these hitting conditions (27) and (26) into which (9) and (25) are substituted. Therefore we introduce a small time delay d

from t_1 of (13). A new hitting time t_{1d} is proposed by using d as

$$t_{1d} = (\pi - \alpha)/\Omega_1 + T_a/2 + d \tag{28}$$

A solid line in Fig. 3 shows the posture of the flexible link hammer at $t = t_{1d}$. Let us determine d from (26) and (27) into which (28) is substituted. In (9) and (25) $\sin\Omega_1(t-T_a/2)$ and $\cos\Omega_1(t-T_a/2)$ are assumed to be $-(\Omega_1 d - \alpha)$ and -1, respectively, since $|\Omega_1 d - \alpha| < 1$. In this case $\tan\kappa(t)$ in (25) turns into

$$\begin{aligned} \tan\kappa(t_{1d}) &= -\frac{h(1+b_1\phi_1'(\ell))}{\ell\{1+b_1\phi_1(\ell)/\ell\}} \\ &\triangleq -k \end{aligned} \tag{29}$$

The value k is a constant determined by the size of a hammer and the first eigen-function $\phi_1(x)$ of the flexible link. By substituting (27) into (9) and using (29) instead of (25), equation (26) is rewritten into

$$\begin{aligned} &\omega_a(b_1\phi_1'(\ell)/\Omega_1)(\Omega_1 d - \alpha) \\ &+G(\phi_1'(\ell)\phi_1(\ell)/\Omega_1^2)\cos(\Phi_Z - k) = k \end{aligned} \tag{30}$$

As seen from (15), the value α in (30) can be approximated for $\alpha < 1$ to $\alpha = G\phi_1(\ell)/(b_1\Omega_1\omega_a) \cdot \cos\Phi_Z'$, where Φ_Z' is shown by a dotted line in Fig. 3 and is equal to $\Phi_Z - (\omega_a d + k)$. It is rewritten for $\omega_a d < 1$ as

$$\alpha = G\phi_1(\ell)/(b_1\Omega_1\omega_a)\cdot\{\cos(\Phi_Z - k) + \omega_a d\sin(\Phi_Z - k)\} \tag{31}$$

The delay time d is expressed by substituting (31) into (30) as

$$d = \frac{k}{\omega_a\phi_1'(\ell)\{b_1 - G\phi_1(\ell)\sin(\Phi_Z - k)/(\Omega_1^2)\}} \tag{32}$$

This can be calculated easily from ω_a, Φ_Z, and G which is related to Φ_X. The new hitting time t_{1d} is obtained from (28) using (31) and (32). A swing-up angle θ_{s1d} is also represented by using d and α as

$$\theta_{s1d} = \Phi_Z - \omega_a(\pi - \alpha)/\Omega_1 - (\omega_a d + k) \tag{33}$$

Finally the hammer can hit an object flatwise without a tangential velocity at $t = t_{1d}$ when the hub is angled at θ_{s1d} of (33) before hitting.

4 Application

Some products are inspected by sound and vibration which are generated by excitation of an impulse force. In these inspection, it is neccesary to generate a specified impulse force repeatedly when an inspected system is non-linear. However a human is not good at generating the specified impulse force by hitting. The proposed robot system can generate a specified impulse force by hitting using a flexible link hammer.

5 Summary

Hitting by a flexible link hammer under gravity has been taken into consideration. The motion of the hammer subjected to gravity has been analyzed in order that the robot can hit an object from any direction in a 3-dimensional space. This paper has derived the hitting conditions that the hammer can strike an object flatwise without a tangential velocity to its face. The conditions are realized by adjusting the swing-up angle of the link and positioning the base of the hammer. It is seen from experiments that the flexible link hammer can drive a nail perpendicularly into a board.

References

[1] W.J.Book and D.Kwon, "Contact control for advanced applications of light weight arms," J. Intelligent and Robotic Systems, Vol.6, pp.121-137, 1992.

[2] D.M.Rovner and R.H.Cannon, "Experiments toward on-line identification and control of a very flexible one-link manipulator," Int. J. Robotics Res., Vol.6, No.4, Winter 1987.

[3] D.Wang and M. Vidyasagar, "Modeling a class of multilink manipulators with the last link flexible," IEEE Trans. Robotics and Automation, Vol.8, No.1, February 1992.

[4] T.Izumi and Y.Hitaka, "Control of impact for a hammering robot using a flexible link," Proc. of the IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems, pp.1347-1352, 1992.

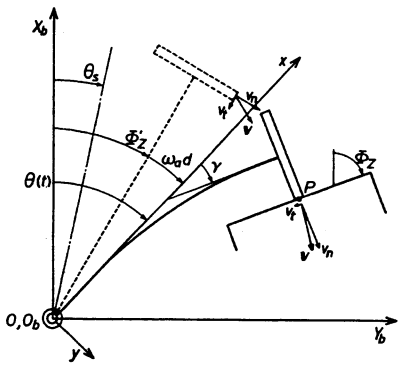


Figure 3: The proposed hitting condition.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Stability Analysis of 3-D Grasps Considering Rolling Contact

Sushanta Kumar Saha *, Takayoshi Yamada *, Takeshi Sakurayama *,
Nobuharu Mimura **, and Yasuyuki Funahashi *

*: Department of Mechanical Engineering
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466, Japan.
Email: sushanta, yamat@eine.mech.nitech.ac.jp

**: Department of Information Engineering
Niigata University
Ninomachi, Igarashi, Niigata 950-21, Japan.

ABSTRACT

This paper addresses stability of 3D grasp considering rolling contact and curvatures at contact point using potential energy approach. The kinematics of rolling contact equations (velocity and acceleration) between the object and finger are derived. Then the stiffness matrix of the grasp is established. The inverse problem, which is to determine the spring stiffness from the positive definiteness of stiffness matrix of the grasp, can be also derived.

***Key words**: stability, rolling contact, inverse problem.*

1. Introduction

While manipulation tasks performed by a multifingered robot hand, grasp stability is an important issue. Multifingered hands have less chance to break at contact points. Many authors have investigated the grasp stability from the viewpoint of potential energy.

Hanafusa et al. [1] analyzed stability of frictionless planar grasps with elastic finger model, and showed that the grasp is stable if potential energy stored in the grasp system is local minimum. Nguyen [2] proposed virtual stiffness model for frictionless 2-D grasps, and evaluated stability of the grasp whether the stiffness matrix is positive definite or not. Kaneko et al. [3] discussed frictional planar grasps by using two-dimensional virtual stiffness model, and derived stiffness condition which make the stable grasp. Mimura et al. [4] extended it to frictional 3-D grasps with three fingers.

However, authors [3] and [4] were assumed that the position of contact points on the object and the fingertip does not move. If contact point movement is allowed, we have to consider the curvature of object and finger at contact points and it will be more significant in practical use. Therefore, the authors [5] discussed 2-D grasps considering the curvatures and the rolling contact.

This paper discusses the stability of 3-D grasp considering rolling contact using potential energy approach. The synthesis of stable grasps is made by three-dimensional virtual springs, which are fixed at the center of the fingertip. One spring is parallel to the normal and the others are parallel to the tangent plane at the contact. The kinematics of rolling contact equations between the object and finger are derived. Then the stiffness matrix of the grasp is established. Stiffness condition for stabilization of the grasp is also analyzed.

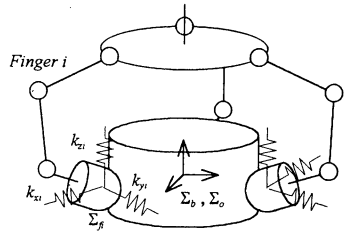


Fig. 1: Coordinates and virtual stiffness

2. Stability Analysis of Multifinger Grasp

The grasping configuration of a cylindrical object by a multifingered hand is shown in Fig. 1. The contact between object and fingers are of rolling contact type, i.e., object rolls without finger slipping.

2.1. Assumptions

To simplify our discussions, we make the following assumptions.

(A1) The shape of the object is known. The object and fingertips are rigid. The fingertip is constant radius ρ_{fi}.

(A2) Each finger is replaced by three virtual springs k_{xi}, k_{yi}, k_{zi} (>0) in Fig. 1.

(A3) Initial grasp configuration is known and is in equilibrium state.

(A4) The motion of the grasp system is quasi-static.

(A5) Friction coefficient is sufficient enough to prevent sliding motion of the fingers.

(A6) The orientation of the fingertip does not change with respect to the base frame because each finger has enough degrees of freedom.

2.2. Coordinate Systems

Coordinates of grasp system are shown in Fig. 1. Base frame, object frame, and i-th finger frame are denoted by Σ_b, Σ_o, and Σ_{fi}, respectively. Σ_{fi} is fixed at the center of fingertip and three-dimensional virtual

spring is fixed along the axes of Σ_{fi}. Σ_o is initially located at Σ_b. The displacement of the object due to external disturbances is denoted by $\varepsilon = [x^T, \xi^T]^T$. $x = [x, y, z]^T$ and $\xi = [\xi, \eta, \zeta]^T$ are translation and rotation, respectively.

2.3. Index of Grasp Stability

After the movement due to disturbances, compression of the virtual springs are $\delta_{xi}(\varepsilon)$, $\delta_{yi}(\varepsilon)$, and $\delta_{zi}(\varepsilon)$, respectively. The potential energy stored in the grasp system is given by

$$U = \frac{1}{2}\sum_{i=1}^{n}\{k_{xi}(\delta_{xi}+\delta_{xoi})^2 + k_{yi}(\delta_{yi}+\delta_{yoi})^2 + k_{zi}(\delta_{zi}+\delta_{zoi})^2\}. \quad (1)$$

where δ_{xoi}, δ_{yoi}, and δ_{zoi} in the initial compression configuration. Then the initial grasping forces are

$$f_{xi} \equiv k_{xi}\delta_{xoi},\ f_{yi} \equiv k_{yi}\delta_{yoi},\ f_{zi} \equiv k_{zi}\delta_{zoi}$$

The grasp system is stable at the initial configuration if and only if the potential energy is local minimum. Using the Taylor series at $\varepsilon = 0$, we have

$$U(\varepsilon) = U(0) + \varepsilon^T \nabla U\big|_{(0)} + \tfrac{1}{2}\varepsilon^T H\big|_{(0)}\varepsilon + \cdots, \quad (2)$$

where

$$\nabla = \frac{\partial}{\partial \varepsilon} \in \Re^6,\quad H = \nabla\nabla^T U \in \Re^{6\times 6}. \quad (3)$$

The grasp is stable if and only if the following two conditions are satisfied.

(i) $\nabla U\big|_{(0)} = 0$.

(ii) $H\big|_{(0)}$ is positive definite.

From Assumption (A3), the condition (i) is satisfied. So the stability depends only on the condition (ii).

$H\big|_{(0)}$ is expressed by

$$\begin{aligned} H\big|_{(0)} = \sum_{i=1}^{n}\{&k_{xi}(\nabla\delta_{xi}\big|_{(0)})(\nabla\delta_{xi}\big|_{(0)})^T \\ &+ k_{yi}(\nabla\delta_{xi}\big|_{(0)})(\nabla\delta_{xi}\big|_{(0)})^T \\ &+ k_{zi}(\nabla\delta_{zi}\big|_{(0)})(\nabla\delta_{zi}\big|_{(0)})^T + f_{xi}(\nabla\nabla^T\delta_{xi}\big|_{(0)}) \\ &+ f_{yi}(\nabla\nabla^T\delta_{yi}\big|_{(0)}) + f_{zi}(\nabla\nabla^T\delta_{zi}\big|_{(0)})\}. \end{aligned} \quad (4)$$

The derivatives $\nabla\delta_{xi}\big|_{(0)}$ through $\nabla\nabla^T\delta_{zi}\big|_{(0)}$ are derived in the following sections.

3. Grasp Analysis

We derive the stiffness matrix of the cylindrical object. Subscript oi and fi mean object and finger, respectively.

3.1. Notations

c : contact position,

n: outward unit normal at contact point,

$u = [u, v]^T$: parameters of contact point, where u and v are longitude and latitude, respectively.

The vectors c_{oi}, n_{oi}, c_{fi} and n_{fi} are given by

$$c_{oi} = \rho_o[\cos v_{oi}, \sin v_{oi}, \tan u_{oi}]^T, \quad (5)$$

$$n_{oi} = -[\cos v_{oi}, \sin v_{oi}, 0]^T, \quad (6)$$

$$c_{fi} = \rho_{fi}[\cos u_{fi}\cos v_{fi}, \cos u_{fi}\sin v_{fi}, \sin u_{fi}]^T, \quad (7)$$

$$n_{fi} = -c_{fi}/\rho_{fi}. \quad (8)$$

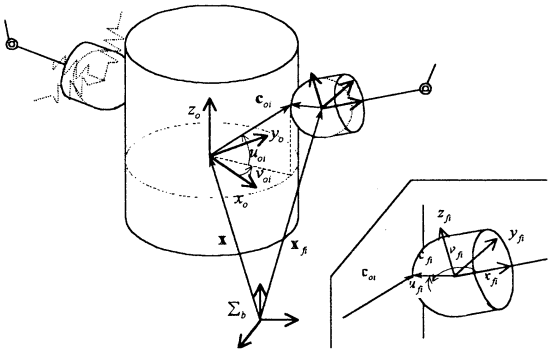


Fig. 2: Grasping of a cylindrical object

3.2. Compression of the springs

This section analyzes the relation between the object displacement and the spring compression. Since the fingertip and object stay in contact as shown in Fig. 2, the following holds.

$$x + R_o c_{oi} = x_{fi} + R_{fi} c_{fi}. \quad (9)$$

R_o and R_{fi} imply the orientation of Σ_o and Σ_{fi} with respect to Σ_b, respectively. The matrix R_o is given by

$$R_o = R_z(\zeta)R_y(\eta)R_x(\xi) \in \Re^{3\times 3}.$$

Using (A3) and (A6), we obtain

$$R_{fi} = [r_{xi}, r_{yi}, r_{zi}] = R_z(\theta_i) \in \Re^{3\times 3},$$

$$\phi_i \equiv u_{oi}\big|_{(0)},\ \theta_i \equiv v_{oi}\big|_{(0)},\ u_{fi}\big|_{(0)} = 0,\ v_{fi}\big|_{(0)} = \pi.$$

Since c_{oi} and c_{fi} depend only on ξ, spring compression is given by

$$\begin{aligned} [\delta_{xi}, \delta_{yi}, \delta_{zi}]^T &= R_{fi}^{-1}\{x(\varepsilon) - x_{fi}(0)\} \\ &= R_{fi}^T\{x + R_o c_{oi} - R_{fi}c_{fi} - x_{fi}(0)\} \\ &= [g_i(x), p_i(x), r_i(x)]^T + [h_i(\xi), q_i(\xi), w_i(\xi)]^T, \end{aligned} \quad (10)$$

where

$$[g_i, p_i, r_i]^T = R_{fi}^T\{x - x_{fi}(0)\},$$

$$[h_i, q_i, w_i]^T = R_{fi}^T(R_o c_{oi} - R_{fi}c_{fi}).$$

Taking the partial derivative of δ_{xi} with respect to ε are

$$\nabla\delta_{xi} = [\frac{\partial g_i}{\partial \boldsymbol{x}}, \frac{\partial h_i}{\partial \xi}]^T = [g_{ix}, g_{iy}, g_{iz}, h_{i\xi}, h_{i\eta}, h_{i\zeta}]^T, \quad (11)$$

$$\nabla\nabla^T\delta_{xi} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ \hline 0_{3\times3} & \begin{matrix} h_{i\xi\xi} & h_{i\xi\eta} & h_{i\xi\zeta} \\ h_{i\eta\xi} & h_{i\eta\eta} & h_{i\eta\zeta} \\ h_{i\zeta\xi} & h_{i\zeta\eta} & h_{i\zeta\zeta} \end{matrix} \end{bmatrix} \in \Re^{6\times6}. \quad (12)$$

The partial derivative of g_i with respect to $\boldsymbol{x}$ *is*

$$\frac{\partial g_i}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}}\{\boldsymbol{r}_{xi}{}^T\{\boldsymbol{x} - \boldsymbol{x}_{fi}(0)\}\} = \boldsymbol{r}_{xi}{}^T. \quad (13)$$

Taking the partial derivative of h_i with respect to α and β respectively yields

$$h_{i\alpha} = \boldsymbol{r}_{xi}^T\left(\frac{\partial R_o}{\partial\alpha}\boldsymbol{c}_{oi} + R_o\frac{\partial \boldsymbol{c}_{oi}}{\partial \boldsymbol{u}_{oi}}\frac{\partial \boldsymbol{u}_{oi}}{\partial\alpha} - R_{fi}\frac{\partial \boldsymbol{c}_{fi}}{\partial \boldsymbol{u}_{fi}}\frac{\partial \boldsymbol{u}_{fi}}{\partial\alpha}\right), \quad (14)$$

$$\begin{aligned} h_{i\alpha\beta} = \boldsymbol{r}_{xi}^T\Bigg\{ & \frac{\partial^2 R_o}{\partial\beta\partial\alpha}\boldsymbol{c}_{oi} + \frac{\partial R_o}{\partial\alpha}\frac{\partial \boldsymbol{c}_{oi}}{\partial\beta} + \frac{\partial R_o}{\partial\beta}\frac{\partial \boldsymbol{c}_{oi}}{\partial \boldsymbol{u}_{oi}}\frac{\partial \boldsymbol{u}_{oi}}{\partial\alpha} \\ & + R_o\frac{\partial}{\partial\beta}\left(\frac{\partial \boldsymbol{c}_{oi}}{\partial \boldsymbol{u}_{oi}}\right)\frac{\partial \boldsymbol{u}_{oi}}{\partial\alpha} + R_o\frac{\partial \boldsymbol{c}_{oi}}{\partial \boldsymbol{u}_{oi}}\frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\beta\partial\alpha} \\ & - R_{fi}\frac{\partial}{\partial\beta}\left(\frac{\partial \boldsymbol{c}_{fi}}{\partial \boldsymbol{u}_{fi}}\right)\frac{\partial \boldsymbol{u}_{fi}}{\partial\alpha} - R_{fi}\frac{\partial \boldsymbol{c}_{fi}}{\partial \boldsymbol{u}_{fi}}\frac{\partial^2 \boldsymbol{u}_{fi}}{\partial\beta\partial\alpha}\Bigg\}, \end{aligned} \quad (15)$$

where $\alpha, \beta = \xi, \eta, \zeta$. In order to derive $\nabla\delta_{xi}$ through $\nabla\nabla^T\delta_{zi}$, we must determine the following four terms.

$$\frac{\partial \boldsymbol{u}_{oi}}{\partial\alpha}, \quad \frac{\partial \boldsymbol{u}_{fi}}{\partial\alpha}, \quad \frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\beta\partial\alpha}, \quad \frac{\partial^2 \boldsymbol{u}_{fi}}{\partial\beta\partial\alpha}.$$

Since these terms appear due to contact point movement, we derive them by using kinematics of rolling contact in the following section.

3.3. Kinematics of Rolling Contact

The contact is called rolling contact if one object rolls on another object without sliding and breaking the contact. The relation between the object and the finger displacement can be obtained by solving these equations.

Taking the time derivative of Eq. (9) and rearranging, we have

$$R_o\frac{\partial \boldsymbol{c}_{oi}}{\partial \boldsymbol{u}_{oi}}\dot{\boldsymbol{u}}_{oi} + \rho_{fi}R_{fi}\frac{\partial \boldsymbol{n}_{fi}}{\partial \boldsymbol{u}_{fi}}\dot{\boldsymbol{u}}_{fi} = 0. \quad (16)$$

Since finger makes contact with the object, the following equation is satisfied.

$$R_o\boldsymbol{n}_{oi} = -R_{fi}\boldsymbol{n}_{fi}. \quad (17)$$

Taking the time derivative of Eq. (17) yields

$$R_o\frac{\partial \boldsymbol{n}_{oi}}{\partial \boldsymbol{u}_{oi}}\dot{\boldsymbol{u}}_{oi} + R_{fi}\frac{\partial \boldsymbol{n}_{fi}}{\partial \boldsymbol{u}_{fi}}\dot{\boldsymbol{u}}_{fi} = -\omega_o \times R_o\boldsymbol{n}_{oi}. \quad (18)$$

ω_o is angular velocity of the object frame with respect to the base frame and is related to ξ as follows.

$$\omega_o = \begin{bmatrix} \cos\eta\cos\zeta & -\sin\zeta & 0 \\ \cos\eta\sin\zeta & \cos\zeta & 0 \\ -\sin\eta & 0 & 1 \end{bmatrix}\dot{\xi}. \quad (19)$$

After manipulating Eqs. (16) and (18), we have

$$\frac{\partial \boldsymbol{a}_{oi}}{\partial \boldsymbol{u}_{oi}}\dot{\boldsymbol{u}}_{oi} = \rho_{fi}\boldsymbol{n}_{oi} \times (W\dot{\xi}), \quad (20)$$

where $\boldsymbol{a}_{oi} = \rho_{fi}\boldsymbol{n}_{oi} - \boldsymbol{c}_{oi}$ and $W\dot{\xi} = R_o{}^T\omega_o$. Considering initial conditions, Eq. (20) becomes

$$\begin{aligned} & \begin{bmatrix} 0 & (\rho_{fi}+\rho_o)\mathrm{s}_{\theta i} \\ 0 & -(\rho_{fi}+\rho_o)\mathrm{c}_{\theta i} \\ -\rho_o/\mathrm{c}_{\phi i}{}^2 & 0 \end{bmatrix}\dot{\boldsymbol{u}}_{oi}|_{(0)} \\ & = \rho_{fi}\begin{bmatrix} 0 & 0 & -\mathrm{s}_{\theta i} \\ 0 & 0 & \mathrm{c}_{\theta i} \\ \mathrm{s}_{\theta i} & -\mathrm{c}_{\theta i} & 0 \end{bmatrix}\dot{\xi}, \end{aligned} \quad (21)$$

where $\mathrm{s}_\theta = \sin\theta$ and $\mathrm{c}_\theta = \cos\theta$. $\dot{\boldsymbol{u}}_{oi}$ is expressed by

$$\dot{\boldsymbol{u}}_{oi} = \frac{\partial \boldsymbol{u}_{oi}}{\partial\xi}\dot{\xi} + \frac{\partial \boldsymbol{u}_{oi}}{\partial\eta}\dot{\eta} + \frac{\partial \boldsymbol{u}_{oi}}{\partial\zeta}\dot{\zeta}. \quad (22)$$

Equating the same coefficient terms from Eqs. (21) and (22), we have

$$\begin{aligned} \frac{\partial \boldsymbol{u}_{oi}}{\partial\xi}|_{(0)} &= -\frac{\rho_{fi}}{\rho_o}\mathrm{c}_{\phi i}{}^2\mathrm{s}_{\theta i}\begin{bmatrix}1\\0\end{bmatrix}, \\ \frac{\partial \boldsymbol{u}_{oi}}{\partial\eta}|_{(0)} &= \frac{\rho_{fi}}{\rho_o}\mathrm{c}_{\phi i}{}^2\mathrm{c}_{\theta i}\begin{bmatrix}1\\0\end{bmatrix}, \\ \frac{\partial \boldsymbol{u}_{oi}}{\partial\zeta}|_{(0)} &= -\frac{\rho_{fi}}{\rho_{fi}+\rho_o}\begin{bmatrix}0\\1\end{bmatrix}. \end{aligned} \quad (23)$$

Taking the time derivative of Eq. (20), we have

$$\begin{aligned} \frac{\partial \boldsymbol{a}_{oi}}{\partial \boldsymbol{u}_{oi}}\ddot{\boldsymbol{u}}_{oi} = & -\left(\frac{\partial}{\partial \boldsymbol{u}_{oi}}\frac{\partial \boldsymbol{a}_{oi}}{\partial \boldsymbol{u}_{oi}}\dot{\boldsymbol{u}}_{oi} + \frac{\partial}{\partial \boldsymbol{v}_{oi}}\frac{\partial \boldsymbol{a}_{oi}}{\partial \boldsymbol{u}_{oi}}\dot{\boldsymbol{v}}_{oi}\right)\dot{\boldsymbol{u}}_{oi} \\ & + \rho_{fi}\left(\frac{\partial \boldsymbol{n}_{oi}}{\partial \boldsymbol{u}_{oi}}\dot{\boldsymbol{u}}_{oi}\right) \times (W\dot{\xi}) \\ & + \rho_{fi}\boldsymbol{n}_{oi} \times \left(\frac{\partial W}{\partial\xi}\dot{\xi} + \frac{\partial W}{\partial\eta}\dot{\eta} + \frac{\partial W}{\partial\zeta}\dot{\zeta}\right)\dot{\xi}. \end{aligned} \quad (24)$$

$\ddot{\boldsymbol{u}}_{oi}$ is represented as

$$\begin{aligned} \ddot{\boldsymbol{u}}_{oi} = & \frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\xi^2}\dot{\xi}^2 + 2\frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\xi\partial\eta}\dot{\xi}\dot{\eta} + 2\frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\xi\partial\zeta}\dot{\xi}\dot{\zeta} \\ & + \frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\eta^2}\dot{\eta}^2 + 2\frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\eta\partial\zeta}\dot{\eta}\dot{\zeta} + \frac{\partial^2 \boldsymbol{u}_{oi}}{\partial\zeta^2}\dot{\zeta}^2. \end{aligned} \quad (25)$$

3.4. Stiffness Matrix

Substituting all the values into Eq. (4), we have the

stiffness matrix for the cylindrical object by using moment equilibrium equations

$$H|_{(0)} = \sum_{i=1}^{n}\left\{ k_{xi}\begin{bmatrix} \boldsymbol{r}_{xi} \\ \boldsymbol{c}_{oi}\times\boldsymbol{r}_{xi}\end{bmatrix}\begin{bmatrix} \boldsymbol{r}_{xi} \\ \boldsymbol{c}_{oi}\times\boldsymbol{r}_{xi}\end{bmatrix}^T + k_{yi}\begin{bmatrix} \boldsymbol{r}_{yi} \\ \boldsymbol{c}_{oi}\times\boldsymbol{r}_{yi}\end{bmatrix}\begin{bmatrix} \boldsymbol{r}_{yi} \\ \boldsymbol{c}_{oi}\times\boldsymbol{r}_{yi}\end{bmatrix}^T + k_{zi}\begin{bmatrix} \boldsymbol{r}_{zi} \\ \boldsymbol{c}_{oi}\times\boldsymbol{r}_{zi}\end{bmatrix}\begin{bmatrix} \boldsymbol{r}_{zi} \\ \boldsymbol{c}_{oi}\times\boldsymbol{r}_{zi}\end{bmatrix}^T \right.$$

$$+ f_{xi}\begin{bmatrix} 0_{3\times3} & 0_{3\times3} & \\ & (\rho_{fi}-\rho_o)s_{\theta i}{}^2 & -(\rho_{fi}-\rho_o)s_{\theta i}c_{\theta i} \quad 0 \\ 0_{3\times3} & & (\rho_{fi}-\rho_o)c_{\theta i}{}^2 \quad 0 \\ & sym. & \quad \frac{-\rho_o{}^2}{\rho_{fi}+\rho_o} \end{bmatrix}$$

$$+ f_{yi}\begin{bmatrix} 0_{3\times3} & 0_{3\times3} & \\ & (\rho_{fi}-\rho_o)s_{\theta i}c_{\theta i} & -\frac{1}{2}(\rho_{fi}-\rho_o)c_{2\theta i} \quad 0 \\ 0_{3\times3} & & -(\rho_{fi}-\rho_o)s_{\theta i}c_{\theta i} \quad 0 \\ & sym. & \quad 0 \end{bmatrix}$$

$$\left. + f_{zi}\begin{bmatrix} 0_{3\times3} & 0_{3\times3} & \\ & -\rho_o t_{\phi i} \quad 0 & -\frac{(\rho_{fi}+2\rho_o)\rho_o}{2(\rho_{fi}-\rho_o)}c_{\theta i} \\ 0_{3\times3} & & -\rho_o t_{\phi i} \quad -\frac{(\rho_{fi}+2\rho_o)\rho_o}{2(\rho_{fi}-\rho_o)}s_{\theta i} \\ & sym. & \quad 0 \end{bmatrix} \right\}, \tag{26}$$

where $t_\theta = \tan\theta$. In Eq. (26), the first three terms are due to stiffness of the three virtual springs respectively and remaining three terms imply rotational stiffness which are influenced by curvatures, initial contact forces and contact position. It is shown that the stiffness matrix is a function of above parameters.

4. Stiffness Conditions

In this section, we derive the spring stiffness conditions from positive definiteness of stiffness matrix. The cylindrical object is grasped by 3 fingers. We assume the values of following parameters are

$$\phi_i = 0,\ \theta_i = 2\pi(i-1)/3,$$
$$k_{xi} = k_x,\ k_{yi} = k_y,\ k_{zi} = k_z,$$
$$f_{xi} = f_x,\ f_{yi} = f_{zi} = 0,\ \rho_{fi} = \rho_f,\ i = 1,2,3.$$

Then we have the stiffness matrix in diagonal form.

$$H|_{(0)} = diag[k_{xx}, k_{yy}, k_{zz}, k_{\xi\xi}, k_{\eta\eta}, k_{\zeta\zeta}]. \tag{27}$$

Diagonal elements are as follows.

$$k_{xx} = k_{yy} = 3(k_x + k_y)/2,\ k_{zz} = 3k_z,$$

$$k_{\xi\xi} = k_{\eta\eta} = \frac{3}{2}\rho_o{}^2\left(k_z - \frac{\rho_o-\rho_f}{\rho_o{}^2}f_x\right),$$

$$k_{\zeta\zeta} = 3\rho_o{}^2\left(k_y - \frac{f_x}{\rho_f+\rho_o}\right). \tag{28}$$

Using the above equations, we can provide the condition of spring stiffness.

$$k_x > 0,\ k_y > f_x/(\rho_f+\rho_o),$$
$$k_z \begin{cases} > 0 & for\ \rho_f \geq \rho_o. \\ > f_x(\rho_o-\rho_f)/\rho_o{}^2 & for\ \rho_f < \rho_o. \end{cases}$$

We can stabilize the grasp system easily, if we set the large values of ρ_f and ρ_o and the small value of f_x.

5. Conclusion

We have discussed the grasp stability of 3-D object by considering the curvature of both hand and object at contact points with rolling contact using potential energy approach. From the kinematics of rolling contact, we derived the velocity equations and the acceleration equations. In stiffness matrix, the first three terms are due to stiffness of the three virtual springs respectively as well as remaining terms imply rotational stiffness which are influenced by curvatures, initial contact forces and contact position. We have also established the stiffness condition for stabilization of grasp by using examples. The advantage of this paper is that we can assign the values of spring stiffness and make the stable grasp.

References

[1] H. Hanafusa and H. Asada: "Stable prehension by a robot hand with elastic fingers," Proc. of 7th Int. Symp. on Industrial Robots, pp.361-368, 1977.

[2] V. D. Nguyen, "The Synthesis of Stable Grasps in the Plane," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 884-889, 1986.

[3] M. Kaneko, N. Imamura, K. Yokoi, and T. Tanie, "Stiffness model based grasp stability analysis of multifingered hand with friction," J. Rob. Soc. Jpn., Vol. 7, No. 3. pp. 161-171, 1989 (In Japanese).

[4] N. Mimura and Y. Funahashi, "Grasp Stability analysis of three fingered hand in three dimensions," Trans. Jpn. Soc. Mech. Eng, Vol. 57, No. 537, Series C, pp. 1631-1638, 1991 (in Japanese).

[5] Y. Funahashi, T. Yamada, M. Tate, and Y. Suzuki, "Grasps stability analysis considering the curvatures at contact points", Proc. of IEEE Int. Conf. on Robotics and Automation, pp.3040-3046, 1996.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Algorithm for Identification of Contact Condition from Contaminated Data of Contact Moment

Tetsuya Mouri, Takayoshi Yamada,
and Yasuyuki Funahashi
Department of Mechanical Engineering
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466, Japan
E-mail: mouri, yamat@eine.mech.nitech.ac.jp

Nobuharu Mimura

Department of Information Engineering
Niigata University
Ninomachi, Igarashi, Niigata 950-21, Japan

Abstract

When a grasped object is in contact with external environment, it is necessary to perform the assembly tasks with identification of contact conditions. This paper treats the case when sensing data are contaminated with noise. We propose an efficient and an analytical algorithm for identifying unknown contact conditions using an active force sensing method. The algorithm can identify soft finger contact type, line contact type, and plane contact type in addition to point contact type. These contact types are characterized by a variance of contact moment. A contact position is estimated by a least-squares method. Contact moment is estimated from noisy observations and its eigenvalues are analyzed. We clarify the relation between the contact types and the eigenvalues of estimated contact moment.
Key Words: robot hand, unknown contact condition, contaminated data, contact point, parameter identification.

1 Introduction

In order to automate industrial production process, it is required that robots can perform complex tasks such as assembly. In assembly tasks, contact takes place between a grasped object and the external environments. It is necessary for industrial assembly tasks to be able to identify and control the contact conditions such as contact position, contact force, and contact type. Human can easily identify the current contact condition by using a sense of forces and accomplish the task. For example, by changing contact condition, ex., from point contact to line contact or from line contact to plane contact, one can make a fine positioning and assembling task. Therefore, methods for identification of the contact conditions by using force sensors have been proposed.

Nagata et al. [1], Kitagaki et al. [2], Mimura et al. [3], and Funahashi et al. [4] proposed methods for identifying the contact position of the unknown object by using active sensing. In these methods, however, it was assumed that the contact condition is of point contact type with friction.

Mimura et al. [5] discussed identification of contact condition including soft finger contact type, line contact type and plane contact type in addition to the point contact type, and showed the possibility of identifying contact position of unknown grasped object. Also, contact condition was formulated by considering that there exist some directions in which contact moment cannot be generated. But, since the equation of contact condition was nonlinear, unknown contact parameters were identified by a numerical calculation. Mimura et al. [6] expressed in an explicit form the constraint condition of contact moment. Moreover, the number of unknown parameters were reduced by subdividing the constraint condition. Also, the nonlinear problem of identifying the contact conditions was transformed to linear one. However, those papers treated the case when the data were noise free.

This paper treats the case when the data are contaminated with noise. Contact condition is identified by using sensing data collected by a 6-axes force sensor. First, point, soft finger, line, and plane contact types are characterized by a variance and an average of contact moment. Secondly, we consider the identification of contact position by the data contaminated with noise. Thirdly, we propose a method of identifying contact moment and analyze the relation between the eigenvalues of estimated contact moment and the contact types.

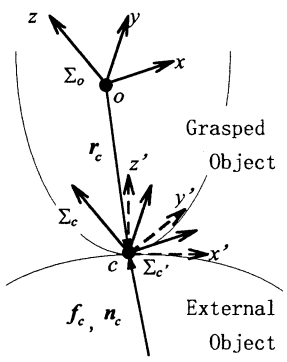


Fig. 1: Interaction between grasped object and external environment

2 Basic conditions

2.1 Symbols

We define the following symbols, as shown in Fig. 1.

c : contact point
o : origin of a sensor mounted on a robot hand
Σ_o : sensor coordinate frame fixed at o
Σ_c, $\Sigma_{c'}$: contact coordinate frame fixed at c
$\boldsymbol{f}_o$, $\boldsymbol{n}_o$: measured force and moment in Σ_o
$\boldsymbol{f}_c$, $\boldsymbol{n}_c$: contact force and moment in Σ_c
$\boldsymbol{r}_c$: position vector of c in Σ_o

2.2 Assumptions

We make the following assumptions for simplicity of discussions.

(A1) The object is firmly grasped by the robot hand and can be manipulated freely by the robot manipulator.

(A2) The force $\boldsymbol{f}_o$, the moment $\boldsymbol{n}_o$, and the position o are measurable and controllable.

(A3) No slip occurs at the contact position c.

(A4) The grasped object is in contact with external environment through point, soft finger, line, plane contact type with friction.

(A5) The number of times of active sensing motion is large enough.

(A6) The average of component of noise $\boldsymbol{\epsilon} = [\ \epsilon_x \quad \epsilon_y \quad \epsilon_z\ ]^T$ are zero. And the variance of them are equal.

(A7) The average of component of contact moment $\boldsymbol{n}_c$ are zero. And the variance of them are equal.

Assumption (A6) is means by

$$E[\epsilon_x] = E[\epsilon_y] = E[\epsilon_z] = 0, \tag{1}$$

$$V[\epsilon_x] = V[\epsilon_y] = V[\epsilon_z] = \gamma^2, \tag{2}$$

where $E[\bullet]$ and $V[\bullet]$ denote an average and a variance, respectively.

2.3 Constraint of contact moment

When we consider four contact types in Fig.2, the moment $\boldsymbol{n}_c$ may be zero in some direction. Namely, there exist some directions in which moment component cannot be generated. For example, when contact is of point contact type, $\boldsymbol{n}_c$ is always equal to zero. When it is of line contact type, the component of $\boldsymbol{n}_c$ in the direction rotating around the contact line is equal to zero. These characteristics can be stated through some appropriate coordinate frame $\Sigma_{c'}$ in Fig.2. The direction of z' axis is normal to the contact plane. In case of line contact type, moreover, the contact line should be taken as x' axis. The following components of contact moment $\boldsymbol{n}'_c = [\ n'_{cx} \quad n'_{cy} \quad n'_{cz}\ ]^T$ are equal to zero in $\Sigma_{c'}$.

$$\left.\begin{array}{ll} \text{Point contact} & : n'_{cx} = n'_{cy} = n'_{cz} = 0 \\ \text{Soft finger contact} & : n'_{cx} = n'_{cy} = 0 \\ \text{Line contact} & : n'_{cx} = 0 \\ \text{Plane contact} & : \text{no constraint} \end{array}\right\}. \tag{3}$$

$\boldsymbol{n}_c$ is expressed by $\boldsymbol{n}'_c$ by the following equation.

$$\boldsymbol{n}_c = R\boldsymbol{n}'_c, \tag{4}$$

where

$$R = [\ \boldsymbol{r}_x \quad \boldsymbol{r}_y \quad \boldsymbol{r}_z\ ] \in R^{3\times 3} \tag{5}$$

is orientation of $\Sigma_{c'}$ with respect to Σ_c.

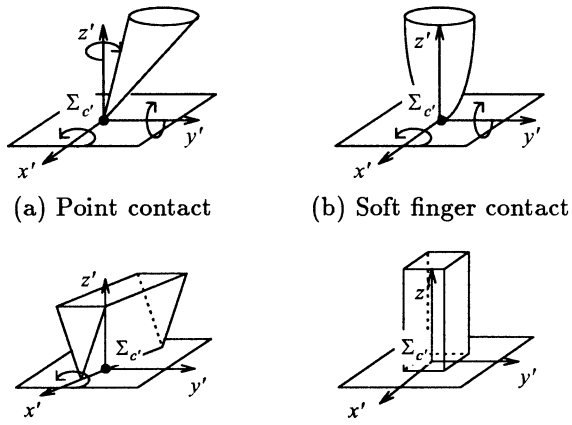
From Assumption (A7), each contact type is characterized by a variance of contact moment in Table 1.

3 Identification of contact condition

We will identify contact position and contact types from the data contaminated with noise. The identification method is established by using Eq. (3) and Table 1.

Table 1: A variance of contact moment

	$V[n'_{cx}]$	$V[n'_{cy}]$	$V[n'_{cz}]$
Point contact	0	0	0
Soft finger contact	0	0	σ^2
Line contact	0	σ^2	σ^2
Plane contact	σ^2	σ^2	σ^2



(a) Point contact (b) Soft finger contact

(c) Line contact (d) Plane contact

Fig. 2: contact types

3.1 Identification of contact point

The equilibrium equations of force and moment are represented by

$$\boldsymbol{f}_c = \boldsymbol{f}_o, \tag{6}$$

$$\boldsymbol{r}_c \times \boldsymbol{f}_c + \boldsymbol{n}_c = \boldsymbol{n}_o. \tag{7}$$

Substituting Eq. (6) into (7) yields

$$\boldsymbol{r}_c \times \boldsymbol{f}_o + \boldsymbol{n}_c = \boldsymbol{n}_o. \tag{8}$$

In Eq. (8), the measurable parameters are $\boldsymbol{f}_o$ and $\boldsymbol{n}_o$, while unmeasurable ones are $\boldsymbol{n}_c$ and $\boldsymbol{r}_c$ from assumption (A2). In order to make it clear, we use following notations.

$$A = -[\boldsymbol{f}_o\times], \quad \boldsymbol{b} = \boldsymbol{n}_o, \quad \boldsymbol{x} = \boldsymbol{r}_c. \tag{9}$$

From k times of active sensing motion, we have

$$A_i\boldsymbol{x} + \boldsymbol{n}_{ci} = \boldsymbol{b}_i - \boldsymbol{\epsilon}_i, \quad i = 1, 2, \cdots, k, \tag{10}$$

where the subscript i implies i-th measured value. In Eq.(10), unknown parameters are contact position $\boldsymbol{x}$, contact moment $\boldsymbol{n}_{ci}$, and noise $\boldsymbol{\epsilon}_i$. Here, contact moment with noise is represented by

$$\boldsymbol{\eta}_i = \boldsymbol{n}_{ci} + \boldsymbol{\epsilon}_i. \tag{11}$$

Eq. (10) becomes

$$A_i\boldsymbol{x} + \boldsymbol{\eta}_i = \boldsymbol{b}_i, \quad i = 1, 2, \cdots, k. \tag{12}$$

First, we identify $\boldsymbol{x}$ by minimizing a performance index of

$$J = \frac{1}{k}\sum_{i=1}^{k} \|\boldsymbol{\eta}_i\|^2 = \frac{1}{k}\sum_{i=1}^{k} \|A_i\boldsymbol{x} - \boldsymbol{b}_i\|^2 . \tag{13}$$

The estimate is given as

$$\hat{\boldsymbol{x}}(k) = \left(\sum_{i=1}^{k} A_i^T A_i\right)^{-1} \sum_{i=1}^{k} A_i^T \boldsymbol{b}_i. \tag{14}$$

From Assumption (A2), an average of component of force $\boldsymbol{f}_{oi}$ is controlled to be equal to zero. Therefore, the estimate $\hat{\boldsymbol{x}}(k)$ can be made to be asymptotically unbiased.

$$\lim_{k\to\infty} \hat{\boldsymbol{x}}(k) = \boldsymbol{x}^*, \tag{15}$$

where $\boldsymbol{x}^*$ is true value of contact position.

3.2 Identification of contact type

This section analyzes the relation between the contact types and the eigenvalues of the estimated contact moment. The estimate, $\hat{\boldsymbol{\eta}}_i$, of contact moment is

$$\hat{\boldsymbol{\eta}}_i(k) = \boldsymbol{b}_i - A_i\hat{\boldsymbol{x}}(k). \tag{16}$$

If it happens that $\hat{\boldsymbol{\eta}}_i = \boldsymbol{n}^*_{ci}$, a contact type can be detected by examining the rank of the matrix:

$$M = \left[\ \hat{\boldsymbol{\eta}}_1 \ \cdots \ \hat{\boldsymbol{\eta}}_k\ \right]. \tag{17}$$

The relation between the contact types and the rank of the matrix M is described by

$$\mathrm{rank}M = \begin{cases} 0 & \text{for a point contact type} \\ 1 & \text{for a soft finger contact type} \\ 2 & \text{for a line contact type} \\ 3 & \text{for a plane contact type} \end{cases}. \tag{18}$$

In general, however, $\hat{\boldsymbol{\eta}}_i \neq \boldsymbol{n}^*_{ci}$. So we calculate a following matrix and analyze its eigenvalues.

$$N = \lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \hat{\boldsymbol{\eta}}_i(k)\hat{\boldsymbol{\eta}}_i(k)^T \in R^{3\times 3}. \tag{19}$$

From Assumptions (A6) and (A7), the matrix N is expressed by

$$N = \lim_{k\to\infty} \frac{1}{k}\left(\boldsymbol{r}_x\boldsymbol{r}_x^T\sum_{i=1}^{k} n'^{*2}_{cxi} + \boldsymbol{r}_y\boldsymbol{r}_y^T\sum_{i=1}^{k} n'^{*2}_{cyi} + \boldsymbol{r}_z\boldsymbol{r}_z^T\sum_{i=1}^{k} n'^{*2}_{czi}\right) + \gamma^2 I_3, \tag{20}$$

where I_3 implies a 3×3 identity matrix, and $\boldsymbol{n}'^*_{ci}$ is constrained by the contact types in Table 1.

We denote the eigenvalues of the matrix N by λ_1 , λ_2 , and λ_3 , where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$. From Eq. (3)

and Table 1, the contact type can be determined from these eigenvalues in the following way.

In case of point contact type, the eigenvalues are

$$\lambda_1 = \lambda_2 = \lambda_3 = \gamma^2. \tag{21}$$

In case of soft finger contact type, we have

$$\lambda_1 = \sigma^2 + \gamma^2, \quad \lambda_2 = \lambda_3 = \gamma^2. \tag{22}$$

In case of line contact type, we have

$$\lambda_1 = \lambda_2 = \sigma^2 + \gamma^2, \quad \lambda_3 = \gamma^2. \tag{23}$$

In case of plane contact type, we have

$$\lambda_1 = \lambda_2 = \lambda_3 = \sigma^2 + \gamma^2. \tag{24}$$

Eqs. (21), (22), (23), and (24) are illustrated in Fig. 3. From the above, if a threshold is set between γ^2 and $\sigma^2 + \gamma^2$, we can judge which contact type happens.

The threshold is denoted by θ which satisfies

$$\gamma^2 < \theta < \sigma^2 + \gamma^2. \tag{25}$$

If the eigenvalues λ_1, λ_2, λ_3 satisfy

$$\theta > \lambda_1 = \lambda_2 = \lambda_3, \tag{26}$$

the contact is identified as point contact type. Similarly, the case

$$\lambda_1 > \theta > \lambda_2 = \lambda_3 \tag{27}$$

is identified as soft finger contact type. The case

$$\lambda_1 = \lambda_2 > \theta > \lambda_3 \tag{28}$$

is identified as line contact type. The case

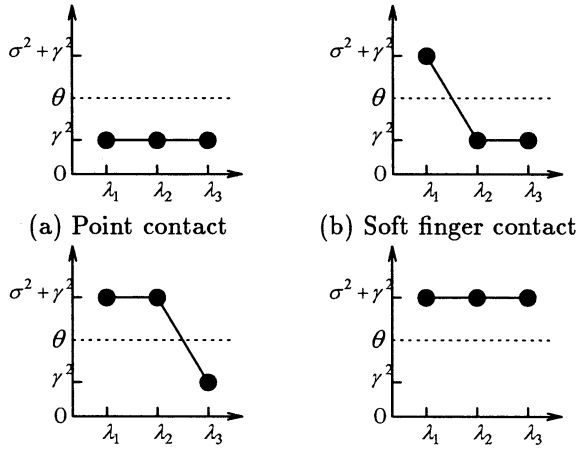
$$\lambda_1 = \lambda_2 = \lambda_3 > \theta \tag{29}$$

is identified as plane contact type.

In this way, all contact conditions can be identified.

4 Conclusions

We have investigated identification of contact conditions under the circumstances that the shape of grasped object is unknown and the data are contaminated with noise. First, point, soft finger, line, and plane contact types are characterized by a variance and an average of contact moment. Secondly, we consider the identification of contact position by the data contaminated with noise. Thirdly, we propose a method of identifying contact moment and analyze the relation between the eigenvalues of estimated contact moment and the contact types. The proposed method can be applied to actual tasks since it handles sensing data contaminated with noise.



(a) Point contact (b) Soft finger contact

(c) Line contact (d) Plane contact

Fig. 3: Relation between contact types and eginvalues

References

[1] Nagata, K., et al., "Pose Estimation of Grasped Object from Contact Force or Joint Data of Manipulator," *SICE*, Vol. 28, No. 7, pp. 783-789, 1992 (in Japanese).

[2] Kitagaki, K., et al., "A method for Detection of Contact Conditions by Force Sensing," *JSME Robotics and Mechatronics '91*, Vol. A, pp. 45-48, 1991 (in Japanese).

[3] Mimura, N. and Funahashi, Y., "Parameter Identification of a Grasp by a Planar Two-Fingered Robot Hand," *Trans. Jpn. Soc. Mech. Eng.*, Vol. 58, No. 554, Series C, pp. 175-181, 1992 (in Japanese).

[4] Funahashi, Y. and Tate, M., "Contact Point Identification by Active Sensing in Enveloping Grasp," *Trans. Jpn. Soc. Mech. Eng.*, Vol. 61, No. 589, Series C, pp. 3607-3613, 1995 (in Japanese).

[5] Mimura, N. and Funahashi, Y., "Parameter Identification of Contact Conditions by Active Force Sensing," *Proc. IEEE. Conf. Robotics and Automation*, pp. 2645-2650, 1994.

[6] Mimura, N., et al., "An Algorithm for Identification of Contact Conditions," *Trans. Jpn. Soc. Mech. Eng.*, Vol. 63, No. 610, Series C, pp. 2061-2068, 1997 (in Japanese).

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Stiffness Condition for Stabilization of Grasps with Interference Between Translational and Rotational Displacements

Yasuyuki FUNAHASHI*, Takayoshi YAMADA*,
Yasumitsu SUZUKI*, and Nobuharu MIMURA**

*: Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya 466, JAPAN

**: Niigata University,
Ninomachi, Igarashi, Niigata 950-21, JAPAN

Abstract

This paper establishes the general condition of the spring stiffness that makes a spatial grasp stable by allowing interference. Three orthogonal virtual springs, whose stiffness is accomplished by computer control, are fixed at the fingertip. The condition of the stiffness that makes the stiffness matrix positive definite is analyzed. And an assignment procedure of the spring stiffness is established. It is shown that the set of admissible values of the stiffness is greatly extended, and assignment error of the stiffness may be permitted. Therefore, the proposed method needs less energy to grasp. The proposed method is simple and is useful for practical use.

Key Words: grasp stability, potential energy approach, stiffness condition, assignement procedure, less energy.

1. Introduction

While a multifingered robot hand grasps an object, the stability is of essential issue. That is, the hand must not break the contact and not drop the object due to external disturbances. In order to evaluate stability of the grasp, potential energy approach has been proposed by many authors. These methods evaluate whether the grasp will return to the initial state after the disturbances disappeared.

Hanafusa et al. [1] analyzed the stability of frictionless grasp of a multifingered hand with elastic fingers. It was shown that the grasp is stable when the potential energy stored in the grasp is local minimum. Nguyen [2] proposed that stiffness of the real springs is realized by that of virtual springs since the stiffness can be controlled by computers. The stability of the grasp is evaluated by positive definiteness of the stiffness matrix. Kaneko et al. [3] discussed stability of frictional planar grasps and derived stiffness matrix of the grasp. Mimura et al. [4] extended the analysis to 3-D grasps. Funahashi et al. [5] considered the curvature of object and fingertip at the contact points.

Kaneko, Mimura, and Funahashi established spring stiffness that makes the grasp stable, since the stiffness can be assigned any value. In their analyses, however, it is assumed that there is no interference between translational and rotational stiffness. So stiffness of fingertip is restricted.

This paper will establish the stiffness condition of the springs, which makes the spatial grasp stable by allowing interference. Three virtual springs, which are along the normal and the tangential at the contact point, are fixed at the fingertip. First, we derive the stiffness matrix of the frictional grasp. The matrix is influenced by spring stiffness, initial grasping force, and so on. Secondly, the condition of the stiffness that makes the stiffness matrix positive definite is analyzed. Then an assignment procedure of the spring stiffness is proposed. Finally, we will describe the advantage of our method.

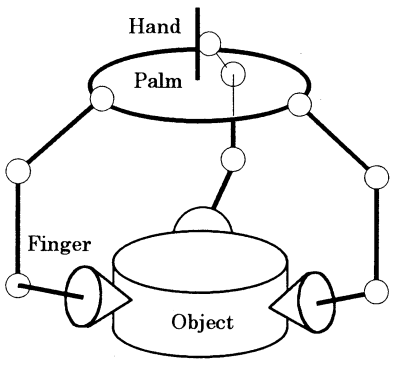


Fig.1 Grasping by a 3-fingered hand

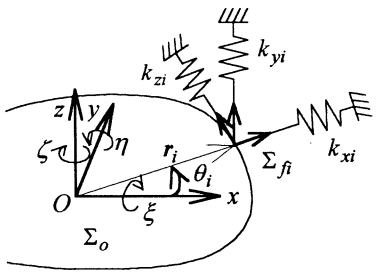


Fig. 2: Orthogonal virtual stiffness at contact point

2. Problem formulation

2.1. Notations

We define the following symbols.

Σ_o : object frame.

Σ_{fi} : i-th finger frame.

R_{fi} : orientation of Σ_{fi} with respect to Σ_o.

f_i : contact force.

r_i : position of contact point with respect to Σ_o.

θ_i : angle between x axis of Σ_o and r_i.

k_{xi}, k_{yi}, k_{zi} : stiffness of springs fixed at i-th fingertip.

$\varepsilon = [x^T \quad \xi^T]^T = [x \quad y \quad z \quad \xi \quad \eta \quad \zeta]^T$:

object displacement due to external distrubances

2.2. Assumptions

An object is grasped by a 3-fingered hand as shown in Fig. 1. Spatial virtual stiffness k_{xi}, k_{yi}, k_{zi} is fixed at the fingertip as shown in Fig. 2. This paper discusses the condition of the virtual springs k_{xi}, k_{yi}, k_{zi} which make the grasp stable. For simplicity of discussions, we make the following assumptions.

(A1) Contact between fingertips and the object is of point contact type with friction.

(A2) Contact position r_i and force f_i are known.

(A3) The grasp system is in equilibrium at the initial configuration.

(A4) 3-points of contact are not aligned.

(A5) The stiffness k_{xi} is assigned to be along the vector r_i, k_{yi} lies in the grasp plane, k_{zi} perpendicular to the grasp plane. k_{xi}, k_{yi} and k_{zi} are positive.

From Assumption (A3), initial grasping forces are internal forces. Σ_o is fixed at which f_i's intersect. The x-y plane of Σ_o lies in the grasp plane. An i-th fingertip coordinate Σ_{fi} is fixed along the springs k_{xi}, k_{yi} and k_{zi}.

2.3. Stability of the grasp

Compression of the springs is given by

$$[\delta_{xi} \quad \delta_{yi} \quad \delta_{zi}]^T = R_{fi}{}^T [x + \{R(\xi) - I_3\} r_i], \qquad (1)$$

where

$$R(\xi) = R_z(\zeta) R_y(\eta) R_x(\xi), \quad R_{fi} = R_z(\theta_i),$$
$$r_i = r_i [\cos\theta_i \quad \sin\theta_i \quad 0]^T,$$
$$f_i = -f_i [\cos\theta_i \quad \sin\theta_i \quad 0]^T, \qquad (2)$$

The potential energy stored in the grasp system is given by

$$U = \frac{1}{2} \sum_{i=1}^{n} \{k_{xi} (\delta_{xi} + \delta_{xoi})^2 + k_{yi} \delta_{yi}{}^2 + k_{zi} \delta_{zi}{}^2\}. \qquad (3)$$

Using Taylor series of Eq. (2) around $\varepsilon = 0$, we obtain

$$U = U(0) + \varepsilon^T \nabla U\big|_{(0)} + \tfrac{1}{2} \varepsilon^T H\big|_{(0)} \varepsilon + \cdots, \qquad (4)$$

where

$$\nabla = \frac{\partial}{\partial \varepsilon} \in \Re^6, \quad H = \nabla \nabla^T U \in \Re^{6\times 6}. \qquad (5)$$

The grasp is stable if and only if the energy U is local minimum at the initial condition $(\varepsilon = 0)$, that is, the following two conditions are satisfied.

(i) $\nabla U\big|_{(0)} = 0$.

(ii) $H\big|_{(0)}$ is positive definite.

From Assumption (A2), the condition (i) is always satisfied. Consequently, the grasp is stable if the condition (ii) is satisfied. The Hessian $H\big|_{(0)}$ is given by

$$H\big|_{(0)} = \begin{bmatrix} k_{xx} & k_{xy} & 0 & 0 & 0 & k_{x\zeta} \\ k_{yx} & k_{yy} & 0 & 0 & 0 & k_{y\zeta} \\ 0 & 0 & k_{zz} & k_{z\xi} & k_{z\eta} & 0 \\ 0 & 0 & k_{\xi z} & k_{\xi\xi} & k_{\xi\eta} & 0 \\ 0 & 0 & k_{\eta z} & k_{\eta\xi} & k_{\eta\eta} & 0 \\ k_{\zeta x} & k_{\zeta y} & 0 & 0 & 0 & k_{\zeta\zeta} \end{bmatrix},$$

where

$$k_{xx} = \sum k_{xi} c_i{}^2 + k_{yi} s_i{}^2, \quad k_{xy} = k_{yx} = \sum (k_{xi} + k_{yi}) c_i s_i,$$
$$k_{x\zeta} = k_{\zeta x} = -\sum k_{yi} r_i s_i, \quad k_{yy} = \sum k_{xi} s_i{}^2 + k_{yi} c_i{}^2$$
$$k_{y\zeta} = k_{\zeta y} = \sum k_{yi} r_i c_i, \quad k_{zz} = \sum k_{zi},$$
$$k_{z\xi} = k_{\xi z} = \sum k_{zi} r_i s_i, \quad k_{z\eta} = k_{\eta z} = -\sum k_{zi} r_i c_i,$$

$k_{\xi\xi} = \sum(k_{zi}r_i - f_i)r_i s_i^2$, $k_{\xi\eta} = \sum(-k_{zi}r_i + f_i)r_i c_i s_i$,

$k_{\eta\eta} = \sum(k_{zi}r_i - f_i)r_i c_i^2$, $k_{\zeta\zeta} = \sum k_{yi}r_i^2 - f_i r_i$,

where $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$.

The elements of $H|_{(0)}$ depend on the stiffness of k_{xi}, k_{yi}, and k_{zi}. In the following sections, we will derive the condition of stiffness k_{xi}, k_{yi}, and k_{zi}, which makes the matrices $H|_{(0)}$ positive definite.

3. Stiffness Condition

Exchanging rows and columns of $H|_{(0)}$ yields

$$K = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix},$$

where

$$K_1 := \begin{bmatrix} k_{xx} & k_{xy} & k_{x\zeta} \\ k_{yx} & k_{yy} & k_{y\zeta} \\ k_{\zeta x} & k_{\zeta y} & k_{\zeta\zeta} \end{bmatrix}, \quad K_2 := \begin{bmatrix} k_{zz} & k_{z\xi} & k_{z\eta} \\ k_{\xi z} & k_{\xi\xi} & k_{\xi\eta} \\ k_{\eta z} & k_{\eta\xi} & k_{\eta\eta} \end{bmatrix}$$

The following two conditions are equivalent.

(i) $H|_{(0)}$ is positive definite.

(ii) K_1 and K_2 are positive definite.

We will analyze the matrices K_1 and K_2.

3.1. Condition of k_{xi} and k_{yi}

The matrix K_1 can be rewritten as

$$K_1 = \begin{bmatrix} -s_1 & -s_2 & -s_3 \\ c_1 & c_2 & c_3 \\ r_1 & r_2 & r_3 \end{bmatrix}\begin{bmatrix} k_{y1} & 0 & 0 \\ 0 & k_{y2} & 0 \\ 0 & 0 & k_{y3} \end{bmatrix}\begin{bmatrix} -s_1 & c_1 & r_1 \\ -s_2 & c_2 & r_2 \\ -s_3 & c_3 & r_3 \end{bmatrix}$$
$$+ \begin{bmatrix} c_1 & c_2 & c_3 \\ s_1 & s_2 & s_3 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} k_{x1} & 0 & 0 \\ 0 & k_{x2} & 0 \\ 0 & 0 & k_{x3} \end{bmatrix}\begin{bmatrix} c_1 & s_1 & 0 \\ c_2 & s_2 & 0 \\ c_3 & s_3 & 0 \end{bmatrix}$$
$$- \sum_{i=1}^{3} r_i f_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}[0\ 0\ 1]$$
$$= AK_yA^T + BK_xB^T - dcc^T \qquad (6)$$

where

$K_x = diag[k_{x1}, k_{x2}, k_{x3}]$, $K_y = diag[k_{y1}, k_{y2}, k_{y3}]$,

$$A = \begin{bmatrix} -s_1 & -s_2 & -s_3 \\ c_1 & c_2 & c_3 \\ r_1 & r_2 & r_3 \end{bmatrix}, \quad B = \begin{bmatrix} c_1 & c_2 & c_3 \\ s_1 & s_2 & s_3 \\ 0 & 0 & 0 \end{bmatrix},$$

$$c = [0 \quad 0 \quad 1]^T, \quad d = \sum_{i=1}^{3} r_i f_i.$$

Since the matrix A is nonsingular from Assumption (A6), we define P as

$$P(k_{xi}, k_{yi}) := A^{-1}K_1A^{-T}$$
$$= K_y + \tilde{B}K_x\tilde{B}^T - d\tilde{c}\tilde{c}^T, \qquad (7)$$

where $\tilde{B} := A^{-1}B$ and $\tilde{c} := A^{-1}c$. Then we have the following condition.

$$K_1 > 0 \quad \Leftrightarrow \quad P > 0$$

Therefore, we derive the condition of k_{xi} and k_{yi} which makes $P(k_{xi}, k_{yi})$ positive definite. From

$$P > 0 \quad \Leftrightarrow \quad l^T Pl > 0 \text{ for } l = [l_1\ l_2\ l_3]^T \neq 0,$$

we investigate $l^T Pl$. Then we obtain

$$l^T Pl = k_{y1}l_1^2 + k_{y2}l_2^2 + k_{y3}l_3^2$$
$$+ k_{x1}\{(\tilde{B}^T l)_1\}^2 + k_{x2}\{(\tilde{B}^T l)_2\}^2 + k_{x3}\{(\tilde{B}^T l)_3\}^2$$
$$- d(l^T\tilde{c})^2 \qquad (8)$$

where $(\tilde{B}^T l)_i$ denotes the i-th element of the vector $\tilde{B}^T l$. Since, from Eq. (8), $l^T Pl$ is linear with respect to k_{xi} and k_{yi} and is non-negative, there exists lowest limit of k_{xi} and k_{yi}. The coefficient of k_{xi} will be zero for $l \in Ker\tilde{B}^T$. The matrix $P(k_{xi}, k_{yi})$ is not guaranteed to be positive definite by enlarging k_{xi}. However, k_{yi} is guaranteed. Consequently, the grasp is stable if stiffness of the springs is assigned as follows. First, k_{xi}'s are assigned any values, then we have

$$Q = \{q_{ij}\} := -\tilde{B}K_x\tilde{B}^T + d\tilde{c}\tilde{c}^T \qquad (9)$$

k_{y1} is assigned to be

$$k_{y1} > q_{11} \qquad (10)$$

Then k_{y2} is assigned to be

$$k_{y2} > \frac{q_{12}^2}{k_{y1} - q_{11}} + q_{22} \qquad (11)$$

Finally, k_{y3} is assigned to be

$$k_{y3} > \frac{q_{23}^2(k_{y1} - q_{11}) + q_{13}^2(k_{y2} - q_{22}) + 2q_{12}q_{13}q_{23}}{(k_{y1} - q_{11})(k_{y2} - q_{22}) - q_{12}^2}$$
$$+ q_{33} \qquad (12)$$

We pick up two of k_{y1}, k_{y2}, and k_{y3}, these are in inverse proportion to each other. Therefore, possible

region of k_{yi} is given by upper area of the bound shown in Fig. 3.

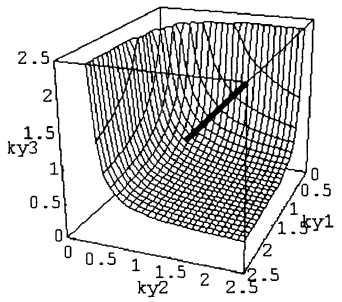


Fig. 3: possible set area of k_{yi}

Refs. [3]-[5] assumed

$$k_{x\zeta} = k_{\zeta x} = 0, \quad k_{y\zeta} = k_{\zeta y} = 0 .$$

These mean no interference between translational and rotational stiffness. So the stiffness k_{y1}, k_{y2}, and k_{y3} are limited to the segment illustrated in Fig. 3. However, our analysis shows that the grasp can be originally stabilized in the wide area of stiffness. This means that we can assign any values of stiffness from the set satisfying Eqs. (10) - (12) and assignment error of stiffness may be permitted when a multifingered robot hand is controlled.

3.2. Condition of k_{zi}

The matrix K_2 can be represented by

$$K_2 = \begin{bmatrix} 1 & 1 & 1 \\ r_1 s_1 & r_2 s_2 & r_3 s_3 \\ -r_1 c_1 & -r_2 c_2 & -r_3 c_3 \end{bmatrix} \begin{bmatrix} k_{z1} & 0 & 0 \\ 0 & k_{z2} & 0 \\ 0 & 0 & k_{z3} \end{bmatrix} \begin{bmatrix} 1 & r_1 s_1 & -r_1 c_1 \\ 1 & r_2 s_2 & -r_2 c_2 \\ 1 & r_3 s_3 & -r_3 c_3 \end{bmatrix}$$
$$- \sum_{i=1}^{3} \begin{bmatrix} 0 & 0 & 0 \\ 0 & f_i r_i s_i{}^2 & -f_i r_i c_i s_i \\ 0 & -f_i r_i c_i s_i & f_i r_i c_i{}^2 \end{bmatrix}$$
$$= A' K_z A'^T - C . \qquad (13)$$

Since, from Assumption (A4), the matrix A is nonsingular, we have the following condition.

$$K_2 > 0 \iff K_z - A'^{-1} C A'^{-T} > 0 .$$

The grasp can be stabilized, if k_{zi} are assigned to be

$$k_{z1} > c'_{11}, \quad k_{z2} > \frac{c'_{12}{}^2}{k_{z1} - c'_{11}} + c'_{22},$$

$$k_{z3} > \frac{c'_{23}{}^2 (k_{z1} - c'_{11}) + c'_{13}{}^2 (k_{z2} - c'_{22}) + 2 c'_{12} c'_{13} c'_{23}}{(k_{z1} - c'_{11})(k_{z2} - c'_{22}) - c'_{12}{}^2} + c'_{33},$$

where $C' = \{c'_{ij}\} := A'^{-1} C' A'^{-T}$.

Therefore all the values of stiffness are determined. Due to lack of the space, numerical examples are not described. From the examples, however, it is ensured that our method can stabilize the grasp more easily with less energy than the methods of Refs. [3]-[5].

4. Conclusion

We have investigated stabilization of the 3D grasp of 3 fingered with friction contact. Three orthogonal virtual stiffness are fixed to stabilize the grasp. We obtained the general solution for stabilization of grasp. The main results of this paper are as follows.

(1) We allow the interference between translational and rotational stiffness. Hence, admissible region of stiffness is greatly extended.

(2) We can assign the values of stiffness from the wide area. The assignment error may be permitted. Our method needs less energy to grasp.

Since the stiffness of the fingertip is accomplished by computer control, the stabilization method proposed in this paper is simple and is useful for the grasp by a multifingered hand.

References

[1] H. Hanafusa and H. Asada: Stable Prehension of Objects by the Robot Hand with Elastic Fingers, Proc. of Int. Symposium on Industrial Robots, pp. 361 – 368, 1977.

[2] V. D. Nguyen, The synthesis of Stable Grasps in the Plane, Proc. of 1986 IEEE Int. Conf. on Robotics and Automation, pp. 884 – 889, 1986.

[3] M. Kaneko, N. Imamura, K. Yokoi, and T. Tanie, Stiffness Model Based Grasp Stability Analysis of Multifingered Hand with Friction, Journal of Robotics Society of Japan, (In Japanese), Vol. 7, No. 3, June, pp. 49 – 59, 1989.

[4] N. Mimura and Y. Funahashi: Grasp Stability Analysis of Three-finger Hand in Three Dimensions, Trans. of Japan Soc. of Mech. Eng., (In Japanese), Vol. 57, No. 537, Series C, pp. 213 – 220, 1991.

[5] Y. Funahashi, T. Yamada, M. Tate, and Y. Suzuki: Proc. of 1996 IEEE Int. Conf. on Robotics and Automation, pp. 3040 – 3046, 1996.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Dynamic Control of Redundant Manipulators Using Artificial Potential Field Approach with Time Scaling

Yoshiyuki TANAKA, Toshio TSUJI and Makoto KANEKO
Faculty of Engineering, Hiroshima University
1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima, 739, Japan

Abstract

This paper proposes a new method for dynamic control of redundant manipulators via artificial potential field approach (APFA). The proposed method is based on the APFA with a combination of a time scale transformation and a time base generator which works as a time scale compressor and can control the dynamic behavior of the robot without any change of the form of the designed controller itself. The effectiveness of the proposed method is verified by computer simulations for a three-joint planar manipulator.

1 Introduction

In the Artificial Potential Field Approach (APFA) [1]-[4], the goal is represented by an artificial attractive potential field and the obstacles by corresponding repulsive fields, so that the trajectory to the target can be generated via a flow-line tracking process with consideration to the obstacle avoidance. This method is often used for the trajectory generation problem of robots because of its simplicity and lower computation than other methods using global information about the task space. However, little attention has been paid to control the dynamic behavior of the generated trajectories such as movement time from the initial position to the goal and velocity profile of the generated trajectory. Although one of the most crucial winning features of the APFA is real-time applicability, it is difficult to use the generated trajectory for the control of the robots in real time.

For the disadvantage of the APFA mentioned above, H. Hashimoto et al. [4] proposed a method using an electrostatic potential field and a sliding mode that can regulate the movement time but not the dynamic behavior of a robot. T. Tsuji et al. [5][6] proposed a method introducing the Time Base Generator (TBG) into the APFA which can regulate the movement time and also the velocity profile of the robot, but can not be applied to the dynamic control.

On the other hand, J.M.Hollerbach [7] developed the trajectory time-scaling method for the torque limited path following problem of the manipulator. The method can lead the end-effector to the goal along the given path by modifying the movement speed. M. Sampei and K. Furuta [8] showed that the stability of a system is preserved for any time scale transformation as long as the defined new time never goes backward against the actual time. More recently, Y. Tanaka et.al. [9] has developed the trajectory generation method for the dynamic control of robots based on the APFA with a combination of time scale transformation and TBG. Then, they applied it to the dynamic control of a holonomic mobile robot.

In this paper, we propose a new trajectory generation method for the dynamic control of redundant manipulators using Tanaka's method [9]. The redundant manipulator has a desirable feature that may lead to more dexterity and versatility of the robot motions, for instance, avoiding obstacles or singular configurations when performing a given task [10]-[12]. The present method can control the spatio-temporal trajectories of the end-effector with significant advantages of redundancy.

This paper is organized as follows: Section II formulates dynamics of a redundant manipulator. Section III points out the general problems of the APFA. Then, the new trajectory generation method based on the APFA is explained in detail in Section IV. Finally, the effectiveness of the proposed method is shown from computer simulations with the dynamic model of the redundant manipulator in Section V.

2 Dynamics of manipulators

The joint space motion equation of an n-degree-of-freedom manipulator whose end-effector is operating in the m-dimensional task space can be expressed as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \ , \qquad (1)$$

where $\boldsymbol{q} \in \Re^n$ is the joint angle vector, $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{n\times n}$ is the non-singular inertia matrix (hereafter denoted by $\boldsymbol{M}$), $\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \Re^n$ is the Coliolies and centrifugal force term, $\boldsymbol{g}(\boldsymbol{q}) \in \Re^n$ is the gravity term, and $\boldsymbol{\tau} \in \Re^n$ is the joint torque vector. On the other hand, the dynamics of the end-effector can be written in the operational space as [12]

$$\boldsymbol{M}_x(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{h}_x(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}_x(\boldsymbol{q}) = \boldsymbol{F}\,, \qquad (2)$$

where $\boldsymbol{x} \in \Re^m$ is the current end-effector position, $\boldsymbol{F} \in \Re^m$ is the end-effector force vector, $\boldsymbol{M}_x(\boldsymbol{q}) = (\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1} \in \Re^{m\times m}$ is the operational space kinetic energy matrix (hereafter denoted by $\boldsymbol{M}_x$), $\boldsymbol{J} \in \Re^{m\times n}$ is the Jacobian matrix, and also $\boldsymbol{h}_x(\boldsymbol{q},\dot{\boldsymbol{q}}) = \bar{\boldsymbol{J}}^T\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{M}_x\dot{\boldsymbol{J}}\dot{\boldsymbol{q}}$, $\boldsymbol{g}_x(\boldsymbol{q}) = \bar{\boldsymbol{J}}^T\boldsymbol{g}(\boldsymbol{q})$, $\bar{\boldsymbol{J}} = (\boldsymbol{M}_x\boldsymbol{J}\boldsymbol{M}^{-1})^T$.

When a manipulator possesses extra degree-of-freedom to execute a given task, i.e. $m < n$, the joint torque of redundant manipulators can be decomposed into two elements; the joint torque $\boldsymbol{\tau}_{effector} \in \Re^n$ to operate the end-effector, and the joint torque $\boldsymbol{\tau}_{joint} \in \Re^n$ to control the additional freedom of joint motion with redundancy of a manipulator. The force/torque relationship between the joint torque $\boldsymbol{\tau}_{effector}$ and the operational force $\boldsymbol{F}$ is given by

$$\boldsymbol{\tau}_{effector} = \boldsymbol{J}^T\boldsymbol{F}\,. \qquad (3)$$

On the other hand, the joint torque $\boldsymbol{\tau}_{joint}$ always satisfies the following condition [12] given by

$$\bar{\boldsymbol{J}}^T\boldsymbol{\tau}_{joint} = \boldsymbol{0}\,. \qquad (4)$$

This equation implies that the joint torque $\boldsymbol{\tau}_{joint}$ must lay in the null space associated with the matrix $\bar{\boldsymbol{J}}^T$ so as not to produce any acceleration at the end-effector. The general solution $\boldsymbol{\tau}_{joint}$ for this condition given by

$$\boldsymbol{\tau}_{joint} = \boldsymbol{G}\boldsymbol{\tau}^*\,, \qquad (5)$$

where $\boldsymbol{\tau}^*$ is an arbitrary n-dimensional vector, and $\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{J}^T\bar{\boldsymbol{J}}^T \in \Re^{n\times n}$. Consequently, the total joint torque $\boldsymbol{\tau}$ for a redundant manipulator can be recomposed of (3) and (5) as follows:

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_{effector} + \boldsymbol{\tau}_{joint} \\ &= \boldsymbol{J}^T\boldsymbol{F} + \boldsymbol{G}\boldsymbol{\tau}^*\,. \end{aligned} \qquad (6)$$

In this paper, we design the feedback control law $\boldsymbol{F}$ and $\boldsymbol{\tau}^*$, respectively. The total joint torque composed of those designed controllers allows a redundant manipulator to perform a given task by utilizing arm redundancy efficiently.

3 Artificial Potential Field Approach

In this section, we attempt to design the feedback control laws $\boldsymbol{F}$ in order to lead the end-effector to the target position and $\boldsymbol{\tau}^*$ in order to control the extra joint motion of redundant manipulators.

Here, we can define the potential function with quadratic form $V_{effector}$ to derive the feedback controller $\boldsymbol{F}$ as follows:

$$V_{effector} = \frac{1}{2}(\boldsymbol{x}^* - \boldsymbol{x})^T\boldsymbol{K}_1(\boldsymbol{x}^* - \boldsymbol{x}) + \frac{1}{2}\dot{\boldsymbol{x}}^T\boldsymbol{K}_2\dot{\boldsymbol{x}}\,, \qquad (7)$$

where $\boldsymbol{x}^*$ denotes the target position of the end-effector, and $\boldsymbol{K}_i$ =diag.$(k_1^i, k_2^i, \cdots, k_m^i)$ under $k_m^i > 0$ $(i = 1, 2)$. When we design the feedback control law $\boldsymbol{F}$ based on the potential function $V_{effector}$ as

$$\boldsymbol{F} = -\boldsymbol{M}_x\boldsymbol{K}_2^{-1}\{\boldsymbol{K}_1(\boldsymbol{x} - \boldsymbol{x}^*) + \dot{\boldsymbol{x}}\} + \boldsymbol{h}_x(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}_x(\boldsymbol{q})\,, \qquad (8)$$

the time-derivative of $V_{effector}$ yields

$$\dot{V}_{effector} = -||\dot{\boldsymbol{x}}||^2 \le 0\,, \qquad (9)$$

with the dynamic equation (2). $\dot{V}_{effector}$ is always non-increasing except at the equilibrium point. It follows that the end-effector can reach the target position by the joint torque $\boldsymbol{\tau}_{effector}$ equivalent to the derived control law $\boldsymbol{F}$ given in (8). For a redundant manipulator, however, the joints continue to move although the end-effector arrived at the target position since the designed controller $\boldsymbol{F}$ can not control the extra freedom of joint motion directly. For this problem in redundancy, we apply the null space on the force/torque transformation to control the internal motion.

Here, we define the potential function V_{joint} in order to design the feedback controller $\boldsymbol{\tau}^*$ as

$$V_{joint} = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}\dot{\boldsymbol{q}} + \kappa(t)Q(\boldsymbol{q})\,, \qquad (10)$$

where $\kappa(t)$ is a positive and non-increasing continuous function, i.e. $\dot{\kappa}(t) < 0$, and $Q(\boldsymbol{q})$ is a differentiable potential function. The first term on the right side of equation (10) is used in order to dampen the redundant joint motion when the end-effector arrives at the goal, and the second one is used to realize the desired posture of the manipulator $\boldsymbol{q}^*$ corresponding to the minimum/maximum of the potential function $Q(\boldsymbol{q})$. If we design the feedback control law $\boldsymbol{\tau}$ based on the potential function V_{joint} as

$$\boldsymbol{\tau} = -\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) - \kappa(t)\,\frac{\partial Q}{\partial \boldsymbol{q}}\,, \qquad (11)$$

the time-derivative of V_{joint} yields

$$\dot{V}_{joint} = -||\dot{\boldsymbol{q}}||^2 + \dot{\kappa}(t)\;Q(\boldsymbol{q}) \le 0\,, \qquad (12)$$

with the joint space motion equation (1) and $\dot{\kappa}(t) < 0$ in the actual time scale. Selecting the designed controller $\boldsymbol{\tau}$ (11) as $\boldsymbol{\tau}^*$, we can obtain the joint torque $\boldsymbol{\tau}_{joint}$ to control the internal motion without altering the generating trajectory of the end-effector.

With the total feedback control law $\boldsymbol{\tau}$ given in (6) composed of the designed controller $\boldsymbol{F}$ (8) and $\boldsymbol{\tau}^*$ (11) , the end-effector can be reached the target position and also the desired posture through an optimization procedure of the potential function $Q(\boldsymbol{q})$.

Moreover, substituting (8) into (2), we can derive the following linear damped system:

$$\ddot{\boldsymbol{x}} + \boldsymbol{K}_2^{-1}\dot{\boldsymbol{x}} + \boldsymbol{K}_2^{-1}\boldsymbol{K}_1^{-1}(\boldsymbol{x} - \boldsymbol{x}^*) = 0 \,. \qquad (13)$$

Obviously, the system in the operational space (2) is asymptotically stable to the equilibrium point $\boldsymbol{x}^*$ by the designed feedback controller $\boldsymbol{F}$ given in (8). Following the above discussion, we can conclude that it is impossible to regulate the convergence time and the dynamic behavior of the end-effector as hoped [5].

4 APFA with Time Scaling

Generally, the stability and dynamic property of systems has no change in any time scale that is a strictly monotone increasing function with respect to the actual time [8]. This indicates that we can design the feedback control law to converge the original system to the equilibrium point at finite time t_f as long as the asymptotic stabilizer for the system in the new time scale where infinite time corresponds to t_f in the actual time is found.

In this section, we present a detail of the proposed method based on the APFA combined with the time scale transformation.

4.1 Virtual time s and TBG

The relationship between actual time t and virtual time s is given by

$$\frac{ds}{dt} = a(t) \,, \qquad (14)$$

where the continuous function $a(t)$, called the time scale function [8], is defined as follows:

$$a(t) = -p\frac{\dot{\xi}}{\xi} \,, \qquad (15)$$

where p is a positive constant and $\xi(t)$ is a non-increasing function called the Time Base Generator (TBG) [5][6] generates a bell-shaped velocity profile satisfying $\xi(0) = 1$ and $\xi(t_f) = 0$ with the convergence time t_f. The dynamics of ξ is defined as follows:

$$\dot{\xi} = -\gamma(\xi(1-\xi))^{\beta}, \qquad (16)$$

where γ and β is a positive constant under $0 < \beta < 1.0$. The convergence time t_f can be calculated with the gamma function $\Gamma(\,\cdot\,)$ as

$$t_f = \int_0^{t_f} dt = \int_1^0 \frac{d\xi}{\dot{\xi}} = \frac{\Gamma^2(1-\beta)}{\gamma\Gamma(2-2\beta)}. \qquad (17)$$

From (14) and (15), the virtual time s can be represented with respect to ξ as follows:

$$s = \int_0^t a(t)\, dt = -p\ln\xi(t) \,. \qquad (18)$$

It is obvious that the virtual time s given in (18) never goes backward against the actual time t. We take this virtual time s as a new time scale in time scale transformation.

4.2 Time Scaling of the System

We can rewrite the two dynamic equations in the joint space (1) and in the operational space (2) into the following linear system with the state variable $\boldsymbol{Z} = (\boldsymbol{x}, \boldsymbol{q}, \dot{\boldsymbol{x}}, \dot{\boldsymbol{q}})^T$ as:

$$\frac{d}{dt}\boldsymbol{Z} = \begin{pmatrix} \mathbf{o} & \boldsymbol{I} \\ \mathbf{o} & \mathbf{o} \end{pmatrix}\boldsymbol{Z} + \begin{pmatrix} \mathbf{o} \\ \boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{F}_t \\ \boldsymbol{\tau}_t \end{pmatrix} \,, \qquad (19)$$

where $\mathbf{0} \in \Re^{(m+n)\times(m+n)}$ is the zero matrix and $\boldsymbol{I} \in \Re^{(m+n)\times(m+n)}$ is the unit matrix. Then

$$\boldsymbol{F}_t = \boldsymbol{M}_x^{-1}\left\{\boldsymbol{F} - (\boldsymbol{h}_x(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}_x(\boldsymbol{q}))\right\} \,, \qquad (20)$$

$$\boldsymbol{\tau}_t = \boldsymbol{M}^{-1}\left\{\boldsymbol{\tau} - (\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}))\right\} \,. \qquad (21)$$

The system given in (19) can be rewritten in the virtual time scale s as follows:

$$\frac{d}{ds}\boldsymbol{\Psi} = \begin{pmatrix} \mathbf{o} & \boldsymbol{I} \\ \mathbf{o} & \mathbf{o} \end{pmatrix}\boldsymbol{\Psi} + \begin{pmatrix} \mathbf{o} \\ \boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{F}_s \\ \boldsymbol{\tau}_s \end{pmatrix} \,, \qquad (22)$$

where

$$\boldsymbol{\Psi} = (\psi_1, \psi_2, \psi_3, \psi_4) = (\boldsymbol{x}, \boldsymbol{q}, \frac{\dot{\boldsymbol{x}}}{a(t)}, \frac{\dot{\boldsymbol{q}}}{a(t)})^T \,, \qquad (23)$$

$$\boldsymbol{F}_s = \frac{d}{dt}\left(\frac{1}{a(t)}\right)\dot{\boldsymbol{x}} + \frac{1}{a(t)^2}\boldsymbol{F}_t \,, \qquad (24)$$

$$\boldsymbol{\tau}_s = \frac{d}{dt}\left(\frac{1}{a(t)}\right)\dot{\boldsymbol{q}} + \frac{1}{a(t)^2}\boldsymbol{\tau}_t \,. \qquad (25)$$

As previously defined in the relationship between actual time and virtual time, stability of the new system given in (22) is the same as the original system in the actual time [8]. Hence, there exists a feedback control law to stabilize the new system asymptotically.

4.3 Design the feedback control law

In this subsection, we design the feedback control law with the APFA to stabilize the new system given in (22) in the virtual time scale.

We can define the potential function with quadratic form $V^{\psi}_{effector}$ for the control of the end-effector in the virtual time scale as follows:

$$V^{\psi}_{effector} = \frac{1}{2}(\boldsymbol{\psi}_1^* - \boldsymbol{\psi}_1)^T \boldsymbol{K}_1 (\boldsymbol{\psi}_1^* - \boldsymbol{\psi}_1) + \frac{1}{2}\boldsymbol{\psi}_3{}^T \boldsymbol{K}_2 \boldsymbol{\psi}_3 \,. \tag{26}$$

If we design the feedback controller $\boldsymbol{F}_s$ based on $V^{\psi}_{effector}$ as

$$\boldsymbol{F}_s = -\boldsymbol{K}_2{}^{-1} \left\{ \boldsymbol{K}_1 (\boldsymbol{\psi}_1 - \boldsymbol{\psi}_1^*) + \boldsymbol{\psi}_3 \right\} \,, \tag{27}$$

the derivative of $V^{\psi}_{effector}$ with respect to s yields

$$\frac{d}{ds} V^{\psi}_{effector} = -||\boldsymbol{\psi}_3|| \leq 0 \,. \tag{28}$$

By inverse transformation of time scale from the virtual time s to the actual time t for the controller $\boldsymbol{F}_s$ with (22) and (24), the controller $\boldsymbol{F}_t$ in the actual time is derived as

$$\boldsymbol{F}_t = -a^2(t)\boldsymbol{K}_2^{-1}\boldsymbol{K}_1(\boldsymbol{x} - \boldsymbol{x}^*) - \left\{ a(t)\boldsymbol{K}_2^{-1} - \frac{\dot{a}(t)}{a(t)} \right\} \dot{\boldsymbol{x}} \,. \tag{29}$$

From (20) and (29), we can obtain the feedback control law $\boldsymbol{F}^{\psi}$ for control of the dynamic behavior of the end-effector as follows:

$$\boldsymbol{F}^{\psi} = \boldsymbol{M}_x \boldsymbol{F}_t + \boldsymbol{h}_x(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}_x(\boldsymbol{q}) \,. \tag{30}$$

The end-effector can be controlled to the target position at the convergence time t_f by means of the joint torque $\boldsymbol{\tau}_{effector}$ equivalent to the feedback control law $\boldsymbol{F}^{\psi}$ given in (30).

On the other hand, we can define the potential function V^{ψ}_{joint} to derive the feedback controller $\boldsymbol{\tau}_s$ as

$$V^{\psi}_{joint} = \frac{1}{2}\boldsymbol{\psi}_4{}^T \boldsymbol{K}_3 \boldsymbol{\psi}_4 + \zeta(s) Q_s(\boldsymbol{\psi}_2) \,, \tag{31}$$

where $\boldsymbol{K}_3 = \mathrm{diag.}(k_1^3, k_2^3, \cdots, k_n^3)$ under $k_n^3 \geq 0$, $Q_s(\boldsymbol{\psi}_2)$ is the differentiable potential function in the virtual time scale, and $\zeta(s)$ is a positive non-increasing scalar function. The derivative of V^{ψ}_{joint} with respect to s yields

$$\frac{d}{ds} V^{\psi}_{joint} = \boldsymbol{\psi}_4{}^T \left\{ \boldsymbol{K}_3 \boldsymbol{\tau}_s + \zeta(s) \frac{\partial Q_s}{\partial \boldsymbol{\psi}_2} \right\} + \frac{d\zeta(s)}{ds} Q_s(\boldsymbol{\psi}_2) \,. \tag{32}$$

If we define the feedback controller $\boldsymbol{\tau}_s$ under consideration that the scalar function $\zeta(s)$ is non-increasing in the new time scale as

$$\boldsymbol{\tau}_s = -\boldsymbol{K}_3{}^{-1} \left\{ \boldsymbol{\psi}_4 + \zeta(s) \frac{\partial Q_s}{\partial \boldsymbol{\psi}_2} \right\} \,, \tag{33}$$

Equation (32) can be calculated as

$$\frac{d}{ds} V^{\psi}_{joint} = -||\boldsymbol{\psi}_4||^2 + \frac{d\zeta}{ds}(s) Q_s(\boldsymbol{\psi}_2) \leq 0 \,. \tag{34}$$

This indicates that the potential function V^{ψ}_{joint} is stabilized to the equilibrium point by means of the feedback controller $\boldsymbol{\tau}_s$ in the virtual time scale.

Here, we define the non-increasing function $\zeta(s)$ as

$$\zeta(s) = \alpha e^{-\frac{2s}{p}} \,. \tag{35}$$

Through inverse time-scale transformation from the virtual time to the actual time for the controller $\boldsymbol{\tau}_s$ with (22) and (25), the feedback control law $\boldsymbol{\tau}_t$ in the actual time is derived as

$$\boldsymbol{\tau}_t = -\left\{ a(t)\boldsymbol{K}_3^{-1} - \frac{\dot{a}(t)}{a(t)} \right\} \dot{\boldsymbol{q}} - \alpha \xi^2(t) a^2(t) \boldsymbol{K}_3^{-1} \frac{\partial Q}{\partial \boldsymbol{q}} \,, \tag{36}$$

where α is a positive constant. From (21) and (36), we can derive the following feedback controller $\boldsymbol{\tau}^{\psi}$ as

$$\boldsymbol{\tau}^{\psi} = \boldsymbol{M}\boldsymbol{\tau}_t + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) \,. \tag{37}$$

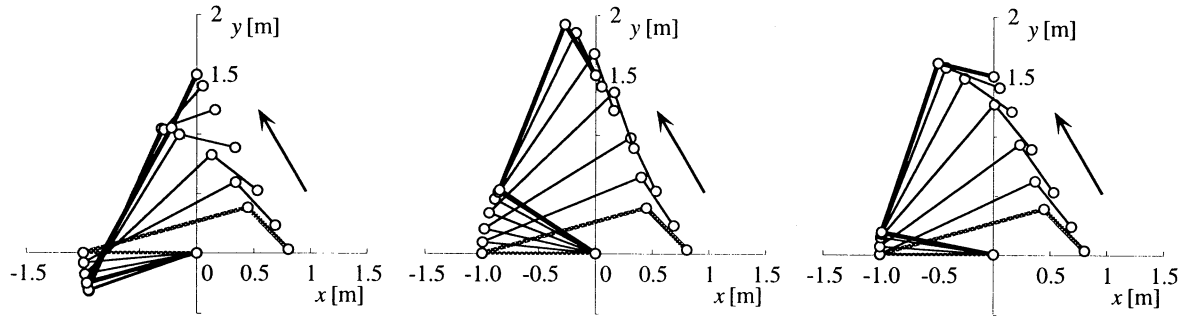
When the joint torque $\boldsymbol{\tau}^{\psi}$ (37) is selected as $\boldsymbol{\tau}^*$, we can obtain the joint torque $\boldsymbol{\tau}_{joint}$ (5) to control the internal motion of the redundant manipulator in the actual time scale.

The total feedback control law $\boldsymbol{\tau}$ (6) composed of the designed controller given in (30) and (37) can lead the end-effector to the target position at the specified time t_f and can also attain the desired posture without altering the configuration of the end-effector.

4.4 Dynamic behavior of the end-effector

In this subsection, the dynamic behavior of the end-effector controlled by the feedback controller designed in **4.2** is analyzed. To simplify the discussion, the target position for the end-effector is set at the origin in the operational space. Substituting the feedback control law $\boldsymbol{F}^{\psi}$ given in (30) into the original linear system equation (19), we have the following differential equation as:

$$\ddot{\boldsymbol{x}} = -p^2 \left(\frac{\dot{\xi}}{\xi} \right)^2 \boldsymbol{K}_2^{-1} \boldsymbol{K}_1 \, \boldsymbol{x} + \left\{ (p-1)\frac{\dot{\xi}}{\xi} + \frac{\ddot{\xi}}{\dot{\xi}} \right\} \dot{\boldsymbol{x}} \,. \tag{38}$$



(a) Case 1 (b) Case 2 (c) Case 3

Figure 1: Changes of the generated end-effector trajectories with different positive functions $Q_i(q)$

Here, we first analyze the behavior of the end-effector on the x coordinate. From (38), the following Euler's equation with respect to x and ξ can be derived as:

$$\xi^2 \frac{d^2x}{d\xi^2} - (p-1)\,\xi \frac{dx}{d\xi} + \frac{k_1^1}{k_1^2} p^2 x = 0\,. \qquad (39)$$

Since the non-increasing function ξ converges to zero at finite time t_f, the necessary and sufficient condition to converge x, $\dot{x}$ and $\ddot{x}$ to zero at the specified time t_f is given with respect to the discriminant of the characteristic polynomial $D_x = 4\frac{k_1}{k_4} - 1$ as follows:

$$(1)\ if\ \ D_x \geq 0\ \ then\ \ p > 4(1-\beta)\,,$$

$$(2)\ if\ \ D_x < 0\ \ then\ \ p > \frac{4(1-\beta)}{1-\sqrt{-D_x}}\,.$$

The dynamic behavior of the other state variables can be analyzed in the same manner.

It can be proven that the feedback controller $\boldsymbol{F}^{\psi}$ can regulate the dynamic behavior of the end-effector and the convergence time to reach the goal.

5 Computer simulations

The proposed trajectory generation method is applied to a redundant manipulator. Figure 1 shows the simulation results with a three-joint-planar manipulator. The initial posture of the manipulator is $\boldsymbol{q}(0) = (\pi, -\frac{5\pi}{6}, -\frac{\pi}{6})^T$ [rad], and the target position of the end-effector is $\boldsymbol{x}^* = (0.0, 1.5)^T$ [m] with the convergence time $t_f = 5.0$ [s] under $p = 8.0$, $\alpha = 1.0$. The gain matrices $\boldsymbol{K}_i$ $(i = 1, 2, 3)$ are set at $\boldsymbol{K}_1 = \mathrm{diag.}(0.25, 0.25)$ [N/m], $\boldsymbol{K}_2 = \mathrm{diag.}(1.0, 1.0)$ [Ns/m], and $\boldsymbol{K}_3$= diag.(1.0,1.0,1.0) [Nm/(rad/s)], respectively. We used the Appel method for the manipulator dynamics [13] and the link parameters of the manipulator as shown in Table 1.

Table 1: Link parameters of the manipulator

	link 1	link 2	link 3
length [m]	1.0	1.5	0.5
mass [kg]	0.8	1.2	0.4
center of mass [m]	0.4	0.6	0.25
moment of inertia [kgm^2]	0.06666	0.22500	0.00833

Figure 1 (a) shows the generated trajectory with the potential function $Q(\boldsymbol{q})$ set at

$$Q_1(\boldsymbol{q}) = 0\,. \qquad (40)$$

On the other hand, the joint angle control of the first joint and maximization of the manipulability [11] is considered as a subtask in Fig.1 (b) and (c). In these cases, the potential functions $Q(\boldsymbol{q})$ are given as

$$Q_2(\boldsymbol{q}) = \frac{1}{2}(q_1^* - q_1(t))^2\,, \qquad (41)$$

$$Q_3(\boldsymbol{q}) = \sqrt{\det \boldsymbol{J}\boldsymbol{J}^T}\,, \qquad (42)$$

where the target angle of the first joint q_1^* is specified as $q_1^* = \frac{5\pi}{6}$[rad].

It can be seen that the generated trajectories are influenced by the corresponding potential functions $Q(\boldsymbol{q})$ defined above. In Fig.1 (a), we can observe that the third joint of the manipulator is outstretched while the end-effector reaches the target position.

In contrast, the end-effector reaches the target position without any singular configurations by utilizing the redundancy control of the manipulator corresponding to locall optimization of the potential functions $Q(\boldsymbol{q})$ in Fig.1 (b) and (c).

Figure 2 shows the time history of the end-effector position $\boldsymbol{x}$ and velocity $\dot{\boldsymbol{x}}$, and the squared sum of the joint angular velocity. It should be noticed that all generated trajectories of the end-effector in Fig.1 completely coincides with the one smooth trajectory

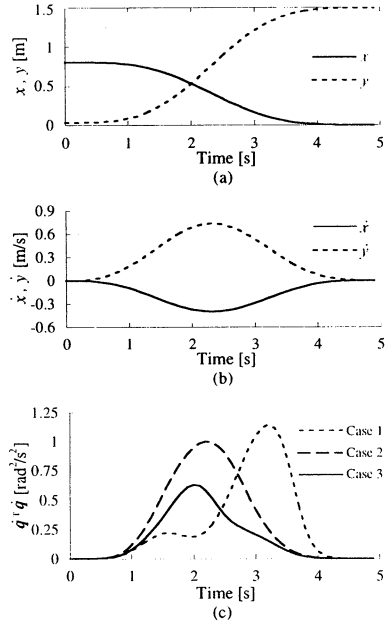


Figure 2: Time histories of the end-effector position, velocity and squared sum of joint velocities

depicted in Fig.2 (a). We can see that the end-effector reached the target position and that the joints of the manipulator do not move any longer after the specified time $t_f = 5.0$[s] in all cases.

6 Conclusions

In this paper, the new trajectory generation method for the dynamic model of redundant manipulators using the concept of the APFA and the time scaling transformation has been presented. We have developed the control strategy for the redundant manipulators that allow achievement of performance with its redundancy. In simulation results with the three joint planar manipulator, the effectiveness of the proposed method was ascertained.

References

[1] L. A. Loeff and A. H. Soni, "An algorithm for computer guidance of a manipulator in between obstacles", Transactions of ASME, Journal of Engineering for Industry, vol.97, no.3, pp.836–842, 1975.

[2] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile robots", int.J.Robotics Res., Vol.5-1, pp.396–404, Spring (1986).

[3] C. I. Connolly, J. B. Burns and R. Weiss, "Path planning using laplace's equation", in Proceedings of IEEE International Conference on Robotics and Automation, pp.2102–2106, 1990.

[4] H. Hashimoto, Y. Kunii, H. Harashima, V. I. Utkin, S. A. Krasnova, and I. M. Kaliko,"Sliding mode control and potential fields in obstacle avoidance", in Proceedings of European Control Conference, pp.859–862, 1993.

[5] T. Tsuji, P. G. Morasso and M. Kaneko, "Trajectory Generation for Manipulators Based on Artificial Potential Field Approach with Adjustable Temporal Behavior", in Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.438–443, 1996.

[6] T. Tsuji, P. Morasso and M. Kaneko, "Feedback control of nonholonomic mobile robots using time base generator", in Proceedings of IEEE Int. Conf. on Robotics and Automation, pp.1385-1390, 1995.

[7] J. M. Hollerbach, "Dynamic Scaling of Manipulator Trajectories", Transactions of ASME, Vol.106, pp.102–106, 1984.

[8] M. Sampei and K. Furuta, "On time scaling for nonlinear systems: Application to linearization", IEEE transactions on Automatic Control, vol.AC-31, no.5, pp.459–462, 1986.

[9] Y. Tanaka, T. Tsuji, M. Kaneko, "Dynamic control of ,mobile robots via artificial potential field approach with time scaling", '98 IEEE International Conference on Robotics and Automation, 1998 (Submitting).

[10] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms", Transactions of SMC, Vol.7, No.12, pp.868–871, 1977

[11] T. Yoshikawa, "Analysis and Control of Robot Manipulators with Redundancy", Robotics Research International Symposium, Brady and Paul eds., MIT Press, pp.735–747, 1984.

[12] O.khatib, "Motion/Force Redundancy of Manipulators, Proc. of U.S.A Symposium on Flexible Automation", Vol.1, pp.337–342, 1990.

[13] V. Potkonjak and M. Vukobratovic, "Two new methods for computer forming of dynamic equation of active mechanisms", Mechanism and Machine Theory, Vol.14, No.3, pp.189–200, 1979.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Force Display for Inter-Atomic Bonds

Minoru Hashimoto
Department of Mechanical Engineering
Kagoshima University
1-21-40 Korimoto, Kagoshima 890, (JAPAN)

Shigeaki Morioka
Dai Nippon Printing Co., Ltd.
Hiroshima, (JAPAN)

Ryoichi Yamamoto
Institute of Industrial Science
University of Tokyo
Tokyo, (JAPAN)

Abstract

We developed a force display system for atomic bonding states in materials. The force display aims to help scientists make discoveries by improving their perception of data describing the atomic world. This paper describes the overview of the project and the control method of a force-display device. The decision procedure of the scaling factors is proposed. The validity of the force display device is proved by the virtual experiment of atomic bonding states with various kinds of material.

Key Words: Robot, Manipulator, Force display, Virtual reality, Materials science, Computer simulation

1 Introduction

We are developing a virtual reality system for atomic bonding states in material. The system is called the virtual material laboratory (VML)[1]. VML aims to help scientists make discoveries by improving their perception of data describing the atomic world. Using the system we can make analysis and design of material efficiently by means of presenting dynamic characteristic of atoms and molecules to human being through force and visual senses.

In the VML it is necessary to present dynamic force behavior of atoms in material. In the mechanical characteristics of material such as deformations and destruction, the dynamic behavior of atoms in material plays an important role. In order to analyze these characteristics we have to know the dynamic behaviors of atoms under stress. As an example of VML experiment, Suzuki et. al [2] made some simulations for superplastic deformation. Although the mechanism of superplasticity seems to be complicated and further information of microscopic transports is required, it is quite difficult to obtain such data from the experiments. Molecular dynamic simulation appears to be useful to obtain these information. If the force display can present the dynamic behavior of the atoms, then the system is useful to analyze the mechanism.

GROPE project [3] has investigated force displays for scientific visualization. The project applied the force display to a complex molecular docking task. The goal was to find the zero-force position and orientation of the molecule. Although this system is very closed to VML, the system deals in static force distributions.

VML consists of a simulation server, a visual display and a force display. We developed the force display device, since Kagoshima group takes charge of the force display part in this system. The aim of this paper is to describe the overview of the force display system in which we can feel the atomic bonding state with the dynamic force.

In the next section, we propose the decision procedure of the scaling factors between atomic world and human world. Then, the control algorithm of the force display is described, and its effectiveness is proved by the experiment of atomic bonding states in Argon, Neon and Xenon.

2 Development of the force display

2.1 Virtual experiment for inter-atomic bonds

Although any computer simulations for material are objects for VML, we selected an example as the first step.

It is important to feel the strength of the inter-atomic bonds and its dynamic behavior, because the inter-atomic bonds are the basis for mechanical properties of material. Fig.1 shows the overview of the virtual experiment. One atom in the pair atoms is fixed on the fixed position (fixed atom), and the operator hold the other of atoms (mobile atom), the behavior of the mobile atom is presented to the operator when

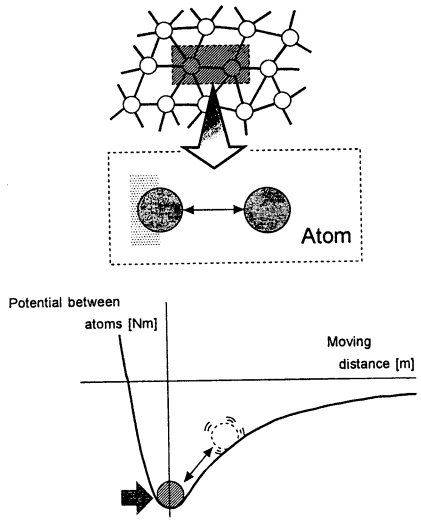


Figure 1: Example of the virtual experiment

a force is applied to the atom.

2.2 Atomic behavior based on a molecular dynamic method

In order to simulate the atomic behavior in material a molecular dynamic method is utilized. In the molecular dynamic method the atoms behaves as particles based on Newton's equation of motion. When the atomic position is expressed in $r_v(t)$, and the applied force of the atom is $F_v(t)$, then the equation of the motion can be written by equation (1).

$$F_v(t) = m_v \cdot \ddot{r}_v(t) \tag{1}$$

Where m_v is mass of the atom. By deforming equation (1), we can obtain the atomic velocity after passing minute time period Δt by the following equation.

$$\dot{r}_v(t + \Delta t) = \frac{F_v(t)}{m_v} \cdot \Delta t + \dot{r}_v(t) \tag{2}$$

Where, F_v is force due to the inter-atomic potential, and F_P is force applied by the operator.

$$F_v = F_P + F_h \tag{3}$$

The inter-atomic force is obtained by the inter-atomic potential. We adopted a Lennard-Jones potential for inert gas crystal.

$$\Phi(r) = 4\varepsilon\{(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6\} \tag{4}$$

Where, ε, σ are the parameters depending on the element. r is the distance between atoms. The inter-atomic force is obtained by partial differentiation of the potential function by the distance r.

$$F(r) = 4\varepsilon\{-12\frac{\sigma^{12}}{r^{13}} + 6\frac{\sigma^6}{r^7}\} \tag{5}$$

2.3 Scaling factor

There is a big difference between the scales of the atomic world and the human world that we are living. Then, the scaling factors must be introduced which converts from the atomic scale into the human scale so that we can recognize various atomic behavior in our scale. What is the best decision of the scaling factors for feeling the atomic behavior? There are limitations in the performance of the force display device and the ranges of force, acceleration and velocity that we can feel. We have to determine the scaling factor with considering these restrictions.

The scaling factor is defined as follows,

$$X_{human} = s_x \cdot X_{atom} \tag{6}$$

X_{human}, X_{atom} are the physical quantities of the human and atomic worlds, respectively. s_x is the scaling factor of the physical quantity, X.

The values of scaling factors are determined by the following procedure to present us correctly the atomic behavior as much as possible.

- Scaling factors satisfy the following equations.

$$S_f = S_m \cdot S_a \quad , \quad S_a = \frac{S_r}{S_t^2} \quad , \quad S_v = \frac{S_r}{S_t} \tag{7}$$

- The operator selects the region of the atomic world and its resolutions that he wants to experience.

Since the equation(7) must be satisfied, the number of independent parameters is three. We selected the scaling factors of position, acceleration and force as the independent variable, because these are basic parameters in force display.

The scaling factors satisfy the following conditions.
[1] The physical values in the human space are within the performance limits of the actuators and the computer. From the limit of the actuators the following relations are obtained.

$$\begin{aligned} S_r \cdot r_{atom}^{max} &\leq r_{actuator}^{max} \\ S_a \cdot a_{atom}^{max} &\leq a_{actuator}^{max} \\ S_f \cdot f_{atom}^{max} &\leq f_{actuator}^{max} \end{aligned} \tag{8}$$

[2] From the limit of the computer the following relation is obtained.

$$S_t \cdot \Delta t_a \geq t_{computer}^{cal} \tag{9}$$

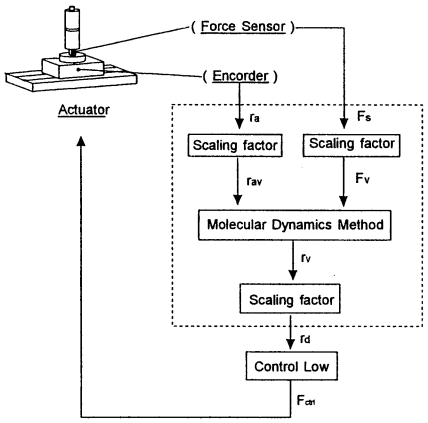


Figure 2: Control algorithm

[3] From the resolution of the atomic world that the operator determined we can obtain the following relations concerning the resolutions of actuators.

$$\begin{aligned} S_r \cdot \Delta r_{atom} &\geqq \Delta r_{actuator} \\ S_a \cdot \Delta a_{atom} &\geqq \Delta a_{actuator} \\ S_f \cdot \Delta f_{atom} &\geqq \Delta f_{actuator} \end{aligned} \tag{10}$$

From the equations (8) and(10), the scaling factors of the position, acceleration and force must satisfy the following relations.

$$\frac{\Delta r_{actuator}}{\Delta r_{atom}} \leqq S_r \leqq \frac{r^{max}_{actuator}}{r^{max}_{atom}} \tag{11}$$

$$\frac{\Delta a_{actuator}}{\Delta a_{atom}} \leqq S_a \leqq \frac{a^{max}_{actuator}}{a^{max}_{atom}} \tag{12}$$

$$\frac{\Delta f_{actuator}}{\Delta f_{atom}} \leqq S_f \leqq \frac{f^{max}_{actuator}}{f^{max}_{atom}} \tag{13}$$

The scaling factors of position, acceleration and force are selected according to the equations (11) to (13). The other parameters (time, mass, velocity) are obtained from the equation (7).

2.4 Control algorithm

A flow chart of the control algorithm is shown in Fig.2. Firstly, the velocity, $\dot{r}_a(t)$, and the force, F_s, of the actuator are measured at time t. Then those values are transformed to the scale of the atomic world by using the scaling factors. The modified values are substituted into equation (2). Using the equation (2), the desired velocity at the next step $\dot{r}_v(t+\Delta t)$ is obtained. Multiplying the scaling factor to the desired velocity in the atomic world, we obtain the desired velocity in the human world. The control input value, F_a, is computed using the following equation.

$$\begin{aligned} F_a \quad = \quad & M_a \ddot{r}_a(t) + D_a \dot{r}_a(t) + \\ & K_p(\dot{r}_r(t+\Delta t) - \dot{r}_a(t)) + \\ & K_i \int (\dot{r}_r(t+\Delta t) - \dot{r}_a(t))dt + F_s \end{aligned} \tag{14}$$

Where $\dot{r}_a(t)$ denotes the present velocity of the actuator. The control is performed with the control input F_a. Then the force, F_s, and the velocity, $\dot{r}_a(t)$, of the actuator are measured in each sampling period, and the processes are repeated..

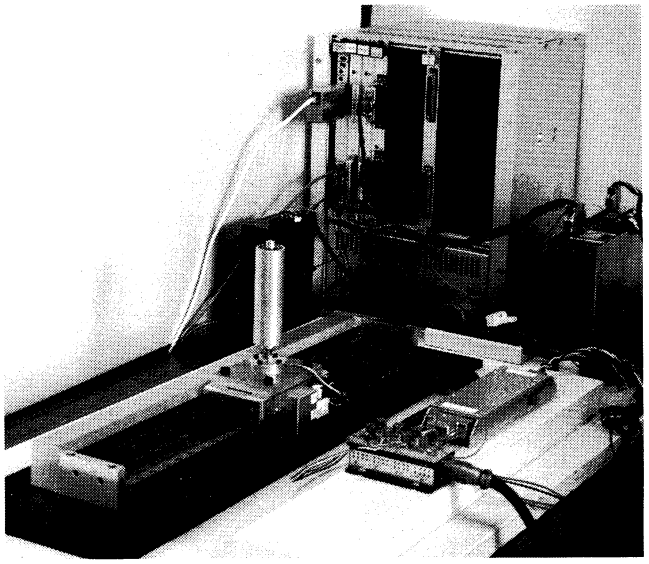


Figure 3: Overview of the force display

2.5 Force display devices

We made a force display device with 1 DOF as the first step, and then the 6 DOF device. The 6 DOF device is built with parallel mechanism, so that the force display has the characteristics of high speed and high accuracy [4].

In the present paper we examine the satisfaction of the required specifications for virtual experiments of atomic bonds using the 1 DOF device.

The overview of the force display devices is shown in Fig. 3. The device consists of an actuator, a force sensor and its controller for the 1 DOF device. We used a direct drive linear actuator (Yokokawa Precision Inc. LM10510J00-0002) for the actuator and a VxWorks operating system for the controller. The controller with VME bus consists of a CPU board (MC68LC40), a D/A converter board (PVME-321), a 12 bits A/D converter board (PVME-301) and a counter board (VPC-44).

Table 1: Atomic parameters

	mass [kg]	σ [m]	ε [N/m]
Argon	6.63×10^{-26}	0.340×10^{-9}	167.0×10^{-23}
Neon	3.35×10^{-26}	0.274×10^{-9}	50.0×10^{-23}
Xenon	21.80×10^{-26}	0.398×10^{-9}	320.0×10^{-23}

Table 2: Scaling factors

S_r	2.00×10^{9}	S_m	3.58×10^{25}
S_a	3.49×10^{-14}	S_t	2.39×10^{11}
S_f	1.25×10^{12}	S_v	8.37×10^{-3}

2.6 Experiments

We made virtual experiments of the inter-atomic bonding states with Argon, Neon and Xenon atoms to examine the validity of the force display system.

In the force display device, the maximum generating force of the actuator is 50 [N], the maximum acceleration is $4.65[m/s^2]$, and the maximum velocity is 0.84[m/s]. The atomic parameters of these elements are shown in Table.1. We selected the maximum values of the force, acceleration and velocity in the atomic world as 1.16×10^{-11}[N],$1.75 \times 10^{14}[m/s^2]$,$8.77 \times 10^{13}$[m/s], respectively. Based on the values the scaling factors are determined as shown in Table.2.

For the static operation the operator moved the actuator very slowly so that the dynamic effect is negligible. From the experimental results it is found that the force sensor output is almost the same as the inter-atomic force computed by the equation (5). Therefor the operator can feel the inter-atomic force through the force display device. The results of the dynamic operation are shown in Fig.4 From the figure we can see that Neon atom shows the fast motion with the small applied force. On the other hand, Xenon atom move slowly under the big external force. These characteristics are due to the inter-atomic bonding states, we can explain these results with the atomic parameters shown in Table 1. From these virtual experiments we can conclude that it is possible to present the dynamic behaviors of the atoms with the force display system.

3 Summary

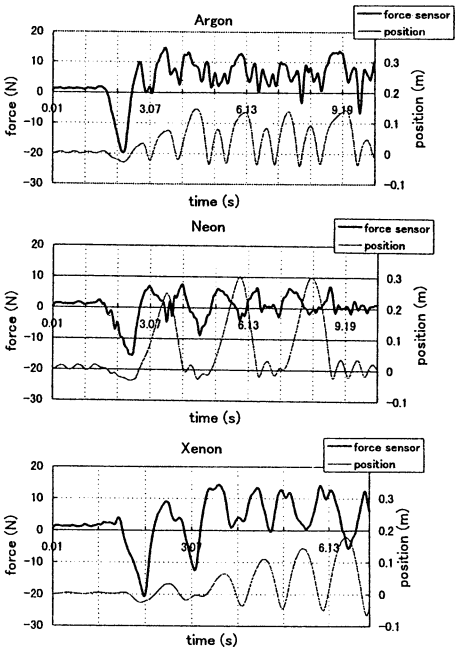


Figure 4: Experimental results for dynamic operations

In this study we made a force display device for atomic bonds. The decision procedure of the scaling factors and the control algorithm are proposed. The validity of the force display device is proved by the virtual experiments.

This research was supported by science and technology promotion adjustment expense entitled "Study on the technique of virtual experiment for matter and material".

References

[1] Tateizumi Y, Yamamoto R(1993) Possibility of the Virtual Laboratory (in Japanese). SEISAN KENKYU, 45(10):53-59

[2] Suzuki A, Kamiko M, Yamamoto R, Tateizumi Y, Hashimoto M(1997) Molecular Simulations in The Virtual Material Laboratory. The 4th International Union of Materials Research Societies, International Conference in Asia, Makuhari, Chiba, Japan, September 16-18.

[3] Brooks FP, Ouh-young M, Batter JJ, Kilpatrick PJ(1990) Project GROPE-Haptic display for Scientific Visualization. Computer Graphics, 24(4):177-185

[4] Hashimoto M, Morioka S, Yamamoto R(1997) Linear Fixed Type Parallel Mechanism using Linear Servo Actuators. Proceedings of the 2nd Asian Control Conference, Seoul, July 22-25, III:603-606

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Approach to the Emergent Design Theory and Applications

Shinzo Kitamura, Yuzuru Kakuda and Hisashi Tamaki
Dept. Information and Systems Engineering
Kobe University
Rokko-dai, Nada-ku, Kobe 657, Japan

Dept. Electrical and Electronics Engineering
Kobe University
Rokko-dai, Nada-ku, Kobe 657, Japan

Abstract

An emergent approach for design of artifacts (artificial systems) is proposed. First, the system design process is reformulated in the framework of set and mapping theory, and the design problem is defined as an inverse mapping from the set of specification depending on environment to the set of components and their connection. The inverse problem can be solved by the iteration of forward mapping. Next, the problem is classified from the viewpoint of the information on specifications and environment. The concepts of evolution, adaptation, learning and coordination can be related to classified problems. Emergent design procedure is defined in the framework of these concepts, especially, by taking evolutionary computing techniques into account. Two examples are shown; the first is a design of complex linear passive filters and the another a design of multi-link moving robots.

1 Introduction

Human designers have played a main role for the design of existing artifacts (artificial systems). However, we have encountered recently many difficulties in designing process for complex and large scaled systems. To cope with such a situation, what a kind of technology can contribute?

In this paper, an approach by employing the concept of emergence is introduced. Design procedure of artifacts is first reformulated in the framework of set and mapping theory. In this formulation, the design procedure is defined as an inverse problem. This inverse problem is normally solved by the iteration of forward mapping where in many cases human designers should play an important role. However, they fails sometimes in solving the problems adequately for the case of ambiguous environment or even ambiguous specifications. In such problems, the concepts of evolution, adaptation, learning and man-machine coordination will be effective. An emergent design procedure is defined in this framework. By implementing evolutionary computing techniques, two examples for the design of complex linear passive filters and multi-link moving robots are shown.

2 System Design

In this section, we reformulate the problem of designing artifacts (in the following, we simply say 'systems') such as electrical, mechanical, chemical systems and computer programs.

We define first the concept of 'structure' and 'function' of a system to be designed.

(a) Structure: determined by specifying both components (or parts) and their connection.

(b) Function: determined by an interaction between the property associated with a structure and the environment around the system.

Referring to the definitions and assuming that a required function is given as a 'specification', we define the problem of system design as follows:

(c) System design: to synthesize the structure of a system so that it satisfies the prescribed specification under known or ambiguous environment.

2.1 Formal definition

Let b, c, s, f and e represent a component, a connection, a structure, a function and an environment, respectively:

component	$b \in \mathcal{B}$	($\mathcal{B}$: the set of components),
connection	$c \in \mathcal{C}$	($\mathcal{C}$: the set of connection),
structure	$s \in \mathcal{S}$	($\mathcal{S}$: the set of structures),
function	$f \in \mathcal{F}$	($\mathcal{F}$: the set of function),
environment	$e \in \mathcal{E}$	($\mathcal{E}$: the set of environment).

Let $\mathcal{B}_s$ be a subset of $\mathcal{B}$. Then, a structure s is generated by applying a connection c to $\mathcal{B}_s$, and this structure s, i.e., the system, exhibits a certain function f by the interaction with an environment e. This relation can be denoted as follows [1]: Let Q represent a mapping from a structure $s = s(\mathcal{B}_s, c)$ to a mapping λ:

$$Q : (\mathcal{B}_s, c) \longmapsto \lambda . \tag{1}$$

where the mapping λ defines a specification. If an environment e is fixed, a function f with the structure

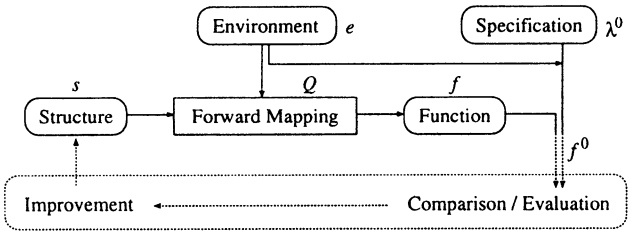


Fig. 1. The framework of system design based on an iterative procedure, where a forward mapping Q is repeatedly applied.

s is determined by

$$Q(\mathcal{B}_s, c)(e) = \lambda(e) = f \in \mathcal{F}. \qquad (2)$$

Based on this representation, the problem of system design is defined:

> System design: to obtain an inverse mapping Q^{-1}, i.e., to solve an inverse problem.

In contrast to the system design, the system analysis, i.e., to trace the causality to clarify the function f of a system, is rather easy by applying related theories, by executing computer simulations, or by doing some experiments. Whereas, as described above, the system design is an inverse problem, i.e., to determine the inverse mapping Q^{-1}. In order to obtain an approximate solution of this problem, we need an iterative procedure where the forward mapping Q is repeatedly applied. Figure 1 shows this framework.

2.2 Difficulties in system design

The difficulties in solving the problem of system design are considered in relation to the information on the environment and the specification. If the information on the environment e is complete, e.g., if its dynamics is completely described, we can separate a system to be designed from the environment. Moreover, if the specification given by human designers is complete, we can separate the system from the human designers. Then, the difficulties in the system design are categorized in the following three cases:

(1) Complete problem: Both the information on the environment and specification are complete. The problem in this case is reduced to the optimization procedure.

(2) Incomplete environment problem: The information on the environment is incomplete, while the specification is complete. Hence, we must consider a framework including the dynamical properties of unknown or ambiguous environment. The key point is adaptation and/or learning to cope with the changes of environment.

(3) Incomplete specification problem: Both the specification as well as the information on the environment are incomplete. Hence, we must additionally consider the human-machine interaction in the case of (2). The key point is the coordination between human designers and machines.

3 Emergent Design

We use, in this paper, the word 'emergence' in the following sense[1].

> Emergence: appearance of unpredictable functions (or structures) through the design process.

The processes as evolution, adaptation, learning and coordination are considered to be essential for revealing the emergent property. According to the definition of the emergent property, we extend the framework in Fig. 1 to:

> Emergent design: to find a structure satisfying (or being closer to) the specification by iterative procedures, i.e., by repeating the evaluation of the function generated, the comparison of it with the specification, and the generation of new candidates.

In Fig. 2, a scheme of emergent design is shown. The 'comparison/evaluation' and the 'candidate generation' blocks are essential for composing the emergent design procedure. Evolutionary computing techniques, e.g., the genetic algorithm and the genetic programming, can be used as a tool to implement these blocks. In this approach, generated entity (individual) corresponds to the 'phenotype' and the information to specify the entity to the 'genotype'. The concept of 'genotype' is rather new in the design process of artifacts.

4 Evolutionary Computing Approach

4.1 Design of passive filters

The design procedure of linear passive filters is a typical inverse process and has been solved in terms of positive real functions. However, there are difficulties in this method to generate the circuits with complex specifications. On the other hand, the genetic programming method have been applied to active analog filters by J. Koza [4] and to digital circuits by H.

[1] Note that the word 'emergence' has been favorably used by researchers in the field of artificial life [2]. Implication here is almost same to their notion.

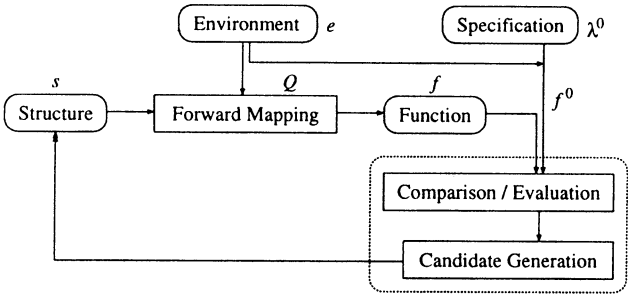


Fig. 2. The basic framework of the emergent design. The key point is a procedure indicated by the dashed box.

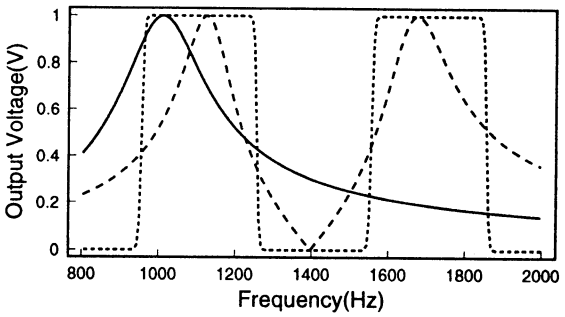


Fig. 3. Frequiency characteristics obtained in the evolutionary process. (The dotted line shows an ideal one, the solid line an initial individual, and the dashed line an individual at the 1000th generation.)

Hemmi [5]. Their results encouraged us to apply the evolutionary computing to design of linear passive filters. We have used the genetic algorithm to make the problem easy and the meaning of genotype in the design process clear.

The components (parts) in this problem are capacitors, inductors and resistors. The ideal frequency characteristic, such as low-pass, high-pass, band-pass or more complicated one, is given as a specification. An incidence matrix whose elements define both the attribute values of components and the connections between components was used as a genotype. The fitness was defined as a difference between the specification and the frequency characteristic of a realized phenotype. We needed to implement only two meta softwares; one is the genetic algorithm and the another a package for circuit analysis (Al's Circuit Simulator in our case).

The implemented system could generate various filters as the generation proceeds. The evolutionary process could be controlled by imposing appropriate constraints for the fitness function; for example, the number of parts could increase monotonically or it could converge to some value, depending on the constraint. Figure 3 is a circuit with double-peak band pass characteristic shown in Fig. 4, where the parts could take E6 standard discrete values.

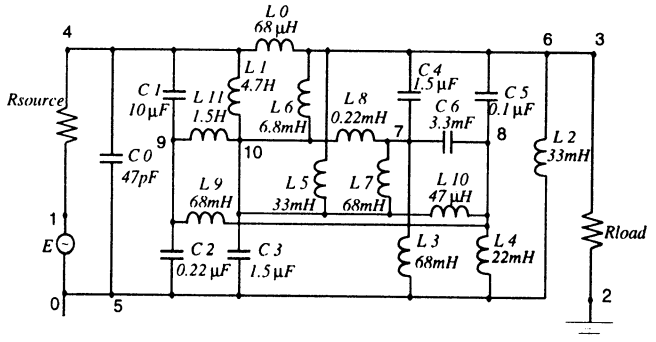


Fig. 4. A filter circuit corresponding to the individual at the 1000th generation.

4.2 Design of multi-joint moving robots

A moving robot which is composed of multiple links and rotating joints was generated by the genetic algorithm. We have been interested in what kinds of structure as a robot and of the motion followed by this structure can emerge in two dimensional plane. A robot was represented as a tree which had a root at the world coordinate. It was assumed that only four links could be connected to a joint, and joint's mass was proportional to the number of links whose mass was proportional to their length. Each joint had a rotating driving torque in sinusoidal motion. A genotype was defined as a string of the parameters for the driving torque, the length of links and their connection. The fitness was defined as a distance the robot could move.

In the simulation studies with the package MOBILE[2], there have appeared various type of robots under the conditions stated above. General remarks in the evolutionary process are; structures of the robots evolved first and next the parameters of components were fixed. This process is similar to that of human designers. The number of components has an optimal value for the robot with the best fitness, since the driving torque of each joint has a limit and also the weight of each link has been taken into account in dynamics. Therefore, the robots evolved so as to minimize the energy consumption rate per a unit of moving distance.

Figure 5 shows a time course of the number of joints in the evolutionary process. Figure 6 shows a walking (moving) pattern of the best scored robot. We have

[2] MOBILE was developed by Prof. M. Hiller and his colleagues at the University of Duisburg, Germany.

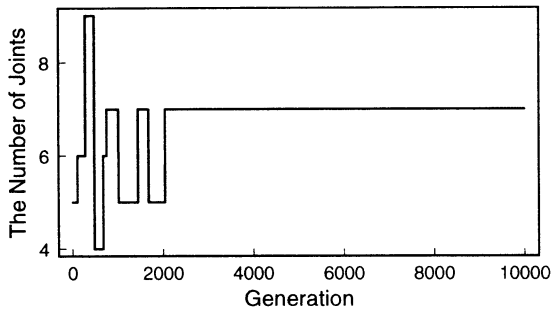


Fig. 5. Convergence process of the numbers of joints.

been interested in noticing this motion pattern looks like that of an inch worm.

5 Conclusion

A bottom up approach for designing artifacts has been proposed under the framework of emergent property and of evolutionary computing. Of course, the advantage of this method is not apparent at present and we do not think it can replace the conventional design procedure in various engineering fields. However, it may show a scope for extending such methods to the design of complex systems.

Acknowledgement

This research was supported by the Grand-in-Aid for Scientific Reseach on Priority Area No.264 from the Ministry of Education, Science, Sports, and Culture of Japan. The authors thank Mr. M. Iwakami and Mr. J. Gotoh for their help in this study.

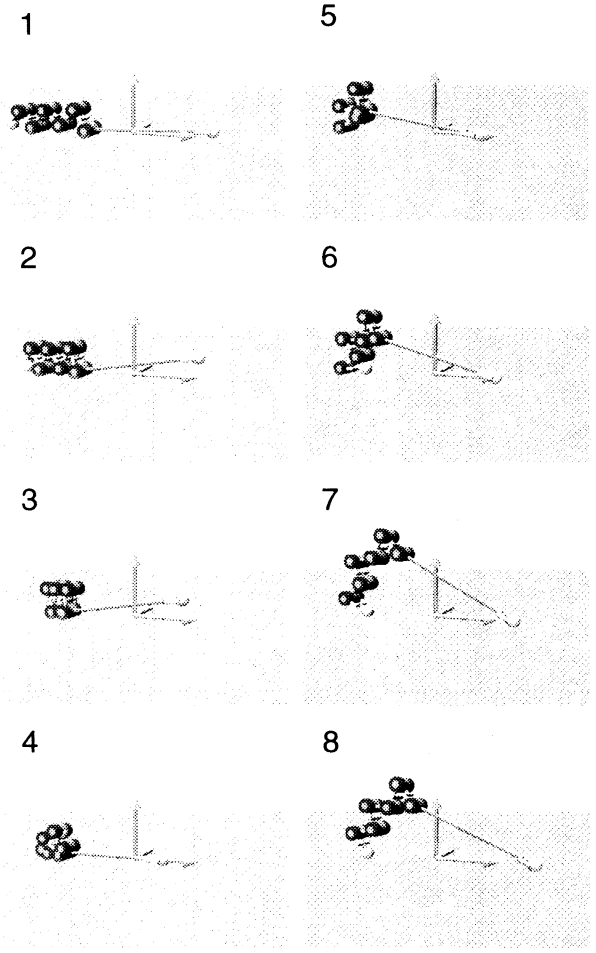


Fig. 6. Walking pattern of a moving robot with seven links and seven joints. (The number shows an elapsed time.)

References

[1] S. Kitamura, Interpretations of Emergent Properties in Engineering, Newsletter for System Theory of Function Emergence (Scientific Research on the Priority Area from the Ministry of ESSC of Japan), No. 5, pp. 1–6, 1996

[2] C. G. Langton, Artificial Life, *Artificial Life* (C. G. Langton, Ed.), pp. 1–48, Addison-Wesley, 1989

[3] H. Yoshikawa, Introduction to General Design Theory, Journal of the Japan Society for Precision Engineering, **45** (8), pp. 20–26, 1979

[4] J. R. Koza et al., Four Problems for Which a Computer Program Evolved by Genetic Programming is Competitive with Human Performance, *Proc. IEEE ICEC '96*, pp. 1–10 (1996)

[5] H. Hemmi, J. Mizoguchi and K. Shimohara, Development and Evolution of Hardware Behaviors, *Proc. Artificial Life IV*, pp. 371–376 (1996)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Evolutionary Design of Lens System

Isao Ono
Dept. of Computational Intelligence & Systems Science, Graduate School of Interdisciplinary Sci. & Eng.,
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku,
Yokohama, 226, Japan.
E-mail: isao@fe.dis.titech.ac.jp

Yoshihiro Tatsuzawa
Dept. of Computational Intelligence & Systems Science, Graduate School of Interdisciplinary Sci. & Eng.,
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku,
Yokohama, 226, Japan.
E-mail: tatsuzawa@fe.dis.titech.ac.jp

Shigenobu Kobayashi
Dept. of Computational Intelligence & Systems Science, Graduate School of Interdisciplinary Sci. & Eng.,
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku,
Yokohama, 226, Japan.
E-mail: kobayasi@dis.titech.ac.jp

Abstract

This paper presents a new lens design method with a genetic algorithm (GA). The GA has been expected as a global search method for complex multi-modal functions with strong dependencies among parameters. The GA also has received attention as a multi-objective optimization technique. In this paper, we consider two criteria for lens systems; the distortion and the resolution. First, we employ a single criterion which is the addition of the distortion and the resolution and design a GA for a single criterion. We show the GA's effectiveness by applying it to a three-element lens design problem and some four-element lens design problems. Next, we design a multi-objective GA that can find a set of solutions which are not dominated by any others on the two criteria. We apply the multi-objective GA to a three-element lens design problem and demonstrate that it can effectively find a lot of triplets that are not dominated by any others.

1. Introduction

The lens design problem is known as a very difficult problem. This is because the lens design problem has many parameters to determine, strong interactions among parameters and many local optima, and also because it has many criteria such as well-known Seidel aberration, chromatic aberration, size and cost.

Most conventional optimization techniques for lens design are based on local search [4]. If a proper starting point is not selected, a local search gets stuck at a local optimum and fails in finding a global optimum. A designer must choose a starting point by trial and error to obtain a desired lens. Moreover, a designer must determine an appropriate tradeoff-ratio among many criteria by trial and error because conventional techniques cannot handle plural criteria at the same time and a designer cannot know the appropriate tradeoff-ratio before a designer actually runs an optimization program. As mentioned above, the lens-design process heavily depends on a designer's experience in finding a proper starting point and an appropriate tradeoff-ratio among plural criteria.

The genetic algorithm (GA) is an optimization technique inspired by the evolution process of natural life. GAs mainly have two characteristics; One is that it is a generate-and-test method and the other is that it is a population-based stochastic search. The first characteristic means that GAs can work if any individuals, or solutions can be evaluated and any two individuals can be compared. GAs do not need information on the derivative of an evaluation functions. GAs are effective against problems that have sophisticated constraints. The second characteristic means that GAs stochastically search using plural individuals. This allows GAs to globally search the problem space and the GA has received attention as a global search technique. Moreover, taking advantage of the both characteristics, GAs can explicitly optimize plural criteria and find a Pareto optimal set, which is a rational solution set in multi-objective optimization, at the same time. The GA has also received attention as a multi-objective optimization technique.

In this paper, we propose a new method for lens design taking advantage of GA's capability of global optimization and multi-objective optimization against two big problems in conventional lens design techniques; 1) choosing a starting point by trial and error, and 2) combining plural criteria to a single criterion. We consider two criteria for lenses, the resolution and the distortion, in this paper. First, we apply the proposed method to a three-element lens design problem and some four-element lens design problems under a single criterion to examine the global optimization capability of the proposed method. Next, regarding the lens design problem as a multi-objective problem by explicitly handling the two criteria, we show the effectiveness of the proposed method in multi-objective lens optimization through applying it to a three-element lens design problem.

2. Lens Design Problems

In this paper, given a focal length, f, an F-number, F and a field size, $2w$ as a specification, we consider searching the curvature of surfaces, r_i, and thickness between two surfaces, d_i, to obtain a lens that has good image forming capabilities (Fig.1). An F-number is expressed by $F = f/D$. The indices of refraction of the medium and the number of surfaces, N, are fixed in this paper. The curvature of the last surface and the distance of it to the image surface are modified to meet a required focal length, f, by applying the paraxial optics, or the Gauss optics. Therefore, an N-element lens design

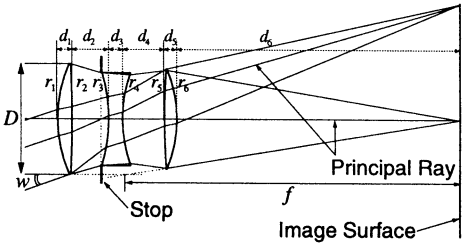


Fig.1 An example of lens system

problem can be considered to be a function optimization problem with $4N$ - 2 parameters.

A lens is evaluated by using spot diagrams made by performing ray tracing (Fig.2). In ray tracing, the three bundles of rays whose entrance angle are 0, 0.65w and w respectively as shown in Fig.2. Each bundle of rays consists of a principal ray, which passes through the center of the stop, and ten rays surrounding and parallel to the principal ray.

In this paper, we use two criteria, the distortion, D, and the resolution, R. The distortion, D, is defined as the distance between the image point of a principal ray and the ideal image point calculated by $f \tan w$. The resolution, R, is the size of the blur of rays surrounding a principal ray.

$$D = \sum_{W\in\{0,0.65w,w\}} \sqrt{(x_0 - x_{ideal})^2 + (y_0 - y_{ideal})^2}$$

$$R = \sum_{W\in\{0,0.65w,w\}} \sqrt{\sum_{k=1}^{10}[(x_k - x_0)^2 + (y_k - y_0)^2]/10}$$

, where (x_{ideal}, y_{ideal}) is an ideal image point, (x_0, y_0) is the image point of a principal ray and (x_k, y_k) is the image point of other ray surrounding the principal ray.

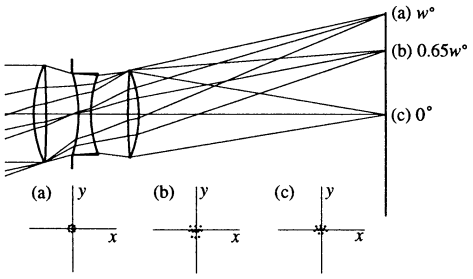


Fig.2 An Example of ray tracing and spot diagrams

3. Designing Representation and Crossover

In this paper, we use a real number vector as a representation. A chromosome is a vector (r_1, r_2, ..., r_n, d_1, d_2, ..., d_n), where r_i is the curvature of surface and d_i is the distance between two surfaces facing each other.

We employ the Unimodal Normal Distribution Crossover (UNDX) [2] as a crossover. The UNDX generates two children obeying a normal distribution around the parents as shown in Fig3. One of the standard deviation of the normal distribution is proportional to the distance between the two parents, Parent1 and Parent2, and the others are proportional to the distance of the third parent, Parent3, to the axis connecting Parent1 and Parent2. In Fig.3, n is the number of parameters.

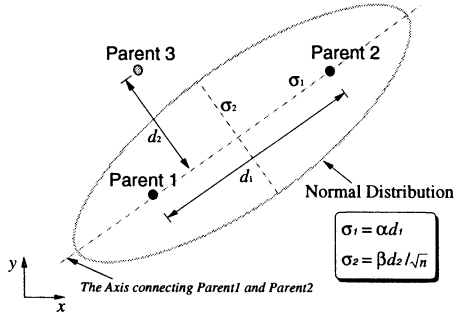


Fig.3 Unimodal Normal Distribution Crossover (UNDX) (No. of Parameters: 2)

It has been shown that the UNDX can efficiently optimize multimodal functions and functions with strong interaction among parameters [2]. We believe that the reasons of the effectiveness of the UNDX are that the UNDX searches globally in the early phase of search where parents are scattered all over the search space and locally in the last phase where parents gather in some promizing areas, and that the UNDX works not depending on the coordinate system very much.

4. Global Optimization by A GA

In this section, we show the effectiveness of GA's capability of global optimization through some experiments. We employ a single evaluation function by setting a tradeoff-ratio between the resolution, R, and the distortion, D.

4.1 Designing A Generation Alternation Model

We employ the Minimal Generation Gap (MGG) model [3] as a generation alternation model shown in Fig.4. In this model, a generation alternation is done by applying the crossover n times to a pair of parents randomly chosen from the population and replacing the original parents with the two individuals chosen from the parents and their children, where one is the best individual and the other is chosen using a roulette wheel selection, or a proportional selection [1].

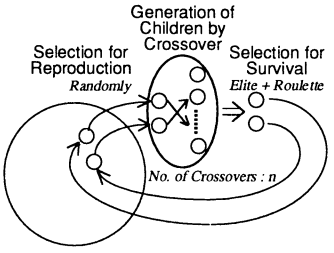


Fig.4 Generation Alternation Model for the single-objective GA

4.2 Experiments and Results

To confirm the effectiveness of the proposed method, we applied it to a three-element lens design problem and three four-element lens design problems. In all experiments, we set a population size to be 100, the number of crossovers to be 100, α and β of the UNDX to be 0.5 and 0.35 respectively. We used the tradeoff-ratio between the distortion D and the resolution R of 1:1.

[Experiment 1] First, we apply the proposed method to a three-element lens design problem. The specifications are the focal length of 100mm, the F-number of 3.0 and the field angle of 38.0 degrees. The GA starts with an initial population consisting of such random lenses as shown in Fig.5 and stops after evaluating 4.0 x 10^6 lenses. Fig.6 shows the layout of the best lens found by the GA and its spot diagrams. The extent of the spot diagrams is from -0.5mm to +0.5mm. A typical convergence curve on a Pentium 200MHz PC is shown in Fig.7.

[Experiment 2] Next, we apply the proposed method to three kinds of four-element lens design problems, where the specifications are 1) a normal lens: f=50mm, F/2.0, 2w=46.0, 2) a telephoto lens: f=135mm, F/2.8, 2w=18.2, and 3) a wide-angle lens: f=20mm, F/5.6, 2w=92.0. The

GA starts with random lenses as well as Experiment 1 and stops at 1.0 x 10^6 evaluations. Examples of the layout of lens obtained by the proposed GA and their spot diagrams are shown in Fig8, Fig.9 and Fig.10.

4.3 Discussions

In Experiment 1, the proposed GA found the lens type, which is called the triplet, shown in Fig.6 in all of ten trials. The triplet is known as the best three-element lens that lens designers have ever found. It is said to be difficult to obtain the triplet by using conventional lens-optimization techniques and starting from a randomly-chosen starting point without the knowledge of lens designers. We can say that the global search by the proposed GA is effective in lens design because the proposed method found the empirically best lens, the triplet, without the knowledge of lens designers.

In Experiment 2, the proposed GA found lenses with relatively good performance under each specification. In the normal lens design problem, the proposed GA found various lenses including famous lens types, the Lee type, the Gauss type and the Ernostar type. In the telephoto lens design problem, we obtained two kinds of lenses including the Ernostar type. The proposed GA obtained two kinds of wide-angle lenses, which show relatively good performance, as shown in Fig.10.

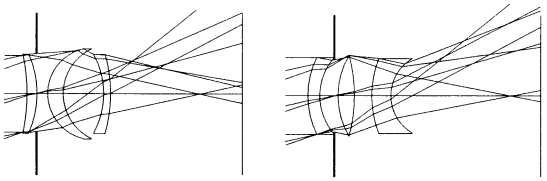


Fig.5 Examples of initial solution in a three-element lens design problem

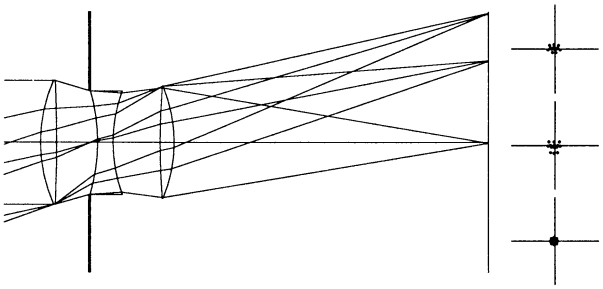


Fig.6 The best solution found by the proposed GA for a single criterion in a three-element lens design problem

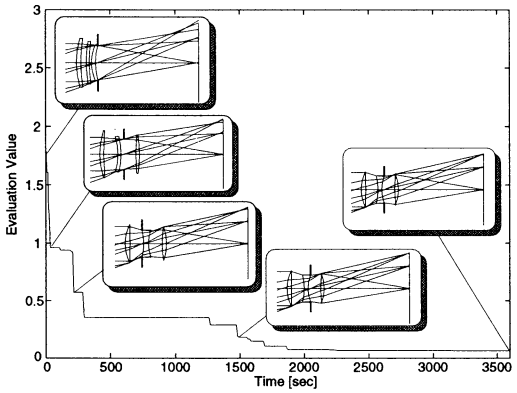


Fig.7 A typical convergence curve in designing a three-element lens on a Pentium 200MHz PC

5. Multi-objective Optimization by A GA

In this section, we explicitly handle the two objectives of the distortion and the resolution and show GA's capability of multi-objective optimization in lens design.

5.1 Designing A Generation Alternation Model

We employ a generation alternation model based on the Pareto optimal selection strategy [5] as shown in Fig.11. The outline of the model is as follows; 1) choose n pairs of

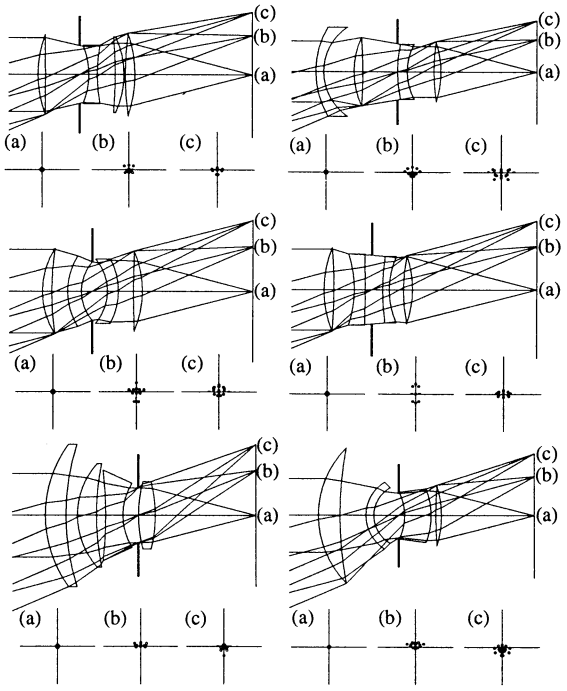


Fig.8 Examples of lenses found by the proposed GA: F=2.0, f= 50mm, $2w$= 46 °

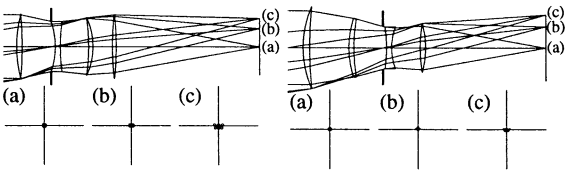


Fig.9 Examples of lenses found by the proposed GA: F=2.8, f= 135mm, $2w$= 18.2 °

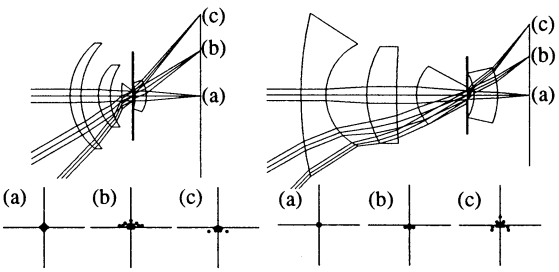


Fig.10 Examples of lenses found by the proposed GA: F=5.6, f= 20mm, $2w$= 92 °

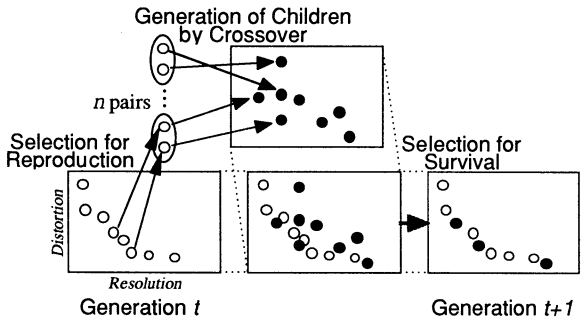


Fig.11 Generation alternation model for the multi-objective GA

parents randomly from the population of the current generation, 2) generate children by applying the crossover to the parents, 3) select non-dominated individuals from the children and the current population, and 4) let the selected individuals to be the population of the next generation. A non-dominated individual is a solution which dominates all other solutions in at least one criterion. The generation alternation model used this strategy can explicitly optimize plural criteria without setting a tradeoff-ratio between the resolution and the distortion.

5.2 Experiments and Results

In order to confirm the effectiveness of the proposed multi-objective GA, we applied it to the three-element lens design problem used in Experiment 1 in section 4.2. We used five non-dominated solutions found by the single-objective GA as an initial population and set n to be 40. The GA runs until 1.6 x 10^6 solutions are evaluated. Fig.12 shows an example of a non-dominated solution set found by the proposed method in a single run. In Fig.12, the horizontal axis and the vertical one show the distortion and the resolution respectively. The number of non-dominated solutions in Fig.12 is 750.

5.3 Discussions

The non-dominated solution set in Fig.12 can be evolved toward the bottom and left direction by resuming the search. However, the solution set must be close to the exact Pareto optimal set because the improvement of the solution set slows down. All lenses in Fig.12 are the triplets and we think that the triplet is the optimal type at any tradeoff-ratio under the specification.

In Fig.12, we can see that we can improve the resolution to some point without deteriorating the the distortion. However, we cannot predict this kind of information. It should be more natural way that a designer picks up a lens which he or she thinks most suitable from a non-dominated lens set found by the multi-objective GA than that a designer searches a lens under a single criterion in which he or she combines plural criteria using some tradeoff-ratio previously determined by intuition.

Fig.13 shows a magnification of Fig.12 with plotting the best lens found in Experiment 1 in Section 4.2 shown in Fig.6. Fig.13 shows that the multi-objective GA found many better solutions that the single-objective GA did. This suggests that making a multi-objective problem a single-objective one by combining plural criteria into a single criterion makes the problem difficult.

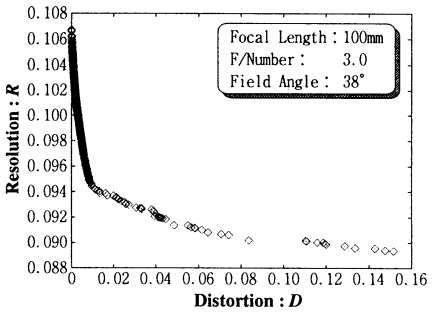


Fig.12 Solutions found by the multi-objective GA in a single run

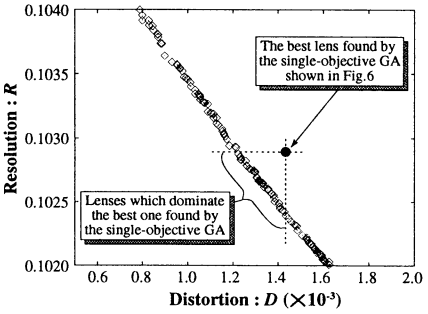


Fig.13 Comparison of the solutions shown in Fig.11 and the one shown in Fig.6

6. Conclusion

In this paper, we proposed a lens design method by the GA and showed its effectiveness. Especially, the multi-objective GA that explicitly optimizes plural criteria is effective in finding plural different lenses in performance at once.

For future work, we would like to enable the proposed method to handle chromatic aberration and to optimize the number of elements. Furthermore, we would like to devise a crossover and a generation alternation model to realize more efficient search.

Acknowlegdements

This research has been done with the cooperation of Nikon Corp. We thank for their offering data and useful comments.

References

[1] Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wsley Publishing Company Inc. (1989).

[2] Ono, I. and Kobayashi, S. : A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, Proceedings of 7th International Conference on Genetic Algorithms, pp.246-253 (1997).

[3] Satoh, H., Yamamura, M. and Kobayashi, S. : Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation, Proceedings of IIZUKA '96, pp.494-497 (1996).

[4] Shannon, R. R. : The Art and Science of Optical Design, Cambridge University Press (1997).

[5] Yoshida, K., Yamamura, M. and Kobayashi, S. : Generating Pareto Optimal Decision Trees by GAs, Proceedings of IIZUKA '96, pp.854-858 (1996).

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Adaptive State Space Design for Reinforcement Learning

Hajime Murao and Shinzo Kitamura
Department of Computer and Systems Engineering
Faculty of Engineering, Kobe University
Kobe, 657 JAPAN

Abstract

In this paper, we propose a method to construct state space for suitable for Q-learning to accomplish tasks in continuous sensor space. In the proposed algorithm, a learning agent starts with single state covering whole sensor space. A new state is generated incrementally by segmenting a sub-region of the sensor space or combining the existing states. The criterion for incremental segmentation and combination is derived from Q-learning algorithm. Simulation results show that the proposed algortithm is able to construct the sensor space effectively to accomplish the task. The resulting state space reveals the sensor space in a Voronoi tessellation.

1 Introduction

Reinforcement learning is an efficient method to acquire adaptive behavior of a learning agent with little or no *a priori* knowledge of an environment where the agent will work. However, there is a problem in applying reinforcement learning to tasks in the real world, *i.e.* how to construct state spaces suitable for the reinforcement learning. A state space is usually designed by segmenting a continuous sensor space using human intuitions. Such state space is not always appropriate to accomplish a task. Coarse segmentation will cause so-called "perceptual aliasing problem" [1] by which a agent cannot discriminate states important to accomplish a task. Fine segmentation to avoid the perceptual aliasing problem will produce too many states to manage with computational resources such as CPU time and memory. It might be a rather reasonable solution to this problem applying a agent not with a state space designed by human but with a method to construct state space using information of its environment.

In this paper, we propose a Q-learning with adaptive state space construction. This provides rather simple but efficient method than other approaches[2]. In the proposed algorithm, a agent starts with single state covering a whole sensor space. A new state is generated incrementally by segmenting a sub-region of the sensor space or combining existing states based on sensor vectors and reinforcement signals. A criterion for the incremental segmentation and combination is derived from the Q-learning algorithm. This is an online method to construct state space without *a priori* knowledge of an environment.

The next section gives a brief review of the reinforcement learning. The basic idea is then introduced, followed by an illustration of the learning algorithm. Some results of computer simulations are reviewed, and finally we conclude.

2 Reinforcement Learning

An N_s sensors learning agent provides N_s-dimensional vector $\boldsymbol{s}$, for which the i-th component is a range value s_i provided by the i-th sensor. For every sensor vector $\boldsymbol{s} \in \boldsymbol{S}$, the agent can take an action a from the action set $\boldsymbol{A}$. The action $a \in \boldsymbol{A}$ for the sensor vector $\boldsymbol{s} \in \boldsymbol{S}$ causes a transition of the sensor vector to $\boldsymbol{s}' = e(\boldsymbol{s}, a) \in \boldsymbol{S}$, where e is a given transition function which defines an environment. We assume that a fitness value $v(\boldsymbol{s})$ is defined for each sensor vector $\boldsymbol{s}$. The agent however can receive reinforcement signals only, which represent a fitness gain or loss between the sensor vector $\boldsymbol{s}$ and the next sensor vector $\boldsymbol{s}'$. Since the sensor vector $\boldsymbol{s}'$ is defined by a previous sensor vector $\boldsymbol{s}$ and an action a, we can define the reinforcement signal as $r(\boldsymbol{s}, a)$. A purpose of the reinforcement learning is to find a policy of selecting an action a for a sensor vector $\boldsymbol{s}$ that maximizes the discounted sum of the reinforcement signals $r(\boldsymbol{s}, a)$ received over time.

The Q-learning algorithm gives us a sophisticated solution to this problem. A sensor space $\boldsymbol{S}$ is quantized into a discrete and finite set $\boldsymbol{X}$ as the Q-learning algorithm requires. A sub-region $\boldsymbol{S}^x$ of the sensor space $\boldsymbol{S}$ is characterised by a corresponding state $x \in \boldsymbol{X}$.

If a sensor vector $\boldsymbol{s}$ is within this region, we say the agent is in the state x. An estimated discounted sum of the reinforcement signals $Q_n(x,a)$ for an action a is assigned to each state x, where n is the number of updates. If the agent is in a state x, an action a is selected from $\boldsymbol{A}$ according to Boltzmann distribution of a $Q_n(x,a)$ value, as follows,

$$P(a|x) = \frac{\exp(Q_n(x,a)/\tau)}{\sum_{b\in\boldsymbol{A}} \exp(Q_n(x,b)/\tau)} \tag{1}$$

where τ is a scaling constant. In the Q-learning algorithm, the agent transits from the state x to a state $x' \in \boldsymbol{X}$ by the action a and this updates the $Q_n(x,a)$ value as,

$$\begin{aligned} Q_{n+1}(x,a) \Leftarrow & (1-\alpha)Q_n(x,a) \\ & + \alpha(r(\boldsymbol{s},a) + \gamma \max_{b\in\boldsymbol{A}} Q_n(x',b)) \end{aligned} \tag{2}$$

where α is a learning rate and γ is a discounting factor.

After a sufficient number of iteration, an action a which maximizes a $Q_n(x,a)$ is the optimal decision policy at a state x.

3 Basic Concept

The basic concept of the proposed algorithm is to find a state x in which a reinforcement signal $r(\boldsymbol{s},a)$ for an action a is constant for every $\boldsymbol{s} \in \boldsymbol{S}^x$ and therefore we can define it as a function $r^x(a)$ of the action a. This is a method to model the fitness landscape by using inclined planes.

If there is the only state x covering the whole sensor space $\boldsymbol{S}$ in which $r(\boldsymbol{s},a)$ for an action a is constant for every $\boldsymbol{s} \in \boldsymbol{S}^x$, $Q_n(x,a)$ for the correct action a (here "correct action" means an action yielding the maximum reinforcement signal) can be calculated as follows,

$$\begin{aligned} Q_{n+1}(x,a) \Leftarrow & (1-\alpha)Q_n(x,a) \\ & + \alpha(r_{max} + \gamma Q_n(x,a)) \end{aligned} \tag{3}$$

where r_{max} is the maximum reinforcement signal given by,

$$r_{max} = \max_{\boldsymbol{s}\in\boldsymbol{S},b\in\boldsymbol{A}} r(\boldsymbol{s},b) \tag{4}$$

Equation 3 converges to,

$$Q_{max} = \frac{r_{max}}{1-\gamma} \tag{5}$$

In general, we need multiple states to cover the whole sensor space. We note here that the following condition is satisfied for every state x and number n,

$$\max_{b\in\boldsymbol{A}} Q_n(x,b) \leq Q_{max} \tag{6}$$

Thus, Eq. 2 can be transformed using $r^x(a)$,

$$\begin{aligned} & Q_n(x,a) \\ & \leq (1-\alpha)Q_{n-1}(x,a) + \alpha(r^x(a) + \gamma Q_{max}) \\ & = (1-\alpha)^n Q_0(x,a) \\ & \quad + \alpha \sum_{i=0}^{n-1} (1-\alpha)^i (r^x(a) + \gamma Q_{max}) \\ & = r^x(a) + \gamma Q_{max} \\ & \quad + (1-\alpha)^n (Q_0(x,a) - (r^x(a) + \gamma Q_{max})) \end{aligned} \tag{7}$$

If the reinforcement signal $r^x(a)$ and the initial Q value $Q_0(x,a)$ satisfy the following conditions,

$$r^x(a) \geq 0, \quad Q_0(x,a) = 0 \tag{8}$$

we can obtain,

$$Q_n(x,a) \leq r^x(a) + \gamma Q_{max} \tag{9}$$

Consequently, we define a agent is in a same state x as long as the reinforcement signal $r(\boldsymbol{s},a)$ satisfies the following equation under the conditions of Eq. 8,

$$Q_n(x,a) \leq r(\boldsymbol{s},a) + \gamma Q_{max} \tag{10}$$

When $Q_n(x,a) - r(\boldsymbol{s},a)$ exceeds γQ_{max}, a new state x' is added, for which the sensor vector $\boldsymbol{s}$ is within the sub-region $\boldsymbol{S}^{x'}$.

4 Learning Algorithm

We define each state x by a set $\mathcal{S}^x$ of representative sensor vectors $\boldsymbol{s}^x \in \mathcal{S}^x$. If $\boldsymbol{s}^x$ is the nearest vector of an incoming sensor vector $\boldsymbol{s}$, a learning agent is said to be in a state x. This criterion will result in a Voronoi tesselation of a sensor space. A procedure of the proposed algorithm using this definition of states is as follows:

1. **[Initialization]** We start with a single state x covering whole sensor space $\boldsymbol{S}$, which is defined by a set $\mathcal{S}^x = \{\boldsymbol{s}^x\}$ where $\boldsymbol{s}^x$ is a randomly intialized representative sensor vector. $Q_0(x,a) = 0$ is assigned to each action a.

2. **[Action based on Q values]** A agent with a sensor vector $\boldsymbol{s}$ in the state x is driven according to the Boltzmann distribution described in Eq. 1 and obtains the reinforcement signal $r(\boldsymbol{s}, a)$ as a result of an action a.

3. When the condition in Eq. 10 is not satisfied under condition of $n > \theta$, (a) otherwise (b). θ is a minimum updates of the state x for segmentation and combination.

   (a) **[Adding a new state]** A new state x' defined by the set $\mathcal{S}^{x'} = \{\boldsymbol{s}^{x'}\}$ is added for which the representative sensor vector $\boldsymbol{s}^{x'}$ is initialized to the sensor vector $\boldsymbol{s}$ of the agent. $Q_0(x', a)$ values for each action a is initialized to $Q_0(x', a) = 0$.

   (b) **[Updating Q value]** The $Q_n(x, a)$ value is updated according to Eq. 2 and the nearest representative sensor vector $\boldsymbol{s}^x$ to the sensor vector $\boldsymbol{s}$ is modified as follows

$$\boldsymbol{s}^x \Leftarrow (1 - \beta)\boldsymbol{s}^x + \beta \boldsymbol{s} \tag{11}$$

   where β is the modification rate of the representative vector with $0 \leq \beta \leq 1$.

4. **[Combining states]** If $n > \theta$ is satisfied for all the $Q_n(x, a)$ value asigned to actions $a \in \boldsymbol{A}$, we calculate a distance $l_{x,y}$ between the state x and every other state y by

$$l_{x,y} = \sqrt{\sum_{a \in \boldsymbol{A}} (Q(x, a) - Q(y, a))^2} \tag{12}$$

   When the distance $l_{x,y}$ satisfies the following condition for a small constant ε

$$l_{x,y} < \varepsilon, \tag{13}$$

   we combine the set $\mathcal{S}^y$ to $\mathcal{S}^x$ as

$$\mathcal{S}^x \Leftarrow \mathcal{S}^x \cup \mathcal{S}^y \tag{14}$$

   and remove $\mathcal{S}^y$.

5 Computer Simulation

We applied the proposed algorithm to a problem navigating a learning agent onto the top of a fitness landscape. We assume the agent with $N_s = 2$ sensors which provide orthogonal coordinates of the agent as $\boldsymbol{s} = (d_0, d_1)$. The reinforcement signal $r(\boldsymbol{s}, a)$ is defined using the fintess value $v(\boldsymbol{s})$ as follows

$$r(\boldsymbol{s}, a) = \begin{cases} 0 & \text{if } v(\boldsymbol{s}') < v(\boldsymbol{s}) \\ 1 & \text{if } v(\boldsymbol{s}') = v(\boldsymbol{s}) \\ 2 & \text{if } v(\boldsymbol{s}') > v(\boldsymbol{s}) \end{cases} \tag{15}$$

The action set $\boldsymbol{A}$ is defined as follows

$$\boldsymbol{A} = \{stop, up, down, right, left\} \tag{16}$$

for which the environmental function $e(\boldsymbol{s}, a)$ is

$$e(\boldsymbol{s}, a) = \boldsymbol{s} + d \cdot \boldsymbol{u}_a \tag{17}$$

with a constant $d = 0.005$ and vectors $\boldsymbol{u}_a$ are defined as follows

$$\boldsymbol{u}_a = \begin{cases} (0, 0) & \text{for } a = stop \\ (0, 1) & \text{for } a = up \\ (0, -1) & \text{for } a = down \\ (1, 0) & \text{for } a = right \\ (-1, 0) & \text{for } a = left \end{cases} \tag{18}$$

Other parameters for simulations are summarized in Table 1.

Table 1. Parameters for simulations of navigating 2-sensors learning agent.

parameters	description
$\tau = 1.0$	the scale constant (Eq. 1).
$\alpha = 0.5$	the learning rate (Eq. 2).
$\gamma = 0.3$	the discounting factor (Eq. 2).
$\beta = 0.1$	the modification rate of the representative vector (Eq. 11).
$\theta = 10$	the minimum updates of the states for segmentation and combination.
$\varepsilon = 0.1$	the small constant in Eq. 13.

A configuration of the fitness landscape shown in Fig. 1 is used for computer simulations. The lighter areas correspond to the higher fitness values and the darker areas to the lower. This is generated by the following equation for every sensor vector $\boldsymbol{s} = (d_0, d_1)$,

$$v(\boldsymbol{s}) = \sin(d_0 \cdot d_1) \tag{19}$$

A procedure of the simulation is as follows:

1. Place a learning agent with single state at randomly chosen coordinates.

2. Apply the proposed algorithm to the agent for 500 steps till it reaches the top of the fitness landscape where $v(\boldsymbol{s}) = 1.0$. We count this 1 trial.

3. Replace the agent at randomly chosen coordinates and repeat trials.

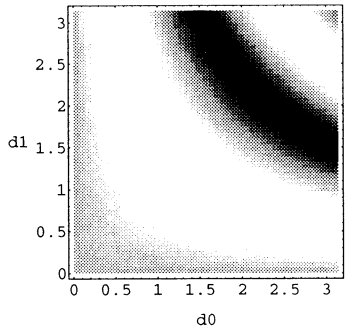


Fig. 1. The fitness landscape generated by $\sin(d_0 d_1)$. The lighter areas correspond to the higher fitness values and the darker areas to the lower.

Table 2. Q values for the major states.

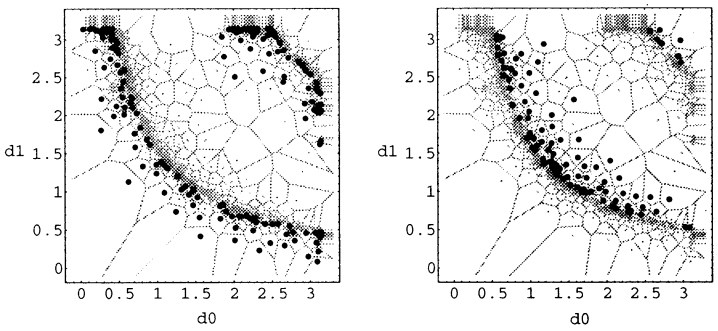
state number	actions				
	stop	up	down	right	left
80	1.86	2.86	0.86	2.86	0.86
169	1.39	0.36	2.86	0.59	2.44

After 100 trials, we obtained 258 states. In which, almost all representative sensor vectors are included in only 2 states. Figure 2 shows the representative sensor vectors included in the 2 major states. Gray solid lines in the diagrams represent boundaries of sub-regions covered by representative sensor vectors. Each dot in the sub-region corresponds to the representateive sensor vector $\boldsymbol{s}^x$. Dark dot means that the representative sensor vector belongs to the corresponding state. Their Q values are summarized in Table 2.

It is clear that the agent can reach the top of the fitness landscape by using the obtained state space and the Q values. Furthermore, fine segmentation of the sensor space is done by fine assignment of the representative sensor vectors around the top of the fitness landscape where are important to accomplish the task and they are combined into small number of states. The obtained state space is free from the "perceptual aliasing problem".

6 Conclusion

This paper presented a combination of Q-learning and adaptive construction of a state space based on reinforcement signals. Compared with other approaches,



(a) The state 80 (b) The state 169

Fig. 2. Representative sensor vectors included in the 2 major states 80 and 169 are represented in large dots. These states complement each other and cover almost the whole sensor space.

the proposed algorithm is realized in a simple way and is nearly free from determining parameters which are used for judgement in segmenting and combining state space. In the proposed algorithm, the judgement parameter γQ_{max} is calculated from the maximum reinforcement signal r_{max} and the discounting factor γ.

The proposed algorithm was applied to the problems navigating the mobile robot in continuous sensor space. Simulation results showed that it could generate a discrete state space efficiently to accomplish these tasks without *a priori* knowledge of environments. Resulting distribution of representative sensor vectors revealed the fitness landscape in a Voronoi tessellation.

Acknowledgement

This research was supported by "Methodology for Emergent Synthesis" Project (project No.96P00702), "Research for the Future" Program of the Japan Society for the Promotion of Science (JSPS).

References

[1] Whitehead, S. D. and Ballard, D. H, Learning to perceive and act by trial and error, *Machine Learning*, Vol.7, pp.45-83, 1991

[2] Murao, H. and Kitamura, S Q-Learning with Adaptive State Space Construction, *Proc. of the 6th European Workshop on Learning Robots (EWLR-6)*, pp.4-13, 1997

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Learning Control of Autonomous Robots Using Instance-Based Classifier Generator in the Continuous State Space

K. Kuroyama, M. M. Svinin, Y. Nakamura*, K. Ueda

Department of Mechanical Engineering,
Kobe University, Rokko-dai, Nada-ku, Kobe 657, Japan.
Email: kuroyama{svinin,nakamura,ueda}@mi-2.mech.kobe-u.ac.jp

Abstract

The paper deals with controlling autonomous mobile robots within the framework of the so-called instant-based reinforcement learning. As a base technique we use an Instant-Based Classifier Generator (IBCG) to produce action rules. Important feature of the proposed IBCG is its ability to work with continuous sensor data representation. Based on the state space analysis we define learning strategy and fix structure of the action rules. Then we outline mechanisms of action selection, rules reproduction, and credit assignment. Feasibility of the proposed IBCG is demonstrated under simulation and experiment.

1 Introduction

The paper continues our research on the use of the IBCG for reinforcement learning. Basically, the IBCG adopts two distinctive mechanisms to deal with wide state space: the instant-based generation of the action-rules; and introduction of generalized rules, which cover multiple regions in the state space. A robot with IBCG explores its environment while determining actions in response to the perceived states, and IBCG memorizes only the actually experienced state-action pair as an action rule. The generated rules compete with one another, and only the rules having high utility are survived. Utility estimates usefulness of the rule for achieving a task. In so doing, the survivor can acquire some useful behaviors.

In this paper, we undertake an attempt to modify and generalize the IBCG in such a way so that it will be possible to apply the concept to a wider class of dynamic systems, where the input sensory state is represented in the continuous form [1]. Note that in the original formulation [2] the IBCG was stated for the "discrete" description of the states and the actions. Extension of the IBCG to the continuous description is achieved mainly by introducing a continuous function to evaluate the matching rate of the rules, which operates under a continuous transformation of the sensory inputs

*Now with Technical Research Institute of Hitachi Zosen Corporation, 2-11-2, Funamachi, Taisho-ku, Osaka 551, Japan. Email:nakamura@mech.lab.hitachizosen.co.jp

This paper is organized as follows. First, we outline the "continuous" version of the IBCG in Section 2. Sections 3 and 4 show simulation and experimental results. Finally, conclusions are drawn in Section 5.

2 Instance-Based Classifier Generator

Let n_s be the number of sensors our robot is equipped with, and let $X = \{x_1, \ldots, x_{n_s}\}^{\mathrm{T}}$ be the sensory input of the robot. It is assumed that the readings x_i can take any real number, i.e., sensors are not restricted to be 1 bit ones. No explicit knowledge on the robot internal model is available in the control system, the vector X defines a particular point in the state space of the robot.

Let us conditionally define the following three regions in the space of all possible $X \in \Re^{n_s}$. S_{S} is a structurally observable region determined by the sensors' resolution only. It does not depend on the particular environment the robot is in. As such, it does not evolve in time. S_{V} is a region where the robot actually functions. It depends on the particular environment but does not depend on time. S_{A} is an actually observable region. It is defined by a set of states encountered by the robot in a particular environment in a certain moment of time. It is evident that $S_{\mathrm{A}}(t) \subset S_{\mathrm{V}} \subset S_{\mathrm{S}}$, $\max S_{\mathrm{A}}(t) = S_{\mathrm{V}}$. To ensure stable behavior acquisition, the size of S_{A} should shrink as the learning process progresses.

IBCG operates on a set of action rules, R. A rule $r \in R$ is described as $r =< C, u, f >$, where C is the classifier, u is the utility of the rule, and f is the credit assignment function. The classifier is an IF-THEN type rule. We define it as follows

$$C := \text{if } \| X - V \|_W \approx 0 \quad \text{then activate } a \in A, \quad (1)$$

where $V = \{v_1, \ldots, v_{n_s}\}^{\mathrm{T}}$ is a state vector related to and memorized in the rule r, and $A = \{a_1, \ldots, a_{n_a}\}$ is a set of the control actions. If V matches in some sense the current sensory input X, the rule r becomes active and—under some conditions to be discussed later—can trigger its action a.

The weight vector $W = \{w_1, \ldots, w_{n_s}\}^{\mathrm{T}}$ in (1) is used for comparing V and X. We set $w_i \in [0, 1]$ to fuzzyfy the Holland's "don't care" symbol # [3].

Thus, the closer w_i to zero, the less important the measurement of the ith sensor. From this viewpoint the set of action rules R can be decomposed as $R = R_{\text{IND}} \cup R_{\text{DEF}}$. Here, R_{IND} is the set of indefinite rules for which $W = 0$. These rules can be activated anywhere in S_{A} regardless of the current state X the robot is in. The other part of R is the set of definite rules, R_{DEF}. Definite rule memorizes a particular state V. It can be activated in a vicinity of V, and the vicinity is defined with the use of weight W.

Next, we define the module $M(a_i)$ as collection of rules encoding same action a_i. $M(a_i) = \{r_1, \ldots, r_{n_i}\}$, $i = 1, \ldots, n_a$, where $r_{n_i} \in R_{\text{IND}}$, $r_j \in R_{\text{DEF}}, j = 1, \ldots, n_i - 1$. Each module has always one indefinite rule. The number of rules of the rules n_i in every module $M(a_i)$ varies by reproduction and extinction.

2.1 Action Selection

To select an appropriate action, we start with defining the competing rules in every module $M(a_i)$. All the rules $r_j \in M(a_i)$ compare their own state vector $V^j = \{v_1^j, \ldots, v_{n_s}^j\}^{\text{T}}$ with current sensory state X. Using the weight vector $W^j = \{w_1^j, \ldots, w_{n_s}^j\}^{\text{T}}$, the matching rate m_j is evaluated as follows:

$$m_j = 1 - \frac{1}{n_s} \sum_{k=1}^{n_s} \frac{\mid x_k - v_k^j \mid}{d_k} w_k^j, \tag{2}$$

where d_k is the maximum difference of the observed value of the k-th sensor until now.

The rules passing a certain threshold θ_m form the set $R_i^{cut} = \{r_j \in M(a_i) \mid m_j > \theta_m\}$ from which the local competitors are to be selected. It is logical to relate the threshold θ_m with the rule's utility u_j. For example, we can set

$$\theta_m = \frac{c_1}{1 + \exp(\frac{c_3 - u_j}{c_2})} \tag{3}$$

where c_1, c_2 and c_3 are constants.

To proceed further, we sort out the rules in R_i^{cut} by a performance index τ_j, and select first n of them to form the set of local competitors

$$R_i^{com} = \{r_1, \ldots, r_n \mid r_j \in R_i^{cut}, \tau_1 \geq \tau_2 \ldots \geq \tau_n\}. \tag{4}$$

Thus, among all the rules with $\theta_m < m_j$, only the n rules with highest values of τ_j participate in the competition. As to the performance index τ_j it is reasonable to relate it to the rule's utility u_j, matching rate m_j, and the specifity λ_j. In our scheme, we set $\tau_j = \lambda_j m_j u_j$. Note that we treat the specifity λ_j as a measure of definiteness of the rule. As such, we can logically define it as the average value of all components of the vector W^j.

It is to be noted that even if no definite rules match the sensory input X, the indefinite rules can still participate in the competition anytime. This is because $W^j = 0$ and the matching rate $m_j = 1$ for all the indefinite rules regardless of the encountered state X.

Having defined the local competitors R_i^{com}, we combine all of them into one set $R^{com} = \cup_{i=1}^{n_a} R_i^{com}$. The final decision on the rule to trigger its action is made stochastically. The winner rule is selected from R^{com} with the selection probability of $r_j \in R^{com}$ given as

$$P(r_j) = \frac{\exp(\tau_j / T)}{\sum_{\forall r_k \in R^{com}} \exp(\tau_k / T)}. \tag{5}$$

2.2 Reproduction of Rules

The winner rule r_j reproduces/generates a new rule r_G except for the case when the action triggered by r_j leads to collision with the external objects and the rule is penalized. Details of the reproduction process are formalized as follows. If the winner is an indefinite rule $r_j \in R_{\text{IND}}$, the reproduced rule memorizes V^j from the parent's rule and all components of W^{G} for the new rule are set to 1. The utility of the new rule $u_{\text{G}} = u_j$, and the action code of the new rule is passed from its parent.

On the other hand, a definite rule $r_j \in R_{\text{DEF}}$ with high utility u_j can also win the competition even if its matching rate $m_j < 1$. In this case, the winner reproduces a new generalized rule r_{G} provided that its matching rate m_j is within a certain reproduction threshold θ_d, i.e., $m_j < \theta_d$. We relate θ_d with the rule's utility u_j by the following expression:

$$\theta_d = 1 - \frac{c_1}{1 + \exp(\frac{c_4 - u_j}{c_2})} \tag{6}$$

where c_1, c_2 and c_4 are constants, with c_1 and c_2 being taken from (3). The vectors V^{G} and W^{G} for the new rule are set as follows:

$$v_i^{\text{G}} = x_i, \quad w_i^{\text{G}} = 1 - \frac{\mid x_i - v_i^j \mid}{d_i}. \tag{7}$$

The utility of the new generalized rule is set as $u_{\text{G}} = m_j u_j$, and the action code of the new rule is passed from its parent r_j. Note that the new generalized rule built as above can match a wider state space, which includes the state of its parent.

2.3 Temporal Credit Assignment

The credit assignment mechanism is almost same as that for the "discrete" version of the IBCG [2]. The utility of the rule r_{A} that triggered an action is updated as follows.

$$u_{\text{A}} \leftarrow (1 - c_f) u_{\text{A}} + \gamma^n P \tag{8}$$

where P is the payoff given in particular states. The payoff is delivered directly only to the winner rules for which $\gamma > 0$ along the sequences of actions. The discount rate of the payoff, γ, is set to 1 when r_{A} triggers its action; and it is gradually discounted as the time

step n increases. A definite rule pays a cost at the rate c_f when it triggers its action, and it reduces its utility at the evaporation rate η when the robot reaches the goal state. The rules decreasing their utility up to a threshold, u^{min}, will be removed.

The current winner r_A hands a part of its utility, Δu, back to the previous winner, r'_A. Hence, the rule r'_A increases its utility: $u'_A \leftarrow u'_A + \Delta u$, where

$$\Delta u = \begin{cases} \delta u & \text{if } \delta u > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$$\delta u = \kappa \lambda_A (u_A - u'_A) \frac{1}{n_s} \sum_{k=1}^{n_s} w_k^A, \tag{10}$$

and $\kappa \in [0, 1]$. Note that the rule r_A does not decrease its utility as normally is in the conventional bucket brigades algorithms [4].

3 A Study Case

Here in this section, we want to verify how IBCG constructs its state space. The simplest environment, robot in a box with no external object, is chosen for the study case. The robot has two light sensors which grow large near the light source. The minimal and maximal sensor readings are set to 0 and 1000, respectively. The set of possible action is defined as follows: Move Forward, Move Backward, Turn Right, Turn Left, Rotate Right, and Rotate Left.

3.1 Task Description

The goal for the robot is to reach the light source. The time interval between the starting motion and reaching the goal is called episode. The robot is set at the starting point, with its heading orientation random at every episode. The episodes are updated when the robot reaches the goal, or when the number of produced actions reaches 500. The robot is rewarded only upon reaching the goal. Note that the robot does not have a priori knowledge of the environment and of the goal coordinates.

3.2 Simulation Results

The simulation shows that the robot can always reach the goal. It was expected and we do not make the point on this feature. What we want to inspect—and what is the most interesting point—is the state space decomposition, which is illustrated in Fig. 1. In this example the region S_V is the rectangle white space and evolution of the actually observed region S_A is shown by circles. The arrows indicate direction of transition of the state space. As the learning progresses, the IBCG produces the action rules more and more consistently and in less number. In episode No. 75, the rules cover S_V almost distinctly. Finally, in episode No. 100 the number of the action rules reaches

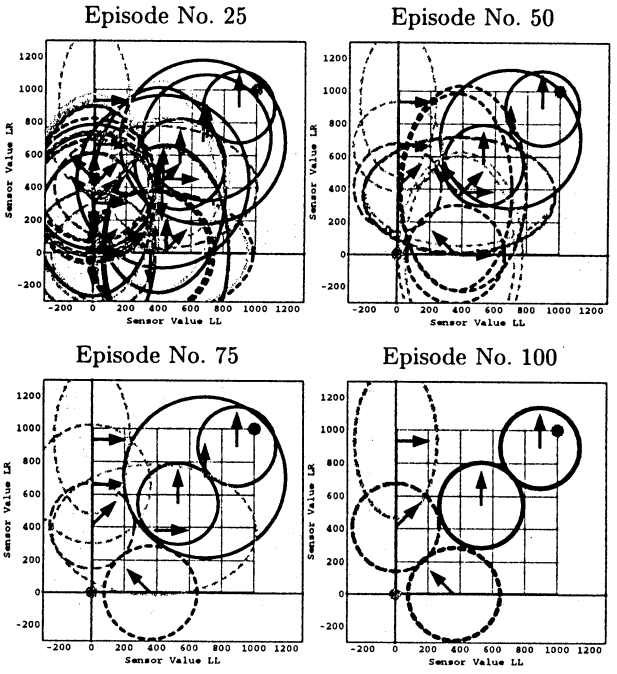


Figure 1: State space decomposition.

five. Consistent action rules attract less states, and the rules irrelevant for our task are evaporated. Thus, we conclude that the IBCG has ability of self organized construction of the state space.

4 Experimental Verification

The Khepera robot is used in the experiment. The robot has eight infrared sensors which can function as the light sensors and as the proximity sensors. Thus, we can set $n_s = 16$. The resolution of the light sensor mode is 512, and that of the proximity sensor mode is 1024. The differential drive mechanism allows 6 elementary actions listed in Section 3. The robot is controlled by and monitored from a work station SS4 through serial communication. Experimental environment is made from white plastic for the obstacles and from acrylic board for the ground.

4.1 Task Description

The experimental task is similar to that described in Section 3. However, the external environment is not empty. Again, the starting direction is changed randomly and the duration of one episode is limited to 500 steps. The robot is rewarded only at the goal and is punished for every collision. The direct payoff $P = 20$ is set for rewarding the rules, and $P = -0.03u_i$ is set for punishment. The total number of rules is restricted to 100. When reproduction exceeds the limit, the definite rule with the minimum utility is replaced.

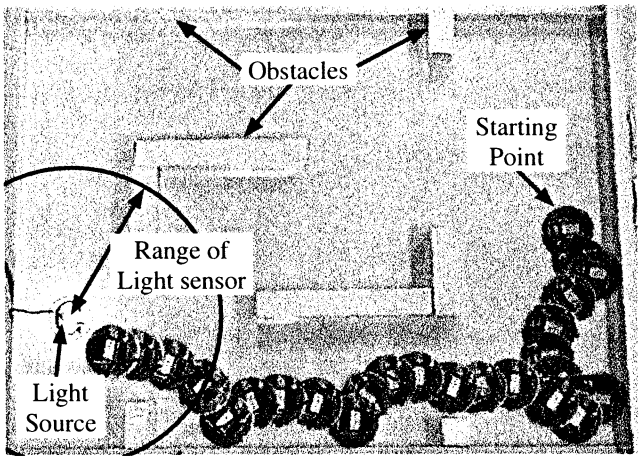


Figure 2: Robot behavior in episode No. 85.

The parameters are set follows: $c_1 = 0.9$, $c_2 = 50$, $c_3 = -100$, $c_4 = 120$, $T = 3.0$, $c_f = 0.01$, $\gamma = 0.9$, $\eta = 0.95$, and $n = 3$.

4.2 Experimental Results

Robot behavior in episode No. 85 is shown in Fig. 2. As can be seen, the robot has acquired the light-seeking and collision avoidance patterns. The number of collisions is plotted in Fig. 3. Here, we also indicate the episodes at which the robot obtains a reward. As can be seen, the number of collisions decreases as the frequency of reaching the light source increases. Fig. 4 plots the number of generations of the definite and generalized rules. As learning progresses, new rules are seldom generated and the total number of rules, also plotted in Fig. 4 for each episode, gradually decreases. This is because only particular definite rules, causing "useful" behavior, trigger their actions and increase their utility. Conversely, the "irrelevant" rules decrease their utility. At certain time their utility evaporates and they vanish. Thus, after some time the behavior acquired by the survivors becomes more stable, or better say dominant.

5 Summary

An attempt to extend IBCG to work with continuous state space representation has been undertaken in this paper. Based on the state space analysis, we have defined the learning strategy and fixed the data structure of the action rules. Basic procedures of the IBCG, such us action selection, rules reproduction and credit assignment have also been reformulated. Convergence features of the newly outlined IBCG has been tested under simulation for the simplest case of no-object environment. Feasibility of the behavior acquisition process has also been demonstrated under experiment where a navigation task for Khepera robot has been solved in real time.

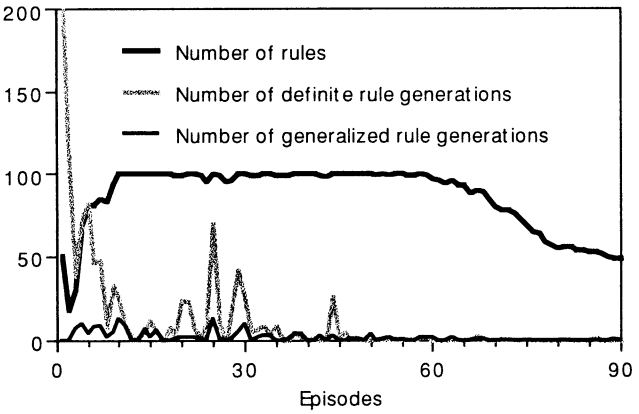


Figure 3: Rule reproduction history.

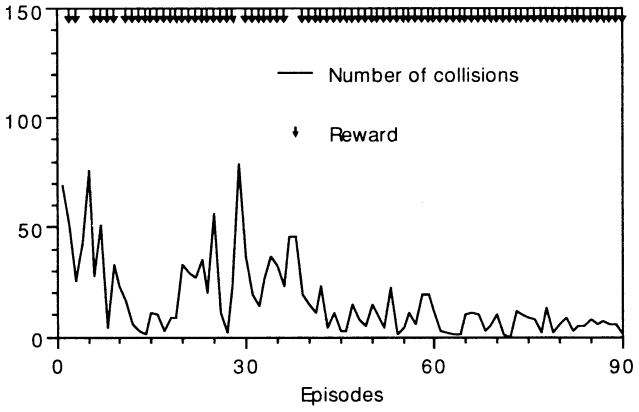


Figure 4: Collision history

Acknowledgments

This study was supported in part by the grant "Methodolgy for Emergent Synthesis", Project No. 96P00702, Program of Japan Society for the Promotion of science (JSPS) "Research for the Future".

References

[1] A.W. Moore and C.G. Atkeson, "The Parti-game Algorithm for Variable Resolution Reinforcement Learning in Multidimensional State Space," in *Machine Learning*, Vol. 21, Dec. 1995, pp. 1–36.

[2] Y. Nakamura, S. Ohnishi, K.Okhura, and K. Ueda, "Instance-Based Reinforcement Learning for Robot Path Finding in Continuous Space," in *Proc. IEEE Int. Conference on System, Man, and Cybernetics*, 1997, pp. 1228–1234.

[3] J.H. Holland, *Hidden Order. How Adaptation Builds Complexity*, New York, Addison Wesley, 1995.

[4] J.H. Holland, "Properties of the Bucket Brigade Algorithm," in *Proc. 1st Int. Conf. on Genetic Algorithms and Their Applications*, 1985, pp. 1–7.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

Intelligent Manufacturing by Simple Robots

Jari Vaario
NTT Human Interface Laboratories
Nippon Telegraph and Telephone Corp.
Hikarinooka 1-1, Yokosuka, 239 JAPAN

Kanji Ueda
Department of Mechanical Engineering,
Kobe University
Rokko Nada, Kobe 657, JAPAN

Abstract

In the paper we describe how a multiagent system consisting of simple self-similar reactive agents could be made to collaborate toward a common goal by self-organization principles. The multiagent system as a whole could be viewed as an intelligent and adaptive system. In the paper it is shown that with a relatively simple method a rather complex manufacturing tasks could be accomplished. The test case is to weld car chases according to the 'lineless' production concept. Robots and car chases on transporters can freely move in the manufacturing area and gather together in order to accomplish the task. The merits of this approach are analyzed and the method for simulating this kind of manufacturing activities is described. The preliminary results proves the approach feasible. The developed system, in addition to provide a tool to analyze such a system, could be used to control the activities in the actual realization of 'lineless' production.

1 Introduction

Most work with autonomous agents considers them to be intelligent at the assigned job. Even not targeting the human-like general intelligence, the agents should be capable of navigating, behaving, and learning autonomously in their environment. The tasks the agents are designed at are solvable by the individual performance. There are a few examples taken from real engineering area that require cooperation in order to solve a system level problem.

In this paper we propose an approach for multiagent systems where the actual performing agents are autonomous self-similar entities without any learning capability or any other intelligence. Although being stupid agents they are capable of operating with simple principles solving efficiently the system level problem. The method is based on the usage of dynamic attraction and repulsion fields that are generated, as well as sensed, autonomously and locally. With simple on-off timing of these fields the system level problem could be solved easily and with a very flexible and adaptive way. The timing is actually the coordination method between the multiagents.

The method has been applied on various problems varying from self-organization of shop-floor layout for bicycle assembly [4] to dynamic scheduling for the drilling of PCBs (Printed Circuit Boards) [1, 5]. In this paper the test case is a new taken from automobile industry. The problem is implementing a 'lineless' welding process of car chases. The basic idea is to avoid the line-wise production concept, and instead let the welding robots and car chases to move freely on the shop-floor and self-organize their activities. From the bird-view the welding hall would look like a chaotic system, but actually all movements are purposefully controlled by the attraction and repulsion fields.

The paper is divided into a discussion on the motivations of the 'lineless' production, following overview of the used method and the preliminary results. At the end there is a discussion of the pros and cons of the method.

2 Motivation for 'lineless' production

There are several reasons for 'lineless' production concept.

Concurrent manufacturing of several models: Current automobile manufactures have to have a great variety of models. It is not feasible to keep all models separated on their own production line. This creates a problem that the line must always proceed at the speed of the most complex model. The task of balancing lines is difficult and requires compromises from placing model to an otherwise better production line.

The 'lineless' concept would allow unlimited number of models coexisting in the same production location. The balancing problem would be much easier by providing simply the necessary amount of processing power for the required production.

Handling the malfunction cases: The line-wise production is depending on the proper function of each workstation. Even a single malfunction at a workstation could stall the whole line. This has been improved by making some backup workstations, but still there might be some critical workstations, that are too expensive to backup. This has let to improve the mean-time of malfunctions, but resulting in the equipments to become more expensive.

The 'lineless' concept would provide an easy backup system where in the case of malfunctions the performance of the whole system would only slightly slow down, but not to stall. This would also enable to use less expensive equipments because the malfunction is not such a critical factor anymore.

Smooth introduction of new models: The biggest nightmare in the line-wise production is the introduction of new models. The normal operation should be stopped, when a new model is taught for each workstation. It will take a lot of time and efforts before the performance of the line is back to the normal.

In the 'lineless' concept the old production could continue while new models are introduced. The robots could be taught on-line without effecting the performance of the current production.

Adjustment of the manufacturing speed: In the line-wise concept the line is designed to progress with a specified speed. It might be able to slow down, but the speed up is difficult. Thus adaptation to the demand is difficult.

In the 'lineless' concept the speed adjustment is a smooth operation by putting more resources on the shop-floor. As far as there is space to operate the speed-up should be linear.

2.1 Test Case Definition

In this paper the used example is the welding process of car chases. The car chases are placed on transporters. Each chase consists of six welding zones: left and right sides, and front, middle and rear zones. Each zone has a docking site for a welding robot with a number of welding spots of two kinds. For each type welding spots a corresponding robot is required. These robots are called 'A' and 'B' type robots. The welding order is not considered in this simulation, but in general increasing the constraints is not a problem. In addition, 'B' robot has a large body, that don't allow the neighbor dock sites to be used simultaneously.

The finished car chases are collected at the one end of hall, and empty transporters goes to pick up new tasks at the other end of hall. Between these two points, while the transporters are carrying the car chases, the robots should gather to do their job. If there is any malfunction, that transporter or robot should be avoided by the others. The problem is now how to select and control the robots to accomplish their job at each transporter carrying the car chases.

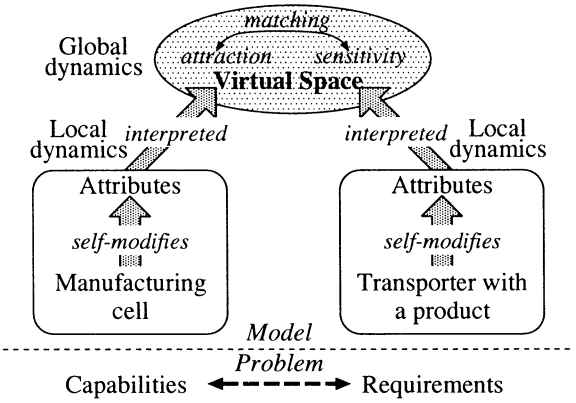


Figure 1: Attraction calculation in virtual space and its relation to manufacturing problems.

3 Method

We are using the self-organization principle inspired by the models of organization of biological cells [3].

3.1 The Principal Concept

Many manufacturing problems could be roughly described as a matching problem between requirements (of products to be processed) and capabilities (of machines to process). The method consists of virtual space simulation of attraction and repulsion fields and on the other hand the sensitivities to these fields. The attractions could be assigned to requirements (attracting machines to process them) or to capabilities (attracting products to be processed) (Fig.1). After this, the simulator repeatedly calculates the attraction forces and moves objects accordingly. The method would result in a stable state, if the attraction fields are stable. However, already by using only simple on-off timing interesting dynamics is achieved, and from the application point of view, a particular task could be accomplished.

3.2 Applying to Welding Robots

In this paper the problem is to attract welding robots to transporters with car chases. Each docking site at

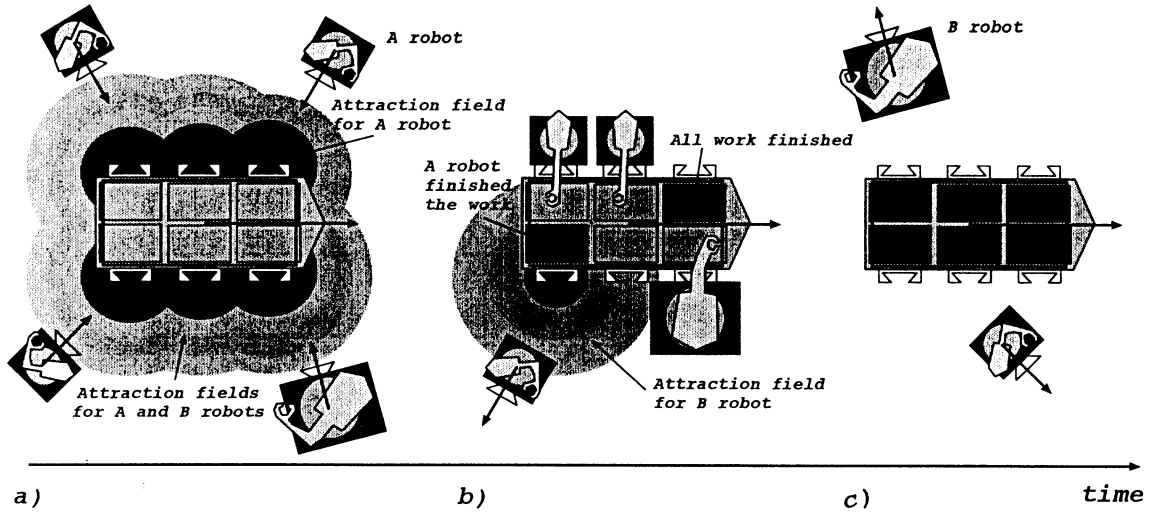


Figure 2: The timing of attractions during docking of robots. a) Transporter attracts at six locations 'A' and 'B' type robots. b) Left-front zone is finished, 'B' robot at the right-front zone blocks the attractions at the right-middle zone. Thus 'A' robot that is finished with right-rear zone cannot continue working here. c) All zones are finished and robots are free to work somewhere else.

the transporter has an attraction generator. Depending on the remaining welding spots at this zone, a corresponding attraction field is generated. If a robot docks at this site, the attraction fields are temporally turned off. When 'B' robot disables the neighbor sites, the attraction fields of these sites are turned off. When the robot has finished its job, the attraction fields are turned back on, providing there are still remaining welding to be done (Fig. 2).

Each robot has two locations where the attraction fields are measured according to the sensitivity specific to this robot (Fig. 3). These values are used to control the direction of robots, that otherwise move with a specific speed. If there are no attraction field, the robot will stop to circulate until new attraction field is detected.

The same principle is used to control the transporters. The transporters become sensitive for 'ready product' attraction when the car chase has been completed. Same way the empty transporters are attracted to the dispatching point of new car chases.

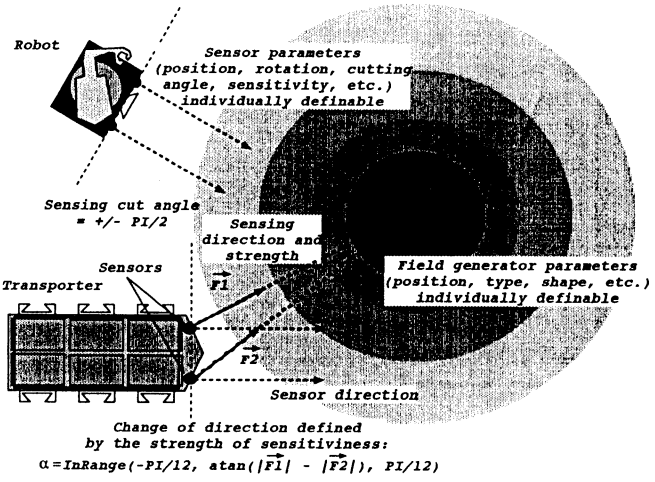


Figure 3: The movement control by two attraction measurements.

4 Results and Discussion

The preliminary results with the simulator proves the method being feasible. The gathering of robots can be controlled by the method (Fig. 4).

The further work includes experiments of the various control methods for the robots and transporters. The current method uses the most simplified control approach. This has a problem to head at the center of attraction fields until the sensitivity cut reduces the number of attraction fields to a remaining one. This causes unnecessary movements for the robots. A more sophisticated method is needed.

Currently no repulsion fields are used between transporters and robots. This causes some unrealis-

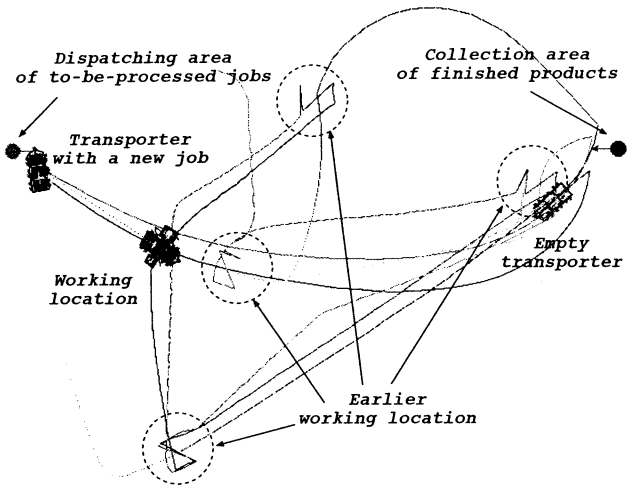


Figure 4: A snapshot of the simulator. The transporter and robot paths are shown by gradually vanishing lines.

tic situations with colliding objects. The dispatching point has a repulsion field, that is used to repel the transporters with a car chase further on the shop-floor. However, this is not yet adequate and other methods should be considered.

The method has some clear pros and cons. The main merit is the implementation of the 'lineless' production with all its advantages. Another merit is to simplify the control system of the shop-floor activities. With a simpler control system the whole production system becomes more adaptable.

The difficulties to realize the system at hardware level might be stated as one disadvantage of the method. For this we have thought a new way to control the future manufacturing systems. This is based on the idea of using virtual computation for all shop-floor activities and then using a real time wireless control. The human participation on such a system happens through the virtual reality interface [2].

5 Conclusion

In the paper attraction fields based simulation of multiagent system for implementing a 'lineless' production was described. The results proved the method feasible to control shop-floor activities of welding car chases in a 'lineless' manner.

For the future work the analysis of the efficiency remains. However, as it was pointed out, the efficiency is not the only evaluation criteria. The productivity should be analyzed in the long run with several model changes, adaptation to demand, and, in general, by the flexibility of the whole production. At the moment the method seems very promising to deal with the ever increasing complexity of production activities.

Acknowledgment

This work is part of IMS Japan Next Generation Manufacturing Systems (NGMS) program (BMS group). Especially thanks are due to Mr. Watanabe, Honda Engineers Ltd., for providing the test case knowledge. The work has been also supported in part by the Grant-in-Aid for Scientific Research on Priority Area of the Ministry of Education, Science, Sports and Culture of Japan; by Nissan Foundation; and by "Methodology for Emergent Synthesis" Project (project number 96P00702) "Research for the Future" Program of the Japan Society for the Promotion of Science (JSPS).

References

[1] Kanji Ueda, Jari Vaario, and Kazuhiro Ohkura. Modeling of biological manufacturing systems for dynamic reconfiguration. volume 46/1, pages 343–346. Annals of the CIRP, 1997.

[2] Jari Vaario, Nobutada Fujii, Dirk Scheffter, Michael Mezger, and Kanji Ueda. Factory animation by self-organization principles. In *Proceedings International Conference on Virtual Systems and MultiMedia (VSMM'97), September 10-13, 1997, Geneve, Switzerland*, pages 235–242. IEEE Computer Society, 1997.

[3] Jari Vaario and Katsunori Shimohara. On formation of structures. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Advances in Artificial Life*, pages 421–435. Springer, 1995.

[4] Jari Vaario and Kanji Ueda. Biological concept of self-organization for dynamic shop-floor configuration. In H. Tamura, editor, *Advances in Production Management Systems – Perspectives and Future Challanges*. Chapman & Hall, 1997. (in press).

[5] Jari Vaario and Kanji Ueda. An emergent modeling method for dynamic scheduling. pages 187–198. The Second World Congress on Intelligent Manufacturing Processes & Systems, June 10-13, 1997, Budapest, Hungary, 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Involvement of cerebellar synaptic plasticity in adaptive control of locomotion of cat.

Dai Yanagihara
Laboratory for Memory and Learning, Brain Science Institute, The Institute of Physical and Chemical Research (RIKEN), Hirosawa 2-1, Wako-shi, Saitama 351-01, Japan.

Abstract

We have studied the controlled locomotion of the decerebrate cat on a treadmill with three belts, one mounted under the left forelimb, another under the left hindlimb and third under the right forelimb and right hindlimb, the speed of which may be independently changed. Exposure of one forelimb to a belt velocity higher than that of the other limbs during locomotion provides an effective means for testing how locomotor systems adapt in response to changing external conditions. During the first 1-50 steps of perturbed locomotion, temporal coordination of limbs was disturbed, and the cycle durations were markedly unstable. However, within 100-200 steps the cycle duration stabilized, and the newly coordinated pattern was performed. Recent studies of molecular mechanisms of the cerebellar synaptic plasticity revealed that some pharmacological reagents inhibit signal transduction processes for the induction of synaptic plasticity, without interfering with the normal synaptic transmission. When a scavenger of nitric oxide or an inhibitor of nitric oxide synthase was applied to the vermal part of the cerebellum, adaptation to the perturbed locomotion no longer occurred, although the normal unperturbed locomotion was not affected. We conclude that the cerebellar synaptic plasticity is involved in the adaptive control of limb movements during locomotion. Our findings support the view that the cerebellum adapts interlimb coordination to the environment by utilizing its synaptic plasticity.

Introduction

To perform smooth locomotion, animals coordinate not only parts of a limb, but also different limbs (interlimb coordination). The dynamics of coordination is modified according to changes of external conditions due to obstacles or other unexpected perturbations. A formation of the dynamics of the coordinated pattern in space-time is viewed as an important process of motor learning and memory[4]. How is the dynamics of locomotion pattern adjusted to the changes of the external conditions by motor learning? And, during those motor learning, what happens in the brain? I will present evidence that adaptive control of locomotion is invloved in the cerebellar synaptic plasticity.

Adaptive changes in interlimb coordination during perturbed locomotion in cat

Adaptive interlimb coordination was studied during perturbed locomotion in experimental chronically decerebrate cats. Perturbations were applied to the stance phases of the left forelimb using a newly developed treadmill which consisted of three compartments, one each for the left forelimb, the left hindlimb and both right limbs (Figure 1). Decerebrate cats were used in this study. There are two major reasons why decerebrate cats were prefered. One is that once the treadmill belt had begun to move, they automatically performed stable locomotion, whereas intact cats rarely exhibited. The other is that decerebrate cats have the advantage of providing greater possibilities for analytical studies, because their lower brainstem-cerebellum-spinal system is isolated from complications that may arise from the cerebrum and upper brainstem. Yet, it is generally agreed that essential locomotor mechanisms are retained in decerebrate cats. The cats were acquired to walk in two different conditions. One is unperturbed locomotion. During unperturbed locomotion, treadmill belts for all limbs were driven at 36 cm/s. Another is perturbed locomotion. During perturbed locomotion, the left forelimb alone was exposed to a belt velocity of 61 cm/s, and the other belts were kept in 36 cm/s. During the first 1-50 perturbed steps, the step cycles of both forelimbs showed marked fluctuations; thereafter, the animals achieved stable locomotion by slightly shortening step cycle durations, and also by adjusting durations of bisupport phases asymmetrically in the left and right forelimbs[8]. From a nonlinear dynamical view, adaptation and learning are pattern formation processes rooted deeply in the concept of stability[4]. Scatter diagrams in Figure 1 easily show the stability or instability of locomotion. Ordinate is duration of bisupport phases. Abscissa is duration of step cycles. During the bisupport phase, both forelimbs come into contact with the ground to switch the weight-bearing limb from right forelimb to left forelimb and vice versa. The reason why we analyse the duration of bisupport phases is, firstly, the phase relationship between both forelimbs is obvious at a glance, and secondly, bisupport phase is important for interlimb coordination and smooth and stable execution of locomotion. Squares represent step cycles for the left forelimb versus bisupport phase from left forelimb touch-down to right forelimb lift-off, and circles represent step cycles for the right forelimb versus bisupport phase from right forelimb touch-down to left forelimb lift-off. Figure a illustrates the pattern of unperturbed locomotion. Stability of the locomotion is represented by convergence of the plotted points within a small area for both forelimbs. In subsequent sessions, the left forelimb alone was exposed to a belt velocity of 61 cm/sec, and other belts were kept in 36 cm/sec. In figure b, 1st trial of perturbed

locomotion is characterized by a large variation in both step cycle and bisupport phases. In figure c, in the second trial of perturbed locomotion, plotted points are still scattered. In figure d, in the third trial of perturbed locomotion, stable locomotion was achieved with step cycles slightly shortened, and with bisupport phases asymmetrical.

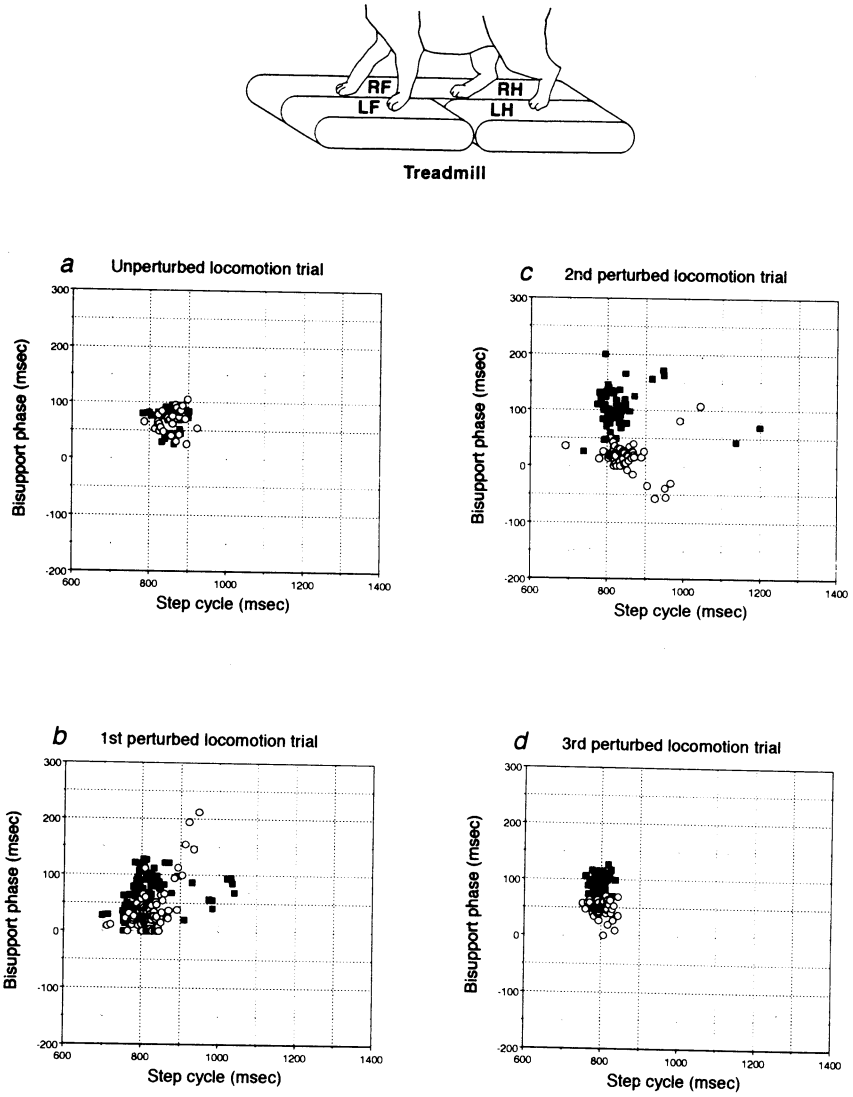


Figure 1. Adaptive changes in interlimb coordination during perturbed locomotion.

The site of locomotive adaptation presently demonstrated in decerebrate cats is in the brainstem and/or cerebellum. The spinal cord may be excluded, since the bisupport mechanism is seriously impaired in spinal cats[1]. Since the cerebellum generally regulates coordinated movements of locomotion and posture, and since lesions of the cerebellum induce disturbances of interlimb coordination and body equilibrium[2], it is suggested that the presently demonstrated locomotive adaptation involves the cerebellum rather than the spinal stepping generator and reflex systems.

Functional linkages between adaptive changes in interlimb coordination and the cerebellar synaptic plasticity

In the cerebellum, long-term depression (LTD) is persistent reduction of transmission efficacy at synapses from parallel fibers to Purkinje cells, which occurs when the parallel fibers are activated in conjunction with a climbing fiber converging onto the same Purkinje cell[3]. LTD has been thought to be a cellular basis of motor learning[3]. Recently, it was demonstrated that nitric oxide (NO) plays an important role in signal transduction processes underlying LTD[7]. Manipulations of NO concentration or its synthesis in cerebellar tissues therefore provide a means for investigating roles of NO in cerebellar functions at both cellular and behavioral levels. In our study[10], we investigated the involvement of NO in the adaptation of locomotion by direct localized application of either an inhibitor of NO synthase or a scavenger of NO to the cerebellum, the adaptation to perturbed locomotion was abolished when NO was deprived from vermal lobule V in cerebellum. However, NO deprivation does not affect normal, unperturbed locomotion. From these results, we concluded that NO-dependent cerebellar function is critically involved in adaptive control of locomotion.

The cerebellum receives information through the spinocerebellar pathways about the ongoing activities both in the spinal stepping generator and at the somatosensory receptors during locomotion (see Figure 2)[2]. This information is conveyed by mossy fiber afferents to Purkinje cells via granule cells and their axons, i.e., parallel fibers. Purkinje cells transform the mossy fiber input signals to output signals that in turn modulate activities in descending tract neurons involved in locomotion. On the other hand, Purkinje cells in vermal lobule V receive enhanced climbing fiber signals during perturbed locomotion[9]. These climbing fiber signals are expected to induce LTD at parallel fiber synapses mediating "ongoing locomotion"-related mossy fiber signals. If the mossy fiber-to-Purkinje cell signal transformation is modified by LTD at parallel fiber-Purkinje cell synapses, this will lead to changes in activities in the descending tract neurons, and thereby to adaptation in locomotion. Since adaptation or conditioning is a simple form of learning, our results, considered together with the results of previous studies[5, 6] can be expanded to motor learning in general.

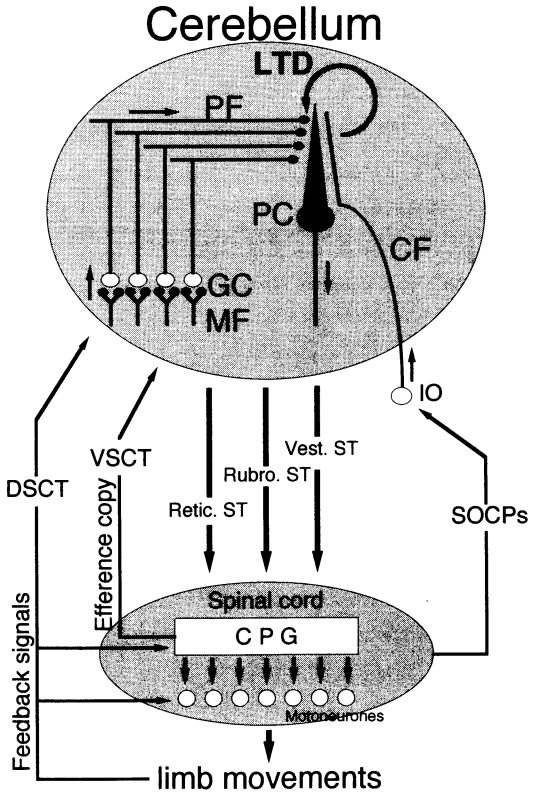


Figure 2. Systems view of the role of the cerebellum in locomotion.

References

1. Forssberg, H., Grillner, S. and Rossignol, S. Phasic gain control of reflexes from the dorsum of the paw during spinal locomotion. Brain Res., 132(1977)121-139.
2. Ito, M. The Cerebellum and Neural Control. Raven Press, New York, (1984).
3. Ito, M. Long-term depression. Annu. Rev. Neurosci., 12(1989)85-102.
4. Kelso, J. A. S. Dynamic Patterns. The MIT Press, Cambridge, (1995).
5. Li, J., Smith, S. S. and McElligott, J. G. Cerebellar nitric oxide is necessary for vestibulo-ocular reflex adaptation, a sensorimotor model of learning. J. Neurophysiol., 74(1995)489-494.
6. Nagao, S. and Ito, M. Subdural application of hemoglobin to the cerebellum blocks vestibuloocular reflex adaptation. NeuroReport, 2(1991)193-196.
7. Schuman, E. M. and Madison, D. V. Nitric oxide and synaptic function. Annu. Rev. Neurosci., 17(1994)153-183.
8. Yanagihara, D., Udo, M., Kondo, I. and Yoshida, T. A new learning paradigm: adaptive changes in interlimb coordination during perturbed locomotion in decerebrate cats. Neurosci. Res., 18(1993)241-244.
9. Yanagihara, D. and Udo, M. Climbing fiber responses in cerebellar vermal Purkinje cells during perturbed locomotion in decerebrate cats. Neurosci. Res., 19(1994)245-248.
10. Yanagihara, D. and kondo, I. Nitric oxide plays a key role in adaptive control of locomotion in cat. Proc. Natl. Acad. Sci. USA, 93(1996)13292-13297.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Adaptive Locomotion to Periodic Perturbation

Adaptation Mechanism with Coupling of Oscillator and Link Dynamics

Satoshi ITO[1)] Hideo YUASA[2),1)] Zhi-wei LUO[1)]
Masami ITO[1)] Dai YANAGIHARA[3)]

[1)] Lab. for Bio-Mimetic Control Systems, Bio-Mimetic Control Research Center, RIKEN
[2)] Dept. of Electronic-Mechanical Engineering, School of Engineering, Nagoya Univ.
[3)] Lab. for Memory and Learning, Brain Science Institute, RIKEN

Abstract

Quadrupeds can acquire new gait patterns with respect to the environmental changes. Yanagihara et al. have demonstrated this adaptability by experiments of a decerebrate cat. These experiments indicate that quadrupeds gradually adapt to their environment by repeating locomotion in the steady environment, and that the acquired gait pattern is persistently memorized after the locomotion.

Our research aims at formulating the mathematical model of these cats' behaviors and constructing a quadrupedal walking robot to realize such an adaptive behavior. By now, we have proposed a mathematical description for adaptation at the level of gait pattern generation using the neural oscillators. In this paper, we extend it to take into account the limb dynamics. We study on how to design the interaction of the oscillator and limb dynamics

key words : quadrupedal locomotion, gait pattern, adaptation, periodic perturbation, coupled oscillators, limb dynamics

1 Introduction

Adaptation in quadrupedal locomotion can be observed in the experiment of decerebrate cats. As a result of adaptation, the gait pattern changes to another steady one. Yanagihara et al. [1][2] have demonstrated this adaptation process using a specially designed treadmill (Fig. 1). Their treadmill consists of three moving belts. Each belt can be driven independently with the different speed. Firstly, they drove all the treadmill belts with the same low speed. After some training, the decerebrate cat naturally showed the 'walk' gait as the intact cats. Next, they changed the speed of one treadmill belt (on which the cat places its left forelimb) 1.7 times faster than the others, and observed the cat's behavior through three trials. Each trial includes 60 to 100 steps, and the interval between each trial was about a few minutes. At the first trial, the gait pattern of the cat did not converge to the steady one. But at the second trial, it converged to a new gait pattern after some steps. At the last trial, the cat showed the acquired new gait pattern from the beginning of the locomotion experiment.

In order to examine the stability of the gait pattern, it is efficient to impulsively disturb the locomotion. However, under the environment of the non-regular perturbation, a new gait pattern can not be acquired: The gait pattern soon goes back to the original one due to its stability. In the experiments by Yanagihara et al., the cat locomotion is perturbed whenever the cat places its left forelimb on the treadmill belt. It means that the locomotion is perturbed periodically. It is under such periodic perturbation that the new gait pattern was generated.

In this paper, we aim at formulating the mathematical model of these cats' behaviors for constructing a quadrupedal walking robot. We have already proposed a mathematical description for adaptation using the neural oscillators [3]. Here, we extend it to take into account the limb dynamics. Then, the coupling

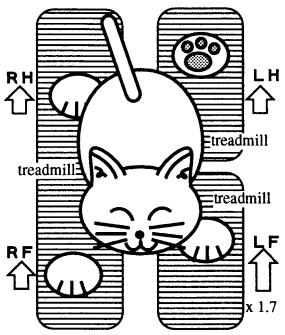
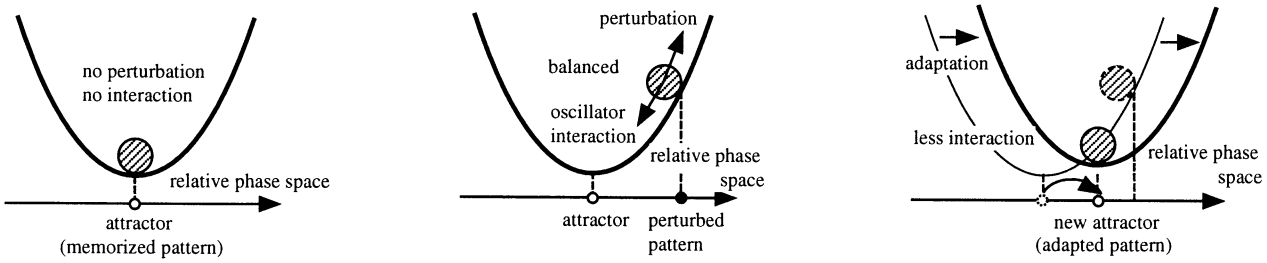


Figure 1: Experiment with a decerebrate cat.



(a) Normal locomotion.

(b) Perturbed locomotion without adaptation.

(c) Perturbed locomotion with adaptation.

Figure 2: Mechanism of adaptation in perturbed locomotion.

of oscillator and limb dynamics becomes an important problem.

2 Adaptation mechanism

The experimental results by Yanagihara et al. provides us the following ideas: Firstly, the cat had memorized the walk gait for the normal environment. When the speed of the treadmill belt changed, this walk gait was not suitable to the changed environment. Accordingly, the cat adjusted the memorized gait pattern into a new one by repeating locomotion. The fact that the new gait pattern emerged at the beginning of the third trial of experiment implies the evidence that the cat memorized the new locomotion pattern.

We formulate this adaptation process as follows: Firstly, when we think about the gait pattern, we focus on the relative phases of four limbs' movements. Although each limb movement is dynamic and periodic, the relative phases will be constant if the gait pattern becomes steady. Then, we can regard the gait pattern as the fixed point in relative phase space. Using the potential function for simplicity, the steady gait pattern corresponds to the minimum point of the potential function in the relative phase space.

Initially, the cat walked on the treadmill with the 'walk' gait. Accordingly, the cat memorized the walk gait for the normal conditions (i.e., all the treadmill speed is the same). So, we can firstly represent the walk gait as the minimum point of the potential function. From the characteristic of potential function, the memorized motion pattern is stable, in other words, it is an attractor in relative phase space. Even though locomotion is perturbed impulsively, its pattern will soon comes back to the original one. However, if perturbation becomes periodic, the next perturbation will disturb the motion pattern again before the motion pattern completely comes back. Consequently, the perturbation and gradient force of potential will balance each other where the relative phase is different from the memorized one.

Note that the gradient force is a force which tends to realize the memorized motion pattern. Even though such a gradient force is always working, this gradient force fails to produce the memorized pattern. It means that the memorized motion pattern is not suitable to the environment. Therefore, the memorized pattern should be adjusted so that the gradient force can achieve it. In other words, the attractor in motion pattern should be adjusted to decrease the gradient force. We consider this adjustment process as 'adaptation'.

3 Formulation of adaptive behavior

3.1 Assumptions

Our final goal is to realize such an adaptive behavior of the decerebrate cat by a quadrupedal walking robot. In the previous paper, we have formulated the adaptation mechanism with mathematical expression [3]. In this study, we extend it to include the limb dynamics. Here, the coupling of the oscillator and limb dynamics, i.e., how to design the interaction of two dynamics, becomes an important problem. Figure 3 is a sketch for our quadruped model examined in this study. We summarize our important assumptions for this problem:

- The walking motion is restricted within the sagittal plane.
- The balance of the body is not considered, i.e., the body is supported at the fixed position on the treadmill.

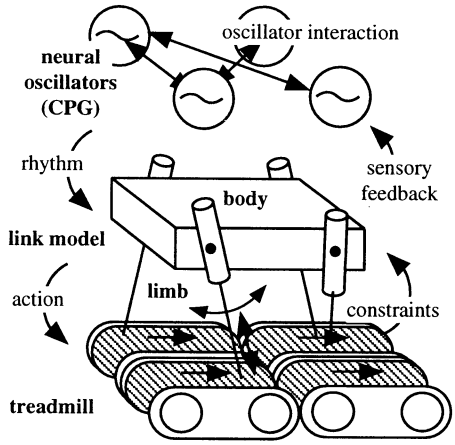


Figure 3: A model of adaptive behavior in the cat locomotion.

- Four limbs have the same structures and perform 2 DOF of the motion: the rotation of the joint between body and limb (α_i), and the contraction/extension of the limb (r_i).
- The step length is constant ($\alpha_{sp\_s} \leq \alpha_i \leq \alpha_{sp\_f}$) for each limb with no relation to the walking speed.
- The ground reacting force from the treadmill is detectable.

The CPG (Central Pattern Generator) is a spinal neural oscillators responsible for rhythm generation of locomotion. In our model, each limb motion is mainly controlled by the neural oscillator model assigned to the limb.

We should define both the control law for the limb and the oscillator dynamics, separately in the swing phase and the support phase. This is because that the nature of the limb movement is essentially different in these two phases. In addition, we must define the dynamics of adaptation process to achieve an adaptive behavior. Therefore, three dynamics are important: the support phase dynamics, the swing phase dynamics and the adaptation dynamics.

3.2 Preparations

We express the phase of oscillator by the variable $\theta_i (i = 0, 1, 2, 3)$. The phase space becomes the one dimensional torus space. We split the phase space to two regions: the region satisfying $\cos\theta_i > \cos\pi\beta = \gamma$ and the remainder, as shown in Fig. 4. The constant parameter β, which denote the ration of the support phase in the one locomotion step, can control the duty factor.

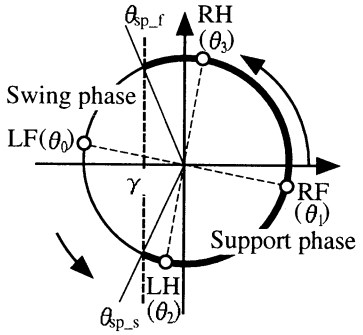


Figure 4: Stance phase and swing phase. In this case, only the LF (θ_0) is in the swing phase and the others (LH, RF, RH) are in the stance phase.

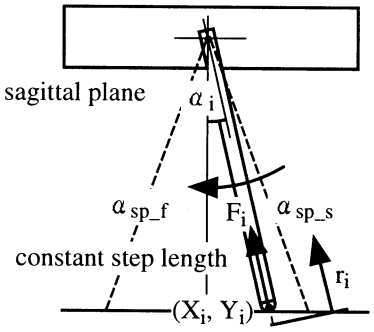


Figure 5: The structure of the limb. Each limb has 2 DOF: rotation (α_i) and contraction/extension (r_i). F_i is a ground reacting force.

The limb has two link structure attached the body, as shown in Fig. 5. We express its dynamics for it as

$$M_l(q_i)\ddot{q}_i + H_l(q_i, \dot{q}_i) + G_l(q_i) = \tau_i + J^T(q_i)F_i \quad (1)$$

where i represent the number of the limb (i=0,1,2,3), $q_i = [\alpha_i, r_i]^T$ is a joint angle (superscript T denotes the transpose), τ_i is a joint torque, M_l is an inertia matrix, H_l denotes the coriolis and centrifugal force, G_l denotes the gravitational force, F_i is a reacting force from treadmill, and J is a Jacobian matrix.

3.3 Support phase dynamics

In the support phase, the limbs support the body, and their movements are constrained by the treadmill. The limb has to generate the large force along the gravitational direction while follow the treadmill movement along the horizontal direction. Accordingly, we apply the impedance control such that high-impedance in the gravitational direction and low-impedance in

the horizontal direction. We can compute the joint torque τ_i so that the response to the ground reacting force F_i becomes

$$M_d\ddot{X}_i + D_d(\dot{X}_i - \dot{X}_{di}) + K_d(X_i - X_{di}) = F_i \quad (2)$$

where X_i is the position of the toe in the absolute coordinate, X_{di} is its desired position, M_d, D_d and K_d are matrices denoting the desired inertia, viscosity and elasticity.

In order for the CPG to generate the locomotion rhythm appropriately with respect to the environment, the position or moving velocity of the limb should be fed back to CPG. Therefore, we change the oscillator phase in the support phase according to the limb movement. We assumed that the step length of the limb movement is constant ($\alpha_{sp\_s} \leq \alpha_i \leq \alpha_{sp\_f}$) as well as the range for the support phase in the oscillator phase space ($\theta_{sp\_s} \leq \theta_i \leq \theta_{sp\_f}$), as shown in Fig. 4 and Fig. 5. It provides the initial and final condition to the relation between the phases of the oscillator and limb movement. Under this condition, we compose some continuous mapping from the limb position to oscillator phase,

$$\theta_i = P(\alpha_i), \quad (3)$$

such that,

$$\theta_{sp\_s} = P(\alpha_{sp\_s}),\ \theta_{sp\_f} = P(\alpha_{sp\_f}). \quad (4)$$

Here, we focused on the joint angle α_i between body and limb, when considering the phase of the periodic limb movement (Fig. 5). Using the above equation, we define the oscillator dynamics in the support phase,

$$\dot{\theta}_i = \frac{\partial P(\alpha_i)}{\partial \alpha_i}\dot{\alpha}_i. \quad (5)$$

3.4 Swing phase dynamics

In the swing phase, the limb is released from the constraint of the treadmill and can move freely. The timing of limb movement, in other words, the relative phases of limb movement, can be modified in this swing phase. So, in the swing phase, CPG governs the dynamics of the limb movement. We can calculate the desired value of α_i, i.e., α_{di} from the oscillator phase θ_i. It is because that, as is the case of the support phase, we have already known the initial and final condition for the relation between θ_i and α_i. we control the limb movement so that the limb movement follows this desired value α_{di}.

The CPG, on the other hand, must produce the oscillation pattern, i.e., phasic relation of four oscillators, which should be memorized as the preferred gait pattern. Using the method proposed by Yuasa and Ito [4], we can control the relative phases $\phi_0 = \theta_1 - \theta_0$, $\phi_1 = \theta_2 - \theta_1$ and $\phi_2 = \theta_3 - \theta_0$, to their desired value D_0, D_1 and D_2 by the potential function defined in the relative phase space,

$$V = \frac{1}{2}\tau_\theta[(\phi_0 + \phi_1 - D_0 - D_1)^2 + (-\phi_2 + D_2)^2 + (-\phi_0 + \phi_2 + D_0 - D_2)^2 + (-\phi_1 + D_1)^2]. \quad (6)$$

According to this potential function, the dynamics of the relative phase become the gradient system,

$$\dot{\phi}_j = -\frac{\partial V}{\partial \phi_j}, \quad (j = 0, 1, 2). \quad (7)$$

Further calculation provides the following oscillator dynamics:

$$\dot{\theta}_i = \omega_i + f_i, \quad (i = 0, 1, 2, 3). \quad (8)$$

Here ω_i $(i = 0, 1, 2, 3)$ denotes natural frequency of neural oscillator, and f_i $(i = 0, 1, 2, 3)$ are oscillator interaction given as

$$f_0 = \tau_\theta(\theta_1 + \theta_3 - 2\theta_0 - D_0 - D_1), \quad (9)$$

$$f_1 = \tau_\theta(\theta_0 + \theta_2 - 2\theta_1 + D_0 - D_2), \quad (10)$$

$$f_2 = \tau_\theta(\theta_1 - \theta_2 + D_2), \quad (11)$$

$$f_3 = \tau_\theta(\theta_0 - \theta_3 + D_1). \quad (12)$$

Note that the memorized pattern is described not only by the desired relative phases D_0, D_1 and D_2 but also the natural frequency of neural oscillator ω_i.

3.5 Adaptation dynamics

At the normal locomotion, CPG produces the walk gait using the potential function eq.(6) for the appropriate $D_j (j = 0, 1, 2)$ and $\omega_i, (i = 0, 1, 2, 3)$. Then, from the characteristics of the potential function, the oscillator interaction does not work, because the walk gait is a minimum point of potential function. However, if the treadmill speed increase, the perturbation push away the motion pattern from a minimum point, and the steady pattern emerges at the different point from the minimum point. Consequently, oscillator interaction always works to realize the memorized pattern.

Then, as mentioned in the previous section, we adjust the memorized motion pattern so that the oscillator interaction decrease. We prepare two type of evaluation function,

$$F_i = \int_T f_i dt, \quad (13)$$

and

$$V_d = \int_T \sum_{i=0}^{3} \left\{ \frac{1}{2\tau_\theta} f_i^2 \right\} dt. \quad (14)$$

To decrease eq. (13), we can derive the dynamics

$$\omega_i^{(n+1)} = \omega_i^{(n)} + \tau_\omega \int_T f_i dt \quad (i = 0, 1, 2, 3), \quad (15)$$

while for the eq. (14), we can obtain the following equations:

$$D_0^{(n+1)} = D_0^{(n)} + \tau_D \int_T (f_0 - f_1) dt, \quad (16)$$

$$D_1^{(n+1)} = D_1^{(n)} + \tau_D \int_T (f_0 - f_3) dt, \quad (17)$$

$$D_2^{(n+1)} = D_2^{(n)} + \tau_D \int_T (f_1 - f_2) dt, \quad (18)$$

where n denotes the number of step cycles, T is the duration of one step cycle, and τ_ω and τ_D are parameters that influence the convergence of ω_i and D_j, respectively.

These parameters should be adjusted slowly, because we can not adjust them is until we know the evaluation for these parameters. Therefore we define the adaptation dynamics discretely at every locomotion cycle. In addition, we set the time constant τ_D and τ_ω smaller enough than that of oscillator dynamics, i.e., τ_θ.

4 Simulation

We executed the computer simulation according to the defined dynamics. Firstly, we confirm that the CPG certainly produce the walk gait at the normal environment. The gait pattern obtained in this simulation is shown in Fig. 6 (a).

Next, we increased the parameter corresponding to the treadmill speed for the left forelimb, 1.7 times faster than the others. Then, the effect of treadmill movement was reflected to the oscillator dynamics through each limb motion, and the adaptation dynamics played a role of adjusting the memorized gait pattern. The gait pattern finally obtained in this simulation is shown in Fig. 6 (b). This pattern is similar to the experimental result with a decerebrate cat.

5 Conclusion

In this study, we extended our mathematical model of adaptive behavior in the cat locomotion to include the actual limb dynamics. The similar results to the experiment can be obtained from the computer simulations.

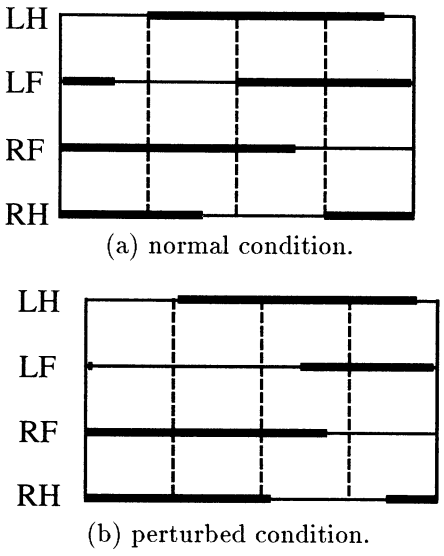


Figure 6: The gait pattern results of the computer simulations.

As the future works, we will practically examine a quadrupedal walking robots on the treadmill, and make the robot to move adaptively on the treadmill as a real cat.

References

[1] Yanagihara D, Udo M, Kondo I, Yoshida T, "A new learning paradigm: adaptive changes in interlimb coordination during perturbed locomotion in decerebrate cats", *Neuroscience Research*, Vol. 18, pp. 241-244, 1993

[2] Yanagihara D and Kondo I, "Nitric oxide plays a key role in adaptive control of locomotion in cat ", *Pro. Natl. Acad. Sci. USA*, vol. 93, pp. 13292-13297, 1996

[3] Ito S, Yuasa H, Luo ZW, Ito M and Yanagihara D, "A mathematical model of adaptation in rhythmic motion to environmental changes", *Proceeding of IEEE ICSMC* Vol. 1, pp. 275-280, 1997

[4] Yuasa H and Ito M, "Coordination of Many Oscillators and Generation of Locomotory Patterns," *Biological Cybernetics*, Vol. 63, pp. 177-184, 1990

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

The optimal gait pattern in hexapods based on energetic efficiency

J. Nishii
Lab. for Neural Modeling
RIKEN
2-1 Hirosawa, Wako
Saitama 351-01

K. Ogawa
Production Engineering Department
Nissan-motor co., ltd
1-1 Enoki, Musashi-murayama
Tokyo 208

R. Suzuki
Matto-Laboratory
Kanazawa Institute of Technology
3-1 Yakkaho, Matto
Ishikawa 924

Abstract

The gait transition during locomotion in legged animals is an attractive phenomenon from the point of view of both robotics and biology. In this paper we will show by means of a mathematical model that the optimal gait pattern is selected according to energetic efficiency, including the energy loss due to the heat emissions of actuators.

1 Introduction

Quadrupeds, such as the horse, show a variety of gait patterns, e.g., walking, trot, and gallop, according to their locomotion velocities, and hexapods, such as the cockroach, also show a variety of gait patterns. Experimental results, in which the oxygen consumption of horses is measured during locomotion, have suggested that the gait pattern is selected so as to minimize the energy consumption to move a unit distance[1]. Although some mathematical studies have treated the relationship between gait pattern and energy consumption from the point of view of robotics [2, 3], these results do not adequately explain the gait selection in legged animals.

In the following sections, we summarize the characteristics of the locomotion pattern in insects, and show the relationship between the gait pattern and the energetic efficiency obtained by computer simulations by using a simplified dynamics of a hexapod. The energy loss due to the heat emission of actuators, which has been ignored in previous studies, is considered to be part of the consumed energy.

2 Locomotor patterns of insects

The following characteristics have been observed in insects[4].

1. The movement of the legs on the same side is propagated from the hindlimb to the forelimb by the same time delay.
2. Contralateral legs of the same segment show anti-phase movements.
3. The period of the leg movement decreases as the walking velocity increases, but the duration of the swing phase is constant.

Another interesting feature of the locomotion pattern is that it changes discretely and can be categorized into some patterns by using the duty factor β, which is defined by the ratio of the duration of the stance phase to the period of the movement of a leg. If all legs have the same duty factor, $N\beta$ (N:number of legs) shows on average the number of legs in the stance phase. In hexapods, $\beta = 5/6$ indicates that five legs are always in the stance phase, which is called a "wave gait." Most of the locomotor patterns of insects are categorized as $\beta = 10/12, 9/12, 8/12, 7/12, 6/12$, where the pattern $\beta = 1/2$ is called the "tripod gait," and the characteristic (3) indicates that the duty factor decreases as the walking velocity increases. In order to walk statically at least three legs must always be in stance phase, hence the lowest possible value for the duty factor in static walking is $\beta = 1/2$.

3 The energetic efficiency

3.1 Definition of the energetic efficiency

The moving velocity of the body V is given by $V = S/\beta T$, where S is the stance length, which implies that two parameters in the parameter set (β, S, T) must be specified in order to move at a given velocity. The purpose of the following sections is to obtain the parameter set which will minimize the energetic efficiency defined by

$$\epsilon = \frac{E}{WVT} = \frac{\beta E}{WS,} \tag{1}$$

where T is the period of movement of a leg, W is the weight of the moving object, and E is the consumed energy to move the distance S. The above definition was developed by G.Gabrielli and T. von Karman (1950) and shows the consumed energy to move a

mass unit distance per unit mass, so a small ϵ indicates high efficiency.

3.2 Model of an actuator

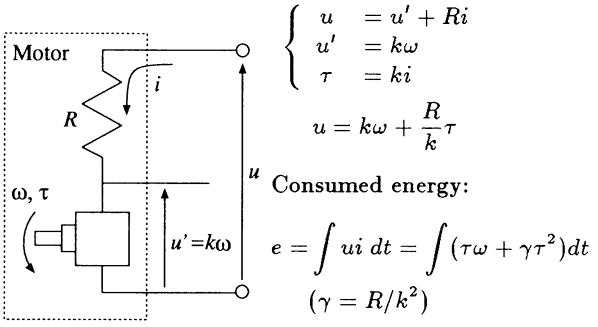


Fig. 1. Energy consumption by an electric motor. The energy consumption e in the motor is given by the sum of the mechanical power $\tau\omega$ and the heat emission $\gamma\tau^2$. u' is the inverse electromotive force and k is a constant.

Here we explain the model of an actuator which is necessary to obtain the energetic efficiency. In general, the generated force and the torque decrease for a constant energy input as the velocity increases, and a portion of the input energy is consumed as heat energy, which is a common feature in electric motors, gasoline engines, and muscles. As a model of an actuator, consider a joint driven by an electric motor (Fig. 1). The consumed energy e in the motor is given by

$$e = \int u(t)i(t)dt = \int \{\tau(t)\omega(t) + \gamma\tau^2(t)\}dt, \qquad (2)$$

where ω and τ are the angular velocity and the torque, respectively, and γ is a constant value depending on the properties of the motor. The first term shows the mechanical power, and the second shows the energy loss by heat emissions. Although a negative value for the first term indicates a gain in energy supplied by external forces, most actuators cannot store this energy. Therefore, we define the energy consumption at a joint by

$$e = \int \{\delta(\tau(t)\omega(t)) + \gamma\tau^2(t)\}dt, \qquad (3)$$

$$\delta(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0). \end{cases}$$

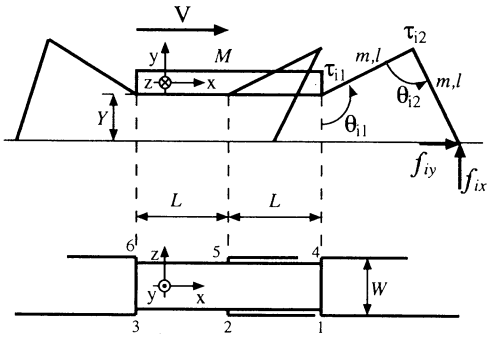


Fig. 2. The side view (upper) and top view (lower) of a hexapod model. In the side view only the right legs are shown.

4 A model of a hexapod

4.1 The dynamics of the legs

We will now discuss a hexapod model. The leg is composed of two joints, and the trajectory is constrained in a two-dimensional plane which is parallel to the xy-plane. The dynamics of a leg is given by

$$M(\boldsymbol{\theta}_i)\ddot{\boldsymbol{\theta}}_i + \boldsymbol{h}(\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i) + \boldsymbol{g}(\boldsymbol{\theta}_i) = \boldsymbol{\tau}_i + J_i^T \boldsymbol{f}_i, \qquad (4)$$

where M is the inertia matrix, $\boldsymbol{h}$ shows the centrifugal force and the force of Coriolis, $\boldsymbol{g}$ shows the effect of the gravitational acceleration, $\boldsymbol{x}_i = (x_i, y_i, z_i)^T$ is the position of the foot on the coordinates of the body, $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2})^T$ is the angle of the joint, and $J_i = (\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{\theta}_i})$ is the Jacobian matrix. If the ground reaction force $\boldsymbol{f}_i(t)$ and the trajectory $\boldsymbol{\theta}_i(t)$ are given, $\boldsymbol{\tau}_i(t)$ is determined by eq.(4). We therefore can calculate the energetic efficiency by eq.(1), (3), and $E = \sum_{i=1}^{6}\sum_{j=1}^{2} e_{ij}$, where e_{ij} is the energy consumption at each joint.

4.2 The trajectory of the legs and the body

The relative velocity of the foot to the body in stance phase is $-V$ if the velocity of the body is V. We assumed that the body moves horizontally at a constant velocity V, and the foot in the swing phase returns close to the ground, the trajectory of which is shown in Fig.3. The motion of the shank is vertically symmetrical and the relative velocity of the foot to the body in swing phase is given by a sinusoidal function so as that the velocity changes continuously. The thighs of the forelimb and the midlimb face forward, and the thigh of the hindlimb faces backwards.

4.3 Phase relation between legs

The phase relations between legs are determined so as to resemble the feature of the locomotor pattern in

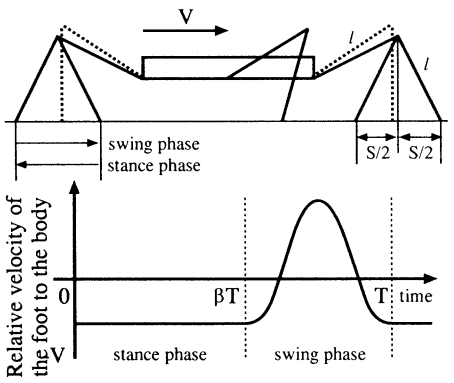


Fig. 3. The trajectory of the legs.

insects mentioned in section 2. The movement of the legs propagates from the hindlimb to the forelimb, and the phase delay between the neighboring ipsilateral legs is given by $(1-\beta)T$.

4.4 The ground reaction force

The ground reaction force $\boldsymbol{f}_i$ is determined by the balance of the force and the moments around each of the axes. Because no forces parallel to the z axis exist in this model, we must obtain at least 2 components (f_{ix}, f_{iy}) of the force for each stance leg (at least 3 legs), i.e., at least 6 components, from 5 equations. Because of the redundancy of the solution, we obtained the ground reaction force by using the pseudo inverse matrix. The inertia force of each leg is neglected in the calculation, which is a small term at low velocities.

5 Simulation results

We calculated the energetic efficiency for each velocity, each duty factor, and each stance length. ($v = 1 \sim 32$ cm/s, $\beta = 12/24 \sim 23/24$, $S/l = 0.1 \sim 0.5$).

Figure 4 shows the relationship between the energetic efficiency and the velocity. When we consider only the mechanical power, the smallest duty factor always gives the smallest energy consumption for all velocities (Fig.4 (b)), but the optimal duty factor decreases as the velocity increases if we take into account the heat energy consumption as a part of the consumed energy (Fig.4 (a)), because the energetic efficiency due to the heat emission which reflects the power to support the body in low velocities shows the opposite behavior to the efficiency due to the mechanical power (fig.4 (c)). The figure 4 (a) also suggests that there is an optimal walking velocity. This optimal value is the fastest velocity in this case, which depends on the parameters of the body, and the constant γ.

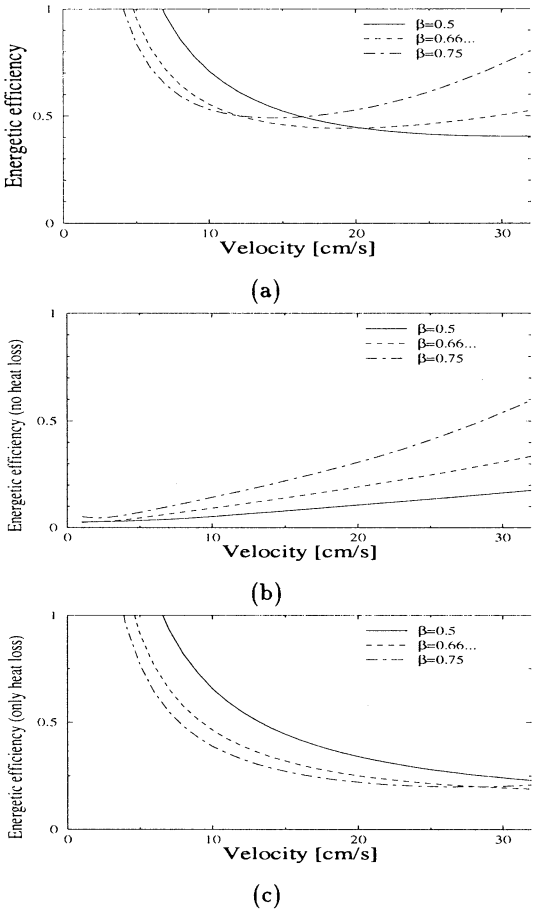


Fig. 4. The relation between the energetic efficiency and the locomotion velocity. (a) Both the energy consumption for mechanical power and the heat energy loss are considered. (b) Without considering the heat energy loss. (c) Only considering the heat energy loss. The stance length is given so as to minimize the energetic efficiency for each velocity and duty factor.

Figure 5 shows the optimal duty factor for each locomotion velocity. Interestingly, the optimal duty factor does not change monotonously, but stays at $\beta = 0.75(= 9/12)$ and $\beta = 0.66..(= 8/12)$, which are observed locomotion patterns in insects. When the ground reaction forces are uniformly given to each leg in stance phase by dividing the weight of the body by the number of the stance legs, different results are obtained and optimal duty factor changes monotonously (not shown here). We therefore conclude that the nonuniform distribution of the vertical forces in legs causes non-monotonous changes in the gait pattern to minimize the energetic efficiency.

The optimal stance length also exhibits complex be-

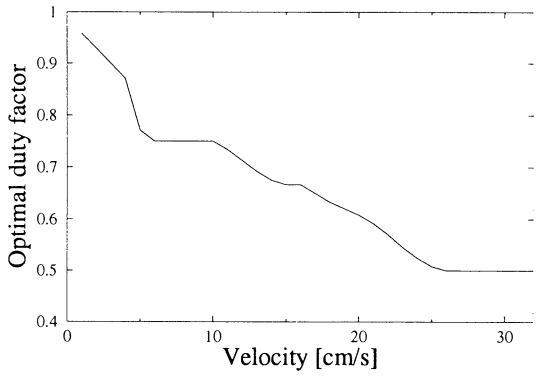


Fig. 5. Optimal duty factor. The stance length takes its optimal value for each velocity which minimizes energetic efficiency.

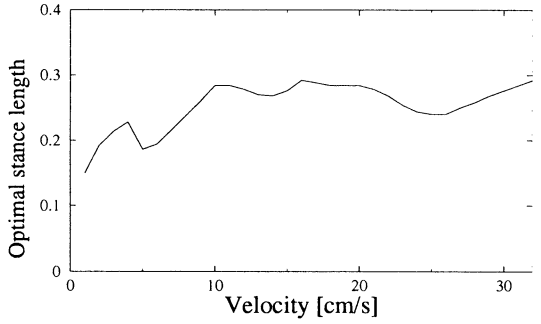


Fig. 6. Optimal stance length (S/l). The duty factor takes its optimal value for each velocity.

havior (Fig.6) which relates to the non-monotonous change in optimal duty factor, and it stays at an almost constant value. This result shows that there is an optimal range for the stance length because large stance length makes the legs horizontal which requires large power to support the body and the small stance length makes the energy to move legs large.

Figure 7 shows the period of the leg motion and the duration of the swing phase given by the optimal duty factor and the optimal stance length for each locomotion velocities which minimize energetic efficiency. These results correspond to characteristics of insect locomotion: the period decreases as the velocity increases, but the duration of the swing phase stays at an almost constant value.

6 Conclusion

The simulation results suggest that many features observed in the gait patterns of insects can be explained by the criteria of energetic efficiency, and that the energy loss by heat emission plays a critical role in gait selection, which has not been adequately considered in previous studies. We are now investigating how the energetic efficiency depends on the trajectory of the leg and other body parameters.

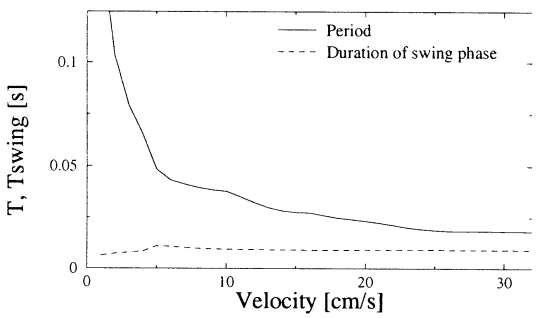


Fig. 7. Optimal period and the duration of the swing phase.

Appendix: Parameters in the simulation

In the computer simulation we set the parameters as, the length of a shank and a thigh are $l = 1.0$ cm, the interval of the root of the thigh is $L = 1.0$ cm, the height of the body is $Y = 0.3$ cm, the weights of the body and the leg are $M = 10$ g and $m = 0.05$ g, respectively, and $\gamma = 0.003$. The calculation was done by Euler method and $\Delta t = T/1200$, $\Delta S/l = 0.002$, $\Delta\beta = 0.1/24$, $\Delta V = 1.0$ cm/s.

Acknowledgement

This study was partially supported by the special postdoctoral researchers program at RIKEN.

References

[1] D. F. Hoyt and C. R. Taylor. Gait and the energetics of locomotion in horses. *Nature*, 292(16):239–240, 1981.

[2] S.Hirose and Y.Umetani. The basic considerations on energetic efficiencies of walking vehicles. *Trans Soc. Instrument Contr. Engineers (in Japanese)*, 15(7):928–933, 1979.

[3] M.Kaneko, S.Tachi, K.Tanie, and M.Abe. Basic study on similarity in walking machine from a point of energetic efficieny. *IEEE Journal of robotics and automation.*, RA-3(1):19–30, 1987.

[4] D. M. Wilson. Insect walking. *Annual Rev Entomology*, 11:103–122, 1966.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Insect Robot Controlled by Emergence of Gait Patterns

Kazushi Akimoto, Shigemichi Watanabe*and Masafumi Yano

Research Institute of Electrical Communication, Tohoku University,
2-1-1 Katahira Aoba-ku, Sendai, Japan 980-77

Abstract

A major problem in walking robots is how to control their walking under unpredictably changing environments. Most walking robots proposed until now can walk under the limited environments, in which the gait patterns are kinematically but not dynamically determined in advance. It means that such robots can not walk adapting to the changes of the world, while animals can walk flexibly and efficiently in the real world. It has been considered that the flexibility and efficiency in animals are originated in the ability of the emergence of the control information. We have already clarified the mechanism of the flexible and efficient generation of gait patterns in animals, so we have tried to make an insect robot based on the mechanisms which can walk adapting the unpredictable changes of environments.

Since these mechanisms are quite new and applicable to other artificial systems, we discuss the emergence system as the control mechanism attaining the aimed state under the constraints of the real world.

1 Introduction

A lot of walking robots have been researched and developed as a advanced moving system, but they can walk well only in the limited environments defined by the designers in advance so as to satisfy the purposes of use. Therefore the changes of circumstances may cause the robot's efficiency of the energy transduction get worse, moreover the unpredictable changes may cause the robot walking breaks down. On the other hand, animals can cope with not only the unpredictable changes of environments but also the changes their purposes by changing its gait patterns. In addition, it is remarkable that the efficiency of the energy transduction is optimized at the any time [1]. These flexiblities are generated by the neural network as a control system, which should be able to organize its dynamical patterns quickly in response to the changes of the walking velocities and load [2].

We have already demonstrated that the neural system composed of active elements such an nonlinear oscillating elements could flexibly self-organize the gait patterns. The gait patterns of the animals are expressed by the coordination of the movement of the muscle fibers, therefor the mechanism of the self-organization of the patterns reflects the characteristics of the muscle contraction. Since muscles have an optimal shortening velocity to provide the highest efficiency, the cooperation and/or competition occur among legs desiring to move at the optimal velocity each other, resulting in the emergence of the most efficient gait pattern [4, 5].

This mechanism can be applied to the artificial systems. The patterns of movement can be self-organized with higher efficiency in response to the changes of the external world. We have made an insect robot based on this mechanism, in which six electric motors attached to six legs are coordinated to efficiently operate under any environmental conditions at any time.

2 Construction of the robot

2.1 Characteristics of walking in insects

In the case of a stick insect moving at high speed, the front leg and the hind leg move simultaneously and the middle leg antiphasic to the others, forming a tripod to support their body, called the tripod gait (Fig. 1). On the contrary, at slow speed, the three legs of each side move metachronally. A pair of legs of the same segment step alternately, called the metachronal gait (Fig. 1). As the walking velocity increases, the insect changes the patterns critically depending on their velocity which resembles a phase transition. The gait patterns also vary with the load [2].

*present address: Networking Laboratory, NTT Multimedia Networks Laboratories, 1-1 hikarinooka, Yokosuka, Japan 239

§ *Keywords:* walking robot, emergence, self-organization, gait pattern, real time.

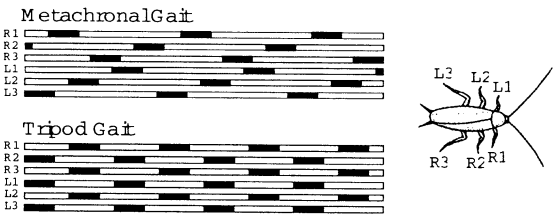


Figure 1: The gait pattern of normal insect. Black bar shows swing phase in which leg lifts and return forward.

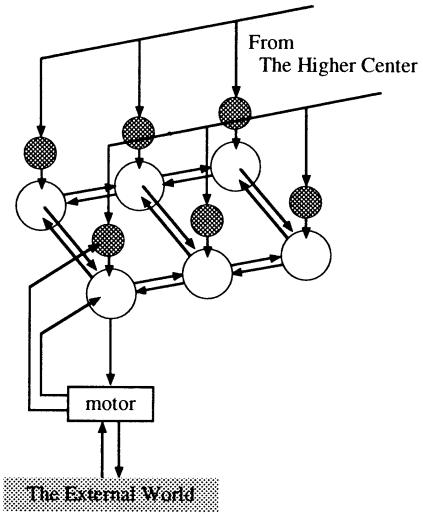


Figure 2: The neural network of the insect robot. Shaded circles and open circles show the rhythmic neuron and nonspiking interneuron respectively.

2.2 Structure of the system for emergence of gait patterns

Since the insects change their gait patterns without their brains, gait patterns are not generated by the central nervous system but by a subsystem below the thoracic ganglions of insects [2]. These ganglions send motor outputs to control leg muscles and receive the external afferents as to position, velocity and force of each muscle. These ganglions are interconnected to each other through a pair of thoracic connectives [2]. Those results imply that the gait pattern may be generated in the subsystem, coordinating the effector organs through active interaction with the real world. Recently it is thought that both the graded potential and the endogenous oscillation play important roles in the neural network to generate the motor patterns

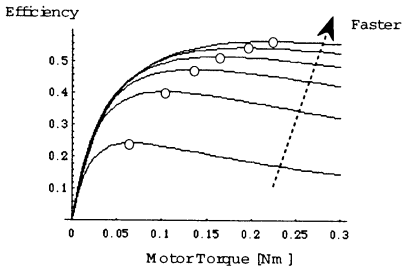


Figure 3: The efficiency curves of a DC motor at various angular velocities. Open circles shows optimal points.

in animals, and the change of parameters defining the two properties cause a wide variation of the motor patterns [3]. Based on their results, we fundamentally composed the neural network for emergence of gait patterns as schematically shown in Fig. 2.

In this network, the feedforward signal from the higher center is given a constant (DC input) for simplicity according to the aimed walking velocity. This signal is sent to every thoracic ganglions, and transformed into endogenous oscillations as the leg cyclic movement by rhythmic neuron. The rhythmic neuron makes direct connection with the nonspiking interneuron which also makes intersegmental inhibitory connections among other nonspiking interneurons. The output of the nonspiking interneuron is transformed into the graded potential as the phase of leg movement along the cycle time of the rhythmic neuron, modulated by the information on the states of leg actuators and intersegmental pathways. Therefore the network of nonspiking interneurons is very important to emergence of the gait pattern.

Here, KYS-oscillators are used as neuron elements, expressed as follows [4, 5]:

$$\frac{d^2x}{dt^2}+(A_1x^2+B_1x+C_1)\frac{dx}{dt}+(A_2x^2+B_2x+C_2)x+D=I$$

where I and x show the input and output terms respectively.

2.3 Feedback for walking efficiently

The gait patterns emerge by coordinating the effector organs through active interaction with the real world. Since the animals can walk efficiently, the gait pattern might be characterized in the insect robot by the efficiency properties of a DC motor as an actuator of a leg. It is well known that a DC motor shows a

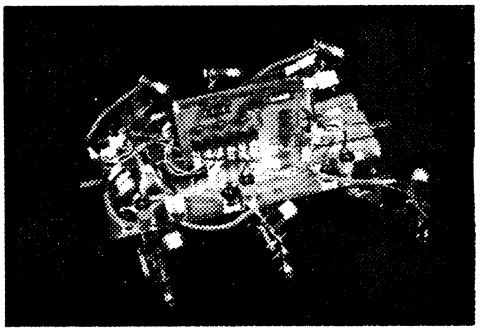


Figure 4: An oblique view of the insect robot.

distinct relationship between the angular velocity (ω) and the torque generation (Tm). Obviously, there is an optimal torque generation for higher efficiency as presented in Fig. 3. In the nonspiking interneuron, the more the parameter C_2 increases, the more the activity grows up. Based on the above, the efficiency of each leg motor feeds back to the parameter of its nonspiking interneuron as follows: When the motor exerts force above the optimal force generation, the activity is inhibited by decreasing C_2 and the activities of others are activated through the inhibitory connections, resulted in hastening the its timing of lifting up and delaying the others, and vice versa.

In addition, the cycle time of the leg movement agrees with the walking velocity and the swing amplitude, so that the information on the cycle time of the leg feeds back to the rhythmic neuron. After comparing the aimed cycle time and the cycle time of leg, the input of the rhythmic neuron is adjusted.

2.4 Total structure of an insect robot

The oblique view of the robot is shown in Fig 4, which is composed of a body and six legs. We use a Chebyshev link mechanism as a leg, which is suitable to produce one dimensional motion parallel to the direction of walking. Each leg has two DC motors, one of which works as a driving actuator during the stance phase and the other is for lifting the leg during the swing phase. In each leg, the position and velocity is detected by a potentiometer, and the driving motor torque by the motor current. The activities of the neurons are calculated with the above informations by using a personal computer during walking, which control the leg motors.

The primary constraint of this robot is to attain the aimed velocity even if the load it is carrying varies. The secondary constraint is to swing its legs with a constant amplitude regardless of its walking velocities.

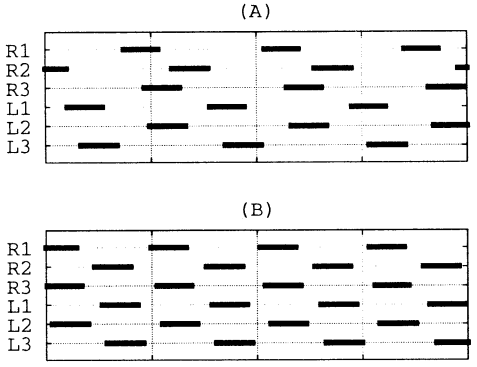


Figure 5: Gait pattern without loads. *A*:At the walking velocity = 2.0 cm/s. *B*:At the walking velocity = 4.0 cm/s. Black bar shows swing phase.

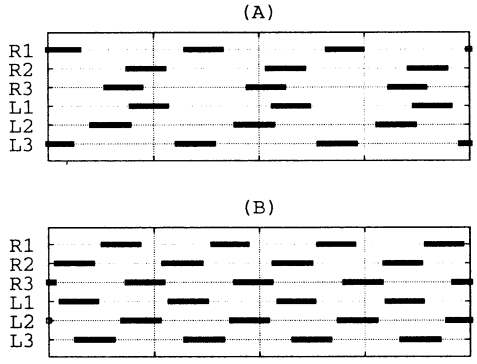


Figure 6: Gait pattern with weight of 600 gw. *A*:At the walking velocity = 2.0 cm/s. *B*:At the walking velocity = 4.0 cm/s.

In this case, the information on the position and velocity of the leg feeds back to DC motor.

3 Results

In the experiments of walking our insect robot, the required velocity is always given as an initial condition, which is the strong constraint for our model system to always attain. Our insect robot can fundamentally generate the two different gait patterns depending on the walking velocities. The gait patterns are characterized by the phase relationship among the six legs. As shown in Fig. 5 and 6.

At higher walking velocity our robot shows that the

front and the hind legs move simultaneously, called tripod gait, but is few influenced by the change of the external load (Fig. 5B and 6B). On the contrary, at a slower walking velocity the external load parallel to the walking direction causes the change of the gait pattern (Fig. 5A and 6A). When our robot walks without loads, the gait pattern looks like the quasi tripod gait (Fig. 5A), but with the external load parallel to the walking direction the six legs of each side move in a metachronal sequence in which the swing phases do not overlap each other, thus showing the metachronal gait pattern (Fig. 6A). These results agree with the experimental phenomena in insects very well.

These phenomena can be explained as follows: At higher walking velocity the efficiency curve of a DC motor shifts right-upper than at slower velocity (Fig. 3). Since a leg works at a lower torque than that at the optimal efficiency of energy transduction, the leg must achieve a higher torque by concentrating the load, resulted in activating itself and suppressing the others. This causes antiphasic movement between the nearest neighboring legs and changes the gait pattern from a metachronal gait pattern to a tripod gait pattern. Thus at higher walking velocity our robot can walk more efficiently at tripod gait.

In the reverse case, the efficiency curve locates left-lower. When a leg works at a higher torque than that at the optimal efficiency, a leg suppresses itself and activates the others, so that sharing the load among legs, then the transition from a tripod pattern to a metachronal pattern occurs. At slower walking velocity the robot with the loads selects the metachronal pattern because it is higher efficiency.

4 Conclusions

We have shown that a new control mechanism installed in the insect robot can operate at an optimal point of energy transduction attaining the purpose of the system under unpredictable changes of the environment. This control mechanism is derived from the walking of animals, in which the efficiency of the energy transduction is very high and not affected by its walking velocity. This can be applied to a control system of a space exploring robot, which need to work in unpredictable changes of the environment and use any resources efficiently.

Here, we have made an insect robot as an example system in which appropriate gait patterns emerge to walk efficiently. Since the gait pattern changes crucially depend upon their walking velocities and loads, animals could generate a great number of diversities of gait patterns to adapt to the unpredictable changes of their surroundings. Several models to generate gait patterns have been proposed. In some models, a command from a higher level, that is a brain, is required to change the connectivity of circuit to produce several gait patterns, so that such models need to learn and solidify several ad hoc patterns in the neural circuit, lacking the flexibility to adapt quickly to the various changes of the external world. This is not the case, because the animals produce a new gait pattern very quickly and flexibly. So we have constructed the control system from a self-organization point of view.

Generally, in order to self-organize some order, it is necessary to have some constraints on the system. In this sense, the biological systems are indefinite until the external world to them is provided. In this case, the required velocity is thought to be a constraint on the system. To operate at an optimal point for the whole, the phase relationship among the legs should be regenerated at real time to minimize the local cost function of each energy transducer, for example the muscle in an animal and the electrical motor in a robot, attaining the purposed velocity simultaneously with various loads. In the case of the gait pattern generation, the emergence of order should not be derived from the reference of the entire system. This is because the reference function could not be determined when the changes of the external world are unpredictable. we call this, "the least dissatisfaction for the greatest number of the elements" rule.

References

[1] Hoyt DF,Talor CR (1981) Gait and the energetics of locomotion in horses, *Nature.* 292(16):239-240

[2] Graham D (1985) Pattern and control of walking in insects, *Adv. Insect. Physiol.* 18:31-140

[3] Getting PA (1989) Emerging principles governing the operation of neural networks, *Ann. Rev. Neurosci.* 12:185-204

[4] Kimura S, Yano M, Shimizu H (1993) A self-organizing model of walking patterns of insects, *Biol. Cybern.* 69:183-193

[5] Kimura S, Yano M, Shimizu H (1994) A self-organizing model of walking patterns of insects, part2, The loading effect and leg amputation, *Biol. Cybern.* 70:505-512

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998, begining page-ending page

The application of autonomous decentralized system to myriapod locomotion robot

Tadashi Odashima
Graduate School of Engineering
Nagoya University
Furocho,Chikusa-ku,Nagoya.464-01

Hideo Yuasa
Graduate School of Engineering
Nagoya University
Furocho,Chikusa-ku,Nagoya.464-01

Masami Ito
Bio-Mimetic control reseach Center
RIKEN
Anagahora,Shimoshidami,Moriyama-ku,Nagoya.463

Mikio Yasubayashi
faculty of engineering
Chubu University
Matsumotocho,Kasugai,Aichi.487

Abstract

The locomotion is said to be generated by the CPG (Central Pattern Generator) that is located at lower level of the nervous system. The CPG consists of some nervous oscillators, which is able to be regarded as an autonomous decentralized system. The gait patterns generated by these oscillators in CPG adapt to various environments and the gaits are high efficiency. We apply the autonomous decentralized system to myriapod locomotion robot. In this algorithm, one subsystem consists of one leg and one oscillator. Though each of them only have functions for controlling its own leg, they can generate harmonious gait patterns. Some simulations based on this algorithm show it.

Key Wards: Local Information, Co-operation, Pattern Generation, Locomotion Robot, Autonomous Decentralized System.

1 Introduction

Behaviour of animals can be changed for their environment and aim suitably. For this ability, animals can move at various speed under any condition. At that time, they change their gait pattern according to the condition and their demanded speed. It is said that though the animals are planning their route in its brain, most of its gait patterns are generated by a lower nervous system locally and autonomously [1]. Animals have the self-excited nervous oscillators which generate periodic movement for each leg. The nervous system at lower level called CPG (Central Pattern Generator) consists of these oscillators. In CPG, the nervous oscillators interact one another and control the periodic movement of each leg and then the gait patterns are generated. These patterns have a property that they are realized as the energy consumption is minimum at the desired speed [2]. This method can make the information from leg reflect to changing gait patterns quickly. That is why the animal locomotion have a efficient and flexible. In this paper, we consider the myriapod locomotion robot as an example of the decentralized autonomous system. The system consists of legs that are regarded as subsystems. We consider the algorithm and functions needed for the realization of three demanded gait patterns by only local interaction. And then computer simulation shows our system based on this algorithm has good abilities.

2 The structure of subsystem

We regard a myriapod locomotion robot as an aggregait of subsystems which consist of one leg and one oscillator. Each subsystem is arranged at two lines along progressive direction shown in **Fig.1**. They link with neighbors (forward, backward and side) and can communicate with them. Each subsystem informs its neighbors of the phase variable (θ) of its oscillator. Then the state space is regarded as an unit circle. It determines the periodic motion of its leg. Note that θ is a state variable which determines oscillator's state but does not show the real motion of its leg. By only a local interaction between neighbors based on θ, the whole system generates some harmonious gait patterns.

Each subsystem has a touch sensor on its foot, and informs neighbors whether it is on the ground or not. And they can get angular velocity parameter (ω) which means the advancing speed of robot. It is necessary that all subsystems can recognize the progressive

direction so that they can work in both sides. Using this information, they can know in which side they are. All subsystems can sense whether their neighbors exist or not, too. If a subsystem sense what no subsystem exists in front, it judge it is head of the line. And vice versa. All functions of this subsystem are as follows:

- This subsystem can sense neighbor's existence.
- This subsystem can inform only its neighbors of its state variable (θ).

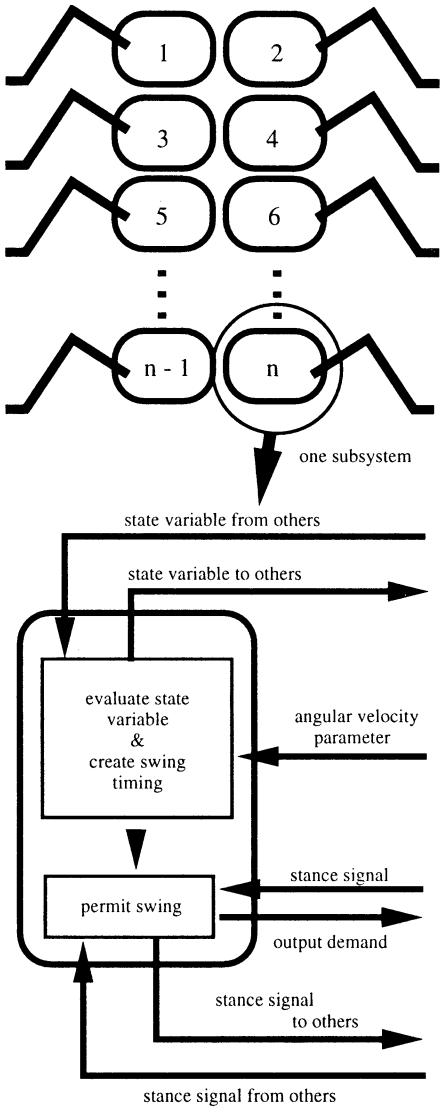


Fig 1: The arrangement and composition of subsystem

- This subsystem can recognize which line it is on.
- This subsystem can get the angular velocity parameter (ω).
- This subsystem has a touch sensor on its foot, and inform its neighbors whether it is on the ground or not.

The aim of the whole system is to realize these desired three gait patterns shown in **Fig.2**.

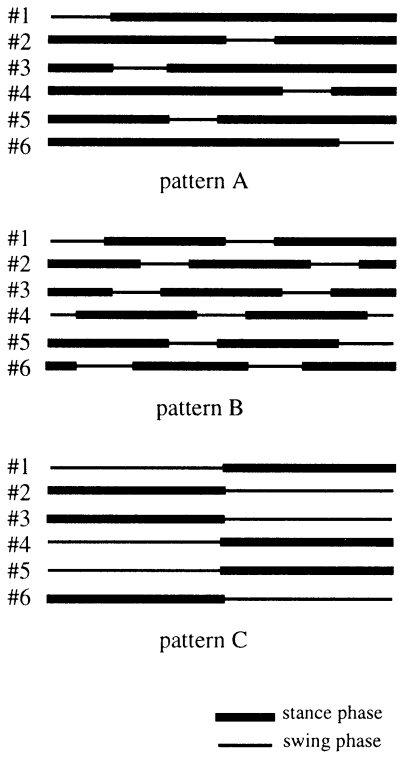


Fig 2: The three desired gait patterns

In the steady state, the phase difference between one subsystem and the opposite side one is π in these three gait patterns. We call them gait A, gait B and gait C respectively. Gate C is the pattern which can move faster than gait B and gait B can move faster than gait A.

These gait patterns are based on those of insect [5] [6]. The vertebrate animals change their walk speed by changing gait pattern as a swing period is constant, whereas insects grow its speed by reducing stance period and increasing stride [6]. In our simulation, we employ the gait patterns of insects but make the stride and the swing period constant.

All gait patterns can be expressed by a phase difference. The demanded gait pattern can be realized by controlling the phase difference to the target value. But the whole size of system is needed to determine the target value. If all state variables are arranged at random in state space, it is difficult for subsystems which can treat only local information to recognize the whole size (See Fig.3). To solve this problem, the state variables should be rearranged at the same order of physical position. After this step, by using the size of state space, all subsystems can recognize the size of whole system from only local information. **Fig.5** is the simulation result of this system, and this shows that generated gait patterns are very simiar to demanded patterns. In following part, these processes for acquiring gait patterns are explained simply [7]. In this paper, we number subsystems but these numbers are not needed for executing our algorithm.

2.1 Phase rearrangement

The initial value of θ is determined in the range from 0 to 2π at random. First, the order of state variable have to be rearranged to the same order of physical position. We prepare the Gaussian function into torus space as potential function.

$$W_i(\psi_i) = -h_i \exp[-T_i \{1 - \cos(\psi_i - \psi_{ip})\}] \qquad (1)$$

In this function, T_i and h_i mean the following.

$$T_i = \frac{1}{2\sigma^2} \quad , \quad h_i = \frac{1}{\sqrt{2\pi\sigma^2}}$$

The σ corresponds to standard deviation. The ψ_i in equation (1) means the phase difference between number i subsystem's and its forward's. And the ψ_{ip} is the target value of ψ_i. In the gradient system (equation (2)) whose potential is equation (1), the ψ_i is converged to ψ_{ip} and then the state value of each subsystem is determined relatively against others'.

$$\frac{d\theta_i}{dt} = -\frac{\partial W_i(\psi_i)}{\partial \psi_i} \qquad (2)$$

When the rate of change of θ becomes small enough, these subsystems start the next step.

2.2 Recognition of size

Each subsystem recognizes the size of whole system by repelling other state variables, which can do from only local information. We prepare such potential function as follows:

$$U_{ip} = \frac{c}{\psi_{id}^2} \qquad (3)$$

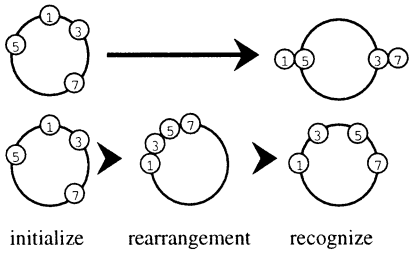


Fig 3: The expanding of state variables which is arranged at left line in state space

The subscript d shows the direction of operation. This potential function works on the front and the rear subsystems, then all state variables are arranged at regular interval in the unit circle. The gradient system following this potential function (3) is as follows:

$$\frac{d\theta_i}{dt} = -\frac{\partial U_{if}(\psi_{if})}{\partial \psi_{if}} - \frac{\partial U_{ib}(\psi_{ib})}{\partial \psi_{ib}} \qquad (4)$$

Subsystems positioned at the head of line and the end of line play a special role in this step. They stop their interaction for inhibiting more extension beyond the half of the state space [7]. The number of subsystems in one side is recognized by the phase difference between one subsystem and its neighbor, when the extension is finished. Because the half of unit circle is π then this number (nu) is calculated by:

$$nu = N\left[\frac{\pi}{\psi_{abs}} + 1\right] \qquad (5)$$

where ψ_{abs} is a final value of ψ_{if} or ψ_{ib}. And the function of $N[\ \cdot\ ]$ means to round off a number to integer.

This recognition method cannot be achieved without the phase rearranged step. **Fig.3** shows the reason. The lower figure in **Fig.3** shows the case that all steps explained before have been done. Four state variables are arranged at regular interval in half of the unit circle. And then, they can recognize the size of whole system correctly (In this case, it is 4). But the upper one in **Fig.3** shows the case that "rearrangement" has not been done. These state variables cannot show the size of whole system correctly. They incorrectly recognize 2 as the half number of legs. This shows that the "rearrangement" step is important for correct recognition of the size of whole system.

2.3 Generation of swing timing

In this step, desired gait patterns are generated by operating local potential functions which bifurcate according to the angular velocity parameter ω [4]. The equation (6) is the final form of oscillator dynamics.

$$\frac{d\theta_i}{dt} = \omega - \sum_P q_{iP}(\omega)(E_{if} \cdot f_{if} + E_{ib} \cdot f_{ib}) - E_{is} \cdot f_{is} \quad (6)$$

In the above equation, the f_{i*} means as follows.

$$\begin{aligned} f_{if} &= \frac{\partial W_{if}(\psi_{if})}{\partial \psi_{if}} \\ f_{ib} &= \frac{\partial W_{ib}(\psi_{ib})}{\partial \psi_{ib}} \\ f_{is} &= \frac{\partial W_{is}(\psi_{is})}{\partial \psi_{is}} \end{aligned} \quad (7)$$

The f_{if} means the interaction between i subsystem and $i-2$ subsystem. Similarly, the f_{ib} and the f_{is} mean the interaction with backward and side. The W_{i*} in equation (7) is the same form with equation (1). If the subsystem is not linked with the specified subsystem, the E_{i*} forces the local potential function not to give any effect to the rate of change of the θ_i. That is, E_{i*} takes 0 if it is not linked, and conversely, E_{i*} takes 1 if it is linked. E_{if} means the direction of forward. In similarly, E_{ib} and E_{is} mean the backward and side. The subscript P means one of the demanded pattern A, B and C. The term of q_{iP} is a support function of ω. According to the change of ω, this function selects the most desired local potential function. The q_{iP} consists of two sigmoidal function shown in (8).

$$\begin{aligned} q_{iP}(\omega) &= \frac{1}{1+\exp[-S_{iP\_}\{\omega - (\omega_P - L_{iP\_})\}]} \\ &- \frac{1}{1+\exp[-S_{iP+}\{\omega - (\omega_P + L_{iP+})\}]} \end{aligned} \quad (8)$$

These parameters S_{iP_*} and L_{iP_*} determine the shape of q_{iP}. These support functions (q_{iA}, q_{iB} and q_{iC}) are regarded as the weights of local potential functions. **Fig.4** shows the notion of a role of equation (6).

It is necessary to inhibit some legs from swinging so that the whole system does not tumble down. So subsystems are allowed to swing their legs only when at least one neighbor is in stance phase. Employing this rule, a neighborhood of any subsystems includes at least one stance leg, and whole system does not tumble down.

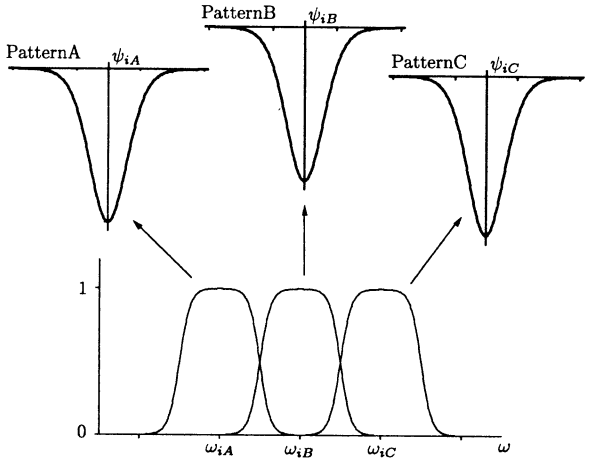


Fig 4: The notion of relation of potential functions and support functions

3 Result of simulation

The result of simulation is shown in **Fig.5**, which is in a case of eight legs system. The numbers with # mark on vertical axis mean the position of subsystem shown in **Fig.1**. The line segments parallel with horizontal axis mean the stance phase. The horizontal axis represent progress of time. The ω is changed according to time (t) as follows:

$$\omega = 0.7t/40000 \quad (9)$$

Fig.5 shows the change of swing timing. In the case of eight legs system, the transition from gait A to gait B occurs at $\omega = 0.175$, and that from gait B to gait C does at $\omega = 0.35$. So each transition occurs at $t = 10000$ and $t = 20000$. The larger the ω becomes, the shorter the line segments become. But the stance phase of #2 becomes short suddenly at the point that t is about 20000. It shows that the #2 delays its swing timing. When ω is extremely small, the rate of change of θ may be a negative value, which means that the state variable may turn reversely in unit circle. Because this reverse motion of legs disturb the whole locomotion, then the rate of change of θ is forced to be 0 if it becomes negative value. This is a realistic method, because the real robot cannot move legs too slow but only can stop them.

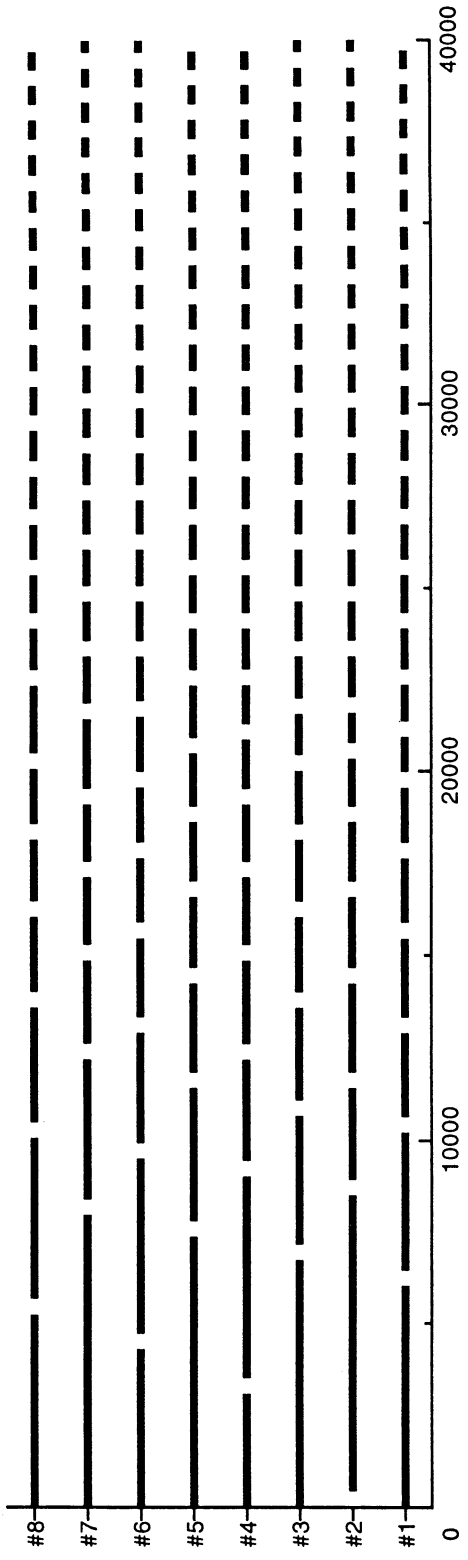


Fig 5: Generated gait pattern

4 Conclusion

Using these method, it is possible to generate some desired gait patterns for miriapod locomotion robot which is consist of subsystems regarded as self-excited nervous oscillator. These gait patterns demanded in this paper may not be the best pattern for locomotion. But we can change them easily by changing target value in potential function. As stated above, these three patterns are based on that of insect [5] [6]. But if the number of legs becomes more than 10, the ω value at transition (ω_A and ω_B) becomes very small, that is, the locomotion speed of gait A and B becomes too slow to walk practically. It is one of our theme how desired gait patterns are defined according to the size of whole system.

The simulation in this paper is programmed for the real locomotion robot which we are designing and making now. We will study how to generate more efficient gait pattern through experiments with real robot.

References

[1] U.Bassler(1986)"On the definition of Central Pattern Generator and its Sensory Control," *Biological Cybernetics*,vol.54,pp.65-69

[2] D.F.Hoty, C.R.Taylor(1981), "Gait and the energetics of locomotion in horses," *Nature*,vol.292,no.16,pp.239-240

[3] H.Yuasa, M.Ito(1991), "A Theory on Structures of Decentralized Autonomous Systems(in Japanese)," *Transactions of the Society of Instrument and Control Engineering*, vol.25,no.2,pp.180-187

[4] Y.Ito, H.Yuasa, M.Ito(1991), "Autonomous Distributed Systems which Generate Various Patterns Using Bifurcation(in Japanese)," *Transactions of the Society of Instrument and Control Engineering*, vol.27,no.11,pp.1307-1314

[5] K.Ogawa, J.Nishii, Y.Uno, R.Suzuki, "The optical hexapod'sgaitpattern based on energetic efficiency(in Japanese)," *Technical Report of IEICE*, MBE92-113(1993-03) 9/16

[6] M.Wells(1968), "Lower animal," George Weidenfeld and Nicolson Ltd,London

[7] T.Odashima, H.Yuasa, M.Ito(1997), "The application of autonomous decentralized system to myriapod locomotion robot(in Japanese)," *Journal of Robotics Society of Japan*,(submitted)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998, begining page-ending page

Evolving Neural Controllers that Elicit Non-trivial Behaviors Using Developmental Process

Akio Ishiguro, Hideaki Iwaki, Akinobu Fujii, Yoshiki Uchikawa*

Abstract

Recently, Evolutionary Robotics (ER) has been attracting much concerns in the field of robotics, artificial life and so on. ER approaches are expected to provide feasible methods to design controllers for autonomous mobile robots with less human intervention. However, most of the conventional studies in ER have been aiming at obtaining very simple (trivial) behaviors such as obstacle avoiding, wall following and target approaching. To make ERs approach more fruitful, we should pay close attention to obtaining non-trivial behaviors. Based on this considerations, in this paper, we propose a method for obtaining non-trivial behaviors using a developmental process with carefully arranged grafting method. To verify the validity, we apply our idea to construction of neural controllers that can cope with rough terrain.

1 Introduction

Recently, *Evolutionary Robotics* (ER in short) has been attracting much concerns in the fields of robotics, artificial life and so forth. In ER approaches, controllers for autonomous mobile robots are designed with less human intervention generally by using evolutionary computation techniques such as genetic algorithms. However, most of the conventional ER studies have been particularly focused on obtaining very trivial behaviors such as obstacle avoiding, wall following, target approaching and so on.

To make ER approaches more useful, we should develop a new method that can realize *non-trivial* behaviors. From point of view, the following studies step toward the right direction.Nolfi *et al.* evolved a neural controller for a garbage collecting robot by introducing a carefully arranged *canalization process*[1][2]. They can successfully obtain the neural controller that realizes the following sequence of behaviors: explore the environment avoiding the walls, recognize the target object, pick-up the target object, move toward the walls without avoiding, and then release the grasped object.

On the other hand, Gomez *et al.* constructed a neural controller that enables evading multiple chasing robots, *i.e.* predators, without colliding with the walls of the arena[3]. They determined the parameters of the neural controller by a genetic algorithm with use of the devised coding, called *delta-coding*.

These are the pioneering works which successfully obtain non-trivial behaviors with the use of evolutionary techniques. However, in both methods, the structures of the neural controllers have to be determined in a top-down manner, and the evolutions are carried out by only changing parameters, *i.e.* synaptic weights, of neural networks. Therefore, possible number of stored behaviors has limitations due to the fixed structure of the neural controllers.

In this paper, unlike the above, we take an approach based on the changing not only parameters but also structures of neural controllers by introducing the concept of developmental process. Thanks to allowing the structural changes, restrictions of a set of behaviors stored in neural controllers can be relaxed. However, structural changes are often in danger of impairing the previously obtained abilities. To overcome this problem, we introduce a developmental process with carefully arranged *grafting method*. We apply this idea to construct neural controllers of legged robots that can cope with uneven terrain. Simulation results show this approach is promising.

2 Problem

Before explaining our proposed method, in this section, we show the problem used in the study. As mentioned above, the long-term objective of this study is to construct a method that realizes obtaining non-

*Department of Computational Science and Engineering, Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-01, Japan. E-mail:ishiguro@bioele.nuee.nagoya-u.ac.jp

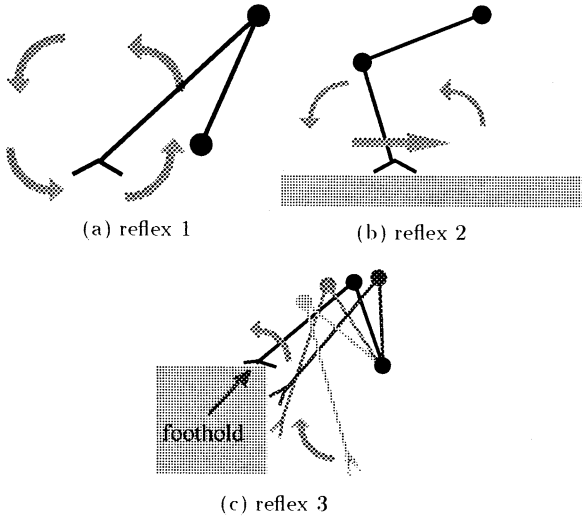
trivial behaviors through evolutionary techniques. In this paper, we take evolutionary construction of a neural controller for a multi-legged robot as a practical example. To date there are several researches aiming at obtaining controllers for legged robots with ER approaches. However, most of these conventional studies just realized neural oscillator with fixed pattern[4]. This implies that their controllers are suitable only for structured environments, *i.e.* even terrain.

Thus, in this paper, we try to construct neural controllers that can cope with uneven terrain. Based on Ding's study, we particularly focus on the following reflexes, called *substrate-finding reflex*, that can be observed in most arthropods[5]:

reflex 1 As soon as a leg has lost contact with a substrate, it instantaneously begins periodically movement for searching for a foothold. (This stereotyped reflex can be easily observed when we pick up insects in mid-air.) (see Fig.1(a))

reflex 2 Once a foothold is established, the leg immediately starts swing-backward behavior to propel its body. (see Fig.1(b))

reflex 3 When a tibia of a leg encounters an object during searching for a foothold, the leg slides up along the object. After clearing up, the leg immediately comes down to establish a foothold. This reflex ensures a foothold correctly even when the object available for a foothold is quite narrow. (see Fig.1(c))



(a) reflex 1 (b) reflex 2

(c) reflex 3

Figure 1: Substrate-finding reflex.

Insects have obtained these reflexes through their long history of evolutionary process. It must be noted that insects use their legs as not only effectors but also sensors. This is very crucial for realizing an agile locomotion even on rough terrain.

In this study, as a report of our preliminary investigations, we try to construct a neural controller for one leg that can realize the above three reflexes. The structure of the leg used in this study. This leg has two joints, namely two degrees of freedom in total. Each joint has a angle sensor and a stress sensor, and each motor is assumed to be driven by torque demand. Therefore, the goal of this study is to construct a neural controller that correctly mapping from four sensory inputs to two motor outputs to realize the three reflexes.

3 Proposed method

3.1 Task decomposition

We should notice that it is quite difficult to evolve a neural controller which enables all reflexes mentioned in the previous section all at once. Therefore, we decide to obtain these reflexes in a step-by-step manner. By using this approach, neural controllers finally to be constructed are expected to be easily obtained rather than a straightforward way. This can be regarded as a sort of canalization process[1][6].

Based on this consideration, in this study, firstly we evolve a neural controller realizes *reflex 1*, then we successively incorporate the abilities of *reflex 2* and *reflex 3* in that order into the previously obtained ability (see Fig. 1).

3.2 Description of developmental process

So far various methods have been proposed for the developmental construction of neural networks[7]. In this study, we adopt a *geometry-oriented approach* for the developmental process since the geometry-oriented developmental process inherently have the advantage of generating various network architectures such as recurrent connections and so on. To put it concretely, we describe the developmental process by devising the method proposed by Kodjabachian *et al.*[8].

In the following, we show how the developmental process are carried out in detail. Fig.2 denotes an example of the genotype. As shown in the figure, the genotypes used in this study have so-called binary tree structures. Each node represents instruction for the developmental system. The instructions **DI**, **GO** and

Table 1: Instruction set of the developmental process.

instruction	content
$DI(x.y)$	create a daughter cell at the position $(x.y)$
ND	no operation
$GI(x.y.w)$	drawing axon with synaptic weight w from other cell around the position (x, y)
$GO(x.y.w)$	outgrowin axon with synaptic weight w to other cell around the position (x, y)
EN	terminate the developmental process of the cell concerned

GI execute cell division, creating efferent and afferent axon, respectively. On the other hand, **ND** means the non-divide instruction. This works as non-sense code, *i.e. intron*. **EN** is the instruction that terminates the developmental process. Details of these developmental instructions are listed on Table 1. Developmental

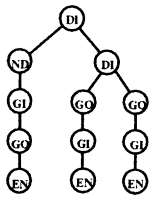
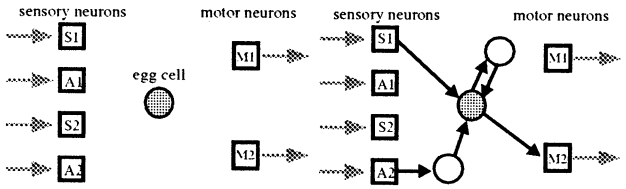


Figure 2: An example of the genotype.

processes are carried out from one egg cell. As shown in Fig.3(a), the initial egg cell is placed around the center of the nervous space. Note that the four sensory and two motor neurons are also placed in this nervous space in advance. The egg cell reads and executes the instructions described in the genotype from the very top of the tree. Throughout this process, final genotype is obtained as illustrated in the figure (b).



(a) before the developmental process (b) after the developmental process

Figure 3: Geometry-oriented developmental process.

3.3 Fitness function

To obtain three reflexes mentioned in the previous section appropriately, we firstly have to determine fitness function for each reflex.

3.3.1 Fitness for *reflex 1*

As mentioned above, *reflex 1* elicits periodical searching behavior when the leg has lost contact. To realize this behavior, we use the following fitness function F_1:

$$F_1 = \sum \vec{f_1} \cdot \vec{v}, \quad (1)$$

$$\vec{f_1} = \begin{bmatrix} -(y - L_1) \\ x \end{bmatrix} \exp\left\{ -\frac{x^2 + (y - L_1)^2}{{L_1}^2} \right\} \quad (2)$$

where $\vec{v}$ denotes the velocity vector of the tip of the leg. $\vec{f_1}$ expressed in equation (2) is a circular vector field as shown in Fig.4. In short, this fitness function will have high value if the leg moves periodically.

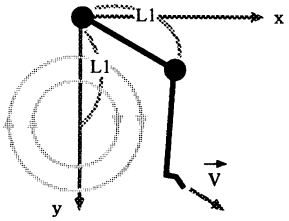


Figure 4: Vector field of f_1.

3.3.2 Fitness for *reflex 2*

This reflex works to support and propel its body when the tip of the leg has contact with the substrate. In this study, we simply use the following function as the fitness for this reflex:

$$F_2 = \textit{distance of moving forward.} \quad (3)$$

3.3.3 Fitness for *reflex 3*

This reflex requires most complicated control among these three reflexes. To elicit sliding-up behavior keep-

ing contact with the object and immediately establishing a foothold after clearing up, we use the following fitness function with the vector field f_3 illustrated in Fig.5:

$$F_3 = \begin{cases} \sum_{\substack{\text{while keeping contact} \\ \text{(if not establish a foothold)}}} \vec{f_3} \cdot \vec{v} \\ \sum_{\substack{\text{while keeping contact} \\ \text{(otherwise)}}} \vec{f_3} \cdot \vec{v} \left\{ 2 + \exp\left(-\sqrt{\frac{D}{L_1}} \right) \right\} \end{cases} \quad (4)$$

$$\vec{f_3} = \begin{bmatrix} sgn[d-x] & - & sgn[x-d] \\ sgn[l-y] & - & sgn[y-l] \end{bmatrix} \quad (5)$$

$$sgn[p] = \begin{cases} 1 & (p \text{ is true}) \\ 0 & (otherwise), \end{cases} \quad (6)$$

where D denotes the distance between the foothold point and the edge of the substrate (see Fig.5).

By integrating the above fitness functions, we use the following equation as the fitness to evaluate the total ability:

$$F = F_1 \times (F_2 + 1) \times (F_3 + 1). \quad (7)$$

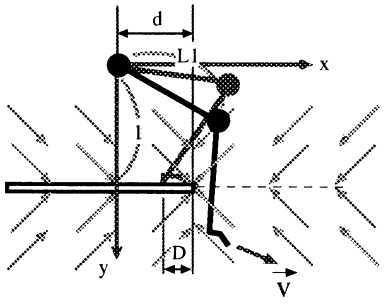


Figure 5: Vector field of f_3.

3.4 Incremental evolution by grafting

As described in this section, we firstly evolve a neural controller which elicits *reflex 1*, then we successively incorporate the abilities of *reflex 2* and *reflex 3* in that order into the previously obtained ability. To realize such an incremental evolution, we propose a *grafting* method.

Fig.6 schematically represents the concept of this grafting. In the figure, suppose that the slightly-shaded portions denote the obtained genotype and its corresponding phenotype that realize the reflex, say *reflex 1*, through the evolutionary process. To incorporate the additional ability, in this case *reflex 2*, we add a subtree to the leaf of the previously obtained tree (heavily-shaded portions). By this, new structures, *i.e.* new cells and connections, can be added while the previous structures are preserved. To keep the previously obtained abilities, genetic operators are carried out only at the newly added subtrees.

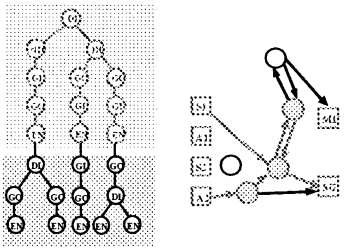


Figure 6: Grafting method.

3.5 Variation of the grafting methods

In the canalization process based on the incremental evolution, it is quite important how we incorporate new abilities into the existing neural controller keeping the destruction of the previously obtained abilities as small as possible. To realize this requirement, we will investigate the following two grafting methods (for convenient, we call these *method A* and *B*, respectively) depicted in Fig.7. It must be noted that unlike *method A* most nodes in the added subtrees (shaded portions) in *method B* are occupied by non-sense developmental instructions, *i.e.* **ND**s. We may recall that this **ND** instruction works as an *intron* observed in genes of living organisms. The effects on the resultant phenotypes caused by these grafting methods will be investigated in the next section.

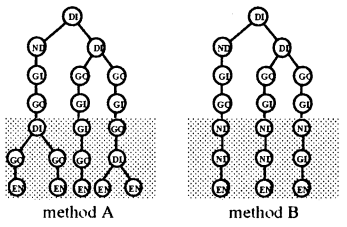


Figure 7: Variations of the grafting method.

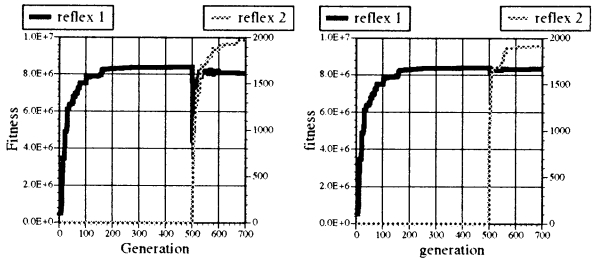
4 Simulation results

To verify the validity of our proposed method, we carried out simulations. In the following simulations, each part of the leg, *i.e.* femur and tibia, is simply assumed to be in the same shape of a stick with uniform density. We set mass and length to 10 and 100, respectively. Movable range of each part is determined based on that of insects.

In the evolutionary process, the initial 100 population are randomly generated. Each individual has the tree structure with depth 8. In the following simulations, we set the depth of each subtree added in the canalization process of obtaining *reflex 2* and *reflex 3* is 5.

Fig.8 shows the comparison of the fitness transitions between *method A* and *method B* in the canalization process from *reflex 1* to *reflex 2*. In this simulation, we evolved neural controllers for *reflex 1* from the 0th until 500th generation. Then we added subtrees to the best genotype obtained at the 500th generation for incorporating the ability of *reflex 2*. This process lasted until the 700th generation.

From the figures, unlike *method B* we can see that *method A* shows drastic decrease of the fitness value of *reflex 1* when the subtrees for *reflex 2* are added. This decrease in *method A* can be fairly recovered through the successive evolutionary process, however, remarkable deviation still exists at the 700th generation. On the other hand, in *method B* prominent decrease in the fitness value of *reflex 1* can not be observed around the 500th generation. From these considerations, it seems that *method B* can suppress destruction of the previously obtained abilities as small as possible rather than *method A*.



(a) *method A* (b) *method B*

Figure 8: Fitness transition in the canalization process from *reflex 1* to *reflex 2*.

Next, we added subtrees again to the best genotype of *method B* obtained at the 700th generation to elicit *reflex 3*. This evolutionary process was continued until the 900th generation. Fig.9 shows the resultant behavior of the neural controller obtained at the final generation. We can see that this neural controller realizes all the reflexes. Fig.10 depicts the structure of the best neural controller obtained at the 500th, 700th and 900th generations. From the figure, it can be observed that the new abilities are incorporated by newly creating neurons and connections.

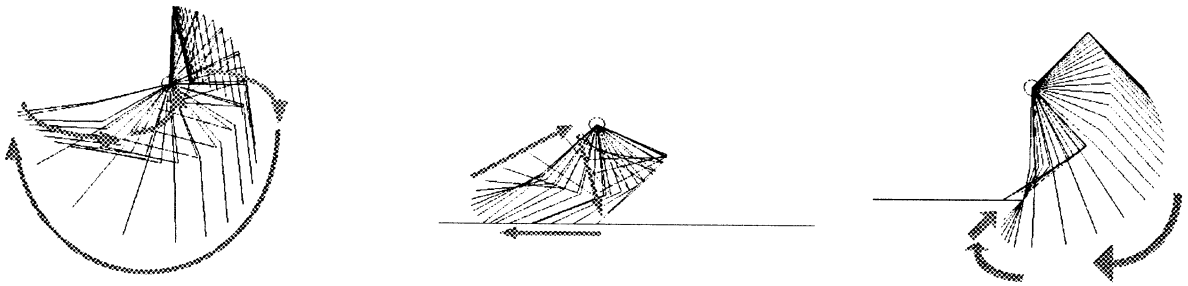
5 Discussions

As shown in the previous section, the grafting method based on *method B* can provide more seamless canalization process rather than *method A*. This can be simply explained as follows. We must recall that added subtrees in *method B* contains many non-sense developmental instructions, *i.e.* introns. Therefore, when we add subtrees to incorporate a new ability, the structures of neural controllers obtained so far are not so changed. This ensures to prevent the existing abilities from destruction. Meanwhile, in the successive evolutionary process meaningful developmental instructions are gradually increased through genetic operators among the individuals. Thanks to this fact, *method B* can incrementally obtain new abilities without drastic impairment on the previously obtained abilities.

6 Conclusions and further work

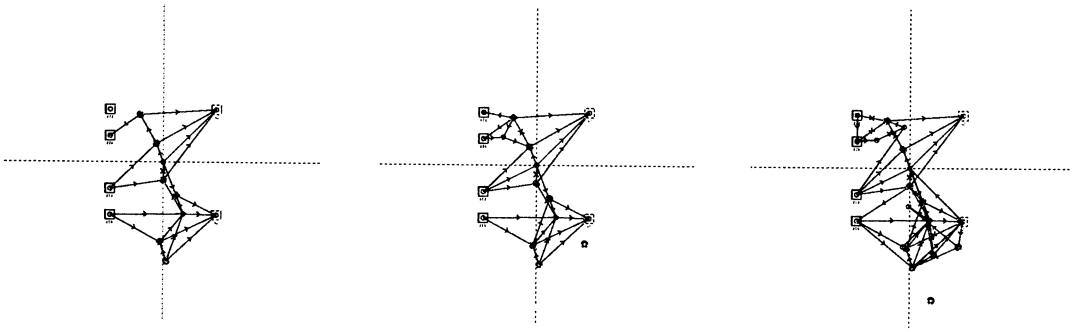
In this paper, we proposed an incremental evolution by introducing a geometry-oriented developmental process. We applied this idea to construction of neural controllers for a legged robot. And we showed that our canalization method can successfully prevent the previously obtained abilities from drastic impairing. Further works are listed below:

- We will expand this approach so that multi-legged locomotion on rough terrain can be realized. We are also planning to construct a real experimental six/four legged robot to verify the validity of our proposed method.
- To clarify the advantage of our proposed method, we will investigate the difference and the limitation of the canalization process by comparing between parametric and structural changes quantitatively.
- We will construct neural controllers by this canalization process on hardware level with the use



reflex 1 reflex 2 reflex 3

Figure 9: Resultant behavior of the neural controller obtained at the 900th generation.



500th generation 700th generation 900th generation

Figure 10: Transition of the neural structures through the proposed canalization process.

of FPAA (Field Programmable Analog Array). FPAA is expected to be more suitable for constructing neural networks on electronic circuit level rather than FPGA (Field Programmable Gate Array). We envision that this approach will be another fruitful stream in the Evolvable Hardware (EHW) community.

These are currently under investigation.

References

[1] S.Nolfi, D.Parisi: "Evolving non-Trivial Behaviors on Real Robots: an Autonomous Robot that Picks up Objects", *Proc.s of the Fourth Congress of the Italian Association of Artificial Intelligence*, pp.243-254 (1995)

[2] S.Nolfi: "Evolving non-Trivial Behaviors on Real Robots: a garbage collecting robot", *Technical Report, Institute of Psychology, Rome*(1996)

[3] F.Gomez, R.Miikkulainen: "Incremental Evolution of Complex General Behavior", *Technical Report AI96-248*, Department of Computer Sciences the University of Texas at Austin (1996)

[4] M.Lewis, A.Fagg, and A.Solidum: "Genetic programming approach to the construction of a neural network for control of a walking robot", *Proc. of the IEEE International Conference on Robotics and Automation*, pp.2618-2623 (1992)

[5] Z.Ding, M.Nelson: "A Neural Controller For Single-Leg Substrate-Finding: A First Step Toward Agile Locomotion In Insects And Robots", *The Neurobiology of Computation, Proc. of the 3rd Annual Computation and Neural Systems Conference*, ed. J. M. Bower, pp. 379-384 (1995)

[6] T.Omata, M.Matsudaira: "Step by Step Learning of a Skill with Assistance", *Proc. of the 24th Intelligent System Symposium*, pp.39-44 (1997) (in Japanese)

[7] J.Kodjabachian, J.Meyer: "Evolution and Development of Control Architectures in Animats", *Robotics and Autonomous Systems*, 16:2-4, pp.1-29 (1996)

[8] J.Kodjabachian, J.Meyer: "Evolutionary Design of an Artificial Neural Network Controlling the Locomotion of a Six Legged Animat", *Notes of the ICML'96 Workshop on Evolutionary Computation and Machine Learning*, pp.87-94 (1996)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Towards a Paradigm for Information Integrated Control System

Shigeyasu KAWAJI
Dept. of Computer Science
Kumamoto University
Kurokami 2–39–1
Kumaoto,860 Japan
TEL: +81-96-344-3631
FAX: +81-96-344-3631
E-mail:kawaji@eecs.kumamoto-u.ac.jp

Nobutomo MATSUNAGA
Kumamoto R&D Center
OMRON Corporation
Mashikimachi Tabaru 2081–17
Kumamoto, 861-22 Japan
TEL: +81-96-289-2226
FAX: +81-96-289-2235
E-mail:matunaga@aipc.kml.omron.co.jp

Abstract

The term "intelligent control" has been a fashionable topic in control literature. But there is an inherent difficulty in answer of what intelligent control is. Integration of all the information about the control problem will be an important key to construct an intelligent control system. In this paper, we will recall what control problem is and discuss a paradigm for Information Integrated Control System.

1 Introduction

Recently there has been a strong upsurge of activity in the field of "intelligent control" with various attempts at a definition of the term, e.g., Shoureshi[1], Antsklis[2], White and Sofge[3], Gupta and Sinha[4]. In the background of this fashion is the fact that, with increasing demands for high precision autonomous control over wide operating envelopes, conventional control approaches have been successfully applied to well-defined and well-structured system and are unable to adequately deal with an ill-defined and ill-structured system. So, it is strongly required to establish a control stratedgy for such an ill system.

Intelligent control is a new emerging discipline that is designed to deal with problems. In the minds of many people, the term intelligent control has come to mean some form of control using fuzzy, neural network, genetic algorithm etc. This perception has been reinforced by a number of papers. However, does intelligent control restrict itself only to those methologies? In fact, according to some definitions of intelligent control, not all fuzzy/neural/GA controllers would be considered intelligent. In fact, there exist control problems which can not be formulated in those framwork.

Ill-defined and ill-structured system has complexity, nonlinearities, spatial and temporal parameter variations, and with uncertainty. Thus, the information about the plant is piecewise, uncertain, incomplete, variable, and the information about environment is alike. Also, control objective is diverse. Therefore, it is necessary to establish a technology which puts all the information about the control subject to good use in control system design. We call this approach as "Information Integrated Control (IIC). A general characterization of IIC is given as follow: IIC system has the ability to act appropriately in an uncertain environment, where an appropriate action is that which increases the probability of the achievement of behavioral subgoals that support the system's ultimate goal. For IIC, a new paradigm is required to deal with various information parallely, disparsively, and integratedly. And for construction IIC system, addition to two approaches of model-based approach and heuristic approach such as fuzzy/nural/GA, new sophisticated functions are needed depending on the problem.

In this paper, to discuss a paradigm for IIC system, we first review the traditional control methologies from the view point of general design equation. Then, we discuss a hierachical architecture for IIC system.

2 The need for a new approach

As is well known, "control" is to add necessary operations (decision making) to the given plant so that the given objective is achieved. The general design equation is then descrived as

$$[S] \ : \ \{\underline{O}bjective\} = \{\underline{C}ontroller\} \ \times \ \{\underline{P}lant\}$$

To solve the equation, we must have the best choice of the descriptions of plant $[P]$ and the objective $[O]$. As a result, we get a realizable solution of the controller $[C]$. But, in general various descriptions are appeared in $[P]$ or $[O]$, qualitatively or quantitively.

IIC system or so-called intelligent control system provides a general architecture for [C] to achieve highly autonomous behavior specified for every ill-defined and ill-structured system. The principal features of such com-

plex control control problems appear to include:

- high degree of nonlinearity
- models are uncertain
- many equilibrium points may exist
- linearized systems are uncontrollable or weakly controllable
- the dynamics changes abruptly
- multiple time scale system
- various type of model descriptions, complexty of information
- distributed sensors and actuators
- large amount of sensory data
- heiararchical and distributed desision architechures
- only partial dynamics is identified

For a closed-loop system,the general design equation is described as

$$[O] = inv[par[I, inv[ser[P, C_c]]]]. \quad (1)$$

where $[O]$ represents closed-loop specifications and $inv(\#)$ is the inverse system of $\#$. $ser(a,b)$ and $par(a,b)$ are serial and parallel conection of a and b, respectively. Solving (1), we get

$$C_c = ser[inv[P], inv[par[-I, inv[O]]]] \quad (2)$$

We remarked from (2) that in general design equation the following properties hold:

1. Any description of $[P]$, $[C]$ and $[O]$ are acceptable, but the operation in equation must be possible and the solution realizable.
2. The structure of the controller $[C]$ reflects the specification of the control objectives$[O]$ and characteristic properties of the plant P.
3. The non-causal model of the plant, $inv[P]$, is used.

By description arrangement of $[P]$, $[O]$, many kinds of control methods such as robust, fuzzy, neural network, genetic algorithm, etc., have been proposed. These methods are characterized as shown in Fig.1.

In the first quadrant where the plant information is rich and the objective is described quantitively, model based approaches have been applied. In the fourth quadrant where the plant information is rich but the objective is qualitiv, classic control approaches such as PID control have been applied. The advantage of the traditional approaches in first and fourth quadrant is to use the rich information of the plant and to derive the implicit results for well-defined plant using the mathematical model.

In the third quadrant where the plant information is poor and the objectives is qualitive, the system is needed to be controlled by human interference. Therefore, heuristic or experienced algorithm such as fuzzy

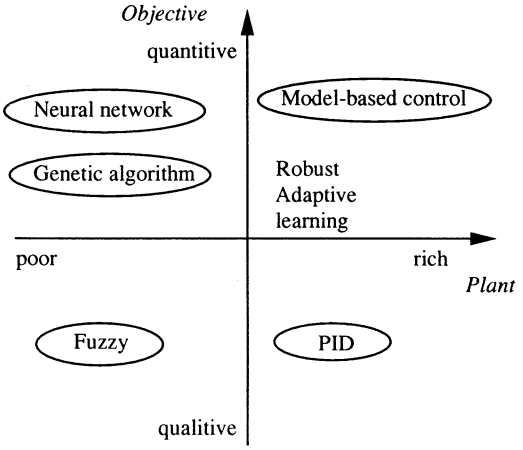


Fig.1: Various control metholodies

control has been applied. In second quadrant where the plant information is poor and the objectives can be described quantitively, the control with the mechanithm of the learning or adaptation, e.g. NN and GA, have been applied. In those controls, the evaluation of the performance will be a key.

It is remarked that the conventional intelligent control approach is almostly positioned in second and third quadrants. But, information integration to supply the poor information about the plant will be important.

3 The candidate tools for IIC

In this section, we examine robust/fuzzy/neural/GA functionally as candidate tools for IIC by applying an inverted pendulum which has been widely used as a bentimark to verify control algorithms.

3.1 Model based control

With increasing demands for high precision control over wide range envelopes, the behavior of plant must be completely expressed, i.e., the mathematical framewark may be needed. If the plant model containes any uncertainties, the deteriorated performance is set as robust control, adaptive and learning control.

For an inverted pendulum[5], the control objective $[O]$ is "to keep the state variables $\{ x, \dot{x}, \theta, \dot{\theta}\}$ at the origin". To get a solution, we usually derive the exact mathematical model for $[P]$ and set the evaluation function for $[O]$.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt. \quad (3)$$

Then, the controller $[C]$ is obtained as the optimal solution of J as follows

$$[C]: \quad u = -R^{-1} B^T P x \quad (4)$$

where P is the solution of the Ricatti's equation. In

the controller design, Q and R are specified by the designner. Remark that the controller has the structure which minimize the evaluation function J and the plant characteristics.

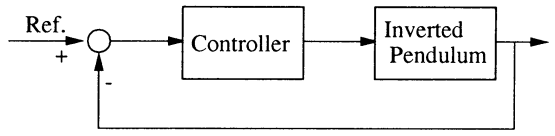


Fig.2: Control system using model-based control

3.2 Fuzzy control

In fuzzy control, $[O]$ is the linguistic expression as " to keep the pendulum in its upright position and the cart in the specified position of the monorail simultaneously". And $[C]$ is represented as production rules R_i

$$R_i : If\ e\ is\ A_i\ and\ \Delta e\ is\ B_i\ then\ u\ is\ C_i \tag{5}$$

where A_i and B_i are fuzzy quantities representing process mesurements and C_i is a fuzzy quantity representing a control signal.

For an inverted pendulum, if the objective is restricted to keep the pendulum in its upright position, a production rule is simply given as "if the pendulum topple to the right, then move the cart to the right". But in the case of stabilizing both the pendulum and the cart, the structual property of the system must be clear to refine the knowledge.

In [6], fuzzy controller for the pendulum was first designed based on VSS and the virtual reference for the pendulum from the structual property was generated to control the cart position (Fig.3). The constructed control system has a hierachical structure. This implies that in the control of complicated system a hiearchical structure is requisite to reflect the dynamical property of controlled object in control rules.

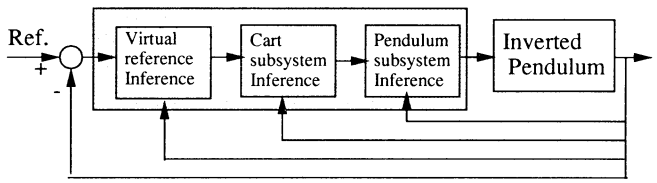


Fig.3: Control system using fuzzy control

3.3 Neural network

In neural network control, no knowledge of the controlled object is usually assumed to be available, i.e., it is treated as the black box. This means the relation between the controlled object and their controller is not obvious. The function of NN is described as follows.

$$\begin{aligned} \Delta C_n j &= BP[\Delta\{evaluation\}_n] \\ \{C\}_{n+1} &= \{C\}_n + \Delta C_n \end{aligned} \tag{6}$$

where $\{C\}_n$ is the controller at the n-th step, ΔC_n is the correction term. The expression infers the improvement of initial value of C and the evaluation for backpropagation(BP).

For an inverted pendulum, it is not realistic to learn the control law using NN as the black box, because the dynamics of the pendulum is complex and the generation of the evaluation signal should be tricky. In [7], we embeded the "sub-unit involved the mechanical structure" into the controller in order to make the NN controller simple. The control system using NN controller is shown in Fig.4.

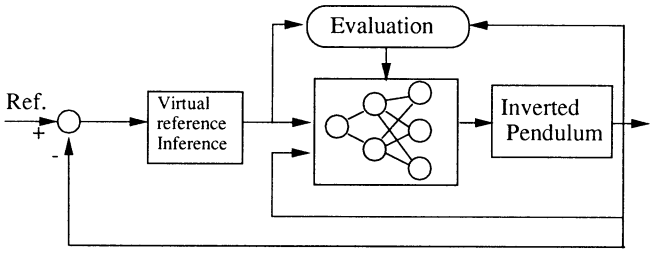


Fig.4: Control system using NN control

3.4 Genetic algorithm

Genetic algorithm (GA) is a kind of random search algorithms modeled after the genetic evolution process. The search process of GA is executed in a population of some search points (individuals) which are renewed stochastically by using genetic operators such as selection, crossover and mutation. Probability with which an individual leaves its offspring in the new population is determnied according to its fitness value aasigned by the fitness function(FF). In order to apply GA to control problems, the problem should be transformed into an optimization problem. Moreover, a proper fitness function must be designed which leads to the solution of the problem.

For an inverted pendulum, a sequence of bang-bang-zero inputs is coded to a chromosome,and the fitness function is designed so as to assign larger fitness value to the individuals that move the pendulum colser to the upright position[8]. The function of GA is described as follows.

$$\begin{aligned} \Delta C_n &= FF[\Delta\{evaluation\}_n] \\ \{C\}_{n+1} &= \{C\}_n + \Delta C_n \end{aligned} \tag{7}$$

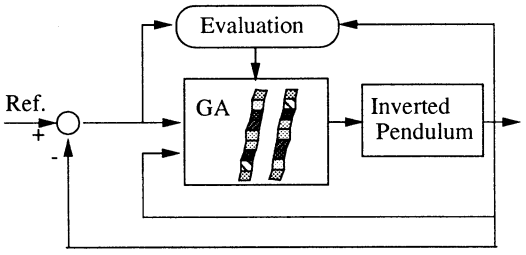


Fig.5: Control system using GA control

4 A basic architecture of IIC system

In order to cope with complexity of control problem, IIC system must have an appropriate functional architecture or structure for efficient analysis and evaluation of control strategies. As is understood in Section 3, when the plant information is poor, some integration of information is needed to supply poor information for soluting the problem. And "hierarchy" is an essential structure to cope with complexity. The term hierarchy refers to functional hierarchy where concepts and methods in many fields are combined and extended such as control, computer science, sensing etc.. Basically, three-level hierachical system has been considered in the literature: execution level, coordination level and organization level. But how three levels are constructed for the control problems is an open problem.

In this section, we outline a hierarchical control system for bibed locomotion robot[9] as an example of IIC system, and make clear "how to construct a hierarchical control system?" and "how to incorporate the human's knowledge in the design of control system?".

In general, controller is combined with reference generator and servo controller. Therefore design of reference generator desides the system's performace. In oder for the robot to walk stably, the following control tactics are proposed.

T1) to modify the reference of supporting foot in order to control the inclination,
T2) to modofy the given path based on the inclination information.

The tactics T1) aims to control the inclination of robot when the robot stands on one leg, because the kick motion of the robot control only the initial movement in frontal plane. The tactics T2) aims to learn the characteristics of the floor, because the reactive force changes according to the characteristics of the floor.

These tactics stand on the motivation that the reference path is determined roughly based on the gross movement of biped rocomotion and the path should be modified hierarchically based on the gross information. In real cases, the exact model of the robot including the environment is merely obtained, so fuzzy inferences was introduced to modify the path on two levels in the hierarchical control system (Fig.6).

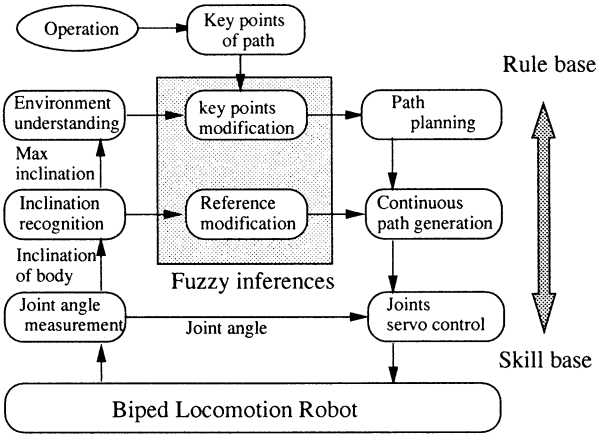


Fig.6: Hierarchical control for biped locomotion

5 Conclusion

To realize Integrated Information Control System, a number of major milestoneswill have to be passed on the way. The authors believe that a hierarchical approach will be most effective and useful to construct IIC system, in which the top level is concerned with "human intelligence" and the lowest level with " machine intelligence". How the middle level is designed using information about control problem is a great problem and this may be true intelligent.

References

[1] R.Shoureshi et al., Special Section on Intelligent Control, IEEE Control Systems Magazine, pp. 33-46, 1991.
[2] P.Antsaklis et al., Defining Intelligent Control, IEEE Control Systems Magazine, pp.64-66, 1994.
[3] D.White et al., Handbook of Intelligent Control, van Nostrand Reinhold, 1992.
[4] M.M.Gupta, N.K.Shinha,*Intelligent Control System*, IEEE Press, 1996
[5] K.Furuta et al.,Mechanical System Control, Ohm-sha, 1984
[6] S.Kawaji et al.,"A Systematic Design of Fuzzy Control for Inverted Pendulum", *J. of Robotics and Mechatronics*, vol.7, pp.45-51, 1995
[7] S.Kawaji et al.,"Learning Control of an Inverted Pendulum using Neural Network", *Proc.31st IEEE CDC*, Tucson, pp.2734-3739, 1992
[8] S.Kawaji et al."Swing-up Control of a Pendulum using Genetic Algorithms", *Proc. of 33rd IEEE CDC*, Florida, pp3530-3532, 1994
[9] S.Kawaji et al.,"Hierarchical Control of Biped Rocomotion Robot using Fuzzy Inferences", *Proc. AMC '94*, San-Francisco, pp421-430, 1994

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Intelligence based on the adaptive control

Seiichi Shin
Course of Mathematical Engineering and Information Physics
School of Engineering, the University of Tokyo
Tel +81-3-3812-2111 ext. 6891
Fax. +81-3-5689-5751
e-mail shin@crux.t.u-tokyo.ac.jp

Abstract

This paper considers the problem of intelligence in adaptive control. It is treated here as the inverse problem, successive estimation, interaction of estimation and control. Moreover, we consider an interaction between adaptive systems based on the decentralized adaptive control systems.

1. Introduction

It is the problem here what is intelligence in the control. This paper will consider the problem from the viewpoint of adaptive control. The adaptive control is counted as one of intelligent control systems, because of its learning ability.

We start the consideration at the point of estimation and we presents a first definition of intelligence of adaptive control as the inverse problem. The intelligence is strictly necessary to solve the problem.

There are only two method to solve the inverse problem. One is to use an inverse operation as the subtraction to the addition, the division to the multiplication, the exponential function to the logarithm, and so on. Another way is successive calculation driven by the error.

Estimation of the adaptive control is done ordinary in online manner, so that the estimation is a sort of successive calculation. It is a try and error method and we can also find here an intelligence, because the intelligence is needed when we are failed and we do not need it when we are succeeding.

Next, we seeks the intelligence in an interaction between the control and estimation, from CE type to Dual adaptive control. Moreover, we treat here an interaction between adaptive system through the decentralized adaptive control system with fault tolerance. The other controllers cooperate to keep the total system stable if one of the controllers is failed.

Finally we reveal the self similarity is one of key point of intelligence in control system.

2 Inverse Problem

The adaptive controller can be divided in two parts (Fig. 1). One is an estimator for unknown parameters in the controlled object and another is a generator, where the control input is synthesized based on the estimated parameter given by the estimator part.

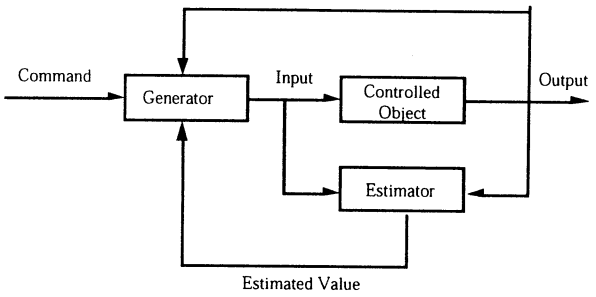


Fig. 1 Adaptive controller consists of the estimator part and the generator part.

The generator part does not seem to be intelligent, because all controllers have the part. Therefore, estimation is the intelligence of the adaptive control systems. The part is also included in the neural control and the repetitive learning control, so that they are family of the adaptive control.

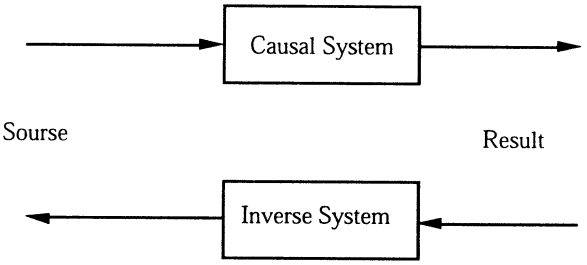


Fig 2. Intelligence stays at the inverse problem.

Kitamori defines the intelligence as an inverse problem [1], where a source is obtained from a result. This is a reverse way, since source induces result in the natural way. We do not need any intelligence if we live along the natural way and intelligence emerges if we think a reason or source of a result.

Estimation is one of the inverse problems since a system characteristic is calculated from input and output data of the system in the estimation problem, where the characteristic to be estimated is the source and the data are results obtained.

3 Successive Estimation

However, the adaptive control has a special estimation property, that is an online estimation. The estimation part

updates its estimated value at every instance when new data are obtained. It works continuously or intermittently. We can get a solution of the inverse problem by an inverse operation or successive calculation. The inverse operation gets a right solution directly, as a subtraction to addition, a division to multiplication and so on. However, almost inverse problemes can not be solvede in this manner.

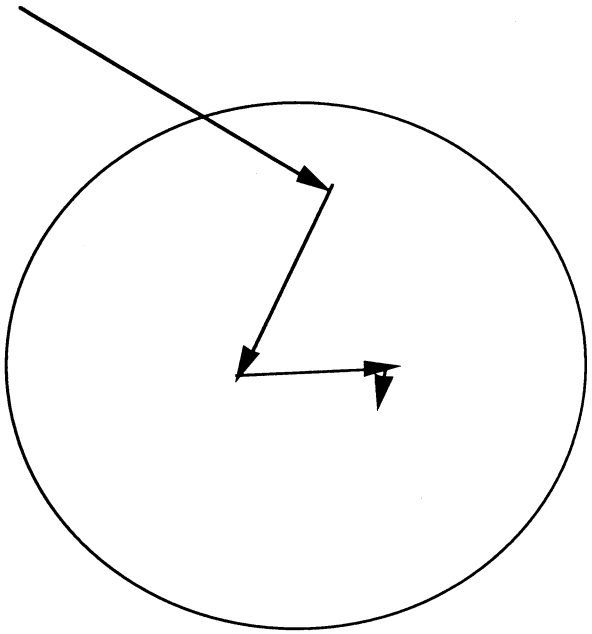


Fig. 3 Successive calculation in the estimator.

Successive calculation ordinary needs infinite calculations and presents mere an approximation. Error is significant in this successive estimation method. The estimator part gives an estimation as a trial and the estimation is updated at the next instance based on the error between a real value and the estimation. Therefore, this is a so cold "try and error" method. We can feel an intelligence on the way of the calculation caused by the error. In this case, the intelligence seems how to cope with the error and not to avoid the error.

4 Interaction

Interaction between the estimator part and the generator part seems to be another intelligence which we can find in the adaptive control. Feldbaum revealed that there are three types of adaptive control along the intelligence [2]. One is CE (Certainly Equivalence) type, where the generator synthesize the input believing the estimated value as a true vale. This is the simplest method and widely used in many adaptive control systems even in the industrial applications. However, insufficient estimation may cause a big transient error. Therefore, we should be careful enough in the selection of initial estimates when we will apply the CE type to the industrial applications. The generation part starts to synthesize the input after the estimates are converged in reasonable values where the estimation part has been started in advance.

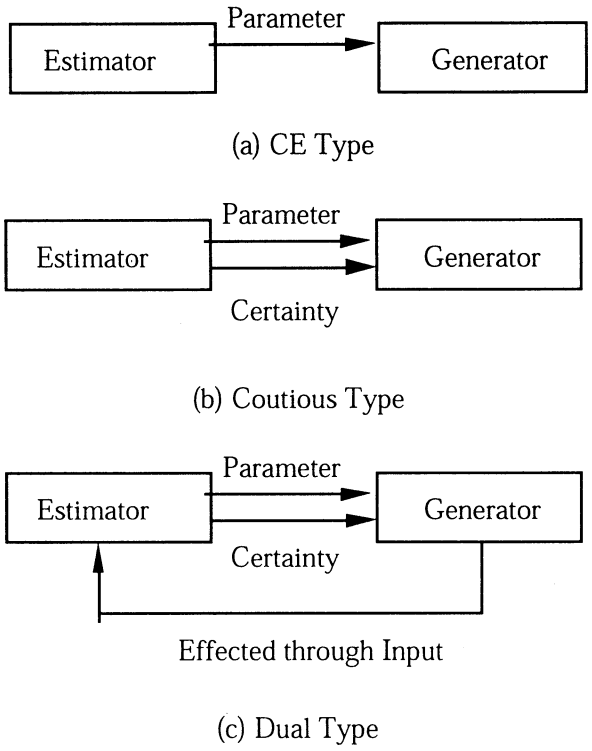


Fig. 4. Three types of adaptive control system.

The next type of adaptive control is cautious type. This control works automatically in the careful start as mentioned above. The generation part does not rely on the estimates perfectly, especially in the early stage of estimation. The control input is suppressed in this stage based on the information of covariance matrix. The suppression may avoid the big transient error.

However, this cautiousness may cause a dead lock in the adaptation, because the estimation will be done based on the error and the generation part of the cautious adaptive controller suppress the error. Estimation sometimes remains poor in this type of control system. Therefore, the third type of adaptive control is necessary. This is the dual type control. A feature of the control is a bidirectional interaction between the generator and estimator. The generator makes the control input not only for the suppression of the control error and also for estimation. The generator makes the big transient error if the estimation part needs the error and suppress the error if not. These two parts watches each other at every instance, so that we say the controller as dual.

5 Distributed intelligence

Next, we focus on the interaction between adaptive controllers, whereas the interaction in one adaptive controller is mentioned above. This is a sort of decentralized adaptive control systems (Fig. 5). These

controller cooperates to control multivariable controlled object, which consists of subsystems mutually interacted. The adaptive controllers ordinary estimate parameters of the corresponding subsystems and interactions and synthesize control inputs independently based on these estimates.

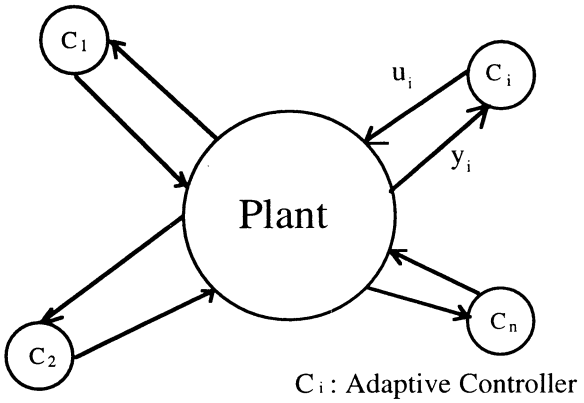


Fig. 5. Decentralized adaptive control.

In this case, an intelligent exists in each controller and we can see cooperation of intelligent machines through estimation and control of the decentralized adaptive control system. We added fault tolerance property to the decentralized adaptive control system and say it Autonomous Decentralized Adaptive Control System (ADACS) [3-4]. The cooperation can be seen in ADACS in comparison with the ordinary decentralized adaptive control, because a substitution of a certain function is necessary in ADACS when a controller is failed.

Intelligence is distributed in these decentralized control system since each controller has the intelligent part, which is plasticity based on environment. A controller in the decentralized adaptive control system gathers sensor information from the controlled object, which is an environment in this case. And the controller has the tunable gains corresponding to estimates of parameters in the environment. Therefore, the unable gains stand for intelligence in the decentralized adaptive control system.

The adaptive control rises a new problem of intelligence in the adaptive system, because the other adaptive systems are included in the environment. The adaptive controller faces to the other adaptive controllers since sensor information is affected by behavior of other controllers. The controllers should consider decisions of other controllers. It has an aspect of game.

6. Decentralized Adaptive Control

Here, the decentralized adaptive control system proposed in [3] is briefly introduced. The decentralized adaptive control system is further than the simple collection of one loop controllers. It consists of those independent controllers whose own parameters are adjustable by oneself based on the local input and the output information.

The plant to be stabilized is,

$$y_i(t)= g_i(s)\{u_i(t)+d_i(t)+ \sum_{j=1,\ j\neq i}^{n} y_j(t) \qquad i= 1,2,\cdots,n \tag{1}$$

where, s is the time differential operator and $g_i(s)$is an unknown transfer function of the corresponding local system. Scalars, $y_i(t)\in R$ and $u_i(t)\in R$ respectively, are the local output and the local input of the controlled object, where R means the field of real numbers. $d_i\in R$ are bounded disturbance.

The control objective considered here is to decrease $y_i(t)$ as small as possible under the unknown disturbances d_i. The control law used here is,

$$u_i(t)= k_i(t)y_i(t) \tag{2}$$

where $k_i(t)$ is an adjustable gain. Therefore, we treat only the proportional local feedback. This brings us to a simple analysis of the total control system. The gain is tuned as,

$$sk_i(t)= y_i(t)^2-\sigma_i k_i(t) \tag{3}$$

where σ_i is positive constant.

This is a σ-modified adaptive law, so that this is a sort of the robust adaptive law. The first term of the right hand side of (3) makes the gain become high. If $\sigma_i= 0$, the gain $k_i(t)$ would diverge under the disturbances. To avoid this divergence, we should take a positive value of σ_i. The second term forces the gain to zero, so that this law takes a trade-off between magnitudes of the error and the gain. It means the individual controllers suppress their optimization on the error minimization and this self-suppression leads to the global stability of the whole system

When a fault occurs in the i-th controller with the gain $k_i(t)$, the output of the controller becomes zero and remains zero as follows,

$$u_i(t)= 0 \tag{4}$$

We can prove the global stability of the total system [3,4] under a suitable condition even if the fault occurred in one of the controllers, so that we say that the decentralized adaptive control system can cooperate each other and we can find an interaction of intelligences here.

We are force to considere the self-similarity in analysis and design of these decentralized control system because

one controller finds another controllers similar to each other. The controller exisiting inside find itself in outside.

7. Self-Similarity

Based on the result obtained in the former sections. we consider ADACS generally. The self suppression is one of crucial points of ADS as mentioned before. The other points is self-similarity in the whole.

The self-similarity is a feature appeared in fractul and chaos. A fractul is shown in Fig. 7. The fractul X is defined by a subset satisfies

$$X = f_1(X) \cup f_2(X) \cup f_3(X) \tag{5}$$

in R^2. This definition of contains itself, so that it is one of self-similarity.

We have shown that the autonomous Decentralized adaptive system is effective if the plant satisfies decentralized stability condition as Theorem 1. A set of plants which satisfy the condition is defined as S and function $f_F(\bullet)$ stands for one fault. Then, a set of plants which keep stability irrespective of the fault can be described by,

$$X_1 = f_F(S) \cap S \tag{6}$$

In the same way, The stable set irrespective of at most two faults can be described by,

$$X_2 = f_F(X_1) \cap X_1 \tag{7}$$

Although number of faults in the controller is restricted less than number of controllers, we can assume infinite number of faults if the fault means replacement of parts each other. In this case, a stable set of plants irrespective of any number of faults is the set X_F which satisfies,

$$X_F = f_F(X_F) \cap X_F \cap S \tag{8}$$

It is a self-similarity, so that the fault tolerance considered here leads to the self-similarity. In the other words, The fault tolerance can not achieved if the total system do not have the feature of self-similarity, which means no change in the whole is occurred when a part is removed or a new part is added. Therefore, the self-similarity is a key point whether a system is ADACS or not.

8. CONCLUSION

We considers intelligences in adaptive control, that is, intelligence in estimation, intelligence in interactions of intellectual modules, including estimator, generator and whole adaptive controllers.

REFERENCE

[1] T. Kitamori, "The essential feature of measurement and measurement engineering," J. SICE, vol. 26, no. 2, pp. 145-152 (1982)

[2] B. Wittenmark, "Stochastic adaptive control method: A survey," Int. J. Control, vol. 21, no. 5, pp. 705-730 (1975)

[3] K. Ikeda, S. Shin and T. Kitamori, "A design of decentralized high gain adaptive control systems," Tr. of SICE, vol. 28, no. 5, pp. 555-563 (1992)

[4] K. Ikeda and S. Shin, "Design of Decentralized adaptive control system with Back stepping," Tr. SICE, vol.32, no.10, pp.1399-1406, (1996)

[5] R. Tanaka and S. Shin, "Autonomous controllability of hierarchical decentralized systems," Tr. IEE of Japan, vol. 117-c, no. 7, pp. 850-855 (1997)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Intelligence Based on Fuzzy Control

Kazuo TANAKA

Department of Human and Mechanical Systems Engineering
Kanazawa University
2-40-20 Kodatsuno Kanazawa 920 Japan
Email ktanaka@t.kanazawa-u.ac.jp
TEL & FAX +81-76-234-4736

1 INTRODUCTION

This paper presents a model-based fuzzy control which is a new research direction of fuzzy control. The advanced fuzzy control is realized by combining human intelligence with a systematic design method.

2 ADVANCED FUZZY CONTROL

Sugeno [1] emphasizes that the essence of "fuzzy" concept is a "human linguistic" approach. This can be also observed in Zadeh's paper [2], where fuzzy sets are regarded as linguistic variables. Mamdani [3], who first applied fuzzy logic to a control problem, called fuzzy control "linguistic control". Thus, it may be concluded that intelligence in fuzzy control is based on "linguistic rules". Actually, fuzzy controller design begins with constructing linguistic variables (fuzzy sets) and if-then control rules.

Fuzzy control has been applied to many industrial processes. Recently, we have witnessed two directions of new research of fuzzy control. One is related to a model-based fuzzy control technique concerning stability analysis and systematic design methods. The other is related to realization of "super" intelligent control. The author defined the two new research directions as "advanced fuzzy control" [4].

This paper mainly discusses a model-based fuzzy control technique utilizing LMI control performance. The author and co-workers first stated in [5, 6] that the stable fuzzy controller design proposed in [7] can be solved numerically, i.e, a convex optimization technique based on LMIs can be utilized to determine feedback gains. Hence LMIs play an important role in the model-based fuzzy control.

3 MODEL-BASED FUZZY CONTROL

Figure 1 shows the procedure of the model-based fuzzy control. Suppose that we have a nonlinear plant to be controlled. It is possible to design a nonlinear controller using one of nonlinear control techniques. However, nonlinear control techniques require special rather than involved knowledge. A fuzzy controller can be easily designed if the nonlinear plant is represented via a Takagi-Sugeno type fuzzy model (T-S fuzzy model) [8]. In fact, the stability analysis and the design procedure of the model-based fuzzy control are simple and natural [4].

Fuzzy model has two kinds of representation: linguistic (if-then rule) representation and nonlinear equation representation. The linguistic representation helps us (designers and engineers) understand the (global or local) dynamics of the fuzzy model and the (global or local) behavior of the fuzzy controller. On the other hand, the defuzzification process of the T-S fuzzy model is represented by a nonlinear equation. Therefore, analysis and design for the fuzzy control system can be realized using the nonlinear equation. Thus, two kinds of representation link to two completely different worlds: mathematical world and intelligent world. In the model-based fuzzy control approach, fuzzy logic bridges a big gap between the mathematical world and the intelligent world.

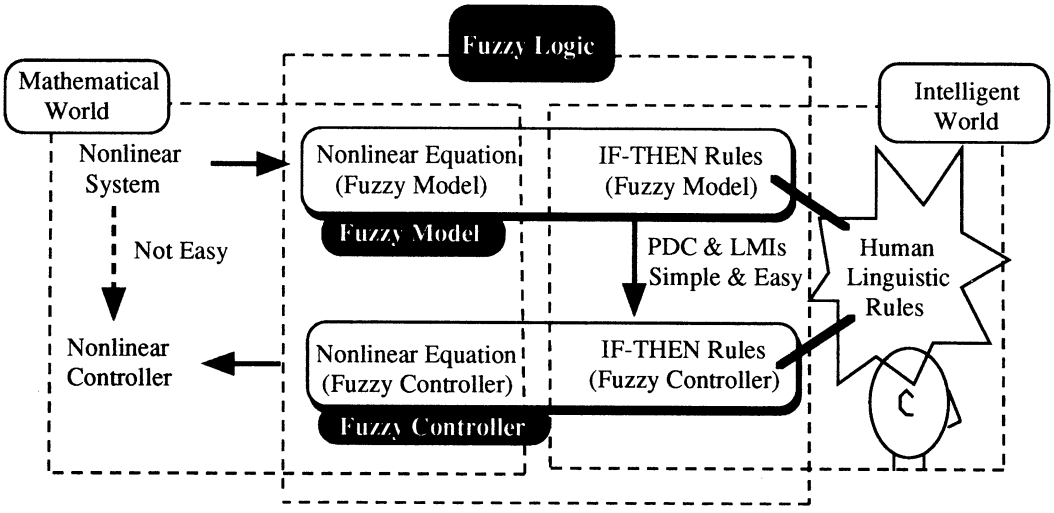


Figure 1 Model-based fuzzy control.

The model-based fuzzy control approach begins with constructing a T-S fuzzy model for the nonlinear plant. Once the fuzzy model is obtained, the control design is carried out via the so-called parallel distributed compensation (PDC) scheme. The design procedure is conceptually simple and natural. The stability analysis and control design problems are reduced to linear matrix inequality (LMI) problems [5,6,10-13] where powerful numerical algorithms can be applied. Moreover the design procedure renders (globally or semiglobally) stabilizing controllers accompanied by desired performance for the closed-loop system.

3.1 T-S Fuzzy Model and PDC

The Takagi-Sugeno fuzzy model [8] is represented as follows:

Plant Rule i :

IF $z_1(t)$ is M_{i1} and $\cdots$ and $z_p(t)$ is M_{ip}

$$\text{THEN} \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_i\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}_i\mathbf{x}(t) \end{cases} \quad i = 1,2, \cdots, r \quad (1)$$

where $\mathbf{x}(t) \in \mathbf{R}^n$, $\mathbf{u}(t) \in \mathbf{R}^m$, $\mathbf{y}(t) \in \mathbf{R}^q$. M_{ij} is the fuzzy set and r is the number of IF-THEN rules. $z_1(t)$~ $z_p(t)$ are the premise variables and are any variables which are independent of the input variables $\mathbf{u}(t)$.

The idea of the PDC is that for each local linear model, there is an associated linear feedback control. The resulting overall controller, which is nonlinear in general, is the fuzzy blending of each individual linear controllers. Hence the PDC approach is a multiple control with automatic switching via fuzzy logic rules. The PDC provides the fuzzy rule structure (2) for the fuzzy model (1).

Control Rule i :

IF $z_1(t)$ is M_{i1} and $\cdots$ and $z_p(t)$ is M_{ip}

$$\text{THEN} \quad \mathbf{u}(t) = -\mathbf{F}_i\mathbf{x}(t) \quad i=1,2,\cdots,r \quad (2)$$

The fuzzy controller design is to determine the local feedback gains $\mathbf{F}_i$ in the consequent parts. Thus, the PDC is simple and natural. Other nonlinear control techniques require special and rather involved knowledge.

3.2 Stability and Relaxed Stability Conditions

The stability conditions derived in [7, 9] can be converted into the following LMIs [6]:

$$\mathbf{X} > \mathbf{0}$$
$$-\mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} + \mathbf{M}_i^T\mathbf{B}_i^T + \mathbf{B}_i\mathbf{M}_i > \mathbf{0}, \quad (3)$$
$$-\mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} - \mathbf{X}\mathbf{A}_j^T - \mathbf{A}_j\mathbf{X} + \mathbf{M}_j^T\mathbf{B}_i^T + \mathbf{B}_i\mathbf{M}_j + \mathbf{M}_i^T\mathbf{B}_j^T + \mathbf{B}_j\mathbf{M}_i \geq \mathbf{0}, \quad i < j \quad (4)$$

where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_i = \mathbf{F}_i\mathbf{X}$. From the solutions $\mathbf{X}$ and $\mathbf{M}_i$, the feedback gains $\mathbf{F}_i$ can be obtained as $\mathbf{F}_i = \mathbf{M}_i\mathbf{X}^{-1}$.

As shown in (3) and (4), the stability analysis of the fuzzy control system is reduced to a problem of finding a common $\mathbf{X} = \mathbf{P}^{-1}$. If r is large, it might be difficult to find a common $\mathbf{X} = \mathbf{P}^{-1}$ satisfying the above conditions. Recently, the author and co-works [10] relaxed the stability conditions (3) and (4) as follows:

[Relaxed stability condition]

Assume that the number of rules that fire for all t is less than or equal to s, where $1 < s \leq r$.

$$\mathbf{X} > \mathbf{0}, \quad \mathbf{Y} \geq \mathbf{0}$$
$$-\mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} + \mathbf{M}_i^T\mathbf{B}_i^T + \mathbf{B}_i\mathbf{M}_i - (s-1)\mathbf{Y} > \mathbf{0}, \quad (5)$$
$$2\mathbf{Y} - \mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} - \mathbf{X}\mathbf{A}_j^T - \mathbf{A}_j\mathbf{X} + \mathbf{M}_j^T\mathbf{B}_i^T + \mathbf{B}_i\mathbf{M}_j + \mathbf{M}_i^T\mathbf{B}_j^T + \mathbf{B}_j\mathbf{M}_i \geq \mathbf{0}, \quad i < j \quad (6)$$

where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_i = \mathbf{F}_i\mathbf{X}$, $\mathbf{Y} = \mathbf{X}\mathbf{Q}\mathbf{X}$.

Note that the stability conditions (5) and (6) are reduced to the previous stability results (3) and (4) when $\mathbf{Q} = \mathbf{0}$. In other words, (5) and (6) are relaxed conditions.

3.3 LMI-based Design

We show that the decay rate and the constrains on control input and output can be represented via LMIs [11]. By solving these LMIs together, feedback gains of the fuzzy controller accompanied by desired performance for the closed-loop system can be obtained.

[Decay Rate]

$$\underset{\mathbf{M}_1 \cdots \mathbf{M}_r}{\textit{maximize}} \ \alpha$$

subject to

$$\mathbf{X} > \mathbf{0}, \quad \mathbf{Y} \geq \mathbf{0}$$
$$-\mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} + \mathbf{M}_i^T\mathbf{B}_i^T + \mathbf{B}_i\mathbf{M}_i - (s-1)\mathbf{Y} - 2\alpha\mathbf{X} > \mathbf{0}, \quad (7)$$
$$2\mathbf{Y} - \mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} - \mathbf{X}\mathbf{A}_j^T - \mathbf{A}_j\mathbf{X} - 4\alpha\mathbf{X} + \mathbf{M}_j^T\mathbf{B}_i^T + \mathbf{B}_i\mathbf{M}_j + \mathbf{M}_i^T\mathbf{B}_j^T + \mathbf{B}_j\mathbf{M}_i \geq \mathbf{0}, \quad i < j \quad (8)$$

where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_i = \mathbf{F}_i\mathbf{X}$, $\mathbf{Y} = \mathbf{X}\mathbf{Q}\mathbf{X}$.

The above design problem satisfying the decay rate is reduced to the stable controller design problem (3) and (4) if we solve (7) and (8) for $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{M}_i$ with $\alpha = 0$ and $\mathbf{Q} = \mathbf{0}$. In other words, a fuzzy controller that satisfies the LMI conditions of (7) and (8) is a stable fuzzy controller. Therefore, it also satisfies the stability conditions (3) and (4).

[Input Constraint]

Assume that the initial condition $\mathbf{x}(0)$ is known. The constraint $\|\mathbf{u}(t)\|_2 \leq \mu$ is enforced at all times $t \geq 0$ if the LMIs (9) hold, where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_i = \mathbf{F}_i\mathbf{X}$.

$$\begin{bmatrix} 1 & \mathbf{x}(0)^T \\ \mathbf{x}(0) & \mathbf{X} \end{bmatrix} \geq \mathbf{0}, \quad \begin{bmatrix} \mathbf{X} & \mathbf{M}_i^T \\ \mathbf{M}_i & \mu^2\mathbf{I} \end{bmatrix} \geq \mathbf{0} \quad (9)$$

[Output Constraint]

Assume that the initial condition $\mathbf{x}(0)$ is known. The constraint $\|\mathbf{y}(t)\|_2 \leq \lambda$ is enforced at all times $t \geq 0$ if the LMIs (10) hold, where $\mathbf{X} = \mathbf{P}^{-1}$.

$$\begin{bmatrix} 1 & \mathbf{x}(0)^T \\ \mathbf{x}(0) & \mathbf{X} \end{bmatrix} \geq \mathbf{0}, \quad \begin{bmatrix} \mathbf{X} & \mathbf{X}\mathbf{C}_i^T \\ \mathbf{C}_i\mathbf{X} & \lambda^2\mathbf{I} \end{bmatrix} \geq \mathbf{0} \quad (10)$$

4 DESIGN EXAMPLES

4.1 A Mobile Robot with Two Trailers [12]

Backward movement control of articulated vehicles is difficult because of nonlinearity and "jackknife" phenomenon. To succeed backward movement of articulated vehicles, the "jackknife" phenomenon should be avoided. Two trailer case, i.e., backward movement control of a vehicle with two trailers, is much more difficult than the one trailer case. As far as we know, experiment results of the two trailers have not been reported yet. Figures 2 and 3 show the vehicle model with two trailers and its photograph, respectively. With respect to $x_1(t)$ and $x_3(t)$, 90 [deg.] and -90 [deg.] correspond to four "jackknife" positions. The control purpose is to realize the backward movement along the straight line (x_5=0) without forward.

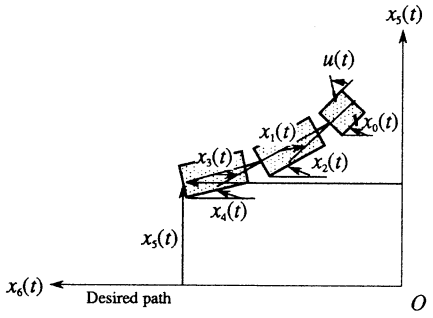


Figure 2 Vehicle model with two trailers.

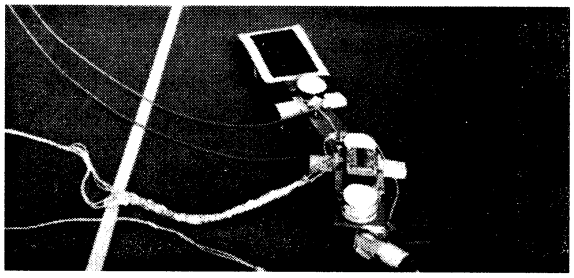


Figure 3 Photograph.

Figure 4 shows an experimental result. The feedback gains are obtained by solving the LMIs with respect to the decay rate and the constraints on the control input and the output. The model-based fuzzy control effectively realizes the backward movement control of the vehicle with two trailers.

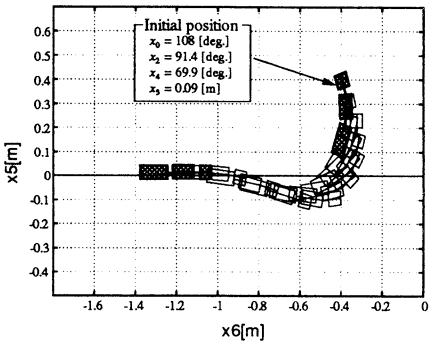


Figure 4 Experimental result.

4.2 Fuzzy Regulators and Fuzzy Observers [11]

Linear regulators and linear observers play an important role in modern control theory and practice. We envision that a systematic design method of fuzzy regulators and fuzzy observers would be important for fuzzy control as well.

A fuzzy control system containing a fuzzy observer and a fuzzy regulator can be also designed in the same control framework. The fuzzy observer (11) is constructed using the PDC, where $\hat{\mathbf{x}}(t)$ denotes the state vector estimated by the fuzzy observer.

Observer Rule i :

IF $z_1(t)$ is M_{i1} and $\cdots$ and $z_p(t)$ is M_{ip}

$$\text{THEN} \quad \begin{cases} \dot{\hat{\mathbf{x}}}(t) = \mathbf{A}_i\hat{\mathbf{x}}(t) + \mathbf{B}_i\mathbf{u}(t) + \mathbf{K}_i(\mathbf{y}(t) - \hat{\mathbf{y}}(t)) \\ \hat{\mathbf{y}}(t) = \mathbf{C}_i\hat{\mathbf{x}}(t) \end{cases} \quad (11)$$

$$i = 1, 2, \cdots, r$$

The fuzzy observer design is to determine the local gains $\mathbf{K}_i$ in the consequent parts. The fuzzy regulator (2) and the fuzzy observer (11) to stabilize the fuzzy model (1) can be designed by solving the following LMIs:

Assume that the number of rules that fire for all t is less than or equal to s, where $1 < s \le r$. The largest bound on the decay rate that we can find using a quadratic Lyapunov function can be found by solving the GEVP in a common positive definite matrix $\mathbf{P}$, a common positive semidefinite matrix $\mathbf{Q}$ and α.

maximize α

subject to $\beta > 0$, $\alpha > 0$

$$\mathbf{P} = \begin{bmatrix} \beta\mathbf{P}_1^{-1} & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} > \mathbf{0}, \quad \mathbf{Q} = \begin{bmatrix} \beta\mathbf{Q}_{11} & \mathbf{Q}_{21} \\ \mathbf{Q}_{12} & \mathbf{Q}_{22} \end{bmatrix} \ge \mathbf{0} \quad (12)$$

$$\mathbf{P}_1\mathbf{A}_i^T - \mathbf{M}_{1i}^T\mathbf{B}_i^T + \mathbf{A}_i\mathbf{P}_1 - \mathbf{B}_i\mathbf{M}_{1i} + (s-1)\mathbf{Y} + 2\alpha\mathbf{P}_1 < \mathbf{0} \quad (13)$$

$$\mathbf{A}_i^T\mathbf{P}_2 - \mathbf{C}_i^T\mathbf{N}_{2i}^T + \mathbf{P}_2\mathbf{A}_i - \mathbf{N}_{2i}\mathbf{C}_i + (s-1)\mathbf{Q}_{22} + 2\alpha\mathbf{P}_2 < \mathbf{0} \quad (14)$$

$$\mathbf{P}_1\mathbf{A}_i^T - \mathbf{M}_{1j}^T\mathbf{B}_i^T + \mathbf{A}_i\mathbf{P}_1 - \mathbf{B}_i\mathbf{M}_{1j} - 2\mathbf{Y} + 4\alpha\mathbf{P}_1 + \mathbf{P}_1\mathbf{A}_j^T - \mathbf{M}_{1i}^T\mathbf{B}_j^T + \mathbf{A}_j\mathbf{P}_1 - \mathbf{B}_j\mathbf{M}_{1i} < \mathbf{0} \quad i < j \quad (15)$$

$$\mathbf{A}_i^T\mathbf{P}_2 - \mathbf{C}_j^T\mathbf{N}_{2i}^T + \mathbf{P}_2\mathbf{A}_i - \mathbf{N}_{2i}\mathbf{C}_j - 2\mathbf{Q}_{22} + 4\alpha\mathbf{P}_2 + \mathbf{A}_j^T\mathbf{P}_2 - \mathbf{C}_i^T\mathbf{N}_{2j}^T + \mathbf{P}_2\mathbf{A}_j - \mathbf{N}_{2j}\mathbf{C}_i < \mathbf{0} \quad i < j \quad (16)$$

for all i and j, where $\mathbf{M}_{1i} = \mathbf{F}_i\mathbf{P}_1$, $\mathbf{N}_{2i} = \mathbf{P}_2\mathbf{K}_i$ and $\mathbf{Y} = \mathbf{P}_1\mathbf{Q}_{11}\mathbf{P}_1$. A common $\mathbf{P}$, a common $\mathbf{Q}$, $\mathbf{M}_{1i}$, $\mathbf{N}_{2i}$ and $\mathbf{Y}$ can be found by using a convex optimization technique [15,16] if they exist. From the solutions, the feedback gains and the observer gains are obtained as $\mathbf{F}_i = \mathbf{M}_{1i}\mathbf{P}_1^{-1}$ and $\mathbf{K}_i = \mathbf{N}_{2i}\mathbf{P}_2^{-1}$. For more detail, see [11].

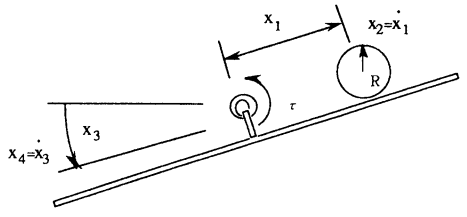


Figure 5 Ball and beam example.

This design is applied to the ball and beam example shown in Figure 5. Figure 6 shows a control result of the ball and beam example, where the dotted line denotes the state variable estimated via the fuzzy observer.

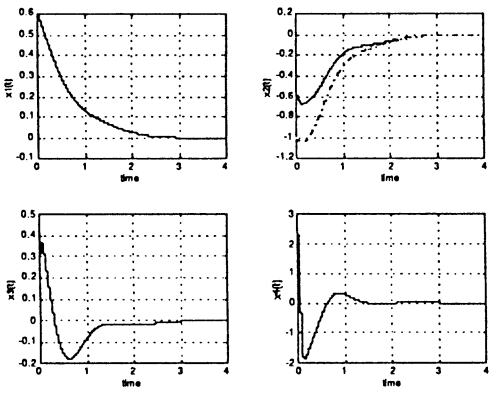


Figure 6 Control result.

4.3 Controlling Chaos [13]

Since the OGY method [14], a method of controlling chaos, was proposed by Ott, Grebogi and Yorke, many methods on controlling chaos have been attempted in linear or nonlinear control framework. We show that even the control of chaotic systems can be also achieved in the same control framework mentioned above. A chaotic system [13], Lorenz's equation with three inputs, is stabilized using the model-based fuzzy control technique. Figure 7 shows the control result, where the control input is added at t > 10 (sec).

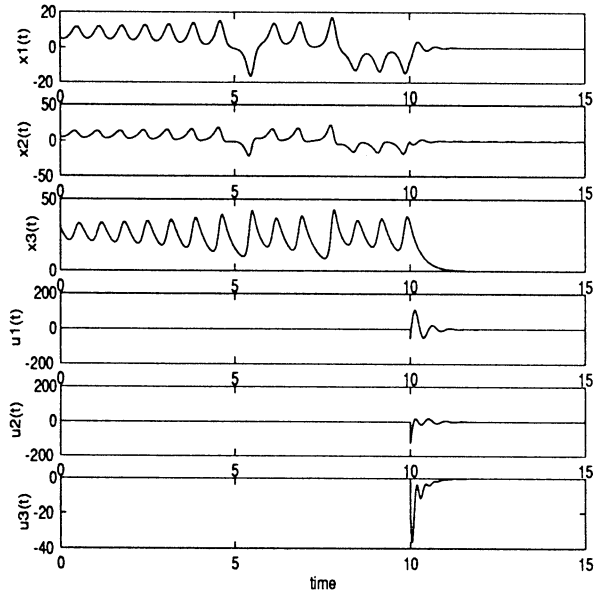


Figure 7 Stabilization.

The chaotic model following control (CMFC), which is more difficult than the stabilization and the standard synchronization problems, can be also realized in the same control framework [13].

5 CONCLUSIONS

This paper has presented a model-based fuzzy control technique using LMIs. We envision that the model-based fuzzy control will be a very important framework for intelligent-nonlinear control theory and practice.

REFERENCES

[1]M.Sugeno, "Fuzzy Control", Nikkan-Kougyou (1988) in Japanese.

[2]L.A.Zadeh, The Concept of a Linguistics Variable and its Application to Approximate Reasoning, Part 1, Information Sciences, 8, pp.199-249 (1975).

[3]E.H.Mamdani:Applications of Fuzzy Algorithms for Control of Simple Dynamic Plant, Proc. IEE, Vol.121, No.12, pp.1585-1588 (1974)

[4]K. Tanaka,"Advanced Fuzzy Cotrol", Kyoritu (1994) in Japanese.

[5]H. Wang, K. Tanaka and M. Griffin, "Parallel Distributed Compensation of Nonlinear Systems by Takagi and Sugeno's Fuzzy Model.", Proceeding of FUZZ-IEEE '95, pp.531-538 (1995).

[6]H. Wang, K. Tanaka and M. Griffin,"An Approach to Fuzzy Control of Nonlinear Systems : Stability and Design Issues", IEEE Transactions on Fuzzy Systems, Vol.4, No.1, pp.14-23 (1996).

[7]K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," Fuzzy Sets and Systems, Vol. 45, no. 2, pp. 135-156 (1992).

[8]T.Takagi and M.Sugeno, Fuzzy Identification of Systems and Its Applications to Modeling and Control, IEEE Trans. on SMC 15, no.1 (1985) 116-132.

[9] Tanaka, K. and M.Sugeno (1990). Stability Analysis of Fuzzy Systems Using Lyapunov's Direct Method. *Proc. of NAFIPS'90*, pp. 133-136.

[10]K.Tanaka, T. Ikeda and H.Wang, Design of Fuzzy Control Systems Based on Relaxed LMI Stability Conditions, in Proceedings of 35th IEEE Conference on Decision and Control, pp.598-603 (1996).

[11]K.Tanaka, T.Ikeda and H.Wang,"Fuzzy Regulators and Fuzzy Observers:Relaxed Stability Conditions and LMI Based Designs",, IEEE Transactions on Fuzzy Systems (To appear).

[12]K. Tanaka, H. Wang and T. Taniguchi, "Fuzzy Control of a Vehicle with Two Trailers", 1997 American Control Conference, New Mexico, pp.2861-2862 (1997).

[13]K. Tanaka, T. Ikeda and H. Wang,"A Unified Approach to Controlling Chaos via an LMI-based Fuzzy Control System Design", IEEE Transactions on Circuits & Systems-1, (submitted).

[14]E.Ott, C.Grebogi and J.A.Yorke,"Controlling chaos", Physical Review Letters, Vol.64, pp.1196-1199, 1990.

[15]Boyd, S., *et al.* (1994). Linear Matrix Inequalities in Systems and Control Theory. *SIAM*. Philadelphia.

[16]Yu. Nesterov and A. Nemirovsky, "Interior-point polynomial methods in convex programming", SIAM, Philadelphia, PA (1994).

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Intelligence based on Neuro-Control

Sigeru Omatu
Department of Computer and Systems Sciences,
College of Engineering, Osaka Prefecture University,
Sakai, Osaka 593, Japan
Email:omatu@cs.osakafu-u.ac.jp

Abstract-----In this paper, we consider intelligence based on neuro-control. Neuro-control is a control scheme to construct the controller based on the neural networks. Neural networks are based on the human brain architechture for information processing and we could expect some additional potentials over the conventional control theory. Some of them are pattern recognition ability, nonlinear representation ability, and learning ability. We will summarize the neuro-control approach to aquire the intelligence from the environment.

1. Introduction

Computational intelligence like fuzzy theory, neural network, genetic algorithm, artificial life, etc. has been well-developed and applied to many real control problems in an efficient way[1]-[2]. As in real control processes, the PID controller has been used as a major control method and the operators are selecting a suitable PID gains from time to time based on their experience and knowledge, it is required to find a suitable PID gains automatically. The main reason is that its structure is simple and PID controllers are robust with noise and parameter variations. But to use the controller, we must tune the PID gains which have been determined by experience and knowledge. In this paper, we propose a tuning method of the PID gains by using neural networks and apply it to torque control of an electric vehicle.

2. Neur-control approach

Recently, many neuro-control schemes have been proposed [1]-[3] where a neural network can be trained to perform as a controller by learning an inverse model of the plant or as an emulator by identifying the forward model. Among many neural network learning methods, the back-propagation algorithm[1] is the most widely used in a wide variety of applications. In control problems, there has been development on neuro-control for robots control problems. Although various intelligent control methods may be applied to this problem, PID control is a major approach since it is robust to noise and stable for parameter change.

Using the back-propagation method, various kinds of neuro-controllers could be trained in such a way that the desired plant output is attained as much as possible. The basic control scheme is depicted as Fig.1 where feedforward controller(FFC) and feedback controller(FBC) are adjusted. The neuro-controller is used in FFC or FBC instead of a conventional controller. In Fig.2 three typical neuro-controllers are illustrated where (a) is called a series type, (b) is called a parallel type, and (c) is called a self-tuning type. In this paper, we consider the self-tuning type of (c) in Fig.2 when FFC is neglected and the PID structure is adopted for FBC.

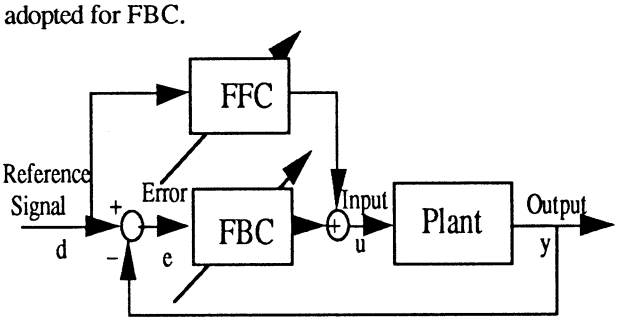


Fig. 1. General control scheme.

In order to construct the neuro-control systems, we use the following three types[1] as shown in Fig.2. The first one is called series type and the neuro-controller realizes the inverse dynamics of the plant. The second one is called series type and the insufficient control inputs could be adjusted by using the neuro-controller. The feedback-errorlearning method by Kawato et al. [4]belongs to this type. The third one is called self-tuning type and the control parameters are adjusted such that the desired signal could be produced. The typical one of this type is self-tuning PID controller which has been developed by the authors. In the following section, we discuss the structure in more detail about the third type.

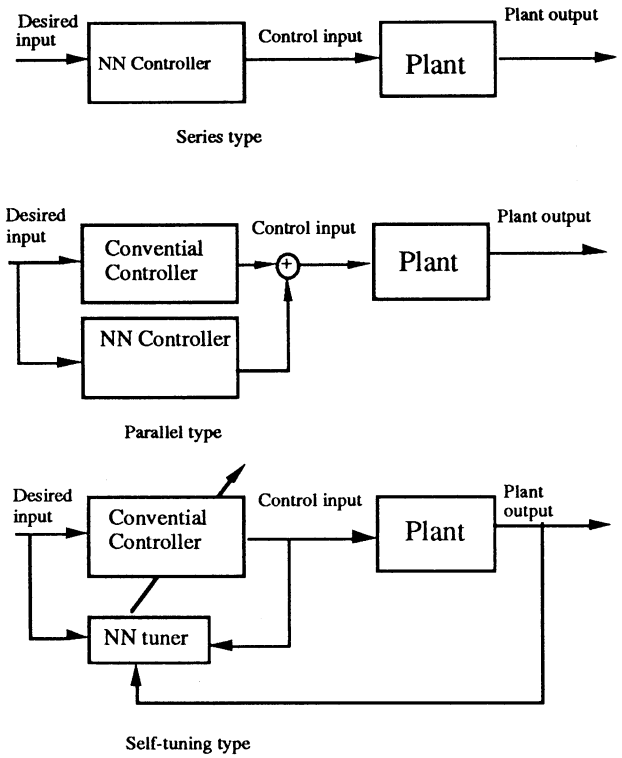


Fig.2. Three types of neuro-controllers

3. Self-tuning neuro-PID Controller

The PID controllers have been adopted in the real plant control, especiaaly, the process control since they have simple control structures and we can select the control parameters based on the physical meanings such as propotional, integral, and derivative actions. The control structure is shown in Fig.3.

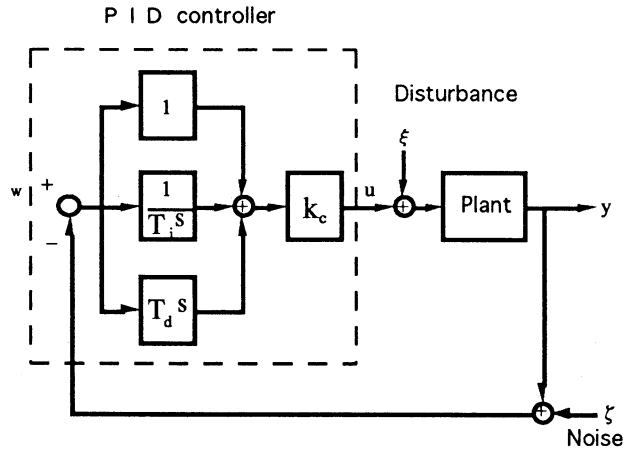


Fig. 3. PID control structure.

The control structure of the self-tuning neuro-PID controller can be shown in Fig.4 where the outputs of the neural network are proportional(P), integral(I), and derivative(D) gains and the inputs are selected in a suitable way according to the specific problem.

In order to derive the self-tuning neuro-PID control algorithm, we will define mathematical notations of neuron models. Total input to neuron j is denoted by *net* j and given by

$$net_j = \sum_{i=1}^{n} w_{ji} O_i + \theta_j$$

where θ_j denotes a threshold. Then the output of neuron i becomes

$$O_j = f(net_j), \quad f(x) = \frac{1}{1+e^{-x}}$$

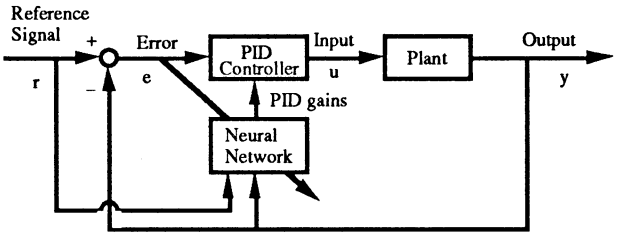


Fig. 4. Self-tuning neuro-PID control scheme.

By connecting many neurons in a layered array, we can derive a learning rule based on the gradient method, which is called the error-back-propagation(BP) method. The BP method is described as follows:

Step 1. Find the generalized error δ_k at the output layer

$$\delta_k = (t_k - O_k) f'(net_k)$$

and the generalized error at the hidden layer

$$\delta_j = f'(net_j) \sum_{k=1}^{K} \delta_k w_{kj}$$

where t_k is the target pattern, O_k is the actual output of neural networks, $f'(x)$ denotes a derivative of $f(x)$ with respect to x, and w_{ji} denotes the connection weight from neuron i to neuron j.

Step 2. Change the connection weights by w_{ji} in the following way.

At the output layer

$$\Delta w_{kj}(n+1) = \eta \delta_k O_j + \alpha \Delta w_{kj}(n) + \beta \Delta w_{kj}(n-1)$$

and at the hidden layer

$$\Delta w_{ji}(n+1) = \eta \delta_j O_i + \alpha \Delta w_{ji}(n) + \beta \Delta w_{ji}(n-1)$$

where a modified BP method has been used to avoid the local minimum problem.

Step 3. If the error becomes smaller than a predetermined value, then stop and if not, then n is replaced by n+1 and go to *step* 1.

Using the above BP algorithm, we can derive the self-tuning neuro-PID control algorithm. First, note that

the PID controller in discrete-time is described as

$$u(n) = u(n-1) + K_P\ (e(n)-e(n-1)) + K_I\ e(n) + K_D\ e(n)-2e(n-1)+e(n-2)) \quad (1)$$

where K_P , K_I , and K_D are proportional, integral, and derivative gains, respectively, u(n) denotes the plant input at nT, and e(n) is the error between a desired value r(n) and the actual output y(n), that is,

$$e(n) = r(n) - y(n).$$

Here, T denotes a sampling interval.

In order to adjust K_P, K_I, and K_D adaptively, we use a three layered neural network based on the BP method. Each layer consists of N_1 , N_2 , and N_3 neurons where N_1 and N_2 can be selected by trial and error according to complexity of the plant and N_3 is three which are equal to the number of PID gains. For the BP method, we need a cost function E which should be minimized. Here, the following E is adopted such that the PID controller can decrease the error e(n+1) at (n+1)T,

$$E = \frac{1}{2}\ e(n+1)^2.$$

The connection weights w_{kj} at the output layer are updated by the following relation.

$$\Delta w_{kj}(n) = \eta \delta_k O_j + \alpha\, \Delta w_{kj}(n-1) + \beta \Delta w_{kj}(n-2)$$

where

$$\delta_k = -\ \frac{\partial E}{\partial net_k}\ .$$

Using the chain rule for derivative, we get

$$\delta_k = e(n+1)\ \frac{\partial y(n+1)}{\partial u(n)}\ \frac{\partial u(n)}{\partial O_k}\ \frac{\partial O_k}{\partial net_k}$$

Furthermore, we have from (1)

$$\frac{\partial w(n)}{\partial O_k} = \begin{cases} e(n)-e(n-1), & \text{for } k=1 \\ e(n), & \text{for } k=2 \\ e(n)-2e(n-1)+e(n-2), & \text{for } k=3 \end{cases}$$

where $O_1 = K_P$, $O_2 = K_I$, and $O_3 = K_D$.

If the plant is unknown, two approximation algorithms to calculate the system Jacobian $\frac{\partial y(n+1)}{\partial u(n)}$.

The first one is to use the following relation:

$$\frac{\partial y(n+1)}{\partial u(n)} = \text{sign}\left(\frac{\partial y(n+1)}{\partial u(n)}\right)\left|\frac{\partial y(n+1)}{\partial u(n)}\right|.$$

Thus, if we know only the sign of the Jacobian , we can adjust the learning rate η such that $\left|\frac{\partial y(n+1)}{\partial u(n)}\right|$ is included into η.

The second way is to build the emulator which can simulate the input/output map of the plant. Then we regard the emulator as the true plant. Using the BP method, we can adjust the connection weights by combining the neural network to tune the PID gains with a neural network to simulate the plant. The total configuration for the second approach is given by adding the emulator to the plant in a parallel way. Using the control scheme, we have obtained fine tuning of the PID gains in many cases. The total configuration of the neuro-PID controller with an emulator is shown in Fig.5 where inputs of the emulator are past plant input and output data.

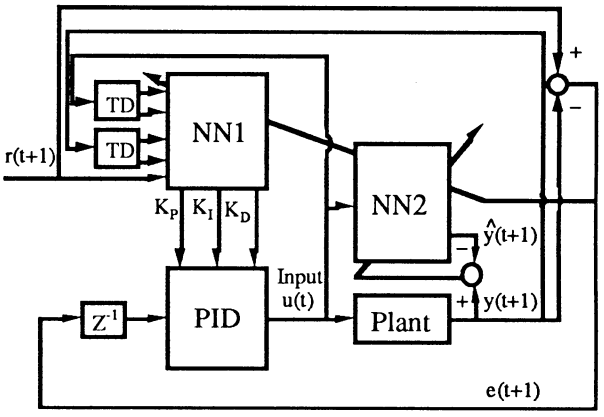


Fig. 5. Self-tuning neuro-PID control structure.

Using the structure of Fig.5, we can adjust the PID control parameter only using the input and output data such that the error between the desired signal and the plant output becomes smaller and smaller.

Note that the NN2 in Fig.5 can be easily realized by using the plant knowledge as shown in Fig.6 where the convetional model means the mathematical model which has been accepted as the physical model of the plant or statistical model. The selection of the physical model or statistical one would be depend on the desiner's will.

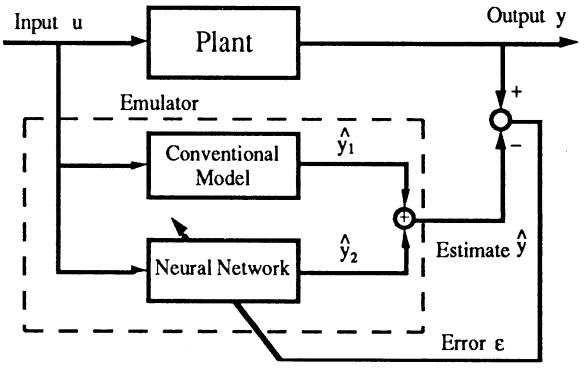


Fig. 5. A method to construct the emulator.

After getting the emulator NN2, then the system

Jacobian can be calcurated by using the the conventional model plus the neural network model. For the latter case, we can use the following relation[5]:

$$\frac{\partial y(t+1)}{\partial u(t)} \cong \frac{\partial \hat{y}(t+1)}{\partial u(t)} = \sum_j w_{ij} O_j'(1-O_j') w_j'$$

where unit i denotes the unit to which the input u(t) is taken as an input in the input layer. Therefore, if we obtain the suitable weights of the emulator, it is easy to find the system Jacobian from the above equation.

To show the effectiveness of the present approach, we show the control results of the inverted pendulum by using the fuzzy control and the proposed control schemes in Figs. 6 and 7, respectively. From these results, the neuro-control appraoch could be used to real control problems in the wide range field applications.

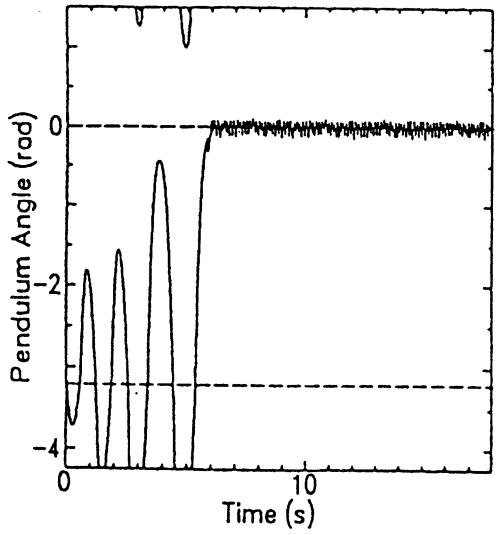


Fig.6. Fuzzy control result of the inverted pendulum.

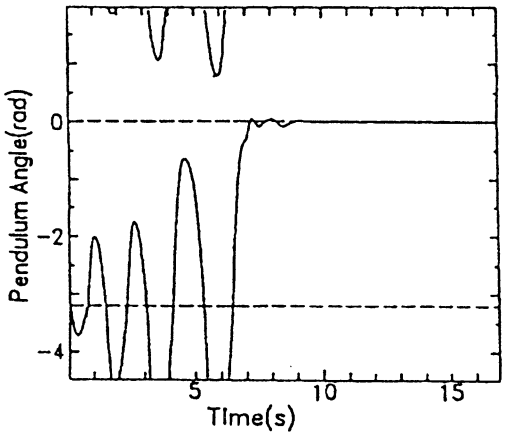


Fig.7. Control result of the inverted pendulum by using the proposed method.

Rerefences

[1] W. T. Miller III, R. S. Sutton, and P. J. Werbos, "Neural Networks for Control", MIT Press, 1990.

[2] D. A. White and D. A. Sofge Edited, "Handbook of Intelligent Control", Neural, Fuzzy, and Adaptive Approaches, Van Nostrand Reinhold, 1992.

[3] S. Omatu, M. Khalid, and R. Yusof, "Neuro-Control and Their Applications", Advances in Industrial Control, Springer-Verlag, 1995.

[4] M. Kawato, Y. Uno, M.Isobe, and R. Suzuki,"Hierarchical neural network model for voluntary movement with application to robotics", IEEE Control Systems Magazine, Vol.8, pp.8-16, 1988.

[5] T. Aoyama and S. Omatu,"Design of a Self-Tuning PID Control System by Neural Networks", Trans. on IEE of Japan, Vol.116-C, No.11, pp.1197-1201, 1996(in Japanese).

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Evolutionary Algorithm for Intelligent Control

Takeshi FURUHASHI
Dept. of Information Electronics, Nagoya University
Furo-cho, Chikusa-ku, 464-01, Japan
Tel.81-52-789-2792, Fax.81-52-789-3166
E-mail: furuhashi@nuee.nagoya-u.ac.jp

Abstract

This paper discusses the feature of application of evolutionary algorithms (EAs) to intelligent control. The distinguishing feature is that EA can discover control know-how. A few examples are shown and the future direction of this research is envisioned.

1 Introduction

Evolutionary Algorithms (EAs) are methodologies inspired by natural evolution. Genetic algorithm, Evolutionary Programming, etc. are the ones of major paradigms of EA. Their power in performing global search have attracted researchers and engineers in many fields. Intelligent control is one of those fields where EA has a high potential for creating new paradigms.

What does EA realize in the field of intelligent control? EA can discover control know-how through interactions with the environment where the control system is immersed. Where there are no experts in controlling a nonlinear plant, EA can find control knowledge.

There have been many research studies on the discovery of control rules using EA. For explicit representation of control rules, production rules are encoded into chromosomes, and the initial primitive rules are evolved to be an sophisticated ones. Fuzzy rules are also employed. Fuzzy logic is considered to be a bridge between numerical information processing and symbolic information processing. Depending on the encoding method of fuzzy rules into chromosomes, the rule finding methods are classified into two groups. One is to encode a control rule into a chromosome. Classifier system [1] is the one to evolve these chromosomes. Each rule is evaluated depending on its contribution to actions. The other is to encode a set of control rules into a chromosome. This approach does not need to evaluate each rule. Pittsburgh approach [2] is classified into this approach. This paper shows two examples of these approaches applied to discovery of control rules [3, 4] and discusses future direction of EA application to intelligent control.

2 Fuzzy Classifier System [3]

2.1 Problem Formulation

The problem here is to steer a ship to reach its goal avoiding other ships. Figure 1 shows examples of success and failure of the steering of the ship. The operator's ship starts from the bottom line, and is directed to the goal on the top line on the screen. Two other ships appear from the left hand side line and right hand side line, and go across in front of the controlled ship. Success means that the controlled ship reaches the goal, and failure means that the ship goes off the screen, or the ship collides with one of the two other ships, or the absolute value of the angle between the direction of the operator's ship and the goal becomes larger than 90 degrees. The FCS is applied to find the steering rules from the results of success and failure.

2.2 Configuration of FCS

The FCS is a learning system that has the following four components: a rule generation mechanism, a fuzzy rule base, a fuzzy inference system, and an apportionment of credit system. The configuration of the FCS is shown in Figure 2. The functions of the components are as follows:

1. **Fuzzy Rule Base** The rule base has fuzzy rules whose inputs are relative distances and angles between the ships, and those between the operator's ship and the goal, and the angular velocity ω. The output of the FCS is the steering angle u. A fuzzy rule in this rule base is, for example, represented in a chromosome as shown in Figure 3. Each locus

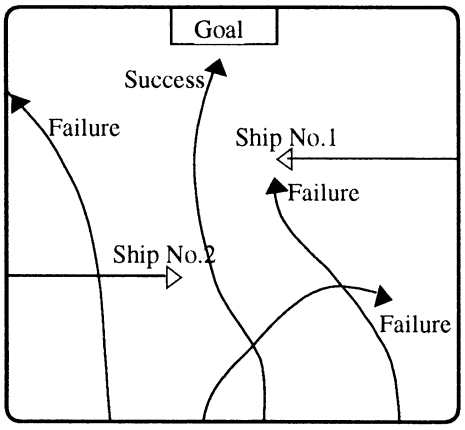


Figure 1: Success and failures of steering

corresponds to one of the membership functions of each variable. The rule in Figure 3 can be read as follows:

IF the distances between the ships
D_1 is Small and D_2 is Big,
the relative velocity angles
ϕ_1 is Positive Small, ϕ_2 is Negative Small,
the angles of the other ships
δ_1 is Negative Small, δ_2 is Positive Big,
the angle to the goal
θ is Negative Big,
the angular velocity of the ship
ω is Zero,
THEN the steering angle
u is Positive Big.

The subscripts 1, 2 mean the variables between the operator's ship and the other No.1, 2 ships, respectively.

2. **Fuzzy Inference System** The steering angle of the operator's ship is outputted from this system. Fuzzy inference is done in this system. The steering of the ship is continued until the steering succeeds or fails.

3. **Apportionment of Credit System** The FCS receives a positive/negative payoff in case of success/failure. This system delivers credits to each rule in proportion to the payoffs. The amount of credit to a rule depends on the contribution of the rule to the actions.

4. **Rule Generation Mechanism** In this component, fuzzy rules are deleted and generated using genetic algorithm (GA). Genetic operations, such as selection, reproduction, crossover and mutation, are applied to the population of chromosomes.

The feature of this system is that the FCS finds rules using such simple payoffs, without being taught detailed rules by humans. Figure 4 shows an example of achieved steering of the ship. The operator's ship succeeded in reaching the goal avoiding two other ships.

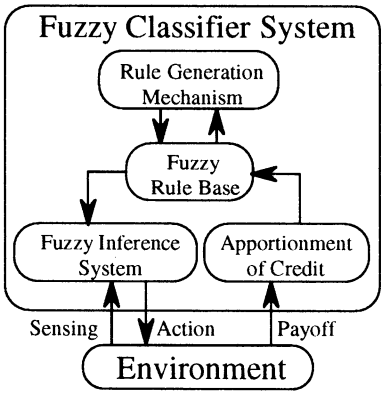


Figure 2: Fuzzy Classifier System

D_1	D_2	ϕ_1	ϕ_2	δ_1	δ_2	θ	ω	u
S	B	PS	NS	NS	PB	NB	ZO	PB

Figure 3: Encoded fuzzy rule

3 Knowledge Discovery by GA [4]

3.1 Encoding Method

An example of application of EA to the discovery of fuzzy control rules is introduced in this section. For fast improvement of local portions of chromosomes, Pseudo-Bacterial GA is used in [4].

Each chromosome in the population encodes the rules of the fuzzy controller as well as the membership functions of the variables. The lengths of the chromosomes are variable. The membership functions used here are triangular, so their parameters are the pairs *(center, width)*. For example, the Rule 4 in Figure 5 means:

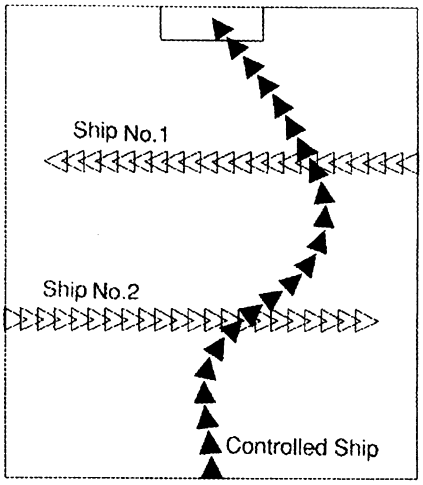


Figure 4: Ship trajectory of acquired steering rules

IF $X_1 = T(4.238, 3.423)$,
$X_3 = T(3.428, 1.213)$,
$X_4 = T(1.382, 0.381)$,
THEN $Y = T(12.112, 9.141)$

where $T(c, w)$ means the triangle-shaped membership function whose center is c and width w and where X_1, X_3 and X_4 are input variables of the fuzzy model and Y is the output variable. This encoding gives a high degree of freedom for the GA, which can define the variables to be used in the rules, the rules themselves and the parameters of the membership functions.

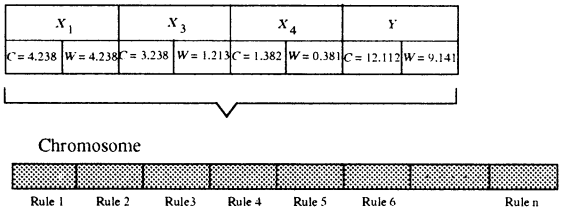


Figure 5: Example of a fuzzy model encoded in a chromosome

3.2 Problem Formulation(Semi-active Suspension System)

Semi-active suspension systems aim to improve the ride comfort and the drivability of cars.

The basic principle of the semi-active control systems is to change in the damping coefficient of the shock absorber (damper) for the ride comfort. The seminal work in the field of semi-active suspension systems was proposed by Karnopp[5] and its objective was only to reduce the vertical acceleration of the body of the car. Human sense of comfort is a nonlinear one. It is not known whether only the acceleration affects the comfort. There have been no theory for the satisfaction of criteria containing not only the acceleration but also various parameters.

GA is promising for discovering control rules for a semi-active suspension system. Figure 6 shows the simulation model of the semi-active suspension system. K is the stiffness of the suspension and C_d is the damping coefficient of the shock absorber.

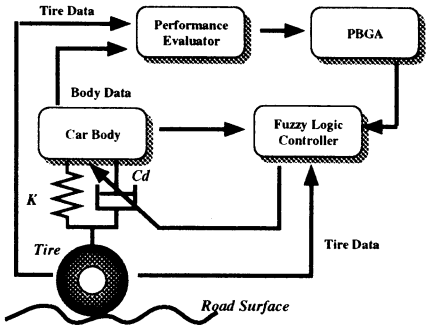


Figure 6: Semi-active suspension model

The dynamics of the simulation process is as follows. Each chromosome encodes a set of rules. All the chromosomes are tested in the model of the suspension system and receives a fitness value from the *Performance Evaluator*, which gives a grade according to the performance index that is being used. Basically, the controllers are supposed to minimize some physical variables of the system, for example, *Body Vertical Position, Body Vertical Velocity, Body Vertical Acceleration, Body Vertical Jerk, etc.* . This process is repeated within the dynamics of the PBGA.

3.3 Simulation Results

Simulations were performed using PBGA with adaptive operator in [4]. 100 chromosomes constituted the population and the process ran through 100 generations. The performance index (PI) used was:

$$PI = peak(B_{def}) + peak(B_{vel}) + peak(B_{acc}) \quad (1)$$

The system tries to minimize the peak values of the parameters of the car body (Deflection, Velocity and Acceleration). In doing so, expectedly higher levels of ride comfort will be achieved.

Figure 7 shows an example of an evolved rule base. The six input variables are, respectively from the left,

the car body vertical position and velocity, tire vertical position and velocity, car body vertical acceleration and tire vertical acceleration. The right most variable is the output variable, namely the damping coefficient of the shock absorber.

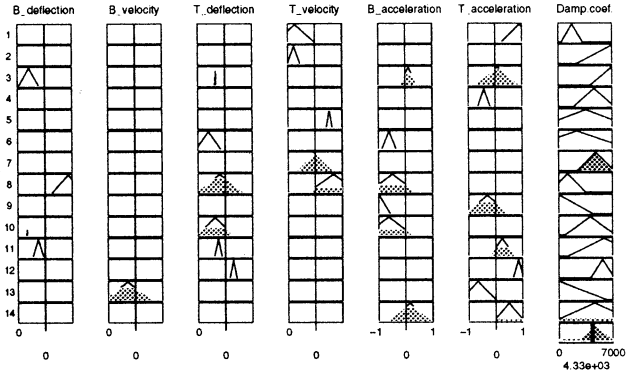


Figure 7: Example of evolved fuzzy control rules

Fourteen rules were acquired. The first rule in the top row, for example, can be read as:

IF $T_{velocity}$ is Small,
$T_{acceleration}$ is Positive Big,
THEN *Damp.Coef.* is Small.

4 Discussions

The examples in sections 2 and 3 showed that EA has a distinguishing feature in that it can find out control rules which satisfy indirect criteria. Intelligent control the author has in mind in this paper is a combination of symbolic information processing and numerical information processing. For this intelligent control, the acquired control rules should be explicitly expressed for combinations with symbolic information processing as illustrated in Fig. 8. The question arises how comprehensible the acquired rules are. The clarity of control know-how and the control performance are often incompatible. The rules in Fig. 7 are not easy for us to understand what kind of control know-how are actually discovered. In the case of large scale systems with many input/output variables, this difficulty will be more remarkable.

Flexible representation of control knowledge for achieving a high control performance and simple expression of control rules for symbolic manipulations should be made compatible.

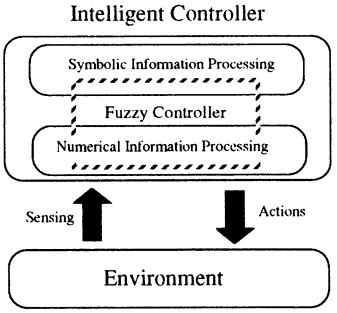


Figure 8: Intelligent control

5 Conclusions

This paper showed the feature of application of EAs to intelligent control was that EAs could discover control know-how. The future challenge clarified in this paper is to make the flexible representation and simple representation of control knowledge be compatible.

References

[1] J. H. Holland, J. S. Reitman, "Cognitive Systems Based on Adaptive Algorithms", in Pattern Directed Inference Systems, D. A. Waterman, F. Hayes-Roth (Eds.), pp.313-329. Academic Press, New York, 1978.

[2] S. F. Smith, "A Learning System Based on Genetic Adaptive Algorithms", Ph. D. Thesis, University of Pittsburgh, 1980.

[3] T. Furuhashi, K. Nakaoka, Y.Uchikawa, "A Study on Fuzzy Classifier System for Finding Control Knowledge for Multi-Input Systems", Studies in Fuzziness and Soft Computing, Vol. 8, pp.489-502, 1996.

[4] N. E. Nawa, T. Hashiyama, T. Furuhashi, Y. Uchikawa, "Fuzzy Logic Controllers Generated by Pseudo-Bacterial Genetic Algorithm with Adaptive Operator", Proc. of 1997 IEEE Int'l Conf. on Neural Networks (ICNN'97), pp.2408-2413, 1997.

[5] D. Karnopp, et. al, "Vibration Control Using Semi-Active Force Generators", *ASME Journal of Engineering for Industry*, vol. 96, No.2, pp.619-626.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A Scaling Law between Number of Multi-robots and Their Task Performance

K. Sugawara I. Yoshihara K. Abe
Graduate School of Engineering
Tohoku Univ.
Sendai 980-77, Japan

Abstract

We research the efficiency of cooperative behavior by interacting active elements. In this paper, we assume a simple robot as an active element and examine the collective behavior through the task of gathering pucks in a field. This robot has a drive system and the simplest means of interaction. The effectiveness of group behavior is investigated for fractal distributions of pucks. To evaluate the efficiency of group behavior, we examine the scaling relation between the task completion time and the number of robots, and the relation between the interaction duration and the efficiency of the group. We also propose a simplified state transition diagram of the group to analyse their characteristics. These results enable us to describe the condition of the field by a variable in the state transition diagram.

1 Introduction

Social insects such as ants and bees establish well-ordered societies even in the absence of particular individual intelligence[1]. Some researchers was interested in their behaviors and studied them using mathematical models[2,3,4,5]. Recently, the cooperative behaviors by many simple elements have been also studied in robotics field. Walter[6] showed that two robots with simple interaction behaves complicatedly, and several experiments upon group behavior of autonomous robots was attempted[7,8,9,10,11]. These researches gave us qualitative aspect of robots movement, however our interest is quantitative aspects of effectiveness of multi-robot group behavior. Simplifying the robot movement and interaction, we attempt quantitative analysis.

2 Foraging Behavior of Multi-robots

In this paper, we investigate the collective behavior and efficiency of simple interacting multi-robots through the task to collect pucks distributed in a field(Figure 1).

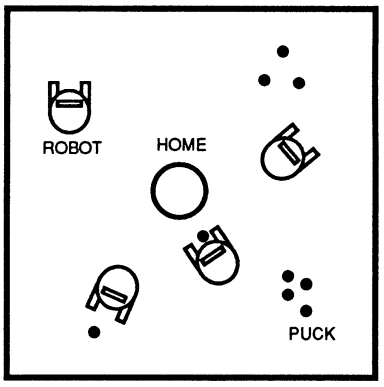


Figure 1: Multi-robots foraging behavior.

This foraging behavior of robots resembles that of ants, but the architecture of the robot is different from that of ant's. Here, simple mechanism is introduced to each robot. Each robot has 3 modes: Searching, Broadcasting and Homing. While searching, each robot moves straight unless it meets a puck, other robots or boundary walls. When a robot encounters a puck, it stops there and radiates a light signal to broadcast its location for a short duration. Other robots, which have no puck, react to this signal and turn towards it using their sensors. Thus robots follow the signal gradient. The robots interact only when a robot radiates a light signal. This duration is called 'interaction duration' in this paper. After the radiation, the robot with the puck stops broadcasting and begins homing. When it arrives home, it drops the puck, changes direction randomly, and goes out to search for pucks again.

3 Robot simulation

3.1 Movement of robots

We studied the efficiency of the group by real robots system and by simulations in a spatially continuous model[12] and discrete[13] model. In this paper, we choose a spatially discrete model to rapid computation. We assume a square field. Home is located at the center. The field is partitioned into 400 × 400 cells. Each cell could have 3 states: 'empty', 'robot' or 'puck'. The position and the direction of each robot are calculated as continuous variables. A cell which includes a robot is occupied and has the state 'robot', implying that the size of the robots is 1 cell. The velocity of the robots is $10 cells/sec$ and the angular velocity is $120 + \theta deg/sec$. When they meet boundary walls and other robots, they change their direction randomly. Moreover, they sometimes change their direction randomly. The probability is 1%. A time step of simulation is $0.1 sec$. The initial location and direction of robots are uniformly random.

3.2 Puck distribution

The task of these robots is to collect pucks distributed in the field. We can assume that various types of puck distribution. Here we chose a fractal distribution because we can discuss the characteristics of the field quantitatively. At first, 400 pucks are distributed by Levy flight method[14]. Levy flight method gives us an arbitrary dimension of distribution. But 400 pucks are too little to make a given dimension of distribution completely. So it is measured again after the flight. The process is as follows: (1) focus on a puck i as a center and consider a circle which radius is r. (2) count the number of pucks inside the circle (denote the number of pucks as $m_i(r)$) by changing the radius r. (3) get the summation $M(r) = \Sigma m_i(r)$ by changing the center puck. (4) determine the dimension D by fitting $M(r) \propto r^D$[14].

3.3 Result of robot simulation

We measure the relation between the number of robots N and a time T to collect all the pucks in the field. A task completion time depends on the number of robots and an interaction duration.

The relation between N and T is not simple because of the finite number of pucks and the frequent collision, but forms a power law relation between N and T approximately:

$$T \sim N^{\beta}. \qquad (1)$$

The exponent β depends on the interaction duration. Figure 2 is an example in case of homogeneously distributed field and extremely localized field.

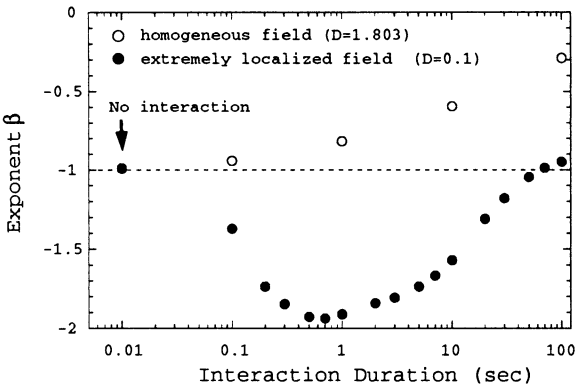


Figure 2: Relation between interaction duration and exponent of the power law, β.

For $\beta = -1$ (dashed line in Figure 2) the completion time and the number of robots are inversely proportional which is expected if each robot works independently. If the exponent β is less than -1, the group behavior is more efficient than multiple single robots. This figure shows there is an optimum value of β. This value of β (here we denote β_{opt}) depends on the dimension of puck distribution. Figure 3 shows the relation between the dimension of puck distribution and the value of β_{opt}.

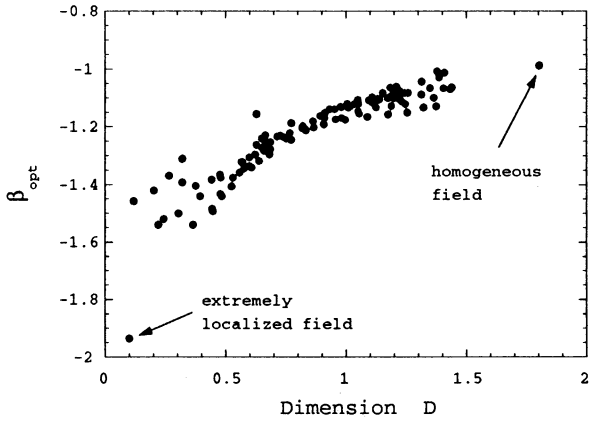


Figure 3: Relation between the dimension of puck distribution and the optimum value of β.

4 Analysis

In this section, presenting a state transition diagram, we analyse the robot behavior. States of the robot are simplified as follows: Searching for pucks, Finding a puck and broadcasting, Returning to home, Moving to signal source. The relation between each state is shown in figure 4.

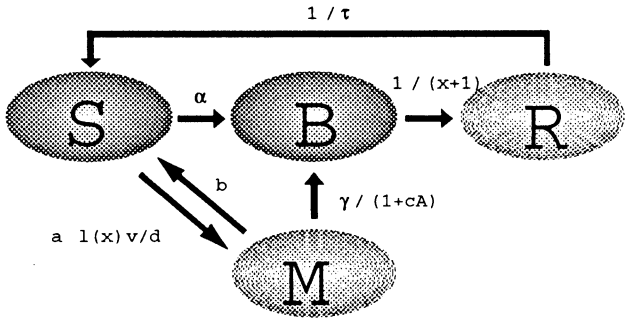


Figure 4: State transition diagram of this system. S,F,R,M represent Searching, Broadcasting, Returning home, Moving to signal source, respectively.

Meaning of variables in this figure is as follows: α: probability to find a puck, b: probability to lose signal source, τ: time to return to home, x: interaction duration, $l(x)$: turn angle, d: average distance between interacting robots, v: velocity of robot, γ: probability to find a puck by following other robots, and S: the number of robots in searching mode, B: the number of robots to find a puck and broadcast, R: the number of robots in returning mode, M: the number of robots that move to signal source. We can derive Master equation from this diagram.

$$\begin{aligned} \dot{S} &= -\alpha S + bM - a \cdot l(x) \cdot \tfrac{v}{d} \cdot S \cdot B + \tfrac{1}{\tau} R \\ \dot{B} &= -\tfrac{1}{x+1} B + \alpha S + \tfrac{\gamma}{1+cM} M \\ \dot{R} &= -\tfrac{1}{\tau} R + \tfrac{1}{x+1} B \\ \dot{M} &= -bM - \tfrac{\gamma}{1+cM} M + a \cdot l(x) \cdot \tfrac{v}{d} S \cdot B \end{aligned} \tag{2}$$

We calculate steady state. It means that

$$\dot{R} = 0, \dot{B} = 0, \dot{S} = 0, \dot{M} = 0 \tag{3}$$

The boundary conditions are as follows,

$$\begin{aligned} &R + B + S + M = N, \\ &R, B, S, M \geq 0 \end{aligned} \tag{4}$$

N is the total number of robots. The number of collected pucks is equal to the number of robots in returning mode. So, we calculate the relation between the total number of robots N and the number of robots in returning mode R. Here, we focus on the range $N < 100$ and calculate β by fitting $R \sim N^{\beta}$. The values used in this calculation are as follows:$\alpha = 0.0025, \tau = 22.0, b = 0.01, a = 1.0, v = 10.0, d = 280.0, c = 0.0033x$. The relation between the interaction duration x and exponent β shows that there is an optimum value of β. This value depends on the value of γ. Figure5 shows the relation between γ and β_{opt}.

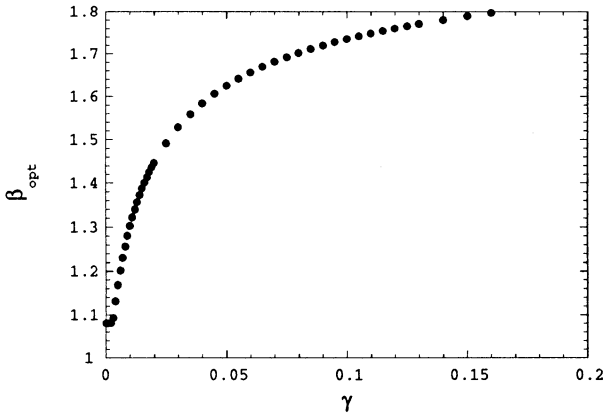


Figure 5: The relation between γ and β_{opt} .

The diagram shown in Figure 4 does not clearly include the condition of field. But Figure 5 implies the field state can be described by γ approximately.

5 Discussion

The range of β can be evaluated by simple estimation. Let's denote P as the number of the puck remaining in the field. The rate of change in P is described as the product of the rate of finding pucks and the rate of puck transport. So, The rate equation for task accomplishment can be written:

$\frac{\partial P}{\partial t} = -$ {finding rate} $\times$ {transporting rate}

The worst case is that only one robot can get a puck and the others cannot. In this case, the finding rate is 1 and transporting rate is 1 and the task completion time is constant. Here we can obtain the relation $T \sim N^0$. It implies the maximum number of β is 0. On the other hand, in the best case, the finding rate and the transporting rate are in proportion to the number of

robot N. It means that $\frac{\partial P}{\partial t} \sim N^2$. It implies that the minimum number of β is -2. These estimation shows us that the range of β is between 0 to -2.

Figure2 showed that β depends on the interaction duration. And it shows that there is an optimum interaction duration. This is because that a short term radiation does not enable all robots to reach signal source and that a too long term radiation causes a lot of collision and traffic jam. Figure2 also showed that there is an optimum value β_{opt}. This value depends on the condition of the field. If pucks are distributed locally, many robots get a puck by interaction. This efficiency is described by high value of β_{opt}. If the pucks are distributed homogenously, a concentration of robots decreases the efficieny of the group. So in the intermediate state, the value of β_{opt} is between 0 and -2.

In section 4, the relation between the value of γ and the value of β_{opt} was obtained. These results enable to describe the puck distribution by a variable γ in state transition diagram. Fitting these result by the function $y = \log x$, we can get the following relation approximately:

$$\gamma \sim D^{\kappa}. \tag{5}$$

The value of κ is -1.418.

6 Conclusion

We investigated collective behavior through puck gathering. We have confirmed an efficiency of interacting multi-robots system by simulations. We evaluated the collective behavior through the relation between the number of robots and their efficiency. Their efficiency was described by the exponent value β. β depends on both interaction duration and the dimension of the puck distribution. From the relation between β and interaction duarion, the optimum value β_{opt} was obtained. And we also obtained the relation between the value of β_{opt} and the dimension of puck distribution D. We proposed state transition diagram to evaluate the group efficiency quantitatively. It was shown that the simple diagram can explain their behavior well. And we found out that there is a relation between the value of β_{opt} and the value of γ in state transition diagram. It implies that γ in the state transition diagram can be described by the dimension of puck distribution D: $\gamma \sim D^{-1.418}$. Now we are studying the meaning of this value.

References

[1] E.O.Wilson, *Sociobiology*, Harvard University Press, (1975).

[2] J. L. Deneubourg, J. M. Pasteels, and J. C. Verhaghe, *J. Theor. Biol.* 105, (1983), pp. 185.

[3] J. L. Deneubourg, S. Aron, S. Goss, J. M. Pasteels, and G. Duerinck, *Physica D*, 22, (1987), pp. 176.

[4] E. Bonabeau, *J. de Physique.*, 6, (1996), pp. 309.

[5] E. M. Rauch, M. M. Millonas, and D. R. Chialvo, *Phys. Lett. A*, 207, (1995), pp. 185.

[6] W.G.Walter, *Scientific American*, 185, (1951), pp. 60.

[7] T. Sato, et al., *J. SICE*, 31, (1992) (in Japanese).

[8] R. Beckers, O.E. Holland and J.L. Deneubourg, *Artificial Life IV*, MIT Press, (1994), pp. 181.

[9] T. Balch and R.C. Arkin, *Autonomous Robots*, 1, (1994), pp. 27.

[10] M.J. Mataric, Ph.D Thesis, MIT, (1994).

[11] L. E. Parker, *Proc. of 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (1995) , pp. 212.

[12] K. Sugawara and M. Sano, *Physica D*, 100, (1997), pp. 343.

[13] K. Sugawara and M. Sano, *Distributed Autonomous Robotic System 2*, Springer-Verlag, (1996), pp. 233.

[14] H. Takayasu, *Fractal*, Asakura-shoten, (1986).(in Japanese)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III '98),
Oita, Japan, 19-21, January, 1998,

Learning cooperative action rules for mobile robots from uncertain human instruction

Hironori Hiraishi and **Fumio Mizoguchi**
Faculty of Sci. and Tech., Science University of Tokyo
2641 Yamazaki Noda Chiba, Japan 278
{hiraishi, mizo}@ia.noda.sut.ac.jp

Abstract

This paper describes a reinforcement learning method for accepting human advice. The unique feature of the method is that human advice is given as a reinforcement value of an action, and a robot can judge whether the advice is correct or not from the change of reinforcement values. As a result, the number of trials is reduced by canceling wrong advice. Experiments are shown in an environment where two mobile robots cooperate with each other.

1 Introduction

Reinforcement learning is a popular method in which a robot can acquire behavioral rules autonomously through trial and error interactions with a given environment[1][3]. Although reinforcement learning has the nature of autonomy, human advice is also useful. Lin's method gives robot advice as an action sequence [4]. In Richard's system called RATLE [5], human advice is included in state-action relations obtained from a knowledge base neural network.

Along this line, we propose a reinforcement learning method in which robots learn from not only an environment but also from human advice. Unlike the previous methods, the proposed method regards human advice as uncertain instruction because we can not always give an appropriate answer. In this situation, human advice is given as a reinforcement value of an action, then a robot judges whether the advice is correct or not from the change of reinforcement values. Thus, the robot can cancel wrong advice. This can be reasonably reduce the number of trials.

The proposed method can be regarded as an extension of a Profit Sharing method[2], which is one type of the exploitation-oriented reinforcement learning. When a robot gets a reward, the reward affects the reinforcement values for all actions included in the associated action sequence. Figure 1 illustrates this situation.

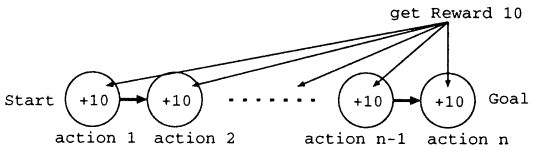


Figure 1: Profit Sharing

The target task here is to learn cooperative action rules for multiple robots. Traditionally, Q-learning is often adopted as a typical reinforcement learning method[1][3]. This is used to identify an environment; thus the environment is treated as static. However, a multi-robot cooperative task is performed in a dynamic environment because the environment changes through a robot action sequence. The profit sharing method allows to reinforcing the action sequence without identifying the environment.

The profit sharing based method also allows us to handle reward effectively. Since human advice has uncertainty by nature, rewards corresponding to wrong advice should be cancelled appropriately. Using our method, a robot can autonomously judge whether the advice is correct or not.

The paper is organized as follows. Section 2 describes the reinforcement learning method using human advice. Section 3 shows the result from an experiment in which two mobile robots cooperate with each other. Section 4 discusses the effect of the method, and Section 5 presents conclusions.

2 Reinforcement Learning Using Advice

2.1 Using advice

Advice is defined as a robot behavior to be carried out in a situation. In order to find preferable behav-

ior that gains rewards in reinforcement learning, the robot chooses an action according to the associated reinforcement value. Thus, advice increases the reinforcement value, and therefore the robot can choose the most preferable action according to the advice.

Suppose that an advice is of the form "Robot should perform action a in state s." The reinforcement value of the action a is updated by Equation (1).

$$F(s,a) \leftarrow \max_{a' \in A} F(s,a') + R \quad (1)$$

$$\Delta F = \max_{a' \in A} F(s,a') - F(s,a) + R \quad (2)$$

where $F(s,a)$ is the reinforcement value of the action a in the state s and R is the value given as the advice. Action a becomes the action with the highest reinforcement value; thus a is selected as the most preferable action. Since R indicates the strength of the advice, a user can specify the degree of belief in this advice. Increment ΔF of $F(s,a)$ in Equation (2) shows such an indicator.

2.2 Judgement of advice

The reinforcement value of an action varies for each learning step, and this can indicate whether the associated advice is correct or not. Suppose that a robot eventually acquires a positive reward after executing the action associated with a given piece of advice. In this case, the reinforcement value increases and the advice is definitely reasonable. In the contrasting case, the robot's action leads to no positive rewards and the reinforcement values of the other actions increase. Since no positive reward means a negative reward, the given advice can be interpreted as wrong.

2.3 Canceling advice

If advice is judged as wrong, it must be canceled so the associated action will not be executed. Suppose that the advice is given to perform action a in the state s. The advice is canceled by decreasing ΔF in Equation (2).

$$F(s,a) \leftarrow F(s,a) - \Delta F \quad (3)$$

2.4 Handling reward

When advice is given to perform an incorrect action, the robot will select this action as the preferable one. To avoid this situation, negative reward or penalty is introduced to decrease the reinforcement value for the action.

Rewards can be handled by a Profit Sharing method which is a type of the exploitation-oriented reinforcement learning and is expressed as follows.

$$F(s,a) \leftarrow F(s,a) + r \quad (4)$$

4 where reward r is assigned to action a of the state s. Reinforcement value $F(s,a)$ is updated by just adding r.

In general, actions are chosen stochastically in proportion to reinforcement values. However, negative rewards may yield a negative reinforcement value, and such action selection can not be done well.

In order to cope with this problem, reward r is used as follows.

$$\begin{aligned} &Reward\ r >= 0, \\ &\qquad F(s,a) \leftarrow F(s,a) + r \\ &Reward\ r < 0, \\ &\qquad F(s,x) \leftarrow F(s,x) + |r| \\ &\qquad\qquad \{x | x \in A, x \neq a\} \end{aligned} \quad (5)$$

When reward r is positive, it is just added to $F(s,a)$. However, when r is negative, the absolute value of r is added to the other actions of the state s. Given a zero or positive initial value, the reinforcement value never becomes negative and the chance of action selection can be calculated. From Equation (5), a negative reward increases the reinforcement values for the other actions. Thus, the negative reward relatively decreases the reinforcement value for the associated action.

A reinforcement value calculated by Equation (5) is meaningful in a relative sense. Let *total* be the sum of all reinforcement values. Since each reinforcement value monotonically increases for each learning step, $r/total$ becomes small, and r itself does not make sense for selecting actions stochastically. Thus, a reinforcement value is updated by using the following equation.

$$\begin{aligned} F(s,x) \leftarrow F(s,x) &- \min_{a' \in A} F(s',a') \\ &+ initial\ value \qquad \{x | x \in A\} \end{aligned} \quad (6)$$

where *initial value* is the smallest reinforcement value of the state s.

3 Experiment and Result

In order to see the effect of our method, we conducted an experiment in an environment in which two mobile robots cooperate with each other as shown in Figure 2. There were six nodes in the environment,

and each robot performs an action such as "go-to" and "come-back" between two destination points (Node 0 and 5). The goal of the robots here is to perform "go-to" and "come-back" actions without collisions.

A state for reinforcement learning is represented in terms of the node currently occupied by a robot. The number of states is as follows:

$$(6\ nodes \times 2\ goals)^{2\ robots} = 144\ states$$

A robot moves to a neighbor node or stops at the current node. It receives 10 as a positive reward if it arrives at the destination node. It receives -10 as a negative reward if a collision occurs. In this case, both robots choose the stop action. They move at the same speed and learn independently.

Advice was given at the 500th and 1000th trials as shown in Figure 3. The reinforcement values for the advice were set to 1000.

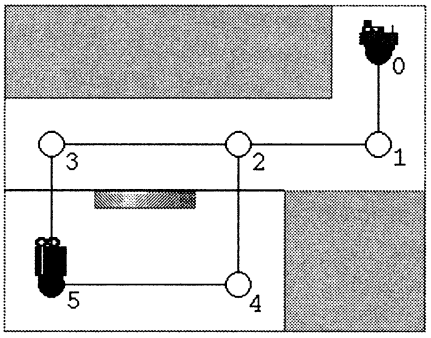


Figure 2: Environment

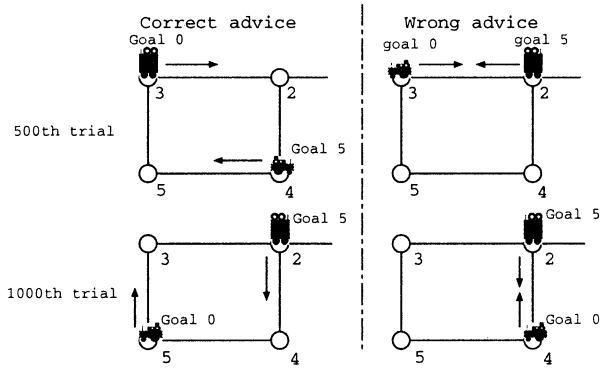


Figure 3: Given advice

We first compare cases in which advice was added or not. Figure 4 (5) shows the results of adding good (wrong) advice.

Figure 6 shows the result on applying advice cancellation to see the effect of wrong advice cancellation. Advice was canceled at the 653rd and 1242nd trials.

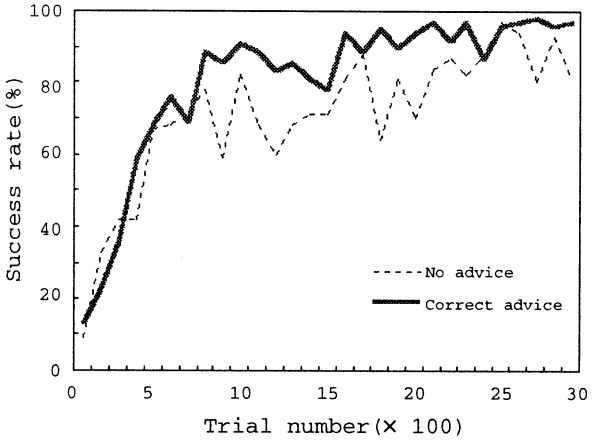


Figure 4: Using good advice

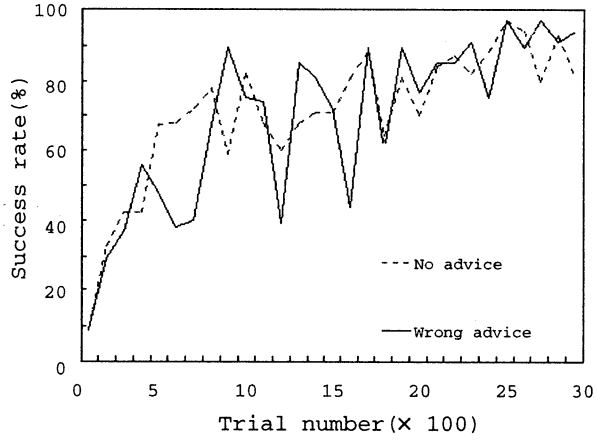


Figure 5: Using wrong advice

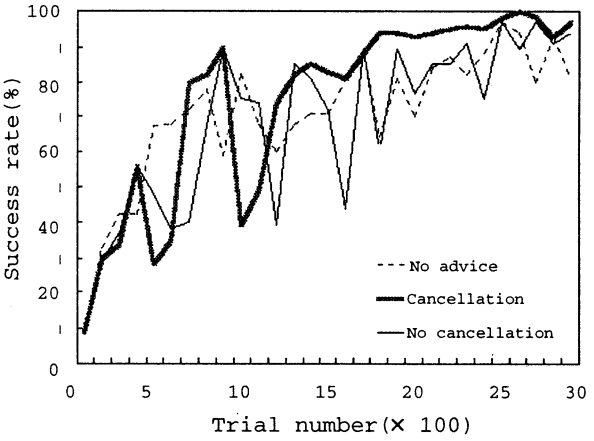


Figure 6: Canceling wrong advice

4 Discussion

Figure 4 simply shows that the case in which correct advice was used has a higher success rate of robot ac-

tions. Thus, we focus on the effect of canceling wrong advice.

4.1 Effect on handling reward

The success rate falls rapidly when wrong advice is given at the 500th and 1000th trials. However, the success rate can be restored within a short term due to our Profit Sharing method in which rewards are distributed to all actions.

4.2 Effect on canceling advice

As shown in Figure 5, the success rate oscillates after wrong advice is given. However, owing to advice cancellation, the success rate became stable at the 700th and 1300th trials. This indicates that advice canceling can be useful for obtaining a higher success rate than achieved in advice-free learning (Figure 6).

Figure 7 illustrates a cancellation process consisting of the following steps: (1) Advice is given for action a. (2) Action a is selected t times. (3) Since another action b has a higher reinforcement value, the previous advice for action a is cancelled.

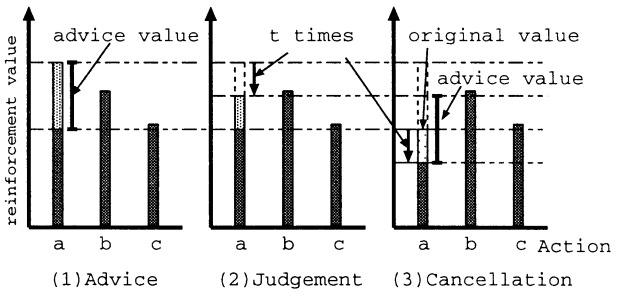


Figure 7: Cancel Advice

As shown in the result, the reinforcement value of the action a is the same value after action a is selected t times when the advice was not given. Therefore, after canceling the advice, the robot doesn't choose the wrong action a, the success rate becomes stable, and a higher success rate is achieved than for advice-free learning.

Finally, we will summarize the advantages of our method.

- Adaptation to dynamic environments.
  The correct advice may be the wrong advice in a dynamic environment. At that time, the advice is canceled, and a higher success rate can be achieved.
- Use of the local advice.
  Advice might be given to a robot from the local scene, in which ease the adviser doesn't need to consider the future behavior of the robot.
- Application to other reinforcement learning.
  In other reinforcement learning such as Q-learning, the action is also selected according to the associated reinforcement value. Our method can thus be applied to other reinforcement learning methods. The value of advice is just added to the reinforcement value or subtracted from it.

5 Conclusion

This paper has described the reinforcement learning method for accepting human advice and has presented an experiment in an environment in which two mobile robots cooperate with each other. Human advice is given as a reinforcement value of an action and a robot can judge whether the advice is correct or not from the change of reinforcement values. The number of trials for learning is reduced by canceling wrong advice.

The result of our experiment showed that our proposed method allows us to detect wrong advice in the early stages and effectively cancel such advice.

References

[1] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda, "Vision-Based Behavior Acquisition For A Shooting Robot By Using A Reinforcement Learning," *Proc. of IAPR/IEEE Workshop on Visual Behaviors-1994*, 1994

[2] John J. Grefenstette, "Credit Assignment in Rule Discovery System Based on Genetic Algorithms," *Machine Learning Vol.3, pp.225-245*, 1988

[3] Mahadevan, S. and Connell, J., "Automatic Programming of Behavior-based Robots Using Reinforcement Learning," *Artificial Intelligence Vol.55,pp.311-365*, 1992

[4] Long-Ji Lin, "Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching," *Machine Learning, Vol.8, pp.293-321*, 1992

[5] Richard Maclin and Jude W. Shavlik: "Incorporating Advice into Agents that Learn from Reinforcements," *AAAI-94, National Conference*,1994

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Task Level Instructions for an Arc Welding Robot

Tasuku Yoshitomi
Department of Mechanical Engineering
Faculty of Engineering, Kyushu Kyoritsu University
Kitakyushu city, Fukuoka, 807 Japan

Abstract

In order to expand the applicable range of the industrial robots, the task level instructions for an arc-welding robot system has been developed.

A teaching program is described with task level instructions. The instruction calls the subprogram named MACRO giving much motions to the robot. A robot controller stores all MACRO and executes them with a interpreter mode.

The MACRO is described with the motion level language which includes geometrical calculations. The MACRO decides excess freedom of the robot appropriately, also choose the most suitable from plural solutions of the joint. By this functions, without changing program to teach, a work-piece can be rotated within a horizontal plane.

A linage of the teaching program is decreased to about 1 / 30. The robot system has been being used actually to the welding of ship structures.

1 Introduction

The applicable range of the industrial robots has been expanded from the field of mass production to the field of small quantity production by means of a progress of an indirect teaching method.

However, the case of the individual production, each work-piece is generally different shape. Since every work-piece of another shape must do a teaching respectively, the teaching time gets large. This matter hinders the application of robots at the field of the individual production. Then, for the economical use of robots in this field, time for teaching must get much smaller.

The robot language of task level is being researched for this aim. This level language includes functions which generate many instructions to motions and conditions.

However, a general-purpose robot language of task level is a step of fundamental development. Therefore, I have developed robot instructions which are task level and are specialized to the arc-welding robot system. The system consists of an articulated robot and a running gantry. The degrees of freedom of the robot and the gantry are 5 and 2 respectively [1].

2 Economy of a robot

2.1 Time and Quantity of teaching

Since the intelligence level of robots for an industry use is low, every motion must be indicated respectively. I have investigated a relation between time and a required accuracy of teaching. The result is shown on Fig. 1.

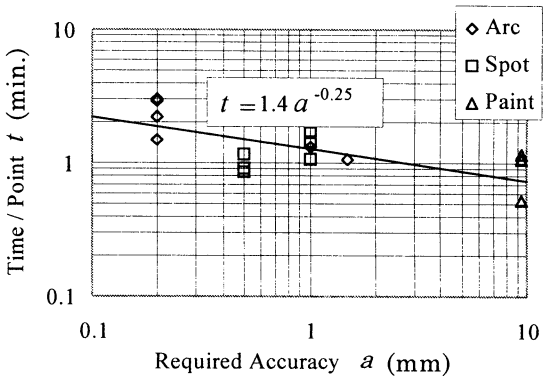


Fig.1 Relation of Time and Accuracy

Time is small with an approximately 1 minute so as to teach one point. However, an easy work to weld a stiffener needs complicated motion being indicated on Fig. 2.

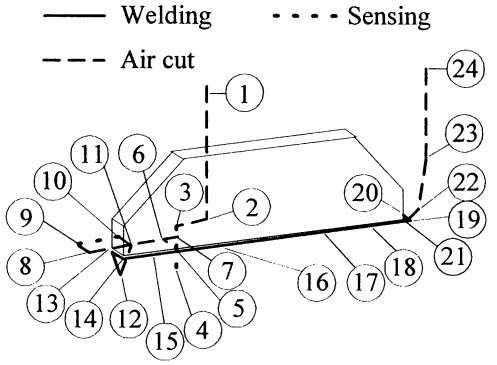


Fig.2 A Path of Straight Welding

Accordingly, number of items to teach is many indicated on Table 1. Consequently, the time for teaching is very long as compared with the time for playback generally.

A time ratio of teaching to playback is shown on Fig. 3.

Table 1 Number of Teaching Items

		Position	Attitude	Speed	Current Voltage	Weaving	Arc sense	Total
Kind of Data		3	3	1	2	15	6	(30)
Number	Air cut	42	15	13	0	0	0	70
	Welding	30	9	6	12	12	18	87
	Total	72	24	19	12	12	18	157

Like this, time for teaching is several dozens of time for playback which is actual working [2].

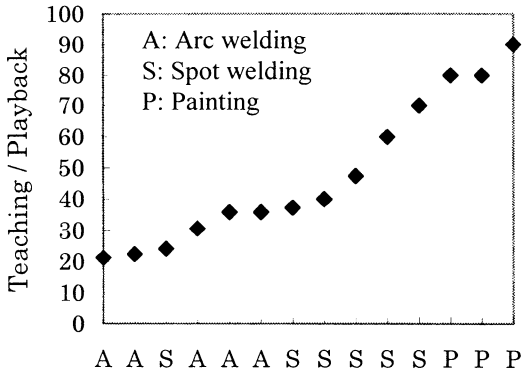


Fig.3 Time Ratio of Teaching to Playback

In the case of mass production, economical influence of long teaching is small because expenses of teaching are shared by many products. However, in the case of individual production, long time for teaching loses much economy of robots because expenses are not shared.

2.2 Number of repeats to keep economically

An investment to equip a robot is returned with an economic effect as a robot produces without an operator. Expenses of teaching by an operator makes economic effect smaller.

To keep economically, the ratio of investment for a robot to annual expenses for one worker should keep the next formula.

$$P / W < \{E(FG - R_D - R_P)\} / (1 + R_D)$$

Here, P is investment for a robot. W is annual expenses for one worker. E is uniform series present worth factor. F is the ratio of operating hours of a robot to working hours of a worker. G is the ratio of productivity of a robot to a worker. R is the time ratio of teaching to playback. The subscript D indicates ratio about direct teaching. Also the subscript P shows ratio about indirect teaching. In the case of an average of each value, the formula indicates that number of repeats is required several hundreds [2].

To reduce time for teaching, effective methods have been developed in the field of off-line teaching [3], self teaching [4], etc.. However, they are not effective to individual production. In the case of individual production, number of repeats about production is quite small. Therefore, time for teaching should be cut more.

3 Reduction of a teaching work

3.1 Intellectual level of a teaching program

The teaching program of a robot may divide into the following five by the intellectual level.

Servo control level
Motion level
Task level
Job control level
Artificial intelligence level

The level which has been actually utilized is the motion level. Japan Industrial Robot Association issued a standard of a robot language in May 1989. It is a motion level. Using a high level language is most useful to reduce time for teaching. However, robot languages having the intellectual level beyond the task level are a stage of concept or research. They are not yet utilized.

Many motion level instructions may be generated by a CAM system. Therefore, the current main method taking easy to teach a robot is the method using CAM system.

Though this is a good method, a programmer is restricted that he must become certain of position to be processed when composing motion level instructions. Because there is a rotation limit of a angle to each joint of a robot, when a position of a work-piece against a robot changes, some motion level instructions must be changed.

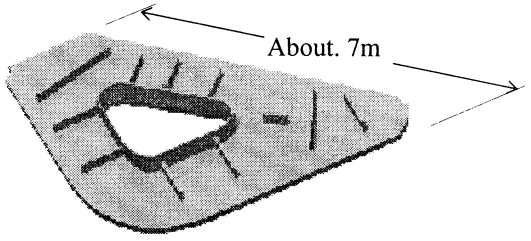


Fig.4 An Examples of A Work-piece

An example of work-piece is shown on Fig.4. A huge work-piece is set at a space in a working area of a robot. When programming a teaching program, a production schedule is not decided in detail. Then, there is difficulties to decide a position of such a huge work-piece because every programmer can not assume the space. Then the CAM system can not be applied to such objects.

3.2 Construction of the task level instruction

Therefore, I have investigated different way

from aforementioned CAM system. A teaching program for this robot system is described with task level instructions. A teaching program consists of following lines. A welding of similar work-piece is executed by this program.

A declaration to name and start of a program
A declaration to start of data
Definitions to coordinate values at many points
A declaration to end of data
A declaration to start of instructions
An instruction to sense work-piece position
An instruction to change coordinate system
Instructions to weld many welding lines
An instruction to stop coordinate changing
An instruction to stop welding
A declaration to end of instructions
A declaration to end of a program

I indicate a format of a instruction to weld to Table 2. The instruction is used to weld a straight line. In the table, **bold** letters are assigned by the system. *Italic* letters mean variable numbers or values. A program has many welding instructions that are written with this format. The formats of instructions or declarations are simple, short and clear. Then, making a program to teach is easy.

Table 2 Format of Instruction to Straight Weld

Format		SW1(&D,P*m*,P*n*,*a*,*b*,*e*)
Arguments	SW1	Calling name of a straight welding
	&D	Sign to keep the wrist rotation
	P*m*	Symbol and number of the start point
	P*n*	Symbol and number of the end point
	a	Sign for a side to be welded
	b	Torch attitude angle at the start point
	e	Torch attitude angle at the end point

3.3 Motion level language

Each task level instruction calls the sub-program giving much motions to the robot. A robot controller stores all sub-programs and generate all motions with interpreter mode. For example, the sub-program for a welding to a straight line generates the 24 motions being shown on Fig.2. Good conditions to the welding is also defined.

In the sub-program, every position of motion is defined by a relative coordinates system. The origin of the coordinates is the start or end point on the welding line. One of the coordinate axes is the perpendicular line. The next axis is welding line. The third is defined with the right hand rule.

The motion level sub-programs decide excess freedom of the robot appropriately. Also, the sub-programs choose the most suitable from plural solutions of the joint angles of the robot. By these functions, a work-piece can be rotated within a horizontal plane without changing program.

The sub-programs are described with the motion level language. The language is fully open to robot users. The language is easy to code like as the BASIC program.

The motion level language includes function of MACRO. You can make necessary sub-program freely with MACRO function. This function is also utilized to save memory on a robot controller. A part of the MACRO to straight line welding is shown on Table 3.

Table 3 Sub-program to straight line welding

Descriptions	Functions
MACRO SW1(&D,P1, P2,⋯)	Entry
SEQ	Declaration
CD1(&H,&M,&T,⋯)	Call welding parameter
WR1(P1,P2,P90,⋯)	Call wrist rotation
POINT16(P1,P2,⋯)	Define approach point
P12=(0,0,&H)P11	ditto
AP1(&B,&BW,⋯)	Call approaching
⋮	⋮
⋮	⋮

Wrist control is programmed with a MACRO named WR1. Detail data for motions and conditions on a welding lines also defined with the MACRO named CD1.

Also, this language includes the function of geometrical calculations being shown on Fig.5. This function is utilized to describe every position of motions. It is also utilized to decide excess freedom as well as to choose plural solutions.

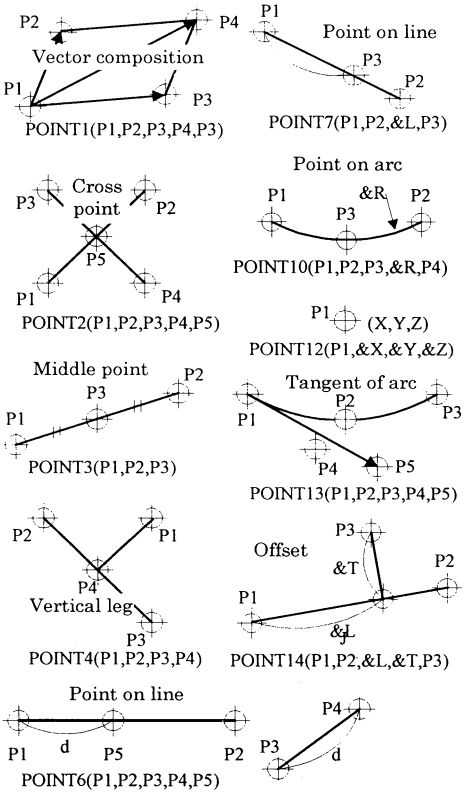


Fig.5 Functions of Geometrical Calculation

4 The characteristic and the effect

The system has been being used actually to weld of ship structure. The task level instructions of the robot gets small time for teaching and also improve flexibility of a production.

4.1 Reduction of Teaching

As the aforementioned, one welding line consists of twenty or more motions. Total number of coordinate values and conditions is beyond one hundred.

If the instructions being proposed here is used, each instruction of the robot calls sub-program respectively. All motions and conditions necessary for welding are generated by the sub-programs which are stored in the robot controller. Motions of air-cut are also generated. Then the linage of instructions is same level with number of welding lines. Also the format of the program is simple and the arguments is a few.

Therefore, the quantity of teaching program as well as the time for teaching become remarkably small. The linage of program decreases to about 1/30 against a motion level program.

4.2 Free setting of work-pieces

The wrist of the robot is adequately controlled with the MACRO for wrist control. The MACRO program keeps every joint within its limit of rotation. The algorithm to control wrist depends on a job, a mechanism and a control of a robot. This MACRO is unique to this robot and the job.

If a work-piece is rotated, directions of rotation of each wrist are automatically adjusted by the function of wrist control. Then, a worker can set work-piece freely without changing program. It get flexibility of production.

4.3 Easy alterations of motions

Macro instructions of robots are often written by the language of system program in general. In that case, a user can not alter such macro instructions.

In the case of this system, every programmer can easily alter the MACRO programs because the programs are written by simple and easy format of motion level.

If a robot interfere with a work-piece, the route or the attitude of the welding torch must be changed. In that case, the user of this robot can easily dissolve a interference. Also, when a new type of work-piece appears at a shop, every programmer can make new instructions with the modifications to original ones.

5 Future applications

The system can be applied to a sorting of work-pieces as well as a sub-assembly at the field of individual production. To supply the parts at the correct time and at the correct position is the precondition to automate.

Numerically controlled gas-cutting is widely used in that field. The data base of the gas-cutting can be utilized to make a program of the system. A numerically controlled robot having the instructions of task level becomes a good solution of the system [5].

6 Summary

1. To extend robot application to the individual production, a time for teaching as well as time for programming should be cut more.
2. To reduce time and to keep flexibility, I have developed the task level instructions.
3. The task level instruction calls the MACRO. A robot controller stores all of them and executes them with interpreter mode.
4. The MACRO is written with a motion level language including geometrical calculations. It is useful for writing various functions.
5. For the free setting of a work-piece, a wrist control is incorporated in a MACRO.
6. All instructions and languages are open to robot users. Programmers can easily modify the instructions and the MACRO programs.
7. The linage of teaching program decreases to approximately 1 / 30.

References

[1] T. Yoshitomi, et al., "The Arc Welding Robot for Hull Sub-assembly Works", Naval Architecture and Ocean Engineering, Vol. 28, 1991

[2] T. Yoshitomi, et al., "The effect of the teaching time on the economy of industrial Robots", Bulletin of Kyushu Kyoritsu Univ., Faculty of Engineering, No.13, pp. 55-60, 1990

[3] K. Matsushima, "Offline Teaching" Journal of the Robotics Society of Japan, Vol. 13 No .5, pp. 29-32, 1995

[4] Y. Mukai, et al., "New Teaching Technology of Welding Robot", Journal of the Robotics Society of Japan, Vol.13 No. 5, pp. 39-40, 1995

[5] T. Yoshitomi, et al., "A Proposal about the Automatic Sorting system of Hull Parts", Transactions of the West-Japan Society of Naval Architects, pp. 283-295, 1996

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Evoluing Computing Theory Brings Hopes to Automatic Programming

Wang Qingchun
Computer Science Department, Wuhan Institute of Chemical Technology
Wuhan, China, 430073

Abstract

Automatic programming is not only one of the central goals of computer science, but also the goal computer science workers are striving for. Some science workers have been exploring and researching in this field. for a long time. Significant advances have been made on evoluing computing theory, especially on Genetic Programming (for short G. P.) theory and method. Those brings hopes to automatic programming.

Keywords: automatic programming, program structure, evoluing method

1. Introduction

Automatic programming means only telling computers what to do instead of giving the computer detailed instrustions to perform programming. The goal will be achieved after it has gone through the phases as follows.

The first one is the period from machine language programming to advanced language programming. During this period, with the development or compiling theory and technology, people were able to design programs using advanced language program instead of ineffecient machine language codes. This resulted in the first leap in automatic programming in the history.

The second one is the period from procedure programming to structure programming. The feature of this time is the application of methodlogy in structure programming. The principle and the method for program structure and style were specified and great progress in programming method had been made. The change in methodlogy led to the second leap of automatic programming in history. During the period, people wished that the program synthesization and the chang in technology could realize automatic programming. But because of the limitedness in extensiblity, repeatability and the theoretical basis, the wish did not come true.

The third one is the period of facing object programming. In the 1980's, fundamental in a cahange internal structure of programming took place. Great progress, especially in the respect of reusability and extensiblity of softwares were also made, which contributed to the development or large-scale science and engineering operation softwares. In turn it led to the automatic production of softwares in the field. In the 1990's the development of parallel and intelligent operation promoted the development of facing agent program methodlogy which combined all the programming advantages in pro-

cess, data and relation (communication) and brought new life to automatic programming, The appearance of G. P. methodlogy marked the new era in automatic programmingin that it is realized with genetic operation method. Thus automatic programming seemed quite promising.

Automatic programming is characterized with such feature as self-adaptablity, self-organizablity and self-studyblity which are necessary for intelligence.

2. Automatic Programming

2.1 The Theoretical Basis of Automatic Programming

The theoretcal basis of automatic programming originated from the theory and methodlogy of G. P. It solved problems with the help of the evolution principle or the evolution principle of nature, especially that of the living creatures . The methodlogy of G. P. orginated from bionic developing operation method, including genetic operation method, developing tactics, developing plan and so on. The theory and methodlogy of G. P. provided automatic programming with basic method, because it has such features as self-adaptability, self organizibility and self—studybility and so on which are necessary for intelligence.

Due to its inner parallelism, the theory and methodlogy of G. P. can perform the function of mordern computer parallel technology in automatic programming. And it is possible to automate programming and to take advantage of parallel technology efficiently. The full use of new parallel technology provides good theoretical basis for automatic programming.

At the same time the theory and methodlogy of G. P. can ideally link electronic appliances with the development of electronic technology and emergence of Programmable Gate Arrays appliances. The combining of them can creat various kinds of new circuits and good programes which have more suitability and practicability. They have attained or even excelled the standard of man—compiled programs in function, and have thus provided the substantial and practical basis for automatic programming.

2.2 Framework of Automatic Programming

In view of the history of programming, any one of computer programes is made up of two kinds of program assembly elements. One is made up of program operation symbols such as +, −, ×, ÷, sin, cos, etc. The other is data assembly such as 3, x, which are used in operation. And the more advanced the program is, the richer the content that can be provided with operation assembly and data assembly is, and the more complicated the compound mapping program is, the stronger the program function is.

This paper provides a program structure method to describe such program structure. That is, operation assembly (including internal function and sub-program warehouse) and data assembly (including data structure) are defined as program structure.

The formula is:

program structure = operation assembly + data assembly

With the concept of program structure having been defined, the fundamental program framework of automatic programming can be realized by means of the theory of the structure, evoluing operation method and evoluing hardware.

The formula is:

Automatic program = program structure + evoluing operation method + evoluing hardware

2.3 The Evoluing Method of Automatic Programming

In standard Genetic Operation Method, genetic group is made up of binary system strings. But the that of automatic programming consists of computer programs. It creats the computer original program at random with program structure element, and applies bionics principle to breed a new (always improved) computer program group. The breeding abides by the principle of natural seleetion, proceeding in a way which has nothing to do with the field. That imitates the genetic operation of the nature: duplication, crossbreeding, mulation. Duplication is the operation that duplicates the existing computer programes to a new group. Crossbreeding is to choose randomly the parts of two existing programes to be reorganized and then added to a new group. Mulation is to change a existing program at random tc creat a new program and add it to a new group.

But the method of automatic programming of the advanced language combines symbolic notation of program structure with intelligent evoluing operation method, adopting the duplication, crossbreeding and mulation to develop, which is described as follows:

Take, for example, the process of automatic programming, used to solve the breeding problem.

(1) Use the operaton assembly and data assembly to duplicate at random andthen produce an original computer program group.

(2) Use the iterative procedure to perform until it meets the prescribed condition.

(A) Perform every program in the group, and choose a fitness number in terms of its suitability.

(B) Use duplication, crossbreeding and mulation to creat a new computer program group. The process is as follows:

(B1) Duplication: Duplcate a existing program to be added to new group.

(B2) Crossbreeding: Choose parts of two existing programes at random to be reorganized and added to a new group.

(B3) Mulation: Randomly choose a part from a program to creat a new program to be added to the new group.

(3) Identify the best program from the computer programming groups as the result of the automatic programming process and repeat the cycle until it has met the rules and obtained the best new program.

3 Conclusion

In this paper, the developing process and theoretical basis of automatic

programming have been discussed, with its basic framework and method-being put forward. Some exploration and investigation have been performed. But there are still some problemes that need further researching and investigating in the sphere of automatic programming to promote the development.

Acknowledgements

I really appreciate the help from Professor Kang Lishan, who is working in State Key Software Engineering Laboratory in Wuhan University .

References:

1. A strict framework on software exploring method, Chen Huowang, Software Journal, etc. Vol. 7. NO. 3. 1996, 187—193.
2. Koza J. K. Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA: MIT Press.
3. Koza J. K. Genetic Programming Ⅱ: Automatic Discovery of Reusable Programs, Cambridge, MA: MIT Press.
4. Holland J. Adaptation in Natural and Artificial Systems. University of Michigan Press, Annbor, 1775.
5. Fogel L. J. Owens A. J. and Walsh M. J. Artificial Intelligence through Simulated Evolution, Wiley, New York, 1966.
6. Rechenberg I. Evolutions strategie: Optimierung technischer Sytsteme nach Prinzipien gerbiologischen Evolution, Frommann — Holzboog Verlag, Stuttgart, 1973.
7. Koza J. R. el al, Four Pro blems for which a Computer Program Evolved by Genetic Programming is Competitive with Human Perfcrmance, Proc. of IEEE Conference on Evolutioonary Computation'96, Nagora, Japan, 1996.

Wang Qingchun graduated from Central China University of Science and Technology. He used to be a visiting scholar of State Key Laboratory of Software Engineering of Wuhan University. He is now an associate professor, teaching in Computer Science Department of Wuhan Institute of Chemical Technology.

He has been engaged in computer science teaching and scientific research for about twenty years, during which he has taught various computer courses. He has also participated in and organized some research items, for example, "communications software", "some key technology research on evoluing computing theory and evoluing software". In addition, he has compiled with other professors the book entitled Basic Program Design, and has published more than ten theses among which are Developing Tendency in Computer Hardware and Potential Breakthrough, Genetic Algorithms Research, etc.

He is unable to attend the international conference due to the problem of cost. But he wishes he could obtain financial aid from the chairman and specialists.

The thesis has been shortened to the requirement to be forwarded. Any help would be greatly appreciated.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Evolution on Folding Landscapes in Combinatorial Structures

S.M. Fraser
Santa Fe Institute
Hyde Park Road
87504 New Mexico

C.M. Reidys
Los Alamos National Laboratory
TSA/DO-SA
87548 New Mexico, USA
Mailstop: TA-0, SM-1237, MS M997

Abstract

In this paper we investigate the evolution of molecular structures by random point mutations. We will consider two types of molecular structures: (a) (RNA) secondary structures and (b) random structures. In both cases structure consists of (i) a contact graph and (ii) a family of relations imposed on its adjacent vertices. The vertex set of the contact graph is simply the set of all indices of a sequence, and its edges are the bonds. The corresponding relations associated with the edges are viewed as secondary base pairing rules and tertiary interaction rules respectively. Mappings of sequences into secondary and random structures are modeled and analyzed. Here, the set of all sequences that map into a particular structure is modeled as a random graph in the sequence space, the so called neutral network and we study how neutral networks are embedded in sequence space. A basic replication–deletion experiment reveals how effective secondary and random structures can be searched by random point mutations and to what extend the structure effects the dynamics of this optimization process. In particular we can report a non-linear relation between the fraction of tertiary interactions in random structures, and the times taken for a population of sequences to find a high-fitness target random structure.

1 Introduction

Evidently, the term "structure" can reflect different levels of coarse graining. In biophysics "structure" is defined in terms of some physical conditions, for example minimum free energy or kinetic parameters; it can also be defined as the set of all affine coordinates of the atoms in a molecule. Alternatively, a structure can be described as a list of all pairs of coordinates of the sequence that are joined by means of chemical bonds. In this paper we will consider "structure" as such a correlation scheme. In particular we will not assume that this scheme has to fulfill constraints that might arise from an embedding in the three dimensional space. In order to investigate different aspects of the process, we consider the following two types of mappings from RNA sequences: first, mappings into RNA secondary structures, and second, into random structures which are described below. For both RNA secondary structures and random structures, the structure consists of (i) a graph (the *contact graph*), in which the vertices correspond to the indices of the nucleotides and the edges correspond to the bonds, and (ii) a set of relations that represent the base-pairing rules.

2 Structures and compatible sequences

Definition 1 *A secondary structure [7] is a vertex-labeled graph on n vertices $\{1,\dots,n\}$ with an adjacency matrix $A=(a_{i,k})_{1\leq i,k\leq n}$ such that*

- $a_{i,i+1}=1$ *for* $1\leq i\leq n-1$
- *for each i there is at most a single $k\neq i-1,i+1$ such that $a_{i,k}=1$*
- *if $a_{i,j}=a_{k,l}=1$ and $i<k<j$ then $i<l<j$.*

We call an edge $\{i,k\}$, $|i-k|\neq 1$ a *bond* or *base pair*. A vertex i connected only to $i-1$ and $i+1$ is called *unpaired*. The number of base pairs and the number of unpaired bases in a secondary structure s is $n_p(s)$ and $n_u(s)$ respectively. Let $\mathcal{A}=\{a_1,\dots,a_m\}$ be an finite alphabet. A *pairing rule* Π over $\mathcal{A}$ is a symmetric, binary relation over $\mathcal{A}$. Let s be a secondary structure and $\Pi(s)=\{\{i,k\}\,|\,a_{i,k}=1,\,k\neq i-1,i+1\}$ its *set of contacts*. The graph $(\{1,\dots,n\},\Pi(s))$ is called the *contact graph* of the secondary structure s. It neglects the backbone bonds that are listed in the corresponding adjacency matrix. Suppose an alphabet $\mathcal{A}$ and a base pairing rule Π is fixed. A secondary structure induce a partition of sequence space as follows: a vertex $P\in\Omega_\alpha^n$ is *compatible to* s_n if and only if

$\forall\{i,j\} \in \Pi(s) : (x_i, x_j) \in \Pi$ i. e. the coordinates x_i and x_j are in Π for all pairs $\{i,j\} \in \Pi(s)$. Let $C[s_n]$ be the set of all compatible sequences, it exhibits a graph structure as follows

$$\mathcal{C}[s_n] = \Omega_\alpha^{n_u(s)} \times \Omega_\beta^{n_p(s)} . \tag{2.1}$$

Accordingly, in $\mathcal{C}[s_n]$ two sequences are adjacent if and only if they differ in (i) in a single position i which is unpaired in s, or (ii) in two positions i and j which form a base pair $\{i,j\} \in \Pi(s)$. Secondary structures have no tertiary bonds. Hence generalized structures have been introduced [3]. Random structures allow for a probabilistic analysis of structural properties which is not feasible for secondary structures. Random structures allow for two types of contacts; first the secondary bonds which form a partial 1-factor graph and second tertiary bonds that are completely random and occur with independent probability p. More precisely let $1 \geq c_1 > 0$ and $1 \geq c_2 \geq 0$ be positive constants. Suppose that $m(n)$ is a monotonously increasing map $\mathbb{N} \to \mathbb{N}$ such that $\lim_{n\to\infty} \frac{2m}{n} = c_1$. $m(n)$ can be viewed as the number of secondary bonds of the random structure. Writing a sequence $V \in \Omega_\alpha^n$ as $V = (P_1, \ldots, P_n)$, let X_1 be a partial 1-factor graph on $2m(n)$ indices, say, $\{\ell_{i_1}, \ldots, \ell_{i_{2m}}\} \subset \{1, \ldots, n\}$. X_1 is the contact graph formed by all secondary interactions. Next let X_2 be the random graph obtained by selecting all possible edges between the n nucleotides except the secondary edges with probability c_2/n. Clearly, the graphs X_1 form a finite *probability space* by assigning to each 1-regular graph uniform probability. Analogously, the graphs X_2 form a finite probability space where a graph (X_2) with k edges has probability $\mu_2\{X_2\} = p^k(1-p)^{\binom{n}{2}-m-k}$ with $p = c_2/n$. The graphs X_1, X_2 induce the random graph $X_1 \otimes X_2$ whose vertex set is $\{1, \ldots, n\}$ and whose edge set $e(X_1 \otimes X_2)$ is the (disjoint) union of all X_1, X_2-edges. $X_1 \otimes X_2$ has probability $\mu\{X_1 \otimes X_2\} = \mu_1\{X_1\}\,\mu_2\{X_2\}$ and is called the random *contact graph*. The probability space formed by the graphs $X_1 \otimes X_2$ will be referred to as Γ^n_{m,c_2}.

Definition 2 *A random structure (r. s.), s_n, on n nucleotides over a finite alphabet $\mathcal{A}$ consists of the following pieces of data: (i) a contact graph $X_1 \otimes X_2$ and (ii) a family of symmetric relations $(\mathcal{R}_*, \mathcal{R}_y)_{y\in X_2}$, where $\mathcal{R}_*, \mathcal{R}_y \subset \mathcal{A} \times \mathcal{A}$, such that for all $a \in \mathcal{A}$ there exists one $b \in \mathcal{A}$ such that $a\mathcal{R}_y b$.*

The relation $\mathcal{R}_*$ is motivated by Watson-Crick *base-pairing rules* observed in RNA secondary structures. For $y \in X_2$ the relation $\mathcal{R}_y$ corresponds to a specific (tertiary) interaction rule that might be context dependent.

Analogous to secondary structures a vertex (sequence) $V \in \Omega_\alpha^n$ is called *compatible* to s_n if and only if (i): for all bonds y of the partial 1-factor graph X_1 its nucleotides indexed by the extremities $\{i,k\}$ have the property $P_i \mathcal{R}_* P_k$ and (ii): its nucleotides fulfill for all tertiary bonds $y \in X_2$: $P_i \mathcal{R}_y P_k$. Again we denote the set of compatible vertices of the r. s. s_n by $C[s_n]$. The contact graph $X_1 \otimes X_2$ induces a partition of its vertex set $\{1, \ldots, n\}$ into the vertex sets of its components $(C_\ell^{(i)})$, where $C_\ell^{(i)}$ denotes component i of $X_1 \otimes X_2$ containing exactly ℓ indices. Let $V = (P_1, \ldots, P_n)$ be a compatible sequence. Each component of the contact graph $X_1 \otimes X_2$, $C_\ell^{(i)}$, induces a multi-set $(P_{i_1}, \ldots, P_{i_\ell})$, consisting of nucleotides whose indices belong to $C_\ell^{(i)}$. This multi-set can be viewed to be an element of a new alphabet, $\mathcal{A}_i$, whose elements are all possible multi-sets $(P_{i_1}, \ldots, P_{i_\ell})$ induced by compatible sequences. Accordingly we can rewrite a compatible sequence as (c.f. (2.1))

$$(A_{i_1}, \ldots, A_{i_l}), \tag{2.2}$$

l being the number of components of the contact graph. We next analyze the graph structure of the contact graph of random structures [3].

Theorem 1 *Let $0 \leq c_2, c_1 \leq 1$, $\frac{2m(n)}{n} \nearrow c_1$ and let $\tilde{T}$ be the r.v. representing the number of vertices of a random graph Γ^n_{m,c_2} that are contained in tree components. Then for $[c_1 + c_2] < 1$ asymptotically almost all vertices of Γ^n_{m,c_2} are in tree components, i.e.*

$$\lim_{n\to\infty} \left[\mathbb{E}[\tilde{T}]/n\right] = 1 . \tag{2.3}$$

There exists a constant $C(c_1, c_2) > 0$ such that a.s. all paths in Γ^n_{m,c_2} have length $\leq C \ln(n)$.
For $c_2 < 1/4$ and arbitrary c_1 there exists a constant $C(c_2) > 0$ such that a.s. all tree components in Γ^n_{m,c_2}, T, have the property $|T| \leq C \ln(n)$.

Accordingly, contact graphs of r.s. decompose with probability 1 into small components and it might in this context be of interest to state the typical ranges for c_1, c_2 that are observed in RNA and protein structures: $0.4 \leq c_1 \leq 0.7$, $0 \leq c_2 \leq 0.2$. In this parameter range Theorem 1 guarantees that a.s. almost all nucleotides of the contact graph $(X_1 \otimes X_2)$ are contained in small components.

3 Neutral Networks as Random Graphs

In this section we introduce a probabilistic model of sequence to structure maps in secondary and random structures as introduced in [6]. The model is formulated in the language of random graph theory and can be sketched as follows: a secondary structure or random structure s_n determines a set of compatible sequences, as described in (2.1) and (2.2). For each compatible sequence we probabilisticly decide whether or not it maps into s_n by selecting it with independent probability λ. The resulting random subset of the set of compatible sequences induces a Ω_α^n-subgraph to which we refer to as $\Gamma_n[s_n]$.

The above modeling of preimages of structures as random graphs bases to large extent on properties of random induced subgraphs of generalized n-cubes. In the following we state some basic properties of random induced subgraphs of generalized n-cubes [6].

Theorem 2 *Let Ω_α^n be a generalized n-cube and Γ_n an induced subgraph with $\mu_n\{\Gamma_n\} = \lambda^{|\Gamma_n|}(1-\lambda)^{\alpha^n-|\Gamma_n|}$ and let $\lambda^* = 1 - \sqrt[\alpha-1]{\alpha^{-1}}$. Then*

$$\lim_{n\to\infty} \mu_n\{\Gamma_n \text{ is } \Omega_\alpha^n\text{-dense and connected}\} = \begin{cases} 1 & \text{for} \quad \lambda > \lambda^* \\ 0 & \text{for} \quad \lambda < \lambda^* . \end{cases}$$

The next result shows that for $\lambda_n \geq \frac{c\ln(n)}{n}$ there exists a.s. a unique giant component in random induced subgraphs of Ω_α^n [5].

Theorem 3 *Let Ω_α^n be a generalized n-cube, $\lambda_n = \frac{c\ln(n)}{n}$, $c > 0$ and μ_n a measure such that $\mu_n\{\Gamma_n\} = \lambda_n^{|\Gamma_n|}(1-\lambda_n)^{\alpha^n-|\Gamma_n|}$. Then there exist $c > 0$, $h \in \mathbb{N}$ such that the largest Γ_n-component, $C_n^{(1)}$, is the induced subgraph of all Γ_n-vertices that are contained in Γ_n-components of size $\geq n^h$, and*

$$\forall \epsilon > 0 \qquad \lim_{n\to\infty} \mu_n\{\Gamma_n \,|\, |C_n^{(1)}| \geq [1-\epsilon]|\Gamma_n|\} = 1 .$$

Accordingly, for remarkably small picking probabilities, asymptotically with probability 1, for a pair of Γ_n-vertices there exists a Γ_n-path connecting them. The next result investigates paths in random induced subgraphs of generalized n-cubes further. It shows, that for the slightly larger picking probability $\lambda_n \geq n^a$, $0 < a < 1/2$ the random graph distance of two vertices P, Q scales linearly with the corresponding distance of P, Q in Ω_α^n [4].

Theorem 4 *Let Ω_α^n be a generalized n-cube, λ_n a probability such that*

$$\exists n_0 \in \mathbb{N},\ \forall n \geq n_0 : \qquad \lambda_n \geq n^{-a} \quad \text{where } 0 < a < 1/2 , \tag{3.1}$$

and $\mu_n\{\Gamma_n\} = \lambda_n^{|\Gamma_n|}(1-\lambda_n)^{\alpha^n-|\Gamma_n|}$ a measure on the set of all induced Ω_α^n-subgraphs. Then, for $k \in \mathbb{N}$ such that $k > \frac{1+3a}{1-2a}$

$$\lim_{n\to\infty} \mu_n\{\Gamma_n \,|\, \text{for } P,Q \in C_n^{(1)} : \text{a.s. } d_{\Gamma_n}(P,Q) \leq [2k+3]\, d_{\Omega_\alpha^n}(P,Q)\} = 1 .$$

4 Sequence to Structure Maps

The intuitive idea of defining probabilistic sequence to structure mappings in combinatorial structures $f : \Omega_\alpha^n \longrightarrow \{s_n\}$ is quite simple: one labels a given set of structures, $\{s_n\}$ and constructs iteratively the neutral networks as random graphs over the remaining sequences. That is, we fix a mapping $r : \{s_n\} \to \mathbb{N}$ having the property $j \leq i \implies r(s_j) \geq r(s_i)$ and set

$$f_r^{-1}(s_0) = \Gamma_n[s_0] \quad f_r^{-1}(s_i) = \Gamma_n[s_i] \setminus \bigcup_{j<i} [\Gamma_n[s_i] \cap \Gamma_n[s_j]] \tag{4.1}$$

One central property of the probabilistic sequence to structure mappings is the embedding of two neutral networks, i.e. what is the Hamming distance between two neutral networks in sequence space? For secondary structures the following is the key result regarding this question [6]:

Theorem 5 *Let Π be a nonempty pairing rule on $\mathcal{A}$ and s_n and s_n' be arbitrary (nonempty) secondary structures. Then*

$$C[s_n] \cap C[s_n'] \neq \emptyset . \tag{4.2}$$

Theorem 5, combined with Theorem 2, guarantees that for secondary structures with large neutral networks $\Gamma_n[s_n]$, $\Gamma_n[s_n']$ there are some sequences $P \in \Gamma_n[s_n]$, $Q \in \Gamma_n[s_n']$ with pairwise small Hamming distances $d_{\Omega_\alpha^n}(P,Q)$. With random structures, we can also study the minimum distance between neutral networks analytically in terms of a new graph, which is obtained by taking the superposition of all edges of the contact graphs of two random structures. Let $X_1 \otimes X_2$ and $X_1' \otimes X_2'$ be the contact graphs with respect to two random structures. Then their union

graph, $(X_1 \cup X_1') \otimes (X_2 \cup X_2')$, is obtained by first taking the union of the secondary edges in X_1, X_1', and then all the tertiary X_2, X_2' edges that are not already in X_1, X_1'. The main result reads [3]

Theorem 6 *Let $X_1 \otimes X_2$ and $X_1' \otimes X_2'$ be two random contact graphs with $\lim_{n\to\infty} \frac{2m}{n-1} = c_1 > 0$ and $0 \leq c_2 \leq 1$. Then for $c_1 < 1$ and $c_2 = 0$ a.s. almost all $(X_1 \cup X_1') \otimes (X_2 \cup X_2')$-vertices are contained in line-graph components. Further, there exists a constant $C > 0$ such that a.s. components of $(X_1 \cup X_1') \otimes (X_2 \cup X_2')$ have size $\leq C \ln(n)$.*
Suppose that $8c_1[2-c_1]c_2 > 1$ and that $\xi \neq 0$ solves $(1-x) = e^{-8c_1[2-c_1]c_2 x}$. Then $(X_1 \cup X_1') \otimes (X_2 \cup X_2')$ has a.s. components $C^{(n)}$ with the property

$$|C^{(n)}| \geq (1-\xi)n[\frac{4m}{n} - (\frac{2m}{n})^2]. \qquad (4.3)$$

According to Theorem 6 there is a distinct change in the graph-structure of $(X_1 \cup X_1') \otimes (X_2 \cup X_2')$. Below the critical value for c_2 (for $c_1 \approx 0.6$ $c_2^{crit} \approx 0.13$) the largest component is $\leq C \ln(n)$, $C > 0$, and above the critical value a giant component emerges.

5 Discussion

In some sense random structures are generalizations of biomolecular secondary structures, which are formally planar knot-free graphs together with rules associated with their bonds. Both, random structures and secondary structures ([2]), induce neutral networks in sequence space, i.e. extended, mostly connected subgraphs consisting of all sequences that are all mapped into the random structure. However, random structures differ from secondary structures in two important regards. First, they may include tertiary interactions, and secondly, they need not satisfy such a knot-freeness condition. Random structures induce a natural, tractable probability space, Γ^n_{m,c_2}, and accordingly allow for the formulation of "almost surely "results, like Theorem 6. They also enable investigation of the influence of tertiary contacts. For the biologically-realistic parameter range, contact graphs of random structures exhibit a similar graph structure to the contact graphs of RNA secondary structures (Theorem 1); their largest component scales with the logarithm of sequence length. The computer generated statistics of contact graphs of random structures illustrate the assertions of Theorem 1. These findings imply that RNA secondary structures and random structures exhibit a significant robustness with respect to point-mutations. As noted in the introduction, the stability of these molecular structures with respect to mutations in the sequences that form them is important in the context of [1], since it provides an explanation for the neutrality of many point mutations at the genomic level. It also explains how diffusion on neutral networks by the accumulation of neutral mutations can, at some later stage, allow novel functionality to emerge (i.e. the population discovers a new neutral network). In the above we have used RNA molecules as the model because of the availability of folding algorithms and formal structural descriptions, because of the clear relationship between structure and function, and the history of *in vitro* RNA experiments. But of course many of the same arguments apply to proteins. In the mapping between DNA or RNA sequences and proteins there is an additional source of neutrality, which is the redundancy in the codon–amino acid coding. The robustness of RNA secondary structures and random structures is the key observation for the modeling of their neutral networks. Neutral networks consist of a few components, whose size depends on the fraction of neutral point mutants (with respect to the structure). Key properties of neutral networks are connectivity and path-structure. For random induced subgraphs of generalized n-cubes there exists a threshold value for connectivity 2 and a giant component exists already for $\lambda_n \geq \frac{\ln(n)}{n}$ Theorem 3. In the case of RNA secondary structures connectivity can be formulated with respect to their corresponding graph of compatible sequences $\mathcal{C}[s] \cong \Omega_\alpha^{n_u} \times \Omega_\beta^{n_p}$ (2.1). Accordingly, connectivity is defined with respect to (i) point-mutations and (ii) base pair-mutations. Since the graph $\Omega_\beta^{n_p}$ is simply a generalized n-cube over the alphabet of base pairs all results of random induced subgraphs derived so far (Theorems 2,3 and 4) apply accordingly. An extension of the connectivity properties is the computation of the length of actual paths between two sequences of a neutral network 4. The main result is that for $\lambda_n \geq n^{-a}$, $0 < a < 1/2$ the random graph distance scales linearly with the distance in the generalize n-cube. Because random structures form a probability space, it becomes possible to analyze in detail the graph structure of the union of two contact graphs, which can only be done for RNA secondary structures with a group theoretic argument [3]. This union of two contact graphs contains information about how close the neutral networks of the two constituent structures come. Considering the edges of the contract graphs as constraints that have to be fulfilled for a sequence to be compatible with a particular structure, the union graph encodes those constraints

for two structures simultaneously; a sequence must fulfill all those constraints to be bicompatible. It is in the region of bicompatible sequences that transitions between neutral networks take place, and the more base pairs in a sequence that are incompatible with the union graph, the greater the number of mutations a sequence must undergo to move between the two networks. We have shown (Theorem 6) that the structure of this union graph changes dramatically while varying the fraction of tertiary interactions c_2 in the two constituent graphs, and shows a phase transition phenomenon at certain critical threshold c_2 with the sudden emergence of a giant component in the union graph. This giant component is very likely to contain cycles, which make it unlikely that that bicompatible sequences will exist, and will increase the distance between the neutral networks. Thus, we expect that populations will find it increasingly difficult to make transitions between neutral nets at higher c_2 values. Larger structures, in which the phase transition is more marked, would show even stronger effects of c_2. For a $c_1 = 0.6$ the critical fraction of tertiary interactions for the emergence of a giant component in the union graph of two random structures has $c_2 = 0.15$ as an upper bound (eq. (2)). For $c_1 = 1$ a lower bound on the critical fraction would be 0.125. Known 3D-structures (for example t-RNA) have values of c_2 which are well below this critical threshold, with about 4-6% nucleotides involved in tertiary interactions. In order to investigate the effects of structure on the ability of populations to search, we induce dynamics over a population of sequences using a replication–deletion process which is described above. In these simulations, sequences are assigned a fitness according to the structure they form, reflecting the assessment of fitness at the phenotypic level. The replication–deletion process is a simply scheme in which sequences are replicated according to their fitness, with a constant population size, in a flow reactor-type process. It should be noted that the only genetic operator that is being used here is point mutation—sequences make only local moves in the sequence space, via their mutant offspring. Yet because of the guarantee that for biologically realistic parameter values neutral networks are dense and connected (Theorem 2), populations of sequences evolving in this way are able to search a large proportion of the shape space. The results demonstrate this—populations of 2000 sequences rapidly find the neutral network of one of the few high-fitness structures in a landscape of 10000 structures.

Acknowledgements

We want to thank Christopher L. Barrett for stimulating discussions. Special thanks to Darrell Morgeson for his continuous support. SF is funded by DARPA under grant ONR N0014-95-1-1000.

References

[1] M. Kimura. *The Neutral Theory of Evolution.* Cambridge Univ. Press, 1983.

[2] C.M. Reidys. *Neutral Networks of RNA Secondary Structures.* PhD thesis, Friedrich Schiller Universität, Jena, Math. Faculty, September 1995.

[3] C.M. Reidys. Mapping in random-structures. *SIAM Journal of Discrete Mathematics and Optimization*, 1996. submitted.

[4] C.M. Reidys. Random Graphs and Sequence to Structure Maps. *Combinatorics Probability and Computing.*, 1997. submitted.

[5] C.M. Reidys. Random Induced Subgraphs of Generalized n-Cubes. *Adv. Appl. Math.*, 19(AM970553):360–377, 1997.

[6] C.M. Reidys, P.F. Stadler, and P.K. Schuster. Generic properties of combinatory maps and neutral networks of RNA secondary structures. *Bull. Math. Biol.*, 59(2):339–397, 1997.

[7] M.S. Waterman. Combinatorics of RNA hairpins and cloverleaves. *Studies Appl. Math.*, 60:91–96, 1978.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Evolution of Herding Behavior of Multiple Autonomous Mobile Robots

Byoung-Tak Zhang
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
btzhang@comp.snu.ac.kr

Yeon-Jun Hong
Dept. of Computer Engineering
Konkuk University
Seoul 143-701, Korea
yjhong@ai.konkuk.ac.kr

Abstract

We study the evolution of herding behavior of multiple autonomous mobile robotic agents. Agents are cloned to form a homogeneous team. Cooperation strategies for a team of the robots are represented in a neural network. An evolutionary algorithm is used to evolve the neural network fitted for herding behavior. The effectiveness of the method is experimentally tested in a simulated environment for solving a furniture transportation problem.

1 Introduction

Some tasks can be done faster or more easily by dividing it up among many agents. Other tasks may not only be solved better by using multiple agents, but can only be effectively solved, by using teams of agents working together. The global behavior of the group of mobile robots is determined by the local interactions.

The problem of cooperation and communication between multiple mobile agents has recently been studied by many researchers studying artificial life and evolutionary computation.

Reynolds [Reynolds, 1993] used genetic programming to evolve "critters" which reacted with a herd instinct to outside preditors. He evolved a single controller which moved each critter based on its position and information about its neighbors and preditors. Each critter used this same controller algorithms to make movement decisions. Haynes *et al.* proposed a genetic programming method to generate a team by considering the whole team as one individual [Haynes *et al.*, 1995]. They considered the subtrees of a genetic program as subindividuals within a main individual, each of which may serve a distinct specialized purpose. Luke and Spector implemented heterogeneous breeding strategies which is functionally very similar to the clear method of producing homogeneous individuals [Luke and Spector, 1996]. In both cases standard genetic programs are grown, and only one "individual" is tested at a given time. For homogeneous teams, individuals are tested by cloning them to form teams, and the resulting teams are tested in the environment. Heterogeneous teams, however, are formed from the individual's collection of subtrees.

We have designed a self-organizing chaotic neural network for the evolution of robot herding behavior. Robots are cloned to form a homogeneous team. Cooperation strategies for a team of the robots are represented in the neural network. Evolutionary algorithms are used to evolve the neural network fitted for herding behavior. The effectiveness of the method is experimentally demonstrated on a furniture transportation problem.

The fitness of each network is evaluated by implanting it into the robots. Each robot is given a bucket of points at the outset. The robots are allowed to move a fixed maximum number of steps. At each movement, the badness of their behavior is evaluated and penalized. When it collides, for example, with its colleagues, it gets K_{coll} points subtracted from the bucket. An amount of K_{away} points is also subtracted when it moves too far away from its colleagues. This factor encourages herding behavior. When the table is out of sight of the robot on each move action, the robot gets K_{sight} points decreased. To encourage the movements of the robots, we also penalized the robots that stay at the same location or move seldom.

The paper is organized as follows. Section 2 motivates the approach and describes the method for evolving herding behaviors. Section 3 presents the neural net architecture. Section 4 provides experimental results.

2 Evolving Cooperative Agents

Some tasks, such as transporting a large table, requires multiple robots to cooperate. To study cooperation strategies in such a context, we constructed a virtual workspace of the robot on a plane of 20m × 20m. Figure 1 shows the workspace occupied by the table and robots to transport it. We assume cylinder robots of 20cm in diameter. It has four sensors positioned 45 degree upward. One move action of the robot brings its body 5cm to the direction it heads at the time. The sight of robots is limited to 2m. An additional sensor detects the target object within 10m. Sensors are connected to the input nodes of a neural network that determines the effector values for move actions. The initial environment contains three obstacles which are located randomly between the robots and the table.

The sensory inputs needed to control the robots include the pressure for measuring the unbalance of weight of the table among the member robots, the detectors for the table and obstacles, a communication signal for synchronization of group behavior, and a stagnation detector for the determination of the direction to move to. The outputs for motor control include the values for driving the left/right motors of the wheels and the signal to send for communication necessary for finding and lifting the table.

In the simulations described below, we have used as input to the network the detectors for the table to transport, colleagues, and obstacles. The network has 9 output units, 8 for determining the direction and the rest for non-moving.

The fitness of each network is evaluated by implanting it into the four robots. Each robot is given a bucket of points at the outset. The robots are allowed to move a fixed maximum number of steps. At each movement, the badness of their behavior is evaluated and penalized. When it collides, for example, with its colleagues, it gets K_{coll} points subtracted from the bucket. An amount of K_{away} points is also subtracted when it moves too far away from its colleagues. This factor encourages herding behavior. When the table is out of sight of the robot on each move action, the robot gets K_{sight} points decreased. In effect, the sum of the penalties on each movement is

$$\begin{aligned} A &= K_{away} \cdot \text{NumAways} \\ &\quad + K_{coll} \cdot \text{NumCollisions} \\ &\quad + K_{sight} \cdot \text{NumOutsights} \end{aligned} \tag{1}$$

where NumAways is a count for the number of being far away from the colleagues, and NumCollisions and NumOutsights are counts for collisions and being too far away from the table.

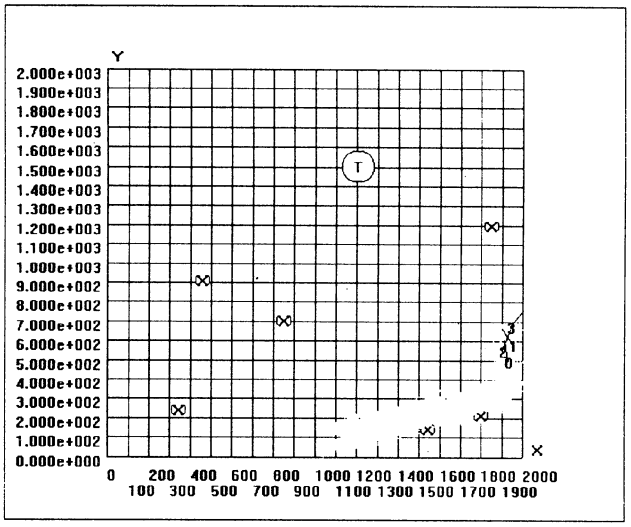


Figure 1: The simulation environment.

To encourage the movements of the robots, the resulting bucket of points is multiplied by the following factor

$$S = K_S \cdot \text{NumStepsMoved} \tag{2}$$

where K_S is a constant amd NumStepsMoved is the total number of steps moved. This term penalizes the robots that stay at the same location or move seldom.

The bucket of remaining points is subtracted again by a fractional amount of the distance from the table:

$$D = K_D \cdot \text{FinalDisplacement} \tag{3}$$

where K_D is a constant. This is to promote moving toward the table.

Overall, the fitness of a robot or its neural network is:

$$F_i = (\text{Bucket} - A) \times S - D \tag{4}$$

where A, S, and D are defined as above.

After being evaluated their fitness, the individuals are selected to be parents for recombination. The selection probability of each individual is given as:

$$R_i = \frac{F_i - F_{min}}{F_{max} - F_{min}} \tag{5}$$

where F_{max} and F_{min} are, respectively, the maximum and minimum fitness values of the individuals in the current population. We used steady-state selection, i.e. 5% of the population are replaced after each evaluation. Thus the population evolves more slowly but continuously than in generational selection.

3 A Neural Net for Herding

The neural network consists of four layers (Figure 2). The bottom layer is the input layer that receives sensory inputs. The next three layers are the recurrent neural network (RNN) layer, the self-organizing map (SOM) layer, and the Cognitron output layer.

The recurrent neural network receives the environmental inputs and processes them to detect the spatio-temporal pattern of the inputs. It contains both excitatory and inhibitory neurons. The recurrent network used in the experiments contains 84 neurons in a virtual cylinder. The neurons are connected with other neurons in the neighborhood more frequently than the neurons in the distant area within the cylinder. We adopted a neuron model with chaotic dynamics [Adachi and Aihara, 1997]. The activation is determined by:

$$x_i(t+1) = f\left[\sum_{j=1}^{M} v_{ij} \sum_{d=0}^{t} k_e^d A_j(t-d) + \sum_{j=1}^{N} w_{ij} \sum_{d=0}^{t} k_f^d x_j(t-d) - \alpha \sum_{d=0}^{t} k_r^d g\{x_i(t-d)\} - \Theta_i\right] \quad (6)$$

where v_{ij} and w_{ij} are synaptic weights to the ith constituent neuron from the jth external input and from the jth constituent neuron, respectively, and k_e, k_f, and k_r are the decay parameters for the external inputs, the feedback inputs, and the refractoriness, respectively. We set $k_e = k_f = k_r$ and $g\{x_i(t-d)\} = x_i(t-d)$. The recurrent network considers the spatio-temporal summation of both external inputs and feedback inputs from other chaotic neurons.

The neurons in the recurrent network learn their weights for each input data. The modified Hebbian learning rule is used to adapt the weights:

$$w_{ij}(t+1) = w_{ij}(t) + r\{1 - w_{ij}(t)\}h\{x_i(t), x_j(t)\} - K\{w_{ij}(t)\} \quad (7)$$

where $K\{w_{ij}(t)\}$ is the weight decay function. The parameters are determined by an evolutionary method. For each input data, the network is activated iteratively to converge to a pattern or oscillate between two or more patterns. For our application, it is useful if the network produces similar output patterns for similar input patterns.

The next layer is the self-organizing map (SOM). Its objective is to recognize the activation pattern of the recurrent network and classify it into a class. The SOM learns for every update of activation at the recurrent network layer:

$$w_{ij}(t+1) = w_{ij}(t) + r_{SOM}\{a_j - w_{ij}(t)\} \quad (8)$$

This implies that the SOM layer learns to recognize the activation patterns of the recurrent network. If the RNN converges to a single pattern, then SOM recognizes it. The ultimate winner of the SOM is the unit whose weight vector has the smallest distance to one of the activation patterns at RNN.

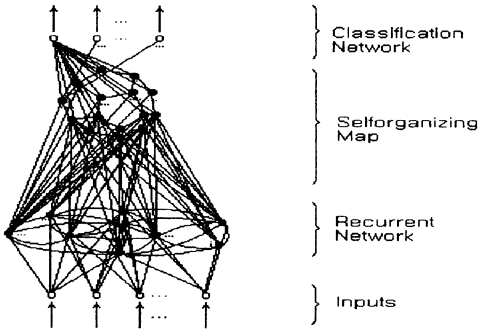


Figure 2: The multinet neural architecture.

The output layer determines the output of the network based on the winner of the SOM network. This layer is trained using reinforcement learning as follows:

$$y(s) = \begin{cases} w_{cij}(t) + r_c\{1 - w_{cij}(t)\} \cdot s \cdot a_j & \text{if } s \geq 0 \\ w_{cij}(t) + r_c \cdot w_{cij}(t) \cdot s \cdot a_j & \text{otherwise} \end{cases}$$

where $s, -1 \leq s \leq 1$, is the normalized difference between the present penalty and the previous one. a_j is the activation value of the winner at the SOM layer.

4 Experimental Results

Figure 3 plots the evolution of fitness values as generation goes on. The thick line is the average fitness of the population and the thin line is the fitness of the best individual at each generation. It can be clearly observed that the average fitness of the robotic agents increases as they evolve in the environment. The best-of-run individual was evolved after 30 generations. After 55 generations the fitness of the best-of-generation individual did not change. The fluctuations in the best fitness reflects the fact that we did not apply the elitist selection strategy.

Though not shown here, we also observed tendency of a convergence of winners at the SOM layer as the

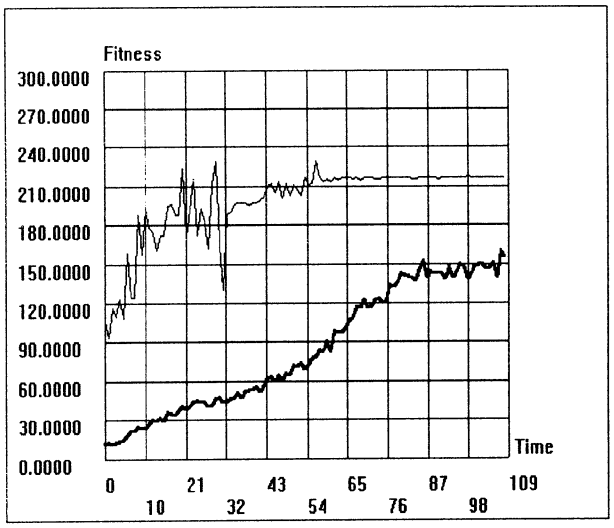


Figure 3: Change of the best and average fitness values.

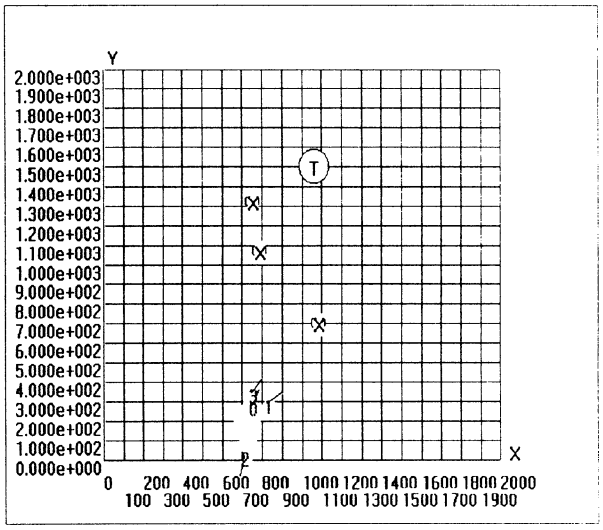


Figure 4: The robots herding to the table.

recurrent neural network updates its activation. Figure 4 shows the behavior of robots controlled by the neural network by the evolutionary algorithm. As generation goes on, improvement in robots' herding behavior was observed in comparison to the behavior of the robots at early generations.

5 Concluding Remarks

We have described a method for evolving herding behavior of a group of small mobile robots. A neural network architecture is presented that processes spatio-temporal patterns of input stimuli, detects convergent patterns, and produces motor outputs corresponding to the stimuli pattern. The network is evolved by situating the robots in the simulated world and rewarding positive actions and penalizing negative actions by a carefully designed fitness function. The performance was demonstrated in the context of a table-transport problem. Future work should refine and improve the current framework. We are studying other efficient learning strategies that assign credits to the component networks to evolve more goal-directed collective behavior.

Acknowledgments

This research was supported in part by a grant from the Korea Science and Engineering Foundation (KOSEF) under contract number 96-0102-13-01-3.

References

[Adachi and Aihara, 1997] M. Adachi and K. Aihara, "Associative dynamics in a chaotic neural network," *Neural Networks* **10**(1):83-98, 1997.

[Haynes *et al.*, 1995] T. Haynes, S. Sen, D. Schoenfeld, and R. Wainwright, "Evolving a team," *Proc. AAAI-95 Fall Symposium on Genetic Programming*, pp. 23-30, AAAI Press, 1995.

[Hong and Zhang, 1997] Y.J. Hong and B.-T. Zhang, "Evolving cooperation strategies for multiple mobile robots, *Proc. ASCC-97*, Seoul, July 1997.

[Kube and Zhang, 1993] C.R. Kube and H. Zhang, "Collective robotic intelligence," *Proc. Second Int. Conf. on Simulation of Adaptive Behavior*, pp. 460-468, 1993.

[Luke and Spector, 1996] S. Luke and L. Spector, "Evolving teamwork and coordination with genetic programming," in *Proc. 1996 Genetic Programming Conf.*, J.R. Koza *et al.* (eds.) Cambridge, MA: MIT Press, pp. 150-156, 1996.

[Mataric, 1993] M.J. Mataric, "Designing emergent behaviors: From local interactions to collective intelligence," *Proc. Second Int. Conf. on Simulation of Adaptive Behavior*, pp. 432-442, 1993.

[Reynolds, 1993] C.W. Reynolds, "An evolved, vision-based behavioral model of coordinated group motion," in *Proc. Second Int. Conf. on Simulation of Adaptive Behavior*, J.-A. Meyer *et al.* (eds.) Cambridge, MA: MIT Press, pp. 384-392, 1993.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

The Species Fitness Method for the Evolution of Cooperative Behavior in a Group Task

Masaki KITAHASHI , Tomohiro YAMAGUCHI and Masahiko YACHIDA
Dept. Systems and Human Science,
Graduate School of Engineering Science
Osaka University
1-3 Machikaneyama-cho, Toyonaka, Osaka, 560, JAPAN

Abstract

This paper studies the evolution of cooperative behaviors as the interaction among agents by GA to improve the performance of the task in a group (*group performance*, in short). Previous research used *the group fitness method* that evaluates the group performance for the evolution of the multiple groups in parallel. However, it takes large simulation costs and the evolution speed is slow.

On the other hand, *the individual fitness method* that evaluates the *individual performance* of the task for the evolution in a group takes a small simulation cost. However, it can't improve the group performance since each agent behaves selfishly.

To optimize the group performance, it is important both the concept of competition and sharing. Therefore, this paper presents *the species fitness method* that has both advantages between the group fitness method and the individual fitness method by sharing the individual performance of the agents belongs to the same species that has same chromosome.

We show the comparative experiments on the evolutionary simulation of a foraging task in a group among the group fitness, individual fitness, and the species fitness method. As the interaction among the agents, four kinds of species on the communication ability whether the agent can send or receive the signal for food are evolved.

Keyword: *Artificial Life, Evolution, Cooperative Behavior, Communication, Species Fitness, Group Preformance*

1 Introduction

The evolution of cooperative behavior among a group of agents is one of the fundamental problems of not only multi-agent or artificial life research but also social science or game theory [4]. For optimizing the performance of the task in a group (group performance, in short), cooperative behavior among a group of agents can be defined as the interaction among agents that improves the group performance. However, in previous researches, a mechanism for acquiring the above cooperative behavior efficiently haven't been make clear. To solve them, this paper describes a method of evolving cooperative behavior that optimizing the group performance.

To optimize the group performance, it is important both competition and sharing. The former is the driving force of the improvement in performance, the latter means the constraint under the competition that develops altruistic behaviors in the interactions among agents. However, in previous methods [1], the balance between them was not considered.

There are two methods for optimizing the group performance, one is the group fitness method that evaluates the group performance itself, the other is the individual fitness method that evaluates the individual performance of the task. The former [1] consider the group performance as the group fitness. It is straight forward for optimizing the group performance since the group performance reflects all the influences of all interactions among agents related to the task in a group. However, it results in inefficient since all agents in a group share the same fitness, no competition among agents occurs so the evolution speed is slow. Besides, it takes large simulation costs than the individual fitness method because it is often used to improve the group fitness by competing several groups in parallel at the evolutionary simulation. For this, the search space for the evolution is much broader than the individual fitness method since the information of all agents in a group is coded as a chromosome of GA as a unit of evolution. Above these reasons, using group fitness is not efficient for optimizing the group task.

On the other hand, as the latter evaluates the performance of an agent in the group as the individ-

ual fitness, the individual performance is optimized. However, the group performance will not be optimized since no agent sharing the group fitness, consequently the interaction that profitable only others among agents is ignored. In other words, it is hard to acquire the cooperative behaviors efficiently since each agent tends to improve the performance of itself selfishly.

For this, the method that directly biases the cooperative behavior by adding it to the individual fitness function can be considered. Yet, following problems occurs;

1. The profitable interactions is a problem dependent.
2. It is no guarantee for optimizing the group performance.

That is the basic query in the evolution as following; "what is a unit for selection of the natural selection? individual, group, or anything else?"

To solve these problems, as biologically inspired by the hypothesis of Hamilton [2] in explanation of the altruistic behaviors, this paper presents the species fitness method that has both advantages between the group fitness and the individual fitness method by sharing the individual performance of the agents belongs to the same species.

The advantage of the species fitness method is that both the competition among the species and the sharing the fitness among the agents in the same species are realized by classifying the agents in a group into the species. Therefore, it is possible to optimizing the group performance efficiently. Besides, in previous reciprocative altruistic behavior [3],it is hard to discriminate the partner sharing the profit from the others, the species fitness method can solve this problem by sharing within the agents that has same genetic information.

2 The Species Fitness Method

The species fitness method is the method of classifying the agents in a group into subgroups that has same chromosome, then sharing the individual fitness of the agents belongs to the same subgroup.

Now, we outline the efficient improvement of the group performance by the species fitness method. When agents in a group is classified into subgroups by the difference of the ability or the behavior patterns because of the different genetic information, a set of the agents those have same genetic information is called species. Sharing the individual fitness among the agents in the same species as the species fitness makes the altruistic behavior reciprocative [3] since it can share the profit by the altruistic behavior within the same species. If the cooperative behavior that increases the species fitness emerges and that is greater than other species fitness, increasing the number of agents in that species makes both improving the performance of a group and spreading the cooperative behavior in the group available without evaluating the group fitness directly.

The Merit of the species fitness method is that though it takes a small simulation cost nearly equal to the individual fitness method, it improves the group performance since it can evaluates the altruistic behaviors within same species, that is cooperative behaviors. Besides, the evolution speed is faster than the group fitness method since it directly evaluates and selects to improve the individual performance as well as the individual fitness method.

3 Evolving the Communication Protocol by a Group Task

3.1 Foraging Task by a Group

In this paper, we define a group task as a foraging task by a group in the simulation. To emphasize the evolution of communication, the foraging task is simplified as follows; there are many agents, many feeding positions and a small number of stacks of food in the environment. The group task is to make agents find and eat all food in the environment as soon as possible. The individual task for an agent is to eat food as much as possible.

As the initial set up for the simulation, the number of feeding positions is 500, the number of agents is 500, the number of feeding positions including the stack of food is only five, and in each stack, the number of food is 100, 200, 300, 400 or 500. For the initial location, each agent is placed at each feeding position for one to one. The feeding position that has the stack of food is located randomly.

Then, the fixed ability of an agent is defined. Each agent is placed a feeding position, at each step, each agent can eat one food if food exists, or can move to another feeding position randomly if there is no food.

3.2 Communication Ability of an Agent

Next, we explain the genetic ability of an agent. In order to improve the efficiency of the group task,

a signal for food is defined. As a genetic ability of an agent, communication abilities whether the agent can broadcast or receive a signal or not is coded as 2bits gene for each agent, then four kinds of species are defined as follows;

- isolated species : no communication ability
- selfish species : only receiver
- altruistic species : only sender
- reciprocative species : both can send and receive a signal

The behavior when to broadcast or the after receiving behavior are fixed as follows; an agent who has a sending ability broadcasts the signal when it eats a food, and an agent without eating who can receive a signal moves into a signal position by a step. If it receives a lot of signals, it randomly selects one within them, so that it moves into a signal position stochastically according to the number of signals.

Above them, species who has both sending and receiving ability become reciprocative in its species since its communication ability makes the task easier. By contrast with selfish species receives profits one-sidedly, altruistic species decreases its future profit indirectly by its broadcast.

4 Experiment

4.1 Experimental Methods

In the experiment by GA, the simulation starts with an initial group in that all agents are isolated species. Group performance is compared by using thee kinds of fitness methods.

- Experiments 1:the group fitness method (20 groups × 1 trial)
- Experiments 2:the individual fitness method (1 group × 20 trials)
- Experiments 3:the species fitness method (1 group × 20 trials)

In each experiment, maximum, minimum, and average of the group performance within 20 groups or trials that is the achieving steps of the group task are measured.

In genetic operation, low fitness agents are deleted by selection rate then the next generation is generated by simple GA making the rest of agents to be a parent.

4.1.1 Experiment 1 : the group fitness method

By the group fitness method, GA is performed with 20 groups in parallel. Gene length of a group is $(2\ bits) \times (500\ agents) = 1000\ bits$. Then the group fitness using GA is following;

$$group\ fitness = \begin{cases} 50 - \text{group performance} \\ (\text{ if 50 - group performance} > 0\ ) \\ 1 \\ (\text{ if 50 - group performance} \leq 0\ ) \end{cases}$$

As GA parameters, population is 20, gene length is 1000 bits, crossover rate is 0.2, mutation rate is 0.01/bit and selection rate is 0.5.

4.1.2 Experiment 2 : the individual fitness method

By the individual fitness method, GA is performed with 500 agents in a group as a population. Gene length of an agent is 2bits. The Individual fitness is following;

individual fitness =

the number of food that an agent ate

As GA parameters, population is 500, gene length is 2, crossover rate is 0, mutation rate is 0.01/bit and selection rate is 0.5.

4.1.3 Experiment 3 : the species fitness method

By the species fitness method, the same method of Experiment 2 is used except the definition of the fitness. Gene length of agent is 2bits. As the individual fitness of the set of agents that have the same gene, that is an agent that belongs to species i, its average is called m_i, and its distribution is called σ_i, then two kinds of species fitness can be defined.

- species average fitness = m_i
- species distribution fitness = $1/{\sigma_i}^2$

Note that both fitness methods are examined, then the species distribution fitness that marked better result is reported in this paper. GA parameters are the same as the individual fitness method.

4.2 Experimental Result

Fig.1,2 and 3 shows the group performance of Experiment 1,2 and 3. In each figure, maximum, average and minimum group performances within 20 groups are shown. Note that each value is averaged every 50 generations.

Compared with the averaged group performance in each experiment, the result of the individual fitness method is worst (from 20 to 25 steps), then the result of the group fitness method is converged about

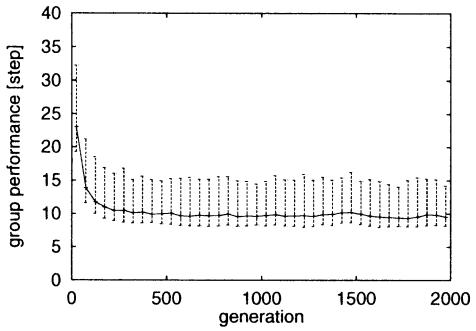


Figure 1: Experiment 1: Result of the group fitness method

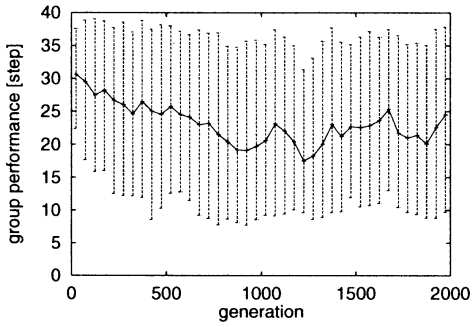


Figure 2: Experiment 2: Result of the individual fitness method

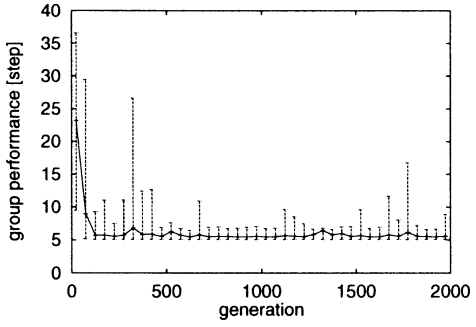


Figure 3: Experiment 3: Result of the species fitness method

10 steps, the best result is the species fitness method converged about 6 steps that is nearly the theoretical minimum steps of four.

Regarding the distribution of the group performance, the result of the individual fitness method is worst so that its group performance become much instable. Even the best case of its group performance is the same as the average of the group fitness method.

In contrast, the distribution of the species fitness method is very small, beyond that its initial convergence is fast. Therefore, the species fitness method is much better than two other methods on the converged performance, the convergence speed, the stability and the distribution of the performance.

5 Conclusions

This paper proposed the species fitness method of sharing the individual fitness among the agents of the same species in a group. By this method, we showed that cooperative behaviors are evolved with less simulation costs than the previous group fitness method by evaluating groups in parallel. Future work is to apply the species fitness method to the foraging task in a 2D space to get a cooperative search behavior.

References

[1] Saunders,G.M. and Pollack,J.B, "The Evolution of Communication Schemes Over Continuous Channels," *Proc. 4th Int. Conf. on Simulation of Adaptive Behavior (SAB-4)*, pp.580-589, 1996.

[2] Hamilton,W.D. "The Genetical Evolution of Social Behavior I, II," *Journal of Theoretical Biology*,Vol.7,pp.1-52,1964.

[3] Sen,S.,"A Foundational Principle for Promoting Cooperative Behavior among Self-Interested Agents," *Proc. Second Int. Conf. on Multi-Agent Systems (ICMAS-96)*, pp.322-329, 1996.

[4] Nowak,M.A. , May,R.M. and Sigmund,K., "The Arithmetics of Mutual Help," *SCIENTIFIC AMERICAN*, June, 1995.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Computational Evolution of Human Bipedal Walking by a Neuro-Musculo-Skeletal Model

K. Hase
National Institute of Bioscience and Human-Technology
Tsukuba 305, Japan

N. Yamazaki
Faculty of Science and Technology
Keio University
Yokohama 223, Japan

Abstract

The acquisition process of bipedal walking in humans was simulated using a neuro-musculo-skeletal model and genetic algorithms, based on the assumption that the shape of the body has been adapted for locomotion. The model was constructed as 10 two-dimensional rigid links with 26 muscles and 18 neural oscillators. Bipedal walking was generated as a mutual entrainment between neural oscillations and the pendulous movement of body dynamics. Evolutionary strategies incorporated for example as fitness in the genetic algorithms were assumed to decrease energy consumption, muscular fatigue, and load on the skeletal system. An initial population of 50 individuals was created, and an evolutionary simulation for 5,000 steps was conducted. As a result, the shape of the body changed from that of a chimpanzee to that of a modern human, and the body size nearly reached the size of a modern human. These simulation results show that improving locomotive efficiency and reducing the load on the musculo-skeletal system are important factors affecting the evolution of the human body shape and bipedal walking. Such computer simulation helps us to understand the process of evolution and adaptation for locomotion in humans.

Key Words: Bipedal Walking, Evolution, Neuro-Musculo-Skeletal Model, Genetic Algorithm

1 Introduction

Erect bipedal walking is unique to hominids, and provides important clues concerning the origin of humans. Although fossils have been studied and anthropoid apes have been analyzed in terms of genetics and biomechanics, in order to explain the process of human evolution, it is difficult to reproduce and to analyze changes in movement, such as bipedal walking using such methods. However, computer simulation techniques are helpful in recreating such evolutionary processes. Recently, research on artificial life has attempted to trace evolutionary processes in many types of organisms using computer simulation. However, these studies have investigated low level or abstract organisms whose structures are relatively simple[1,2]. In order to reproduce more complicated evolutionary changes in the shape of the body and movement, such as bipedal walking in humans, it is important to consider morphological and biomechanical factors such as the structure of the musculo-skeletal system and principles of motor control. In this study, we developed a new computer simulation method and recreated the process of human evolutionary changes in the shape of the body and in patterns of bipedal walking from ancestors to modern man from a biomechanical point of view. The biomechanical factors for such human evolution were also investigated by producing alternative imaginary lines of evolution.

2 Method

2.1 Neuro-musculo-skeletal model

A model to generate autonomous bipedal walking has been proposed[3,4], where bipedal walking is generated as a mutual entrainment between neural oscillations and the pendulous movement of body dynamics. Based on this concept, we structured a neuro-musculo-skeletal model closely approximating the actual body structure of humans, as illustrated in Fig. 1 (see Yamazaki et al.[5] for details).

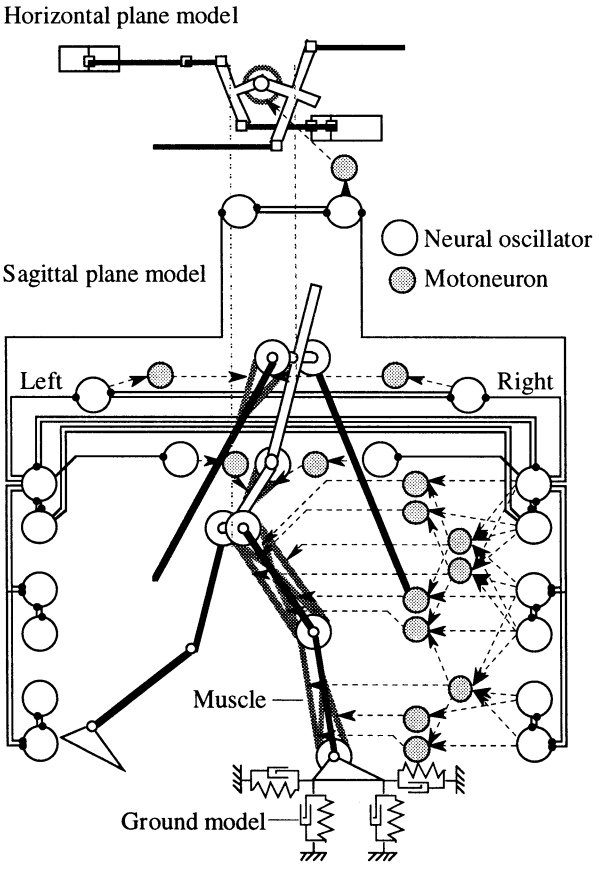


Fig. 1 Neuro-musculo-skeletal model

The shape of the human body was modeled as two two-dimensional models: 10 rigid links representing the feet, calves, thighs, pelvis, upper torso including the head, and upper extremities in the sagittal plane; and 2 rigid links representing the pelvis and the upper torso in the horizontal plane in order to reproduce rotation at the waist. The principal 26 muscles involved in movement in the shoulder, waist, and lower extremities were also modeled. Each joint is driven by the moment due to muscular tension and is also affected by the non-linear viscous and elastic moment representing soft tissues such as ligaments. The nervous system was modeled as a neural network called "rhythm pattern generator" as shown in the figure, which consists of 18 neural oscillators generating the basic rhythm of movement. The output from the neural oscillators becomes the stimulation signals to the muscles through motoneurons. Furthermore, somatic information such as the angle and angular velocity of each segment and the conditions of foot contact with the ground is fed back to the neural oscillators. The ground is approximated by springs and dampers.

These models can be expressed as a total of 64 non-linear, first-order differential equations: 28 equations of motion for the musculo-skeletal system; and 36 equations for the nervous system. A walking movement can be obtained numerically by calculating these equations. The shape of the body and the pattern of movement of this model are evolved by an evolutionary algorithm.

2.2 Algorithms for evolution

As the basic mechanism of organic evolution, we employed the theory of natural selection. That is, evolution is thought to proceed through the repeated process of natural selection, with the diversification of organic characteristics being the product of crossing and mutation of genetic information. In addition to these genetic factors, the shape of the body and patterns of movement can also change through acquired factors such as the development of the nervous system resulting from learning and the adaptation of physical tissues, such as bones and muscles, to mechanical loads. The ability of acquired adaptation and learning is related to survival fitness for the environment, and is also assumed to be a factor in evolution. We developed an evolutionary algorithm based on this concept as shown in Fig. 2.

Although this algorithm is based on genetic algorithms imitating the theory of natural selection, the genetic processes and the acquired processes are differentiated clearly to demonstrate the organic evolutionary course in more detail. That is, this algorithm has a double-loop structure in which the outer loop represents the genetic processes and the inner loop represents the acquired processes. Basic body parameters such as segment length are described in a progression corresponding to a chromosome. According to this chromosome, all body parameters necessary for the neuro-musculo-skeletal model are calculated (an individual is generated). By

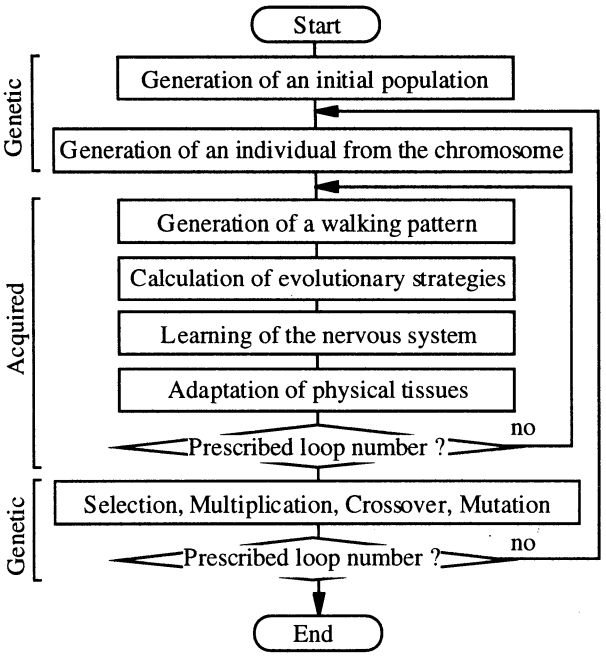


Fig. 2 Flow of the evolutionary simulation.

repeating such this process for the initial stage of evolution, an initial population is generated. Each obtained individual generates a walking pattern, and the values called evolutionary strategies are calculated according to the movement. The direction of evolution in the shape of the body and movement can be determined by evaluating these evolutionary strategies. These values operate to represent levels of survival fitness for the environment in natural selection and can also be used as criteria of movement for the acquired processes. In other words, based on these values, searches of nervous parameters (learning of the nervous system) and mechanical adaptation of physical tissues, such as muscles and bones, are conducted in the acquired processes, whereas selection and multiplication are operated in the genetic processes. In the genetic processes, genetic operations such as crossover and mutation are also carried out. Because body parameters such as segment length are described in the chromosome, the genetic operations enable various body shapes to be produced. Whenever the outer loop is repeated once, selection and multiplication in an individual take place. This repeating of an outer loop is called a "step" in the following text.

3 Results and Discussion

3.1 Conditions of simulation

The evolutionary course traced in this simulation is from human ancestors walking on two feet to modern humans. Although the shape of the body and walking pattern for the common ancestors of humans and great apes are still not completely clear, we created a human ancestor model with reference to the chimpanzee. An

initial population of 50 individuals was created, and the repeated calculations for 5,000 steps were conducted. For these calculations, 6 workstations connected in a network were used in parallel, and it took about 40 hours.

3.2 Reproduced evolutionary process

The following three evolutionary strategies were defined from a biomechanical viewpoint. (1) Metabolic energy consumption was calculated from muscular tension. (2) Muscular fatigue was calculated from muscular stress. (3) Load on the skeletal system was calculated from the maximum moment and reaction forces acting on the joints. These three values were normalized by body weight and travelling distance based on a formula for level locomotive efficiency[6]. Computational evolution was performed to minimize each value. Figure 3 shows the recreated evolutionary process for bipedal walking. Each picture is traced at 0.1 sec intervals. In order to avoid the overlapping of motions, each picture is shifted 15 cm to the right of the previous one. The thickness of each segment is drawn in proportion to its mass. In addition, because the ground is modeled as springs and dampers, the feet sink into the ground.

In the initial stage, the neural parameters were decided by trial and error to generate a pattern of walking that closely matched the bipedal walking patterns of a chimpanzee, with inclined torso, bent knee and hip joints, and relatively slow walking speed. The calculations for evolution started from this initial condition. Changes in walking pattern such as extensions of the knee and hip joints, straightening of the torso and increase in walking speed were observed as selection and multiplication were repeated. The torso became upright at a relatively early stage. Thus, straightening the torso appears to be an important means of improving walking efficiency. The amplitude in the swing of the upper extremities became large at about 1,000 steps because the inertia moment around the vertical axis at the upper torso segment was reduced as the torso straightened rapidly at this stage. As for the activities of the muscles in the lower extremities, there is generally a shift from ape-like walking to human-like walking. That is, extension moment at the hip joints reduces and extension moment at the ankle joint increases. As for the shape of the body, both segment length and mass increased, and the shape of the body approximated that of modern humans after 5,000 steps. The upper extremities which were longer than the lower extremities in the initial stage became almost equal after around 3,000 steps, and thereafter became shorter than the lower ones.

These changes in the shape of the body and patterns of movement are almost identical to what is widely held to be the evolutionary course of humans[7]. In particular, the ratio of length for the upper and lower extremities and the posture with curvature of the lumbar after about 3,000 steps shows characteristics of *Australopithecines* which are assumed to have existed about 3.5 million years ago[7].

3.3 Imaginary evolution

Computer simulation makes it easy to set various calculating conditions, and so allows us to test alternative, imaginary evolutionary scenarios. To investigate human evolutionary strategies further, the following three sets of conditions were set. (1) In addition to the three evolutionary strategies described above, maximization of locomotive speed was added as an evolutionary strategy. That is, the process aimed to acquire a shape of the body and patterns of movement that are adapted to high-speed locomotion. (2) Among the three strategies mentioned above, load on the skeletal system was omitted and only energy consumption and muscular fatigue were minimized in order to improve movement efficiency. (3) For the third set of conditions, only the load on the skeletal system was considered, with the other two strategies being

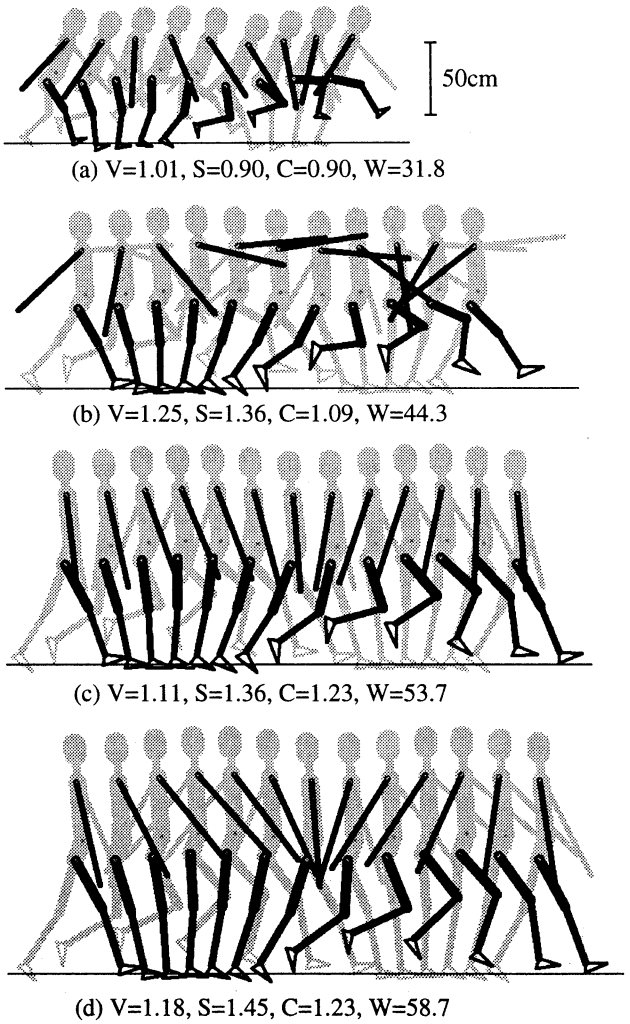


Fig. 3 Changes in the shape of the body and walking pattern from initial stage to evolutionary stage in the simulation. (a) Walking pattern in an individual at the initial stage. (b) Individual after 1,000 steps. (c) Individual after 3,000 steps. (d) Individual after 5,000 steps. In the figure, 'V' indicates walking speed [m/s], 'S' is stride length [m], 'C' is walking cycle [s], and 'W' is body weight [kg].

omitted. That is, reducing the load on the skeletal system was given priority over the others. A calculation of 5,000 steps was conducted for each of these three sets of conditions. The simulation results are shown in Fig. 4. The calculating conditions and display methods are the same as those shown in Fig. 3.

When the locomotive speed was maximized, the shape of the body did not become larger, but a running motion in which the torso remained inclined forward and both feet left the ground emerged (Fig. 4(a)). When minimizing the load on the skeletal system was omitted, the feet and the thigh segments became larger and the stride length became longer (Fig. 4(b)). When only the skeletal load was minimized, the shape of the body did not become so large and the calf length tended to become longer than the thigh length (Fig. 4(c)). As such, under the conditions for evolution described here, a body shape and walking pattern similar to those of modern humans could not be obtained. Thus, it is quite reasonable to assume that the shape of the body and walking pattern of the modern human may have resulted from both evolutionary strategies for higher locomotive efficiency and reduced loads on the musculo-skeletal system as shown in Fig. 3, but not from these conditions for evolution as shown in Fig. 4. Consequently, modern humans have been also adapted to continuous walking for many hours.

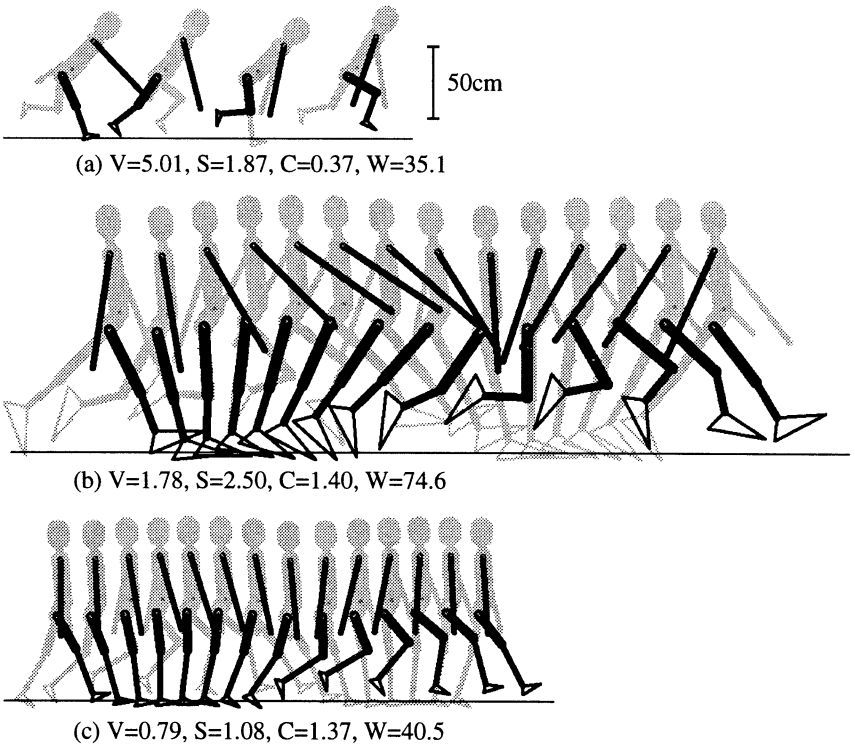


Fig. 4 Simulation results based on various conditions for evolution. (a) When maximizing locomotive speed is added to the evolutionary strategies. (b) When only energy consumption and muscular fatigue are minimized without minimizing load on the skeletal system. (c) When only load on the skeletal system is minimized.

4 Conclusions

In this study, we have presented a possibility to recreate the acquisition process of erect bipedal walking by computer simulation. The simulation results from an imaginary ancestor model with reference to the chimpanzee agrees with assumptions within anthropology about the pattern of human evolution. The evolutionary strategy underlying the acquisition of bipedal walking may be improvement of energy efficiency and decrease of loads on the musculo-skeletal system. Other strategies such as fast walking resulted in different body shapes and walking patterns from those of modern humans.

In this simulation, the shape of the body was modified to adapt to movement. Movement was influenced by changes in the shape of the body. In actual evolution, the shape of the body and movement may form and adapt mutually, and are not independent. Body dynamics such as muscular tension plays an important role in this interaction. The computer simulation which we have developed is useful for explaining the relation of cause and effect in evolution.

References

[1] Toquenaga Y, Ichinose M, Hoshino T, Fujii K (1994) Contest and scramble competitions in an artificial world: Genetic analysis with genetic algorithms. In: Langton CG (ed) Artificial Life III, pp.177-199, Addison-Wesley

[2] Sims K (1994) Evolving 3D morphology and behavior by competition. In: Brooks RA, Maes P (eds) Artificial Life IV, pp.28-39, MIT Press

[3] Taga G, Yamaguchi Y, Shimizu H (1991) Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biol Cybern 65:147-159

[4] Taga G (1995) A model of the neuro-musculo-skeletal system for human locomotion. I. Emergence of basic gait. Biol Cybern 73:97-111

[5] Yamazaki N, Hase K, Ogihara N, Hayamizu N (1996) Biomechanical analysis of the development of infant walking by a neuro-musculo-skeletal model. Folia Primatol 66:253-271

[6] Alexander RM, Goldspink G (1977) Mechanics & Energetics of Animal Locomotion. Chapman and Hall

[7] Campbell B (1985) Human Evolution. Aldine Publishing

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

The genetic algorithms (GAs) to evolve multiple agent cooperative system

Masanori Sugisaka Xiaoshu Wang* Ju-Jang Lee**
Department of Electrical and Electronic Engineering
Oita University, 700 Dannahara, Oita 870-11, Japan
Department of Electrical Engineering**
Korea Advanced Institute of Science and Technology**
msugi@cc.oita-u.ac.jp
wangxs@cc.oita-u.ac.jp*
jjlee@ee.kaist.ac.kr**

Abstract

In this paper the micro-robot soccer-playing game, such as that of MIROSOT (Micro-robot World Cup Soccer Tournament), is adopted as a standard test bed for the research on multiple agent cooperative system. It is considerably complex and requires expertise in several difficult research topics, such as mobile micro-robot design, motor control, sensor technology, intelligent strategy planning and etc., to build up a complete system to play the game. In addition, because it is a kind of antagonistic game, it appears ideal to test whether a method is better than another or not.

Up to now there exists two different kinds of architecture for building up such system. One is called as vision-based or centralized architecture and another is known as robot-based or decentralized architecture. The main differences between them lie in whether there exist, or not, a host computer system, which responds to data processing and strategy planning and etc., and a globe vision system, which can view the whole playground and transfer the environment information to the host computer in real time. We can believe that the latter is a advanced type than the former. But in preliminary step of our study, the former is used for it can lighten overload of the micro-robot design.

In this paper, at first a simplified layer-model of the multiple agent cooperative system is proposed. Based on such model a system for micro-robot soccer-playing game is organized. At the same time the simple genetic algorithms (SGAs) is used for the autonomous evolution of cooperative behavior among micro-robots. In the end a computer simulation system is introduced and some simulated results are explained.

Keywords: *multiple agent system, MIROSOT, genetic algorithms, SGAs*

1 Introduction

In last few years researches on the multiple agent control system have been drawing strong interests for its potent problem-resolving capacity and lots of competitive posters existing till now. Some completely new viewpoints on the recognizing of the nature and some efficacious approaches in practical application may be resulted in along with the through study.

Unlike the conventional one-entity tight-coupled large system, the multiple agent system generally consists of several relatively simple but integrated individuals. The assigned job ought to be completed by the cooperation among these individuals rather than any individual's own activities. But the relation of these individuals are loose and dynamic. It is extremely difficult to coordinate them both at the level of mechanics and of the control system. Moreover, a multiple agents system usually exists within a complicated external environment that varies constantly and can not be predicted in advance. So the interaction between the multiple agent system with the external environment ought to be in real time, which is also an extreme difficulty existing.

For the reasons above, it seems reasonable to use an automatic procedure rather than a pre-planing procedure to keep the path with the variation of environment. The genetic algorithms (GAs), which is a kind of search method based on the mechanics of natural selection and natural genetics, appears just such a kind of automatic procedure. In this paper an autonomous control mechanism based on the genetic algorithms is proposed.

2 The model of multiple agents system

Here a simplified but relatively common model of the multiple agent system is shown in Fig.1.

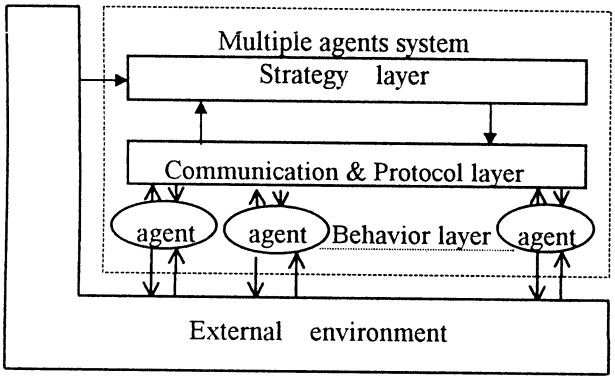


Fig.1 A simplified model of the multiple agent system.

The multiple agent system is divided into several layers in logic. The specific layers' structure may be very different according to the practical system. But following three layers are necessary at least.

Behavior layer: It is composed of agents themselves, which are basic behavioral cells of whole system. These agents may be 'partially blind' or 'blind', which are depended on whether these agents can perceive the external world by themselves or not. In return behaviors of these agents will change the external environment constantly.

Communication and protocol layer: It supplies a information channel among agents and between the layers. By it, the agent can be organized as integration. It will decide which messages will be exchanged and how to be exchanged. The protocol is defined to exactly describe the practical communication specification such as data format, error correcting and acknowledging mechanism and etc. between the host computer and micro-robot.

Strategy layer: It responds to analysis environment information in real time and make optimum decisions to guide the agent to complete the task assigned in advance. In general the autonomous evolution strategy such as neural network, genetic algorithms, fuzzy planning and etc. is needed here.

3 Micro-robot soccer-playing game

Now the micro-robot soccer-playing game is often used as ideal experimental bench for the research on the multiple agent cooperative system. The overview of such micro-robots soccer-playing system is shown in Fig. 2.

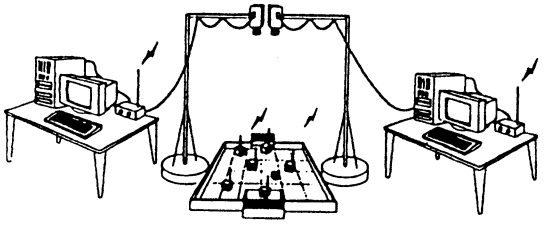
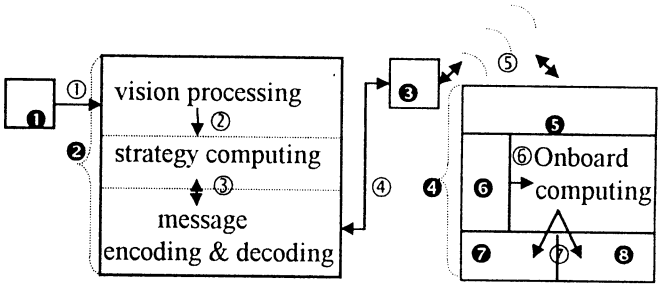


Fig.2 Overview of the micro-robots soccer-playing system.

The game is played by two teams, which respectively consists 3 or more micro-robots, in a playground of 130cm x 90cm in dimension. In addition the size of micro-robot is limited to 7.5cm x 7.5 cm x 7.5cm and each micro-robot ought to be fully independent, which means that power and motor systems must be self-contained. But some appendages such as arms and infrared sensors can be equipped if meets the size restriction. In addition, a CCD or video camera vision system, a host computer system and a wireless communication system are necessary. During the game the CCD or video camera is fixed just over the playground and takes the vision picture of the entire play-ground in real time, then the visual message is transformed into digital frame data and transferred into the host computer. So in fact the computer is real control center of this system. It responds to the vision processing, robot and ball locating, strategy planning, message encoding-decoding and so on. The command is sent to micro-robots by the wireless communication system. So each micro-robot must be equipped with a wireless receiver at least. In end, the micro-robot acts in accordance with the command from the host computer and input from its own sensors. The system structure based on the viewpoint of data flow is showed in Fig.3.



❶: CCD or video camera, ❷: Host computer system,
❸:Wireless transmission unit; ❹:Micro-robot;
❺: Wireless transmission unit embedded in micro-robot;
❻: Sensors embedded in micro-robot;
❼: Left motor; ❽: Right motor

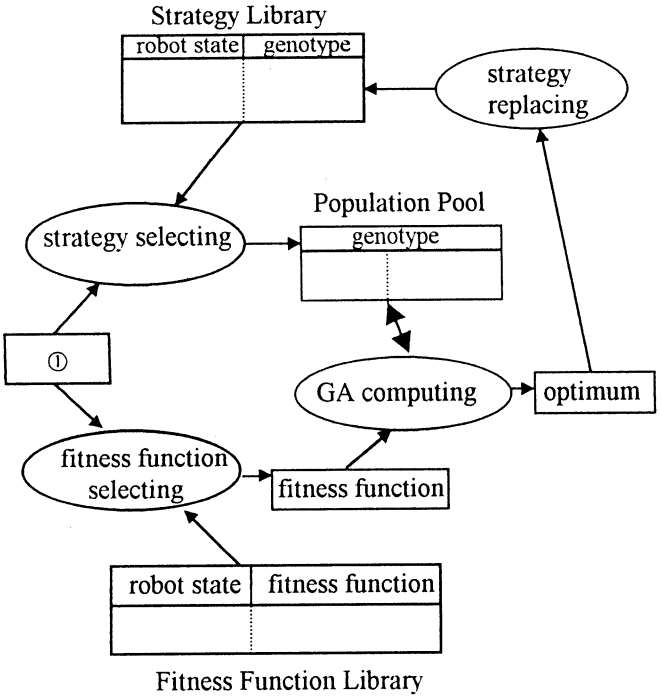
①: Raw vision image; ②Normalized environment data;
③: Command from strategy layer *or*
Requirement from micro-robots
④: Formatted transmission frame data
⑤: Wireless transmission signal;⑥: Sensors data
⑦: Motor driving signal

Fig.3 System data flow-chart.

We can find that the micro-robots soccer-playing system described above can be well corresponded with the multiple agent system's model shown in the section 2. This kind of system architecture is called as centralized control or vision-based architecture. Contrasting with it, another kind of decentralized control or robot-based architecture exists. In latter the host computer and the vision system attached with the host computer can not be used.

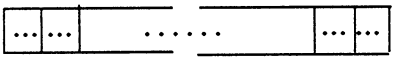
4 The control strategy

The structure of control module is shown in Fig.4.



(①: *state of the robot who is closest to the ball*)
Fig. 4 The structure of the control module.

Here, the genome is organized as a bit string as follow,

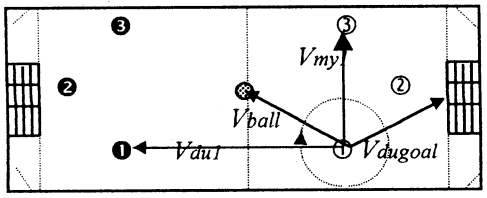


action of robot 1 *action of robot n*

we presume that each micro-robot can move by sixteen directions. Then 4 bit is needed for each micro-robot in the genotype. The length of whole genotype will be $4*n$.The control flow is explained as below.

In addition, The state a micro-robot is described as

$\{\{V_{my1}, V_{my2}...V_{my(n-1)}\}, \{V_{du1}, V_{du2}...V_{dum}\}, V_{ball}, \{V_{mygoal}, V_{dugoal}\}\}$



①②③: *home players;*
❶❷❸: *opponent player;*
⊗ : *ball;*
Fig. 5. The location of micro-robot in the play field.

Here,$\{V_{my1}, V_{my2}...V_{my(n-1)}\}$is set of vectors from its own position to the position of its partners, They are sequenced from *1* to *(n-1)* by clockwise, the sum of micro-robots in home team is n. $\{V_{du1}, V_{du2}...V_{dum}\}$is set of vectors from its own position to the position of opponents. They are sequenced from 1 to m by clockwise, here n is sum of micro-robots in opponent team. Usually n is equal to m. V_{ball} is the vector from its own position to the position of the ball. $\{V_{mygoal}, V_{dugoal}\}$ is set of vectors from its own position to the its own goal and opponent's goal respectively. Here we do not use the coordinate values *(x,y)* directly to describe the position of micro-robot because they can not express the relations well. If the micro-robot involved in the game increases, the memory load will be a problem. If so, we can only select those players who are most close to it to express its state because we can believe that the behaviors of any micro-robot is mainly affected by the environment around itself.

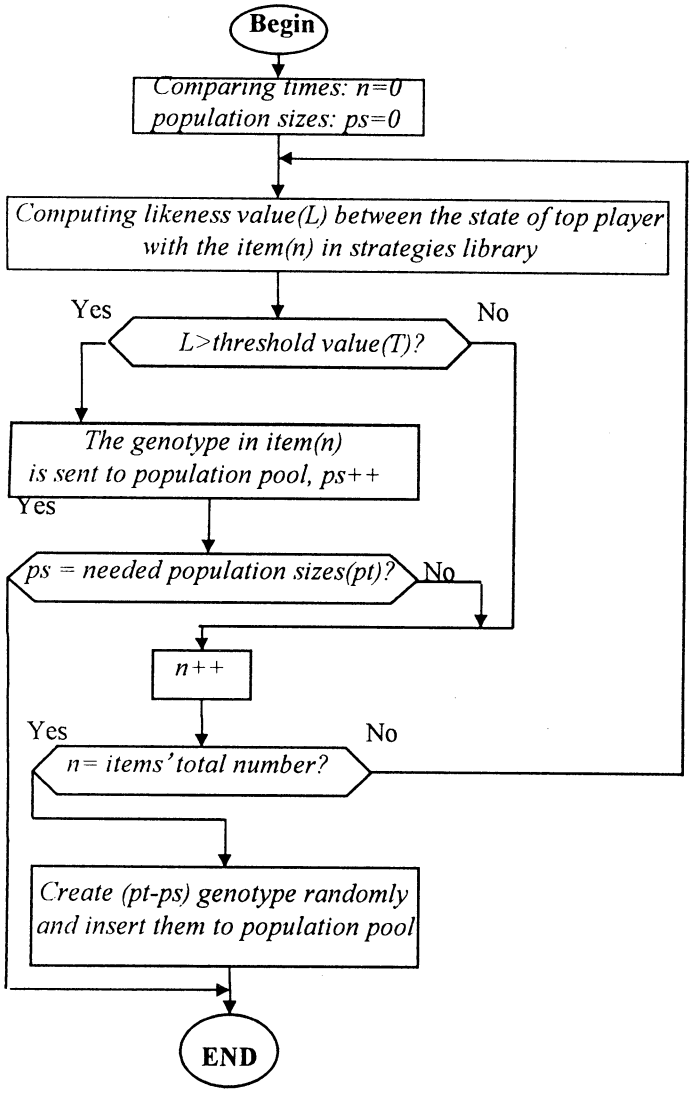
Step 1: The micro-robot who is closest to the ball is selected as top player;

Step 2: The state of the top player is compared with the items in the strategy library. This processing is called as strategy selecting. In fact this processing is a fuzzy computation because it is impossible to gather all states within the library. In initial, the strategy library may be blank or inherits content from former system. The former looks like a person who did not play soccer game at all in before. The latter appears like a person who has been taught how to play a soccer game. The flow chart of such processing is showed in Flowchart 1;

Step 3: The state of the top player is compared with the items in the fitness function library. Thought the end goal of the team is to kick the ball into the opponent goal and keep the ball from its home goal, this object is too general to be expressed and achieved easily. So we can divide this end goal into several sub-goals according to the current environment. Here we require that this fitness function library must be built up before the game begins. It is arduous but very important to the performance of whole system. Like the strategy selecting, the fitness function selecting is also a fuzzy processing. The item which has greatest likeness value with the state of top player is selected as current fitness function to take part the genetic computation in the next step;

Step 4: The genetic computation on the genotype population runs. The genotype population is created during the *step 2* and the fitness function is selected during the *step 3*. Here the simple genetic algorithms (SGAs) is used and those parameters such as *population sizes, crossover probability, mutation probability* and so on can be set manually in advance. The result of this step is that a optimum genotype is created and transferred to communication and protocol layer to create a command in end;

Step 5: A replacing procedure runs on the strategy library to make it more efficient and effective for the game. This feedback channel is very necessary for this system.



Flowchart 1. The procedure of strategy selecting.

5 Simulated experiment

Currently the methodology of computer simulation is often used during the preliminary stage of a system design. The difference between the real environment with the simulated one can be eliminated by designing the simulation model carefully. In our initial stage a computer simulation system is developed for the test of the control method explained above, which appears feasible.

Two simulated results are showed in Fig. 6 and Fig.7. Former shows a typical case of offensive cooperation. The ball and top player locate at the opponent field. But there are two opponents who keep the ball from being kicked to the goal directly. So the top player decides to pass the ball to his partner. Fig. 7 shows a typical case of defensive cooperation. The ball is on home half-field. So top player and his partner try to prevent the ball from home goal.

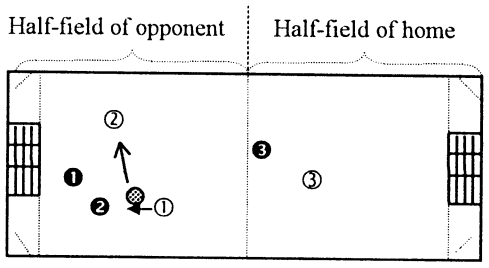


Fig 6. Offensive cooperation.

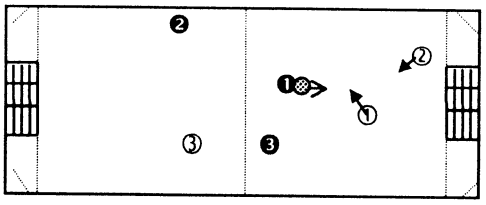


Fig 7. Defensive cooperation.

6. Conclusions

In this paper a simplified layers model of the multiple agents control system is proposed at first. As a experiment example of such model, a kind of micro-robots soccer-playing game is adopted. During the achievement of control strategy the simple genetic algorithms (GAs) is used for the autonomous evolution of cooperation activity of micro-robots. In end a computer simulation system is introduced and some simulated results are shown. From those results we can be sure that genetic algorithms is one of feasible try during the design of a multiple agents system. Of course some problems, such as the computation load and time consumption, exist still. To eliminate these shortcomings, some measures, such as a strategy library and a fitness function library, are involved.. This system will be tested on a real environment in next step to tell whether these measures are efficient or not.

References

[1] T. C. Hsia and M. Soderstrand: "Development of a micro robot system for playing soccer games", Proceedings of KAIST MIROSOT'96, Taejon, Korea, , 9-12 Nov. 1996, pp.149-152.

[2] Sahota, Mackworth Kingdon and Barman: "Real-time control of soccer-playing robots using off-board vision", IEEE Intern. Conf. System, Man and Cybernetics, Canada, 22-25 Oct. 1995, pp. 3090-3093.

[3] K.Ohkawa, T.Shibata and K.Tanie: "Self-generating algorithms of evolution for cooperative behavior among distributed autonomous robots", Proceedings of AROB'97, Beppu Oita, Japan, 18-20 Feb. 1997, pp123-126.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

The Gesture-controlled Interactive Games by Artificial Retina Chips

Kazuo Kyuma, Hiroshi Kage, and Yasunari Miyake

Mitsubishi Electric Corporation
Advanced Technology R&D Center
8-1-1 Tsukaguchi-Honmachi, Amagasaki-city, Hyogo, 661 Japan
TEL:81-6-497-7078, FAX:81-6-497-7288
e-mail: kyuma@qua.crl.melco.co.jp

1. Introduction

Human beings can execute very difficult tasks in real time. This is due to the inherent parallelism of our visual system, particularly of our retina, which combines image sensing and processing functions. In the past few years, we have reported artificial retina chips (AR chips) which mimic the excellent functions of human retinas, that is the simultaneous functions of image sensing and image processing on one chip. In this talk, we describe the AR chips and their applications to the interactive games controlled by gesture input instead of the handheld keyboards/joysticks.

2. Artificial Retina Chips

The Si AR chips consist of an two-dimentional array of variable sensitivity photodetection Cell (VSPC), a random scanner, and a multiplexer, as shown in Fig. 1. The principle operation is based on a novel type of optical vector-matrix multiplication. The two-dimentional optical image **W** is projected onto the AR chip as a matrix. On the other hand, the electric control voltage is applied to the artificial retina chip as a vector, in order to modulate the detection sensitivity **S** of the VSPC. Thus the output current vector **J** is expressed as **J=WS**. The choice of the appropriate control voltage allows the AR chip to perform various kinds of image processing, including simple image sensing, edge extraction, pattern matching, random access, image compression, etc. The some of the image processing functions are shown in Fig. 2. In Fig. 1, the operation principle for the one-dimensional spatial filtering is illustrated for simplicity. However, the extension to a two-dimensional spatial filtering is easily achieved by using the chip architecture shown in Fig. 3, where two kinds of the scanner have been integrated on the chip. The kernel scanner decides the horizontal area to be processed, and the kind of 2D spatial filtering. On the other hand, the row selector scanner decide the vertical area to be processed.

The Si AR chips with 32×32, 128×128 and 512×512 pixels have been fabricated by the CMOS technology. Their profound features of the AR chips are programmable image processing, wide dynamic range (40dB, 10dB wider than CCDs), low power consumption (10mW, 10 times lower than CCDs), and variable frame rate (1-1000Hz). The AR systems have also been developed. One is an AR camera with a 128×128 pixel chip. Another is an AR module with a 32×32 pixel chip and a μP. The μP controls the processing mode of the AR chip, and also executes the recognition algorithm. The Fig.4 shows the AR camera in which an AR chip, a 16-bit microprocessor, and a 1 Mbit SRAM are installed in the multichip module (MCM).

2. Interactive Games

The experiments on the interactive games were performed by using the AR module. Two kinds of real time vision algorithm, which are suitable for the AR chips, have been developed. They are based on the shape analysis and the optical flow model. In the shape analysis algorithm, the input image is transformed to the

equivalent rectangle. The output signals from the module, obtained as the coordinates of the gravity center, the vertical and horizontal length, and the tilt angle of the rectangle, were used for the gesture recognition. In the optical flow model, the input image was divided into 6×6 regions and the 36 optical flow signals were obtained from each region.

We have succeeded in the control of the game character by the gesture input. The AR systems directly detects the player's body movements and translates them into the character's action on the game screen. Figure 5 shows how our interactive game system works, ranging from image capturing by the AR chip to the videogames, through some process of two vision algorithms described above. Two kinds of games, Decathlete and Nights, are used for demonstration of the authorities of SEGA enterprises.

Decathlete, a track-and-field game, has ten events such as 100m hurdles or javelin throw. In the developed interactive game, a player has only to stand in front of the artificial retina module, and jump, run, or make a gesture of throwing a javelin.

In Decathlete, the gesture is analyzed by the optical flow algorithm. In order to convert the calculated optical flow pattern into the effective signal to the machine, all optical flow signals are summed up to take the vertical or horizontal amplitude into amount. If the amplitude is larger than a predetermined threshold, then a command for jump, for example of the hurdle game, is transmitted to the machine. Similar conversions are made in other events. The required time for image detection, recognition and feedback to the game character is less than 16 msec.

In Nights, a fantastic 3-D flying game, a character appears and can fly through 3-D space to any direction by the player's control. By using the center of mass method, its direction can be controlled by the player's hand gesture.

We have demonstrated this game system at various public exhibitions, including Comdex 96 (Las Vegas) and Cebit 97 (Hanover). Figure 6 shows a photograph of a scenery at Cebit'97.

3. Summary

The AR chips execute multiple, easily programmable forms of image processing, which will open the revolutionary image processing systems with small size, fast processing, low cost, and low power consumption. In this talk, The application to the interactive games by gesture/body action control was introduced. As shown in Fig. 7, other applications include human-interface, security, robotics,video/still camera, medical equipments, automobile, etc.

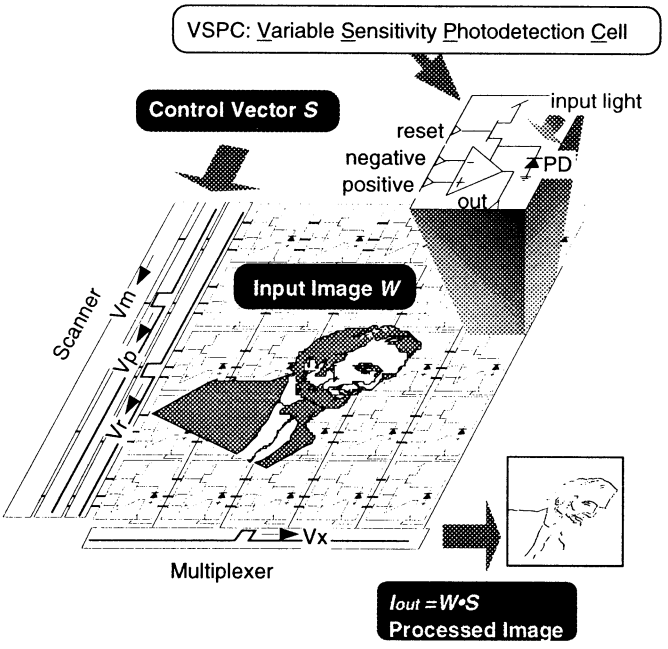


Fig.1: Structure of Si AR chip

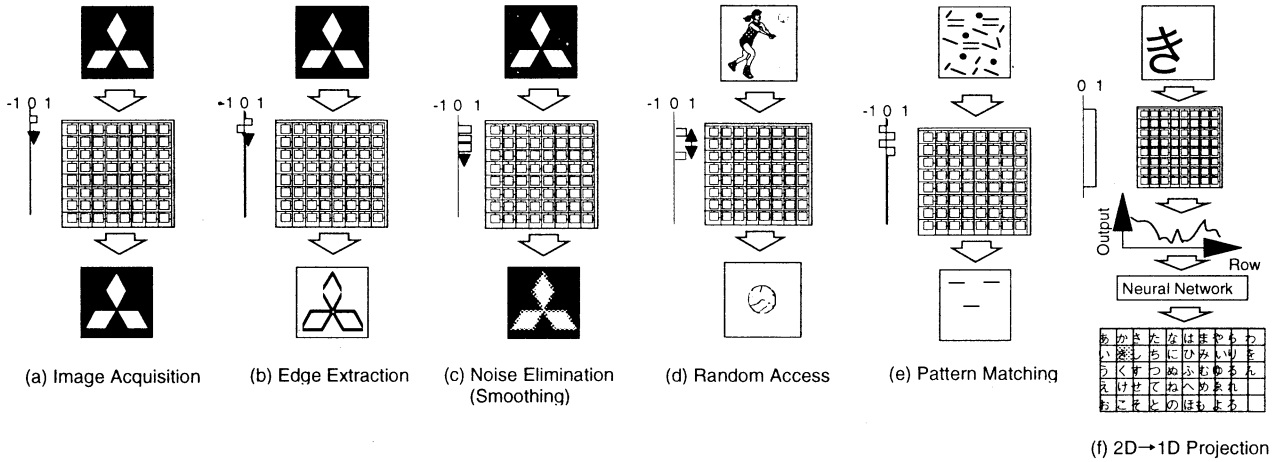


Fig.2: Several examples of processing functions by AR chips.

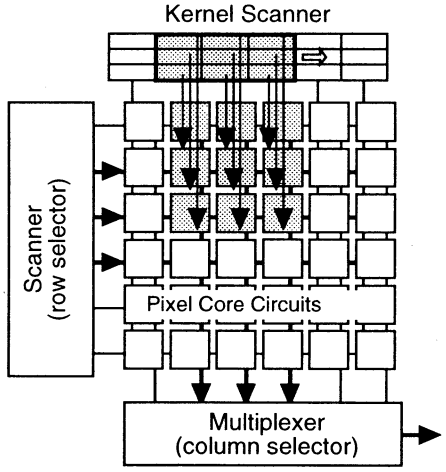


Fig. 3: Structure of AR chip with 2D spatial filtering function.

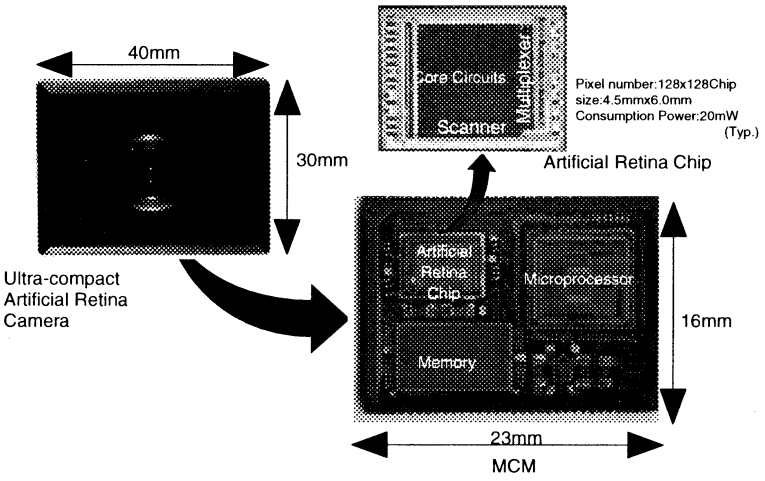


Fig. 4: Photograph of AR camera.

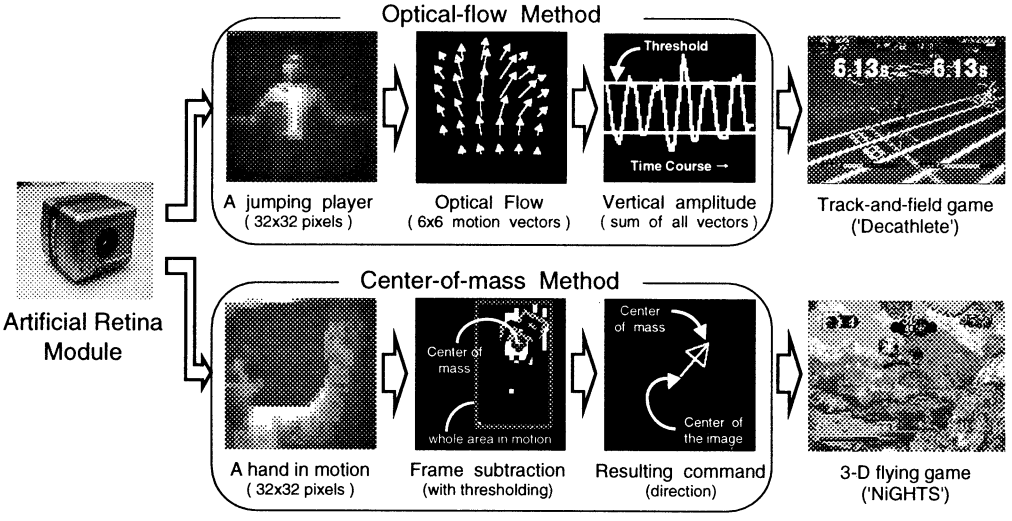


Fig. 5: Image processing flow of interactive game.

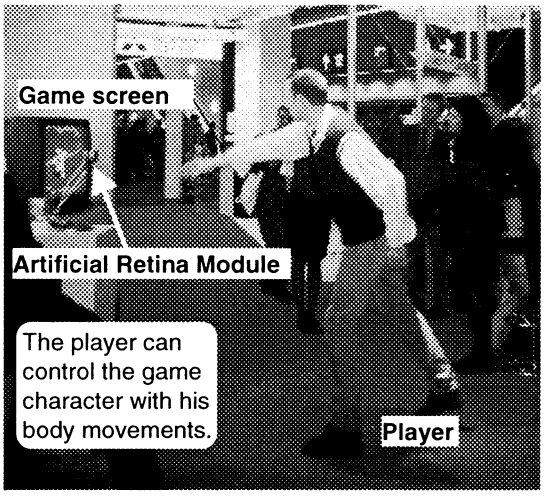


Fig.6: Photograph of demonstration at Cebit '97.

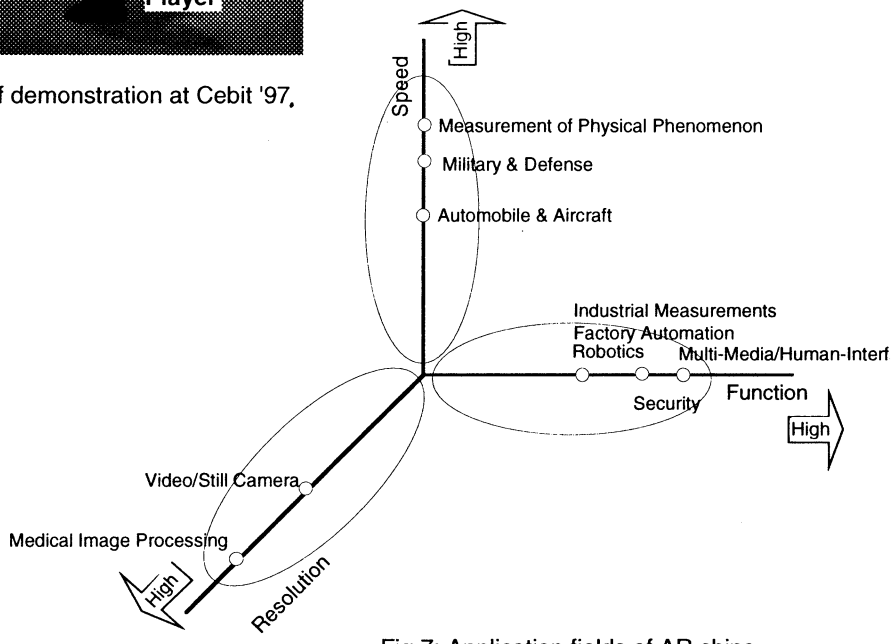


Fig.7: Application fields of AR chips.

(References)

1) E.Lange, Y.Nitta, and K.Kyuma, "Optical Neural Chips" , IEEE MICRO, 14, 6, 29-41, 1994.

2) K.Kyuma et. al., "Artificial retinas - fast,versatile image processors-" , nature, 372, 6502, 197-198, 1994

3) J.Ohta et. al., "Variable sensitivity photodetector for optical neural networks" , IEEE J. Lightwave Technol.,9, 1747-1754, 1991.

4) E.Funatsu et. al., "An artificial retina chip with current-mode focal plane image processing functions" , IEEE J. Electron Devices, 44, 1777-1782,1997.

5) Y.Miyake et. al., "A gesture controlled human interface using an artificial retina chip" , Proc. IEEE LEOS'96 Annual Meeting, Volume 1, TuT3, 292-293, 1996.

6) T.Toyoda et. al., "Man-machine interfaces using image information obtained by a 32×32 pixel artificial retina chip" , Proc. Optics in Computing (IEEE-LEOS,OSA), OThC4-1, 181-183, 1997.

7) W.T.Freeman and M.Roth, "Orientation histograms for hand gesture recognition" , Int'l workshop on automatic face- and gesture-recognition, Zurich, Switzerland, Dept of Computer Science, University of Zurich, CH-8057, 1995.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Building a Machine for Evolution *in Silico*

Edward A. Rietman, Bell Labs, Murray Hill, NJ 07974

Robert Slous, Xilinx, San Jose, CA 95124

Hitoshi Hemmi, NTT, Kanagawa, Japan

Hugo de Garis, ATR, Kyoto, Japan

Katsunori Shimohara, ATR, Kyoto, Japan

Abstract

We review the history of evolvable hardware and show that there was a natural progression from the programmable logic device (PLD) to the field programmable logic array (FPGA). We describe a system we built to test conjectures concerning evolvable hardware. The latest FPGAs can be memory mapped to a CPU. This makes building these systems very easy. We describe a CPU address-space mapped system that uses an Intel 8255 programmable peripheral interface chip between the CPU and a Xilinx 6216 FPGA.

Introduction

In this paper we describe the design and programming of a hardware system that enables one to evaluate conjectures concerning *in silico* evolution; that is, the evolution of hardware circuits using evolutionary search. In the first part of this paper we discuss the evolution of evolvable hardware. In the second part we discuss FPGAs in general and the XC6216 chip in particular; the key enabling component for *in silico* evolution. In the third part of the paper we discuss the software programming at the bit-level. Finally we describe the design of a hardware system we built. From that exercise we discovered flaws in the system design; we describe the flaw and a "work-around." We conclude with a philosophical comment on engineering.

Adaptive Hardware - Historical Comments

The idea of evolvable hardware is synonymous with adaptive hardware. The neural network approach, where the system is being trained by a teacher signal, and the so called genetic approach, where the system is evolved with a fitness function, are examples of adaptive hardware. Evolvable and adaptive hardware showed a natural evolution in the real-world as engineers sought solutions to real-world problems; and as we will show, the concept of evolvable hardware cannot be attributed to one person. The idea of using evolvable hardware or adaptive hardware has been discussed and experimented with for decades. Today we have better tools for our empirical studies.

To our knowledge, the idea of using evolvable hardware with heuristic programming was first coherently outlined by Wolfram (1986). His several page description proposes using genetic algorithms and other heuristic techniques to design circuits using programmable logic devices (PLD).

The first adaptive hardware available was programmable logic devices. Relative to the number of years that programmable logic devices were on the market, Bostock (1988) is a recent review. These systems consist of two arrays of logic gates, for example AND and OR, and a connection matrix of diodes. Circuit design is done with CAD software and the diode connections are then disabled or "burned in" to "wire" the circuit. This is a one-shot solution and is not very useful for heuristic programming. These chips solved the engineering problem where a customer needed only a few samples of essentially a custom chip. They are not a cost-effective replacement for ASICs (application specific integrated circuits).

There was a natural progression from these programmable logic devices to ones with floating gate transistors that would allow one to erase the circuit using UV light, then to ones that could be electrically erased, and finally field programmable devices that could be written to and erased many times. These devices allowed engineers to download circuits "in the field" and are called FPGAs (field programmable gate arrays). Chips of this type use latches in the connection matrix between the logic arrays. First generations of these required special software supplied by the chip vendor for design and downloading of the circuit. Although these did allow one to potentially use heuristic programming for system design, they still lacked the practicality of the latest generation of FPGAs. The problem with this first generation was not only the propriety software but also the requirement of the vendor to supply hardware to interface with the computer that was used to design the circuit (cf. Oldfield and Dorf, 1995; Trimberger, 1994). Also there is

was the possibility of damage to the chip by downloading random circuit configurations.

With the latest generation of FPGA devices it is finally possible to do experiments using heuristic programming to design real-world hardware. Wolfram's suggestion has now become a reality. Simple experiments in evaluating the fitness of a logic circuit show that there is a major speed advantage in doing the system design *in silico* rather than in simulation.

Adaptive Hardware - *in Simula*

One could claim that neural network simulations are early simulations of adaptive hardware, and although this is correct, we consider simulations where genetic algorithms were used as the programming paradigm. One of the earliest suggestions of using genetic algorithms to evolve neural network connections and architecture was given by Ackley (1987). By 1989 the Third International Conference on Genetic Algorithms at George Mason University had six papers on using genetic algorithms to evolve the connections in neural networks.

Torrele (1991) demonstrated the idea of using genetic algorithms to evolve recurrent and dynamic neural networks. Ackley and Littman predate the above by presenting their work at the 1990 Artificial Life (AL) conference in Santa Fe, NM. They vastly extend these ideas and evolved AL forms that show the ability to compete in an ecology and the ability to learn. Each AL form consisted of several neural network modules for various action behaviors. The proceedings for that conference was published in 1992 (cf. Ackley and Littman, 1992).

One of the first logic circuit simulations of evolvable hardware was reported by Hemmi et al., (1994) at the Fourth International Workshop on Artificial Life. They evolved, using a Connection Machine, simple logic circuits for an artificial ant. Their work used a hardware description language (HDL) to describe the logic circuits.

Adaptive Hardware - *in Silico*

As we pointed out above, the idea of using adaptive or evolvable hardware, matured over decades as a solution to real-world engineering problems. As a result evolvable hardware researches can now do experiments with little cost. The key enabling part is the FPGA. These adaptive chips were developed to allow engineers to quickly reconfigure circuits and systems while in the field. It was a natural progression of adaptive hardware to use the FPGA with evolutionary programming. Likely most of the experimenters didn't even know about the paper by Wolfram (1986). By the time the FPGA chips were avaliable, numerous simulations, and some difficult hardware experiments, had already been done.

Reconfigurable computing is analogous to adaptive computing. The difference between reconfigurable and adaptive is that the reconfigurable systems downloaded to the FPGA have been designed by a technique other than heuristic programming or *in silico* evolution. These are the type of hardware systems that engineers developed to solve complex and special purpose computing applications (cf. Box, 1993; Villasenor and Mangione-Smith, 1997). General purpose systolic arrays for custom computing were implemented as early as 1993 in FPGA based systems (cf. Johnson, et al. 1993).

Caudell and Dolan (1989) used a genetic algorithm to evolve neural networks for the parity problem. Clearly they were the first to use a GA to evolve hardware systems. Their work was similar to Friedman et al. (1993). Both groups evolved an electro-optical pattern recognition system with a genetic algorithm manipulating the spatial light modulator between a light source and a CCD camera. They also show population convergence curves. Their solution is quite innovative given the state of the art in hardware in 1993.

Higuchi et al. (1993, 1994) used a first generation FPGA from Lattice Semiconductor (GAL16Z8) to demonstrate evolvable hardware using a genetic algorithm to download circuits in an FPGA. Thompson (1995) used RAM cells and logic gates to evolve the behaviors of a mobile robot. This work was likely a spin-off from Harvey et al. (1993) (see also Thompson et al. 1996).

The most significant advance to date was by Thompson (1996, 1997) using a second generation FPGA from Xilinx (XC6216). His system was the first implementation of address mapping the hardware to a CPU. He demonstrated the ability of the logic gates, essentially high gain amplifiers, to act as analog devices. He genetically evolved a tone discriminator to process analog signals using digital logic gates. Not only is his work significant in demonstrating and applying the analog processing behavior of logic gates, it is also the first demonstration of *in silico* evolution for a circuit with a practical application.

Chip Description

The key component in our system is the new XC6216 chip from Xilinx. This is a second generation chip with an open architecture that can be memory mapped to a CPU. This device can be thought of as a sea of gates with the

gates confined to cells that can be interconnected with neighbors. The gates within the cells can implement any Boolean function of 2 inputs. Also associated with each cell is a D-type register. Figure 1 shows the sea of gates and their interconnections. Not shown are the 4x4, 8x8, 16x16, and 64x64 connections. The entire array is 64x64 with the ability to connect parts together in superarrays. Every logic block is connected through reconfigurable switches so every "wire" is a one-way device. The cells on the edges of the device are wired to IO pads.

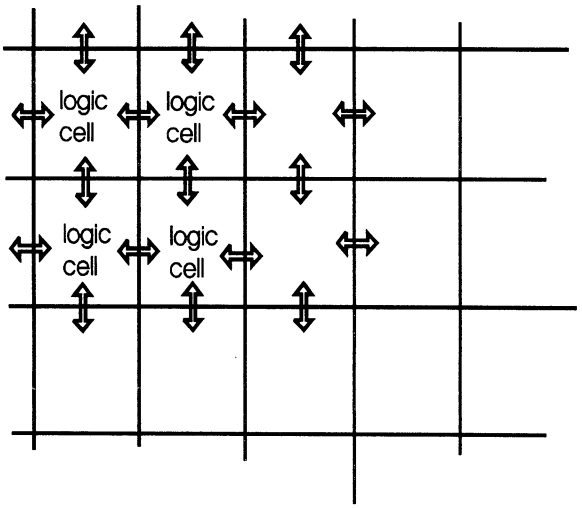


Figure 1. The sea of gates without the communication hierarchical communications.

Bit-Level Software Programming

As we stated above, the configurable switches in the chip can be memory mapped to a CPU for data transfers 8, 16, or 32 bits wide. In order to understand the structure of the data stream sent to the logic cells, consider the cell structure shown in Figure 2.

The cells are configured with 24 bits of data. The first 6 bits are the cell routing. That is the S, W, E, N muxes. The next 6 bits describe the X1, X2, X3 inputs and the CS signal shown in the figure. The highest byte is used for programming the control register for the entire device. Further details can be found in the Xilinx documentation on the XC6200 family of devices.

System Description

Since the part can be memory mapped to a CPU the interfacing is quite easy. Thompson (1996) used TTL glue logic to map the device to the address space of the CPU not

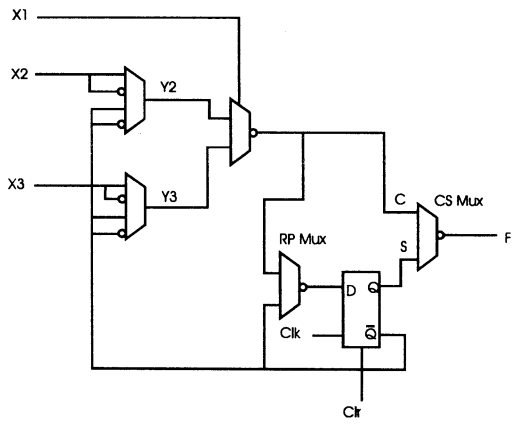


Figure 2. Logic cell contents.

the memory space. Our system is similar except we used an Intel 8255 programmable peripheral interface chip (cf. Goldsbrough, 1979). In our system the 8255 chips were mounted on a PC-AT card with three 8255 chips per card and three headers for external connections. Ribbon cables connect the 8255 boards to the 6216. The 6216 is mounted on a vector board with a combination of solder connections and wire-wrap connections. Also mounted on the board are LEDs for the address and data lines. These LEDs allowed us to debug the software for timing.

In order to evaluate the capabilities of the system we used a genetic algorithm. The task for the *in silico* evolution was to design an oscillator. The results indicate the evolved oscillator was a saw tooth function. On study of the evolved oscillator, we discovered that it exploited the impedance of the connecting cables from the 6216 to the 8255. An obvious "work-around" would be to install bus drivers (e.g. 4050) between the 6216 and the 8255. As Thompson (1996, 1997) discusses the GA will exploit flaws in the system if it is to the advantage of the fitness.

Conclusions

In a systemic approach, we can now evaluate the capabilities and limitations of real-world *in silico* evolution. Before huge systems can be built or designed using genetic algorithms it is necessary to show good engineering results. This would be the foundation on which to build larger systems. Good science and solid engineering both advance by conjectures and refutations (cf. Popper, 1963, 1989). To paraphrase Dennet, (1995, page 21) "There is no such thing as philosophy-free engineering; there is only engineering whose philosophical baggage is taken on board without examination."

Acknowledgments

EAR thanks Robert Frye of Bell Labs for identifying the impedance mismatch problem and Lorenz Huelsbergen, also of Bell Labs, for the early results on oscillators and for many stimulating discussions.

References

1. Ackley, D. H., A Connectionist Machine for Genetic Hillclimbing, Kluwer Academic Publishers, Boston, 1987

2. Ackley, D. H., and Littman, M., "Interactions Between Learning and Evolution", 487-509, Artificial Life II, C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, (editors), Addison-Wesley, Redwood City, CA, 1992

3. Bostock, G., Programmable Logic Devices Technology and Applications, McGraw-Hill Book Co., New York, 1988

4. Box, B., "Field Programmable Gate Array Based Reconfigurable Preprocessor", 40-48, D. A. Buell and K. L . Pocek, (editors) IEEE Workshop on FPGAs for Custom Computing Machines, IEEE Computer Soc. Press, Las Alamitos, CA, 1993

5. Caudell, T. P. And Dolan, C. P., "Parametric Connectivity: Training of Constrained Networks Using Genetic Algorithms", 370-374, Proceedings of the Third International Conference on Genetic Algorithms, George Mason University, J. D. Schaffer, (editor), Morgan Kaufmann Publishers, San Mateo, CA, 1989

6. Dennett, D. C., Darwin's Dangerous Idea, Evolution and the Meanings of Life, Simon and Schuster, New York, 1995

7. Friedman, M., Mahlab, U., and Shamir, J. "Collective Genetic Algorithm for Optimization and its Electro-Optic Implementation", Applied Optics, 32 (23), 4423-4429, 1993

8. Goldsbrough, P. F., Microcomputer Interfacing with the 8255 PPI chip, Howard W. Sams, Indianapolis, IN, 1979

9. Harvey, I., Husbands, P., and Cliff, D., "Genetic convergence in a Species of Evolved Robot Control Architectures", 636, S. Forrest, (editor), Proceedings of the 5th Int. Conf. on Genetic Algorithms, Morgan Kaufmann, San Matio, CA, 1993

10. Hemmi, H. Mizoguchi, J., and Shimohara, K., "Development and Evolution of Hardware Behaviors", 371-376, Artificial Life IV, R. A. Brooks and P. Maes (editors), MIT Press, Cambridge, MA, 1994

11. Higuchi, T., Niwa, T., Tanaka, T., Iba, H., de Garis, H., and Furuya, T., "Evolving Hardware with Genetic Leraning: A First Step Towards Building a Darwin Macine", 417-424, J.-A. Meyer, H. L. Roitblat, and S. W. Wilson, From Animals to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behavior, MIT Press, Cambridge, MA, 1993

12. Higuchi, T., Iba, H., and Manderick, B., "Evolvable Hardware", 399-421, H. Kitano and J. A. Hendler, Massively Parallel Artificial Intelligence", MIT Press, Cambridge, MA, 1994

13. Johnson, K. T., Hurson, A. R., and Shirazi, B., "General-Purpose Systolic Arrays", IEEE Computer, 20-31,

14. Oldfield, J. V. and Dorf, R. C., Field-Programmable Gate Arrays, John Wiley, New York, 1995

15. Popper, K. Conjectures and Refutations, The Growth of Scientific Knowledge, Routledge, London, 1963, 1989

16. Thompson, A. "Evolving Electronic Robot controllers that Exploit Hardware Resources", F. Moran, 640-656, (editors), Advances in Artificial Life: Proceedings from the 3rd Euro. conf. on Artificial Life, Springer Verlag, New York, 1995

17. Thompson, A. "Silicon Evolution", 444-452, J. Koza (editor) Proceedings of Genetic Programming, MIT Press, Cambridge, MA, 1996

18. Thompson, A., "An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics", 390-405, T. Higuchi, M. Iwata, and W. Liu, Evolvable Systems: From Biology to Hardware, Springer, New York, 1997

19. Thompson, A., Harvey, I., and Husbands. P., "Unconstrained Evolution and Hard Consequences", 136-165, E. Sanchez and M. Tomassini (editors), Towards Evolvable Hardware, The Evolutionary Engineering Approach", Springer-Verlag, New York, 1996

20. Torreele, J., "Temporal Processing With Recurrent Networks: An Evolutionary Approach", 555-561, Proceedings of the Foruth International Conference on Genetic Algorithms, George Mason University, R. K. Below and L. B. Booker, (editors), Morgan Kaufmann Publishers, San Mateo, CA, 1991

21. Trimberger, S. M., Field-Programmable Gate Array Technology, Kluwer Acacemic Publishers, Boston, MA, 1994

22. Villasenor, J. and Mangionesmith. W., "Configurable Computing", Scientific American, 66-71, June 1997

23. Wolfram, S., "Approaches to Complexity Engineering", Physica D, 22, 385-399, 1986

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

H^3: High-speed Hardware for Human-like genetic algorithm

Hideaki Asada, Hiroyuki Sugiura, Takamitsu Kawai, Hideki Ando, Toshio Shimada
Department of Information Electronics, Graduate School of Engineering
Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8603, Japan
{asada,sugiura,kawai,ando,shimada}@shimada.nuee.nagoya-u.ac.jp

Abstract

Hardware accelerators called GA engines are required to reduce execution time of genetic algorithms (GAs). Unlike typical engines for other applications, GA engines must be versatile because a great variety of fitness functions is used for different GAs. Field programmable gate arrays (FPGAs) are often used for this requirement because of its reconfigurability. Since only slow circuits can be implemented in a FPGA, the architecture of the GA engine must be sophisticated. Although previously reported engines accelerate the execution of GAs over software implementations, the speedup ratio is limited to only 7 to 18. This limitation arises from difficulty for pipelining the execution. We introduce two techniques to solve this problem. Using steady state GA and a pipelined binary search, the pipeline of our GA engine never stalls, and thus significantly improves performance. We call our GA engine the "H^3 engine" (the high-speed hardware for human-like genetic algorithm engine). We evaluated the H^3 engine from the view points of advantage and performance. We concluded our GA is better than a simple GA (SGA). Our evaluation shows that the H^3 engine achieves approximately 300 times speedup over the software implementation of our GA and approximately 500 times speedup over the software implementation of the SGA that are executed on a 333 MHz DEC AlphaStation. Thus we conclude that the H^3 engine outperforms previously reported GA engines by an order of magnitude.

1 introduction

Genetic algorithms (GAs) [1] based on the process of natural selection are useful for a wide variety of optimization problems, including numerical optimization and combinatorial optimization problems like circuit layout and job-shop scheduling. For solving those problems with software implementations, a large amount of computation is necessary, and we can not solve those problems on real time. So hardware accelerators called GA engines are required to reduce execution time of GAs.

Since GAs are applied to various problems by a change of fitness function, GA engines are required to change fitness function easily. Because field programmable gate arrays (FPGAs) are reconfigurable LSI, the GA engines implemented with FPGAs can satisfy those requirement. Since only slow circuits can be implemented in FPGAs, the architecture of the GA engines must be sophisticated.

Although previously reported engines [2][3] accelerate the execution of GAs over software implementations, the speedup ratio is limited to only 7 to 18. This limitation arises from difficulty of pipelining the execution. We introduce two techniques to solve this problem. Firstly, we introduce steady state GA [4][5] which replace only a few individuals in each generation of a fixed size population. Secondly, we use a binary search for choosing a target gene in roulette selection. Using the steady state GA and the binary search, the pipeline of our GA engine never stalls, and thus significantly improves performance. We call our GA H^3 (the high-speed hardware for human-like genetic algorithm) and the realization of the H^3 is named H^3 engine.

This paper is organized into six sections. Section 2 provides a brief overview of genetic algorithms. Section 3 extracts problems of previously reported GA engines. Section 4 describes our H^3 engine. Section 5 evaluates effects of using the steady state GA with SGA and the execution time between hardware and software. Section 6 provides some concluding remarks.

2 Genetic Algorithms

In 1975, Holland [6] developed an optimization technique, based on the process of natural selection and evolution, which is called the genetic algorithms. A simple genetic algorithm (SGA) is the basis of GA. During its operation the GA maintains a population of candidate solutions. A fitness value is associated with each individual. In the SGA, probability in which individuals are selected is proportional to their fitness

values. The population of the next generation is created by applying crossover and mutation operators. After applying those genetic operators repeatedly, we can get a semi-optimal solution.

Roulette selection is a representative stochastic selection model which is based on fitness-proportionate selection strategy. The principle of roulette selection is a linear search over a roulette wheel with slots weighted in proportion to the individual's fitness value. A target value is set, which is a random proportion of the sum of the fitness values in the population. The population is stepped through until the target value is reached. If best individuals aren't selected to reproduce or if they're destroyed by crossover or mutation, they can be lost. Therefore elitist preserving selection is an addition to many selection methods that forces the GA to retain some number of the best individuals at each generation.

In the single-point crossover, a crossover point is chosen at random and the selected individuals are splitted at the crossover point and the latter parts of each individuals are exchanged to form their offsprings. The offsprings inherit their parents' features, as a result, diversity of the population degrades.

As a means of preventing premature convergence to local minima, an operation known as mutation randomly perturbs a population.

3 Problems of previously reported GA engines

A variety of researchers has reported GA engines (e.g. SPGA [2], HGA [3]). Those GA engines are based on SGA. In the SPGA, one generation consists of selection, crossover, fitness value calculation and mutation. These processes can be executed in a pipeline. In the HGA, one generation consists of selection, crossover, mutation and fitness value calculation. The pipeline execution is possible as well as the SPGA. Although the speedup ratio of previously reported engines implemented on FPGAs is limited to only 7 to 18 in comparison with software implementations on general microprocessors with about ten times faster clock than FPGAs. This limitation arises from difficulty on pipelining the execution. There are two reasons for this pipeline stall. We explain these reasons using Figure 1 and Figure 2. Figure 1 shows the model of conventional GA engines. Figure 2 shows the way stalls occur in the pipeline.

Firstly, we can not start roulette selection until all individuals in a population are associated with fitness values. In the GA engines, a individual is evaluated in orders of selection, crossover, mutation and fitness in a pipelined execution, as shown in Figure 1 and Figure 2. The above processes repeat for population size N

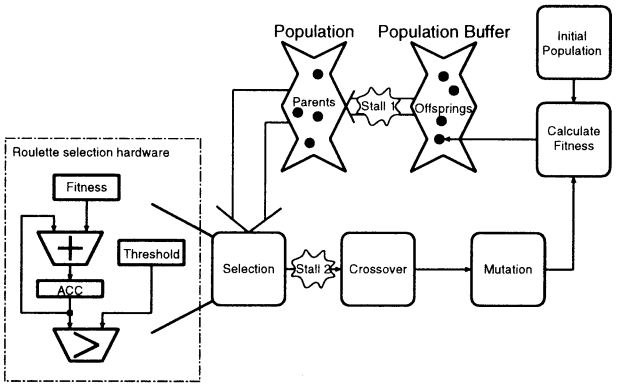


Figure 1: The model of conventional GA engines

times in one generation. But at the transition from current generation (i-th Gen.) to the next generation (($i+1$)-th Gen.), the pipelined execution stalls (stall 1), because data dependencies between a population in the i-th Gen. and in the $(i+1)$-th Gen. exist.

Secondly, the roulette selection hardware can be implemented directly as shown in the bottom left block in Figure 1. As a result, the processes of addition and comparison are repeated until the sum of the fitness value exceeds the threshold. Because this circuit has a loop structure [7], a pipelined execution stalls (stall 2).

In the previously reported GA engines, the pipelined execution stalls shown in Figure 2, and the throughput decreases greatly.

4 H^3

We introduce two techniques to solve the problem of previously reported GA engines. Firstly, we introduce steady state GA [4][5] which replaces only one individual in a fixed size population. Secondly, we use a binary search for choosing a target gene in roulette selection. We call this GA the high-speed hardware for human-like genetic algorithm (H^3). In H^3, a population changes in each time step for the steady state GA. A generation of natural selection changes at a regular period in SGA, on the other hand in H^3 a generation change at each time that an oldest individual is replaced by a new one. It is like a human.

In the following sections we introduce the steady state GA and the roulette selection using binary search and then propose an architecture of H^3 engine.

4.1 Steady state genetic algorithm

Steady state GA works as follows:

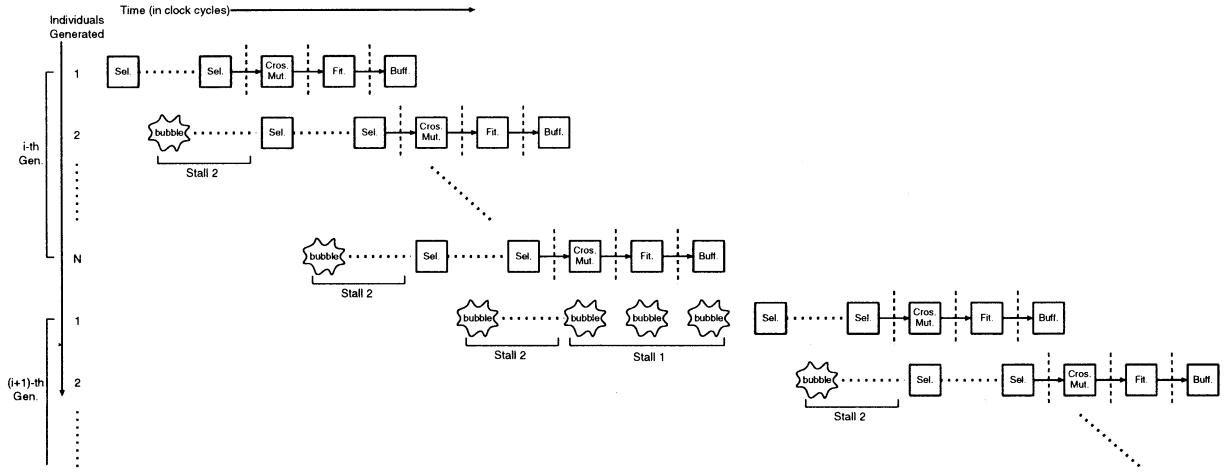


Figure 2: The way stalls occur in the pipeline

Let one iteration consists of step 2 to step 7. Each iteration generates an individual G_i.

Step 1: Initialize a population P with a randomly generated N individuals. Namely, $P = \{G_{-N+1}, G_{-N+2}, \cdots, G_{-1}, G_0\}$. And then calculate the fitness value of each individual in the population P. Set $i = 1$, ${G_A}' = G_0$.

Step 2: Select a individual G_A from the current population P, according to its fitness value.

Step 3: Execute crossover between G_A and ${G_A}'$ at a randomly chosen point by using crossover probability to form an offspring G_B. Set ${G_A}' = G_A$.

Step 4: Mutate the offspring G_B at each locus with mutation probability to form a new individual G_i.

Step 5: Calculate the fitness value of the individual G_i.

Step 6: The newest individual G_i adds to the population P and a oldest individual G_{i-N} removes from the population P. That is $P = P \cup \{G_i\} - \{G_{i-N}\}$.

Step 7: If the termination criteria is met, then a semi-optimal solution is a best individual and exit. If not, set $i = i + 1$ and go to step 2.

Figure 3 illustrates the model of steady state GA. The genetic operators (selection, crossover and mutation) and fitness value calculation are applied to only one individual at each time step. The steady state GA replaces one individual in a population in each iteration, therefore the steady state GA must calculate the sum of fitness values used by roulette selection in each iteration. The necessary procedures are to subtract the fitness value of the oldest individual G_{i-N} from the sum in iteration i, and to add the fitness value of new individual G_i to sum. Therefore these procedures finish in short time and do not cause significant overheads. With the steady state GA the pipeline doesn't stall in the transition from the current generation to the next generation.

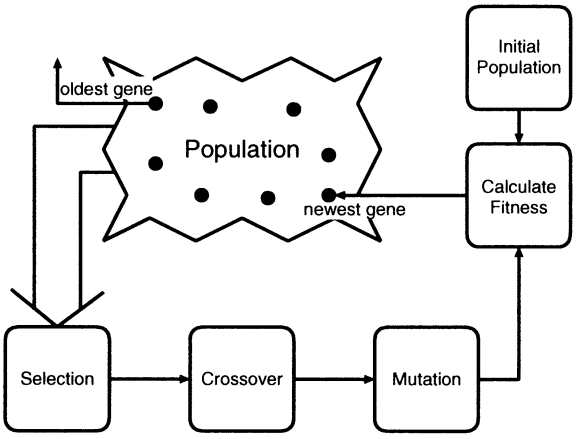


Figure 3: The model of steady state GA

4.2 Roulette selection with binary search

In direct hardware implementation of roulette selection, addition and comparison is repeated until accumulated fitness value exceeds the threshold. Therefore the pipeline stalls and results in throughput degradation. To realize the pipelined execution, we must duplicate the selection unit of Figure 1 and connect it in series with combinational logic circuits. However the size of combinational logic circuits is too large to implement them in FPGA. Therefore we use a binary

search to narrow the search space and select the target in the narrow search space with roulette selection.

The roulette selection with the binary search works as follows:

Let S_j be equal to a sum of fitness values of individuals from 1 to j. We define $S_0 = 0$. The number of divided parts of the population is 2^D. We use the binary search until the divided population size is equal to $N/2^D$. The population size N is power of 2. The maximum value of D is $\log_2 N$.

Step 1: Calculate the sum of the fitness values $f_i (i = 1, \cdots, N)$ of individuals in the population P. Calculate $S_j = \sum_{i=1}^{j} f_i$ $(j = N/2^D, 2N/2^D, 3N/2^D, \cdots, N)$. Continue to calculate S_j until the number of divided population is equal to 2^D.

Step 2: Generate a random number $rand \in [0, 1]$. Set $r = rand \times S_N$.

Step 3: Decide the index x of an individual at the head position in the searching range and calculate the sum y of fitness values of individuals up to x.

- **Step 3.1**: Set $j = N/2, x = 1, y = S_0 = 0$. Set $level = 1$ which is the depth of searching space.
- **Step 3.2**: If $r < S_j$, set $j = j - N/2^{level+1}$. If $r \geq S_j$, Set $x = x + N/2^{level}$, $y = S_j$, $j = j + N/2^{level+1}$.
- **Step 3.3**: Set $level = level+1$. If $level > D$, exit. If not, go to step 3.2.

If $D = \log_2 N$, select the x-th individual and exit. If not, go to step 4.

Step 4: Calculate minimum k such that $\sum_{i=x}^{k} f_i + y \geq r$ and select k-th individual and exit.

Step 1 and Step 2 make preparations for the roulette selection with the binary search. Step 3 uses a binary search to narrow a search space. Step 4 selects the target in the narrow search space with roulette selection. Figure 4 illustrates the Step 3. In Figure 4 the population size is divided into two times number of pieces in every level. Firstly the level 1 compares r with $S_{N/2}$. The comparison decides the range (hatched part) including r. At the same time it decides the index x of an individual at the head position of searching range and the sum y of fitness values up to individual x. Secondly the level 2 divides the first or second half parts into two parts recursively, and decides the position of r. At the same time x and y are set. The process repeats in a similar way until the division number is equal to D.

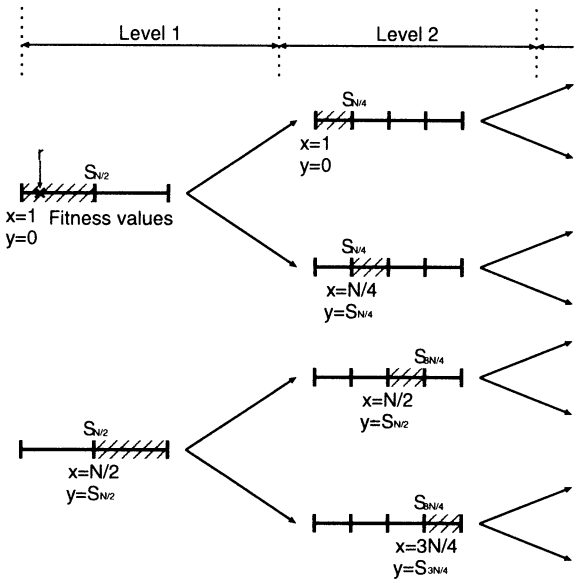


Figure 4: Steps of binary search

With the H^3 engine we set $D = \log_2 \frac{N}{4}$. That is, when the divided population size is equal to 4, the H^3 engine begins to search for the target by Step 4. By using a hybrid selection including the binary search and the combinational logic circuit, selection circuit can realize an efficient pipeline and the throughput does not decrease.

4.3 The architecture of H^3 engine

H^3 engine uses the steady state GA and the hybrid selection. For a genetic operator, the selection uses the hybrid selection and the elitist preserving selection together and the crossover uses the single-point crossover.

Figure 5 illustrates a block diagram of H^3 engine. "Fitness Unit (F Unit)" calculates the fitness values. "Selection Unit (S Unit)", "Binary Search Unit (B Unit)" and "Combinational Logic Circuit Unit (C Unit)" select the target individual with hybrid selection. "Crossover/Mutation Unit (CM Unit)" executes the crossover and the mutation.

A newly produced individual (output of CM Unit) and its fitness value (output of F Unit) is stored in "Register". "Register" consists of shift registers and the oldest individual G_{i-N} goes out and then the newest individual G_i comes in instead. The calculated fitness at F Unit is sent to "Register" and "S Unit" simultaneously.

"S Unit" calculates the sum of the fitness values S_N and S_j. Figure 6 illustrates the circuit of "S Unit". In the figure 6 the population size is supposed to 8. "S Unit" adds the fitness value of G_i to current S_N and

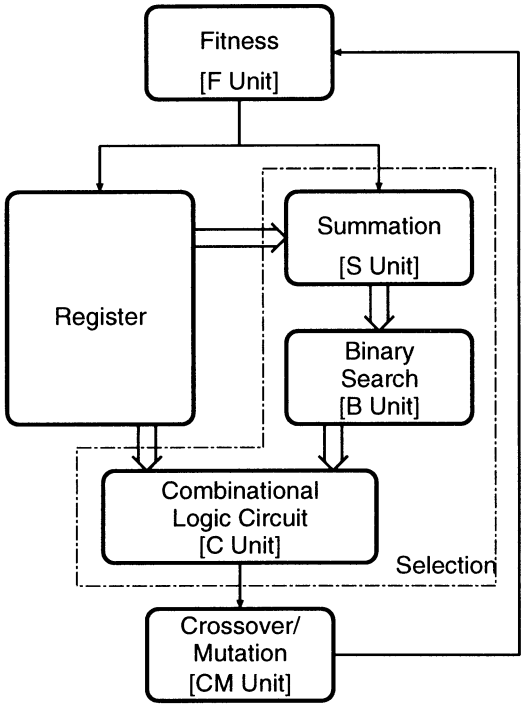


Figure 5: Block diagram of H^3 engine

then subtract the fitness of G_{i-N} from it. In a similar way the fitness value of G_i is added to current $S_{N/2}$ and then the fitness value of $G_{i-N/2}$ is subtracted from it.

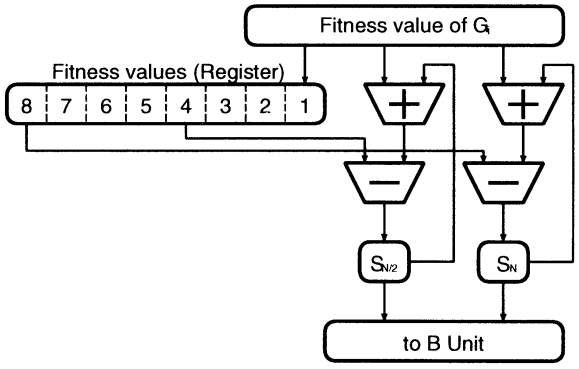


Figure 6: Circuit of S Unit

"B Unit" executes the binary search and "C Unit" searches a target individual with the combinational logic circuit. Figure 7 illustrates circuits of "B Unit" and "C Unit". The upper right block is "B Unit". "B Unit" calculates $r = rand \times S_N$ with random numbers generated by the random number generator (RNG). And then it compares r with S_j and sends control signals to "C Unit" according to the result of comparison. "C Unit" selects four individuals to search by a multiplexor (MUX) with control signals from "B Unit". After the MUX, it executes comparison $\sum_{i=x}^{x+3} f_i + y \geq r$ for each i and selects a minimum i satisfying the above conditions by Minimum Index Search (MIS). It sends the individual selected by MIS to "CM Unit".

The H^3 engine uses H^3, and inserts the pipelined registers into each unit to adjust the clock cycle time. Thus the pipeline never stalls and realizes higher performance.

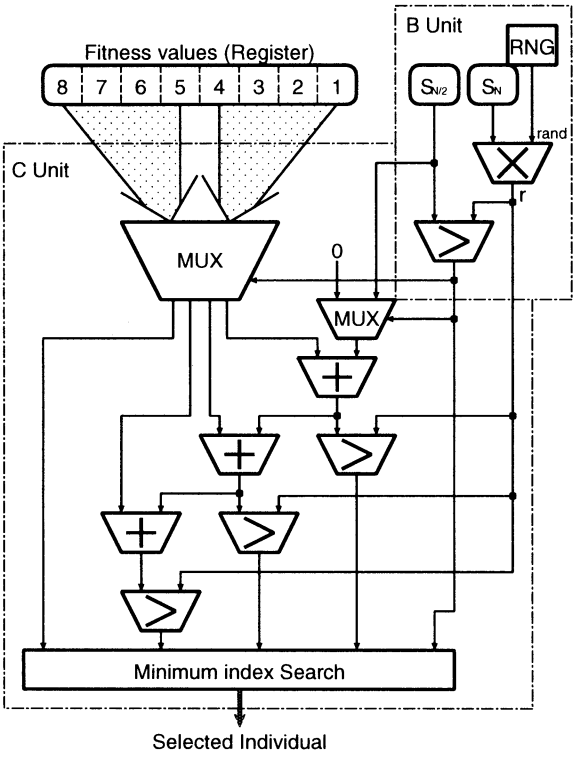


Figure 7: Circuits of B Unit and C Unit

5 Simulation result

We evaluate the effect of using the steady state GA and compare the execution time of a hardware implementation to a software implementation. Simulations run on a workstation. We evaluate a maximum function problem and use a gray code. The fitness function is $f(x) = |-(x-a)^2 + b|$. The gene length is 8, crossover probability is 40% and mutation probability is 10%. We define one execution as one process running until an optimal solution is found. We executed the simulations 100,000 times.

5.1 Effects of Steady state GA

We compared the effects of using the steady state GA with SGA. We also compare H^3 with SGA us-

Table 2: Comparison of execution time between hardware and software

population	execution time (sec)			comparison of execution time	
	H^3 engine	H^3(soft.)	SGA(soft.)	H^3(soft.)/H^3 engine	SGA(soft.)/H^3engine
8	0.3195	95.25	148.60	305	465
16	0.3923	119.69	210.81	298	537

ing same genetic operator. Because the steady state GA replaces one individual each generation, Average number of evaluated individuals of H^3 and SGA is compared until an optimal solution is found. Table 1 shows the results. The average number of individuals of H^3 is about 40% less than SGA. steady state GA.

Table 1: Comparison of the average number of evaluated individuals until an optimal solution is found

pop.	average number of individuals	
	H^3	SGA
8	98	160
16	125	224

5.2 Comparison of execution time

We compare execution time between hardware implementation and software implementation. We evaluate three kinds of implementation: software simulator of H^3 engine, software implementation of H^3 and software implementation of SGA. We assume H^3 engine executes on the 33 MHz FPGA board [8]. The performance of the software implementation executing on a 333 MHz DEC AlphaStation is compared to H^3 engine in Table 2. The results are that H^3 engine achieves approximately 300 times speedup over the software implementation of H^3, and approximately 500 times speedup over the software implementation of the SGA in spite of about 10 times slower in clock rates.

6 Conclusion

We have introduced two techniques to solve the problems of previously reported GA engines. Firstly, we introduce steady state GA which replace only a few individuals each generation from a fixed size population. Secondly, we use a binary search for choosing a target gene in roulette selection. As a result, the pipeline of our GA engine (H^3 engine) never stalls.

The average number of evaluated individuals of H^3 is about 40% less than SGA. steady state GA. We found H^3 engine achieves approximately 300 times speedup over the software implementation of H^3, and approximately 500 times speedup over the software implementation of the SGA in spite of about 10 times slower in clock rates. We could solve the problems of conventional GA engines and conclude that the H^3 engine outperforms conventional GA engines by an order of magnitude.

Acknowledgments

The work reported in this paper was partly supported by Nagoya Industrial Science Research Institute.

References

[1] D. E. Goldberg (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company.

[2] P. Graham and B. Nelson (1995) A Hardware Genetic Algorithm for the Traveling Salesman Problem on Splash 2, Field Programmable Logic and Applications, pp. 352-361.

[3] S. D. Scott, A. Samal and S. Seth (1995) HGA: A Hardware-Based Genetic Algorithm, Proceedings 1995 ACM/SIGDA Third International Symposium on FPGA, pp. 53-59.

[4] Whitly, D. and Kauth, J. (1988) Genitor: A Different Genetic Algorithm, Proceedings 4th Rocky Mountain Conference on Artificial Intelligence.

[5] K. A. De Jong and J. Sarma (1992) Generation Gaps Revisited, Foundations of Genetic Algorithms-2, Morgan Kaufmann Publishing, pp. 19-28.

[6] J. H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press.

[7] P. Graham and B. Nelson (1996) Genetic Algorithms In Software and In Hardware - A Performance Analysis Of Workstation and Custom Computing Machine Implementations, Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, pp. 216-225.

[8] Shuichi Ichikawa and Toshio Shimada (1996) Design and Implementation of Reconfigurable PCI card (in Japanese), Technical Report of IEICE VLD96-85, CPSY96-97, pp. 159-166.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

On Mixed-level Real-time Hardware Evolutionary System

Hitoshi Hemmi, Tomofumi Hikage and Katsunori Shimohara
Evolutionary systems group
NTT Human Interface Research Laboratories
1 - 1 Hikarinooka Yokosuka, 239 Japan
{hemmi, hikage, katsu }@nttcvg.hil.ntt.co.jp

Abstract

A new circuit model dedicated to hardware evolutionary system is proposed here. The proposed model is based on object oriented concept and has high modularity. Therefore it is considered to be congenial to genetic operations like crossover. Modularity of this model is also expected to be good for quick compilation of HDL (Hardware Description Language) program based on this model. This characteristic of the HDL in turn will add real-time property to HDL-based hardware evolutionary system. Generality and efficiency of this model are also discussed.

1 introduction

Using evolutionary technology for its control mechanism, the robot expect to acquire flexibility and robustness in its behavior. Due to the population based characteristic of this technology, software-only system is not suitable for real time operation, because it is very time consuming. So hardware accelaration is indispensable. Fortunately, thanks to recent electronics progression, special devices named FPGAs (Field Programmable Gate Arrays) come to be available for this purpose. Using these devices, electronic circuit systems that autonomous adapt to some environment can be constructed. Such systems are called "hardware evolutionary systems" or "evolvable hardware" ([1], [2], [3], [4], [5]). No doubt, evolvable hardware is very suitable for real time autonomous robot control.

One of the most important issue of the study of evolvable hardware is the strategy to change the circuit.

Higuchi et al. ([3]) used a PLA chip named GAL16. [3] regarded connection bit string of AND-array and OR-array as chromosome of GA (Genetic Algorithms) and evolved combinational circuits. Thompson ([4]) used the latest FPGA chip XC6216. [4] similarly with [3] regarded configuration bit string of the chip as GA chromosome. Generally FPGA chip can change configuration of circuit more flexible than PLA; produced circuit may include loops. Consequence of this fact, [4] evolved analogue behavior of the digital FPGA chip. These two investigation ([3], [4]) utilize elementally gate level configuration capability of the chips. So these are considered low level evolvable hardware.

On the contrary Liu et al.([5]) proposed function level evolutionary system. They are developing so called FPPA (Field Programmable Processor arrays) This direction is considered high level evolvable hardware.

We have developed hardware evolutionary system named AdAM (Adaptive Architecture Methodology)([1], [2]).

This system uses HDL (Hardware Description Languages) and evolves HDL-programs. Owing to the wide range description ability of this special purpose programming language, AdAM system can change and evolve circuits in several levels. For example, it can mutate one gate of a circuit, and exchange functional block of two circuits. So, AdAM system can be considered as mixed level hardware evolutionary system. This multi-level circuit changing capability of AdAM system may good for both fast evolution and gate count savings.

Unfortunately this system has one week point; HDL programs must be compiled, and usually this process is very time consuming.

Main reason of this time consumption is that the HDL model of circuit is not so much congenial to real circuit model; The HDL model is designed for human being to easy to understand.

So the calculation for conversion of HDL model into real circuit model takes long computing time.

In this paper, we try to concur this difficulty; Here we propose a new circuit model which both has affinity to real circuit and is easy to understand for human beings.

Next section introduces the former HDL model used

in AdAM system. Following sections discuss the new model and strategy to evolve them.

2 Former HDL model

2.1 Finite state machine based control flow

HDL used in AdAM system is called SFL (Structured Function Description Language). One of the features of this language is its provisioning of FSM (Finite State Machine) based control flow. Usually in ordinary computer languages, control flow is consists of loops (while, for, until etc.) and conditional jumps (if, then, else etc.). In digital circuit design, however, the designers almost always choose FSM based control flow. One of the reason for the popularity of FSM based control flow is that in hardware design, usually the resources (gate count, routing area, etc.) are limited compared with ones in software design. So the designer of circuit must choose most compact one among functionary equivalent circuit models. Usually, based on FSM the circuit will be most compact. Another reason for circuit designer to choose FSM based control flow is that it is easy to understand as well as loops and conditional jumps combination.

2.2 Implementation of FSM based control flow

FSM based control flow can be implemented with following basic idea. (Fig. 1) In figure 1, some flip flops are provided to store coded state of FSM. The number of flip flops is at least $\log 2N$, where N is number of states of the FSM. The outputs of flip flops are decoded using decoder and produce control signals for control lines. There is one combinational circuit to rise state change. This circuit takes the outputs of flip flops (or equivalently decoded control signals) and external flag signals as inputs, and produce next coded state as outputs.

Among the circuit in figure 1, the flip flops and decoder have regular structures depend on the number of state. However, all other part, including data path part and combinational circuit for state change, differ randomly in circuit to circuit. So in compilation of HDL-programs, execution of time-consuming algorithms such as placement, routing, etc. can not be avoided.

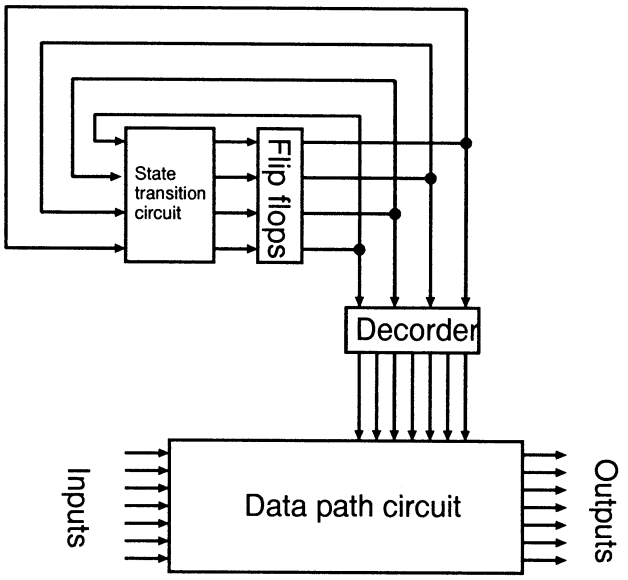


Figure 1: Implementation of FSM based control flow

3 Circuit model based on Johnson counter

In this section we propose a new circuit model based on Johnson counter.

3.1 Johnson counter

Johnson counters or twisted ring counters are constructed by arranging JK flip flops in a line and connect them in twisted ring manner. (Fig. 2) Figure 3 shows the truth table of JK flip flops. Usually FPGAs provide D flip flops as one of basic components and do not have JK flip flops. However JK flip flops can be constructed using D flip flops and some AND and or gates. This construction is also shown in figure 3.

The lover part of figure 2 shows the state transition diagram of the Johnson counter in upper part. As shown in figure 2, the basic operation of Johnson counters are changing FSM states circularly. It is easy to add the "stop motion" to each state of the state transition diagrams. Figure 4 shows an Johnson counter with stop motion. and state transition diagram of it.

Although we did not show the decode circuit of Johnson counters, they are very simple and have regular structures; Actually the decoder consists of only one 2-input AND gate per state.

3.2 Combination of Johnson counters

Although one Johnson counter can only implement circular state transition, it is possible to realize com-

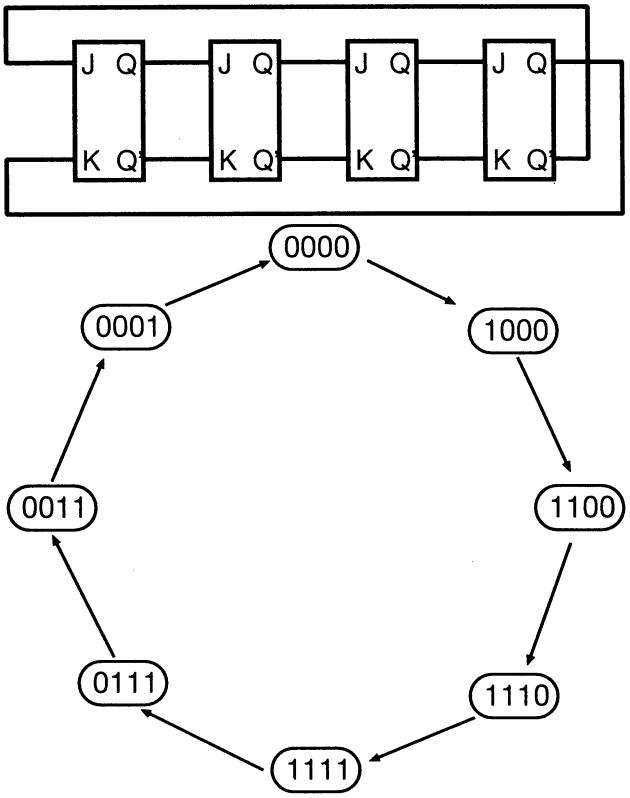


Figure 2: Johnson counter

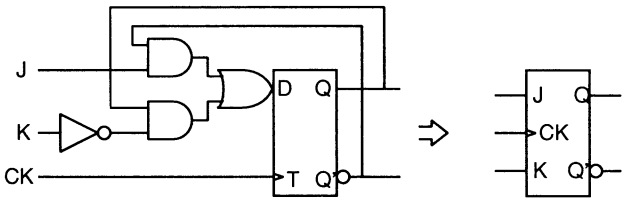


Figure 3: JK flip flop

plex state transition by combining plural Johnson counters. For example, in figure 5, the second state of the first Johnson counter "kicks"the second Johnson counter, and third state of second Johnson counter kicks back the first Johnson counter. Such kicking operation can be achieved by connecting appropriate decode line of first Johnson counter and stop motion control input of second Johnson counter.

By considering a "one hot state machine" (i.e. each state of the FSM is implemented by a 1-bit Johnson counter), it is easy to proof that this combination technique is general (i.e. any FSM can be implemented by Johnson counters using this technique).

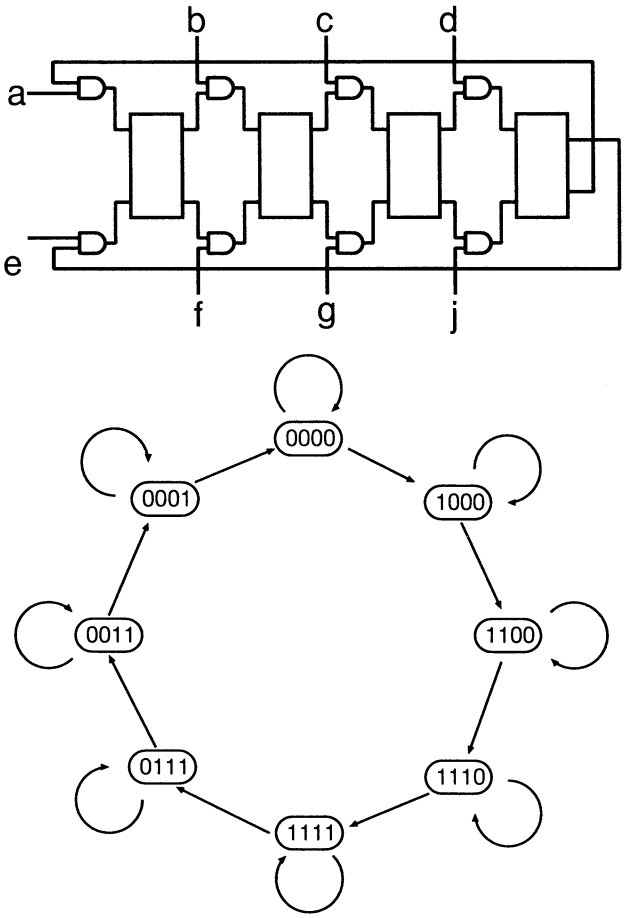


Figure 4: Johnson counter with "stop motion"

4 Hardware Evolutionary System using Johnson counters

To find the combination of Johnson counters that implement a certain FSM, we need to execute some algorithms, and it takes some computing time. For evolvable hardware purpose, this cannot be a problem, because we don't need to find an implementation of a certain FSM. We only make a population of circuit that consists of some FSMs and data path part, test them, and change them in evolutionary technique's manner.

So an important issue here is to determine the placement plan of circuit, so that changing circuit can be achieved fast enough.

Figure 6 shows an example floor plan of evolvable hardware. In figure X, at the top of the FPGA chip, there are several Johnson counters in a line. It is very easy and to change the composition of the Johnson counters; Variation of Johnson counters are only based

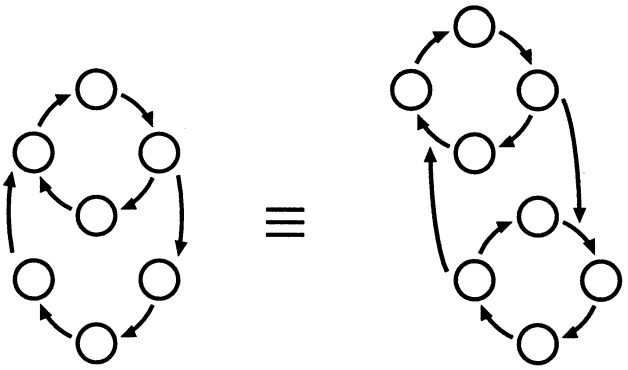


Figure 5: Combination of Johnson counters

on their bit length, and we only need to move borders among them to change the length. Under the Johnson counters area, the routing area exists. This area is for connection among Johnson counters, and between data path circuit and them. We don't need to worry about routing problem here; we simply make connections, test them, and change them. At the bottom of the FPGA chip, there exists data path area. It consists of a number of flip flops, gates, and input and output pins. Some of them (ex. number of inputs and outputs) are predetermined according to the problem applied. Others are subjects of evolution, but usually they have regular structures.

Using a floor plan such as figure X, we can make a hardware evolutionary system that can operate evolutionary process in real time, and flexibly in multiple circuit level.

5 Summary

This paper analyzed the former hardware evolutionary systems including the authors' AdAM system, and proposed a new circuit model using Johnson counters. This circuit model is both easy to understand its behavior for human being and congenial to real circuit model. A floor plan is also shown for real time mixed level hardware evolutionary system.

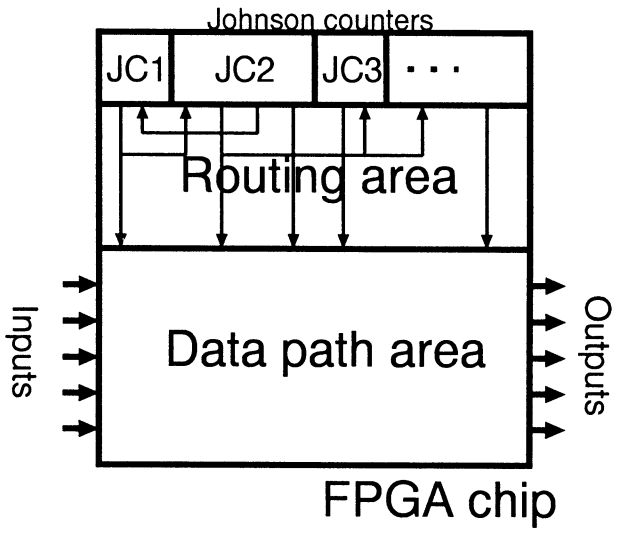


Figure 6: Floor plan for Johnson counter based evolvable hardware

References

[1] H. Hemmi, J. Mizoguchi, and K. Shimohara, "Development and evolution of hardware behaviors", in E. Sanchez and M. Tommassini, editors, *"Towards Evolvable Hardware: The Evolutionary Engineering Approach", Lecture Notes in Computer Science, Vol. 1062*, pages 250–265, Springer-verlag, 1996.

[2] H. Hikage, H. Hemmi, and K. Shimohara, "Hardware Evolution System Introducing Dominant and Recessive Heredity", in T. Higuchi, M. Iwata, and W. Liu, editors, *"Evolvable Systems: From Biology to Hardware", Lecture Notes in Computer Science, Vol. 1259*, pages 423–436, Springer-verlag, 1997.

[3] T. Higuchi, M. Iwata, I. Kajitani, H. Iba, T. Furuya, and B. Manderick, "Evolvable hardware and its applications to pattern recognition and fault-tolerant systems" in E. Sanchez and M. Tommassini, editors, *"Towards Evolvable Hardware: The Evolutionary Engineering Approach", Lecture Notes in Computer Science, Vol. 1062*, pages 118–135, Springer-verlag, 1996.

[4] A. Thompson, "An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics", in T. Higuchi, M. Iwata, and W. Liu, editors, *"Evolvable Systems: From Biology to Hardware", Lecture Notes in Computer Science, Vol. 1259*, pages 390–405, Springer-verlag, 1997.

[5] W. Liu, M. Murakawa, and T. Higuchi, "ATM Cell Scheduling by Function Level Evolvable Hardware", in T. Higuchi, M. Iwata, and W. Liu, editors, *"Evolvable Systems: From Biology to Hardware", Lecture Notes in Computer Science, Vol. 1259*, pages 180–192, Springer-verlag, 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB Ⅲ'98),
Oita, Japan, 19-21, January, 1998,

A New Structure of EHW

Hewei Lishan Kang Baozhong Wang

State Key Lab of Software Engineering
Wuhan University
Hubei, 430072
P.R. China
E-mail: hewei@rjgc.whu.edu.cn

Abstract

EHW realizes circuit function through evolution. It may be viewed as the combination of EAs (evolutionary algorithms) and PLDs (programmable logic devices). At present the main problem is that the speed of evolution is too slow. In order to tackle the problem, we must do work from two aspects: algorithm and device structure. This paper presents a function PLD as a new EHW's structure. So the efficiency of coding is improved, and evolution is speeded up. 4-bit comparator is successfully completed.

Keyword EHW EAs PLDs function PLD

1 introduction

EHW[1] is a new method of designing hardware, which is presented in early 1990s. On the base of PLDs whose structure can be reconfigured, EHW realizes circuit function through evolution. The interesting of this new research area is increased rapidly in recent years.

The reasons why EHW is notified extensively are as follows: 1. EAs has succeeded in varieties of intelligent computation, such as machine learning[2], artificial neural network[3], adaptive control[4], complex combinational optimum question and etc.; 2. With the scale and complex of hardware increasing more and more, it's too hard to implement hardware design with traditional method. So a new design method is urgently needed; 3. The hardware's self-organization and self-adaption is a goal that designers want to reach. It's so difficult to reach with the traditional method, but EHW may be a solution in the area.

The method that EHW adopts is very different with traditional method. It is inspired by biology. To implement hardware design, the method simulates the evolutionary process of creature. In the 1960s, Von Neumann proposed the idea of cell automa, which realizes a machine with biological characteristics such as self-reproduction and self-growth. But limited to the technology in that time, it was mainly discussed on theory instead of practice. With the development of technology and the special ability that EAs show in many areas, the realization of EHW becomes feasible.

This paper introduces EHW's principal idea firstly, and then discusses a new structure, function PLD. Finally experimental results show that the new structure is successful.

2 EHW's idea

PLD's structure is decided by architect bits: different structures correspond to different architect bits. According to the desired function ,we construct architect bits to download a PLD, so the PLD possesses the desired function. When the environment in which hardware is has changed, if hardware's structure is needed to be adjusted, we may

accomplish it through changing architect bits to reconfigure the PLD.

With the characteristic of PLD's reconfiguration, EHW may directly take architect bits as chromosomes in EAs and correctness of evolved circuits as fitness value to implement hardware's evolution (see figure 1).

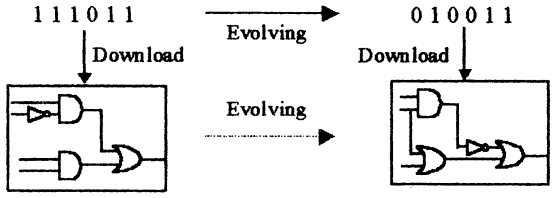


Figure 1 relation between circuits and architect bits

EHW can be viewed as: EHW=EAs+PLDs. At present the main problem existed in EHW is that evolutionary speed is too slow. To solve this problem, we can start from the two aspects: the EAs fitting into hardware and the hardware structure fitting into evolution. Kajitani et. al. proposed VGA[6] from the EAs' aspect. It improves the coding efficiency so that the average length of chromosomes reduces. Therefore the evolutionary speed and scale of circuits is increased. From the aspect of hardware structure, this paper suggests a new structure named as function PLD as EHW's structure.

3 Function PLD

Comparing with traditional PLDs such as GAL and PAL, function PLD has a special characteristic: it owns a math model.

$$T = f(\vec{X}, \vec{C}) \tag{1}$$

$$\vec{X} = (x_1, x_2, \cdots, x_n) \quad \vec{C} = (c_1, c_2, \cdots, c_m) \quad x_j, c_j \in \{0,1\}$$

where $\vec{X}$ is input vector; $\vec{C}$ is control vector, which may be regarded as architect bits. When $\vec{C}$ equals to different values, T has different expressions to decide what logic function PLD realizes.

GAL device realize logic through programmable fuse array (see figure 2). That all switches of fuse array are coded in a chromosome is a simple and direct method apparently, but which results in redundancy of chromosomes, that is, some genes are useless. For example the two switches, which are within the 1st circle in figure 2, respectively decide whether I_0 and its inverse phase $\bar{I}_0$ connect to the first horizon line or not. Because the two switches contradict each other evidently, both of them simply coded in a chromosome is to cause inefficient coding and expand search space insignificantly. VGA[6] tackles the problem with coding the switches that really decide hardware structure instead of all switches in fuse array.

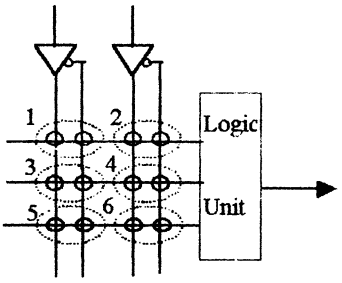


Figure 2 a simplified diagram of GAL's structure

Function PLD avoids inefficiency of chromosome's coding by way of its special structure. The space determined by control vector $\vec{C}$ is the smallest search space, which corresponds to the whole logics that device can realize. So $\vec{C}$ directly taken as chromosome in EAs doesn't expand search space insignificantly.

We introduce a direct programmable tree-structure function PLD in detail, which is one kind of function PLD. Its math model is:

$$\begin{aligned} T_n &= f(\vec{X}, \vec{C}) \\ &= \{\cdots[(c_m x_1 + c_{m-1}\bar{x}_1)x_2 + (c_{m-2}x_1 + c_{m-3}\bar{x}_1)\bar{x}_2]x_3 \cdots\}x_n \\ &\quad + \{\cdots[(c_4 x_1 + c_3\bar{x}_1)x_2 + (c_2 x_1 + c_1\bar{x}_1)\bar{x}_2]\bar{x}_3 \cdots\}\bar{x}_n \end{aligned} \tag{2}$$

where T_n, c_m, $x_n \in \{0, 1\}$, n takes a positive integer, $m = 2^n$, and $n \geq 1$. For example, while n=2, then

$$T_2 = (c_4 x_1 + c_3\bar{x}_1)x_2 + (c_2 x_1 + c_1\bar{x}_1)\bar{x}_2 \tag{3}$$

where $\vec{C} = (c_1, c_2, c_3, c_4)$, $\vec{X} = (x_1, x_2)$. Controlling what value $\vec{C}$ in T_2 takes, we can get sixteen kinds of logic (see table 1).

Table 1 relation between control vectors and logics

Control Vector				Logic
c_1	c_2	c_3	c_4	T_2
0	0	0	0	0
1	0	0	0	$\bar{x}_1\bar{x}_2$
0	1	0	0	$x_1\bar{x}_2$
1	1	0	0	$\bar{x}_2$
0	0	1	0	$\bar{x}_1x_2$
1	0	1	0	$\bar{x}_1$
0	1	1	0	$\bar{x}_1x_2 + x_1\bar{x}_2$
1	1	1	0	$\overline{x_1x_2}$
0	0	0	1	x_1x_2
1	0	0	1	$x_1x_2 + \bar{x}_1\bar{x}_2$
0	1	0	1	x_1
1	1	0	1	$x_1 + \bar{x}_2$
0	0	1	1	x_2
1	0	1	1	$\bar{x}_1 + x_2$
0	1	1	1	$x_1 + x_2$
1	1	1	1	1

Seen from the table 1, every point in the space of $\bar{C}$ corresponds to a kind of T_2's logics. In fact, each bit in a vector $\bar{C}$ decides a minimum term of Boolean product, therefore the whole $\bar{C}$ space represents all logics that T can realize, and each logic has a correspondent point in $\bar{C}$ space. From this point to say, $\bar{C}$ space has no redundancy and is minimum search space.

4 Experiment result

Taking GAL as hardware structure, Kajitani et. al. has evolved out a 4-1 multiplexor with both direct coding and VGA, but the evolutionary speed is faster while adopting VGA. They are failure in evolving 4-bit comparator with GAL. In order to test the efficiency of direct programmable tree-structure function PLD, we also select the two types of experiments: (1) 4-1 multiplexor; (2) 4-bit comparator.

The EA adopts SGA (simple GA), and control vector $\bar{C}$ is directly taken as bit-string chromosome. Crossover operator selects one-point crossover, and mutation operator is bitwise mutator. The fitness is on the base of correctness of evolved circuit. Given test vectors, we verify each circuit configured with the correspondent chromosome, and evolved circuit's number of correctness is regarded as its fitness. For instance, 4-bit comparator has eight binary input variables, so its test vectors equal to 256, and its maximum fitness is 256.

(1) 4-1 multiplexor

4-1 multiplexor has six binary input variables, four of which are data inputs, and the other two decide which data input is outputted. So 4-1 multiplexor realized with tree-structure function PLD needs a control vector with 64-bit, and the whole search space is 2^{64}. The number of test vectors equals to 64, and maximum fitness is 64.

(2) 4-bit comparator

4-bit comparator can realize the function that which of two four-bit binary numbers A and B is bigger. In practice, there are three outputs in 4-bit comparator. We remain one output in this experiment: while $A > B$, then output is 1; while $A \leqslant B$, then output is 0.

There are eight binary input variables in 4-bit comparator. Realized with tree-structure function, it needs a 256-bit control vector, and the whole search space is 2^{256}. The number of test vectors equals to 256, and maximum fitness is 256.

We run ten times over experiment (1) and (2) respectively. The experiments are implemented on PC (P133). Figure 2 and 3 shows the diagram of the relationship between fitness and generations. The results are recorded in table 2 and 3. The results show that experiment (1) and (2) all get ultimate control vector $\bar{C}$. Therefore adopting direct programmable tree-structure function as EHW is feasible.

Table 2 4-1 multiplexor's results

number of test	1	2	3	4	5	6	7	8	9	10	average
convergence generation	377	175	332	232	269	226	213	210	229	229	250
control vector $\vec{C}$	0000100001001100001010100110111000011001010111010011101101111111										

Table 3 4-bit comparator's results

number of tests	1	2	3	4	5	6	7	8	9	10	average
convergence generation	2167	1738	1274	1312	1278	1626	1574	1529	1387	1750	1564
control vector $\vec{C}$	00000000000000001000000000000001100000000000000011100000000000000111100 00000000011111000000000011111100000000001111111000000000111111110000 00001111111110000000111111111100000011111111111000000111111111111000011 11111111110001111111111111110011111111111111110										

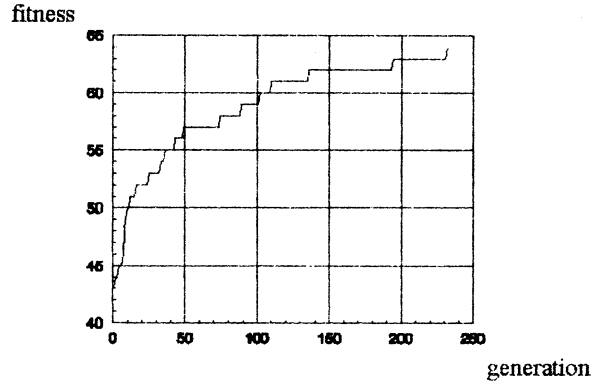


Figure 2 4-1 multiplexor's diagram (fitness vs.generation)

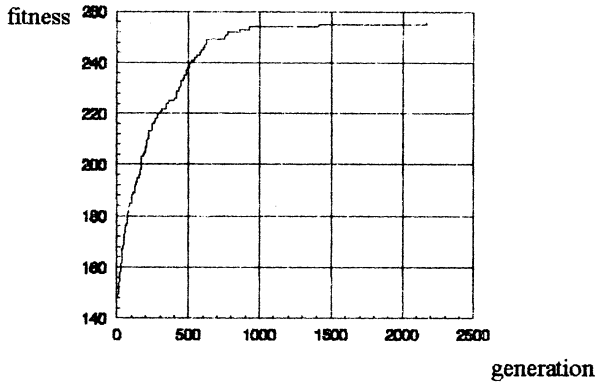


Figure3 4-1 comparator's diagram (fitness vs. generation)

5 Conclusion

EHW is a new research area, which has great applied value. At present there is a distance to practice because the scale of EHW is smaller. This paper presents a new structure, that is function PLD, from the aspect of hardware structure. Compared with GAL, it has gotten better results that show it's worth doing study deeply.

Acknowledgments

This paper is supported by National Natural Science Foundation.

References

[1] Hugo de Garis, "Evolvable hardware workshop report," Technical report, ATR, Japan, October 1995

[2] Goldberg, D. E., & Holland, J. H. (1988), "Genetic algorithms and machine learning," Machine Learning , 3 , 95--99.

[3] Yao, X. (1992b), "A review of evolutionary artificial neural networks," International Journal of Intelligent Systems.

[4] Goldberg, D.E. (1989), "Genetic algorithms in search, optimization and machine learning," Reading, MA: Addison-Wesley.

[5] Boseniuk, T., & Ebeling, W. (1988a), "Evolution strategies in complex optimization: The traveling salesman problem," Systems Analysis Modeling Simulation, 5 (5), 413--422.

[6] I.Kajitani, T. Hoshino, M. Iwata, and T. Higuchi, "Variable length chromosome GA for evovable hardware," In Proc. of the 1996 IEEE Int. Conf. on Evolutionary Computation, pages 443-447. IEEE Press, Piscatawa, NJ, USA, May 1996.

[7] 钱博森. 直接可编程树型结构函数. 计算机学报, 1996, 5: 334 — 341.

[8] 刘勇, 康立山, 陈毓屏. 非数值并行算法（第二册）遗传算法. 科学出版社

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Adaptive gait generation as an analogy to magnetising process

S. Mikami
Faculty of Engineering
Hokkaido University
Kita-13, Nsi -8, Sapporo, Japan
+81-11-706-7225, E-mail: mikami@conplex.hokudai.ac.jp

The manuscript of this talk will be distributed at the symposium.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Realization Model for Brain Computing

Michinori Ichikawa, Hitoshi Yamada and Gen Matsumoto
Brainway Research Group
Brain Science Institute of RIKEN
Hirosawa 2-1, Wako, Saitama, 351-0198, Japan

Abstract

The present paper reports on a newly developed micro-processor for brain-way computing. The new device is designed to improve performance of calculating of a large number of connections and of their self-organizing abitlity. Its instruction set is quite efficient to describe complex neurons and large scale neural networks. A minimum processor system was constructed with a 35k gate custom LSI (gate array) and four 1M-bit SRAM's, which can simulate over thousand neurons and one million connections with real-time execution of calculation and learning.

1 Introduction

Generally, any neuron models so far reported as artificial neurons for neural networks has been too simple in structure, however each living neuron forms dendrites of very complex structure. Recently, physiological experiments have clarified that spikes are generated not only at the cell body but also in the proximal dendrites. These results suggest the conventional neuron models used in artificial neuro-networks are not mimic for living neural networks. We propose a novel the ones of modeling an artificial neuron with multiple dendrites, which can model of various properties of a living neuron. This model neuron is, further, demonstrated as a processor based on current semiconductor fabrication techniques.

2 Modeling

A model neuron consists of four elementary components called Cell, Body, Dendrite and Node. We can synthesize any structured cell by combining some or all of them, as typicaly shown in Figure 1. A component can receive 1 to 4096 inputs and put one output which is resulted after calculation of connections (weights) and accumulation of activities. Cell and Body transmit impulse to any other component away from Cell or Body but Dendrite and Node output continuous values to neighboring components. Impulses are transmitted from a neuron to other neurons. For communications inside a neuron, continuous values are used. Therefore, the most simple neuron consists of a Cell. On the other hand a large neuron normally consists of several number of Dendrites, Nodes and a Body. Dendrites are always the parts that get the input from other neurons, and the Body transmits the output to other neurons. Because of these building blocks, it is possible to make the neuronal cell to some complex degree as the designer desires. One of the important characteristic features of the new processor is that each of the different learning rules can be applied to each of the components independently. By this particular feature, a huge cell can involve many different learning rules at a portion to portion inside the cell simultaneously.

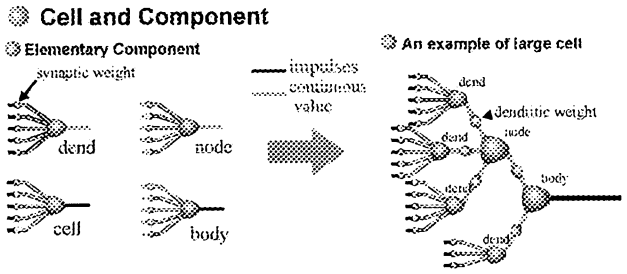


Fig. 1 Cell Modeling

3 A New Micro-Processor

The model described above necessitates both repetitive and fixed calculations in much more numbers than the care for normal computation, ex. simulation for physical study. This means that if some repetitive

operations could be execute by special hardware, execution time would be much shorter than the time used by conventional computer. A new micro-processor is accelerator for neural network simulations of our cell modeling concept.

The chip include multiplier-accumulator and learning engine for calculation. Essential foreground calculation is $Yi=F\ (\ \Sigma\ (Xj\ *\ Wij)\ )$, for the simplest neuron model. Background operation is learning, which is simplified as, $\Delta Wij=G\ (\ Yi*Xj\ )$. Many similar but modified equations can be calculated in a single clock cycle per each connection. Configuration registers contain such parameters , ex time-constant, threshold, learning mode , for mimicing cells and connections. One setup can realize SAM neuron (Shigematsu et.al. 1995). According to this idea, the synaptic weight change takes place for time-resolved learning rule in an output-dependent manner. Another setup can realize , for example simple Hebbian neuron. Other circuitry blocks perform sequence control and network management.

A chip contains about 35K gate logic , executes one connection within 50nsec and realizes to calculate over 1,000 cells and 1,000,000 connections in a few msec. Its potential have ability for parallel connection calculation expand both of cell number and speed. A minimum system can operate, only using 5 main components under controlling by a host-PC.

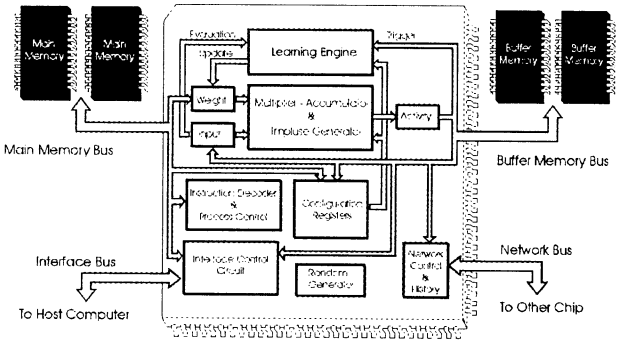


Fig. 2 Block diagram of a new micro-processor.

4 Brain-way Computing

It is our concept of brain-way computing that the system can find solution for any problem by auto(self)-acquiring fashion and tune itself for some given environment by its own learning ability. We assume six main components are required in artificial brain. 1: *Sensory filter* converts the real sensory data to internal format , 2: *Event memory* records events and association, 3: *Behavior planner* generates next motion, 4: *Motor controller* commands actuator for real movements, 5: *Emotion driver* initiates learning control, 6: *Evaluator* controls all learning signals. Learning is managed by two different way. One is emotional strong input, other is evaluation (success or error of output) caused by past events. Emotion makes some constraint of behavior as constraints of some limit and forcing. Evaluation leads reasonable cognition results and better combination control of motor system. They are not independent but control each other tightly. We should know them in more details for total realization of brain function by experiments and modeling.

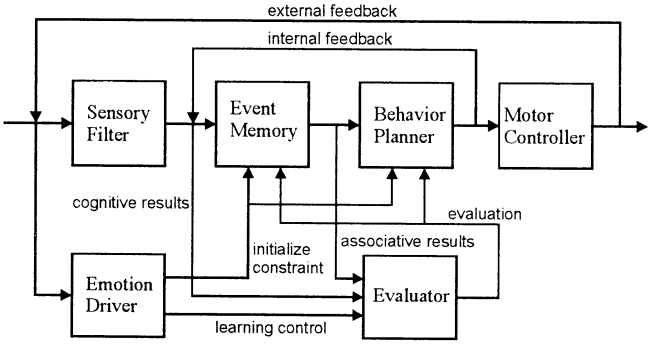


Fig. 3 Schematic diagram of Brain-way Computer

References

Modeling of artificial neurons with Complex dendrite structures. M. Ichikawa, H. Yamada, T. Iijima and G. Matsumoto, Abstract of Neuroscience meeting 1996.

A new learning rule for temporal sequence. Y. Shigematsu and G. Matsumoto, Abstract of ICONIP '94 .

Acknowledgements

Authers would express their cordial thanks to Drs. Toshio Iijima and Yoshifumi Shigematsu (ETL, Tsukuba Japan) for their kind collaborations at the present stage of this study for physiological experiments and modeling respectively.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Large-scale Neural Network Method for Brain Computing

N.Miyakawa
E C Tech. Dev. Dept.
AT&EC Fuji Xerox,
2274 Hongo, Ebina-shi,
Kanagawa 243-04,Japan

M.Ichikawa*
Lab. Head Lab. for Brain-operative Device,
*Brainway Group, Brain Science Institute(BSI),
The Institute of Physical and Chemical Research(RIKEN))
1-2 Hirosawa, Wakou-shi, Saitama, 351-01,Japan

G.Matsumoto*
Group Director,

Abstract

The development of the large-scale neural network method is very important for the realization of high performance and small size Brain Computer. The neural network has efficient data processing architecture based on parallel processing, distribution of memory and learning function. In this paper, a large-scale neural network method which consists of chip elements using mixed analog and digital technology, a 3 dimensional structure connecting chip elements by electronic wiring technology, and an optical connection between each 3 dimensional structure for Brain Computer as a special purpose computer is proposed.

1 Introduction

The neural network circuits have been studied since 1980's [1] and recently the Brain Computer with learning function and with more than 1000 neurons is prototyped and verified.[2] However, the Brain Computer needs a lot of improvements in terms of high density wiring and power consumption etc. The Brain Computer is one of the special purpose computers. The special purpose computer is very useful to easily prepare the countermeasure for the restrictive factors of the system, to adaptively share the hardware portion and software portion in data processing, and to select the optimum circuits technology for high performance hardware. However, the computer needs a software library and a special program. Brain Computer has two important factors for high performance and compact size. The first factor is the circuit design technology. Fig.1 shows the trend of Si semiconductor technology.[3] We will be using the technology of more than 30(MTrs./chip), and more than 10^3(Mbits/chip), if we want to use digital circuits 2 to 3 years later. The large-scale neural network using only digital circuits is larger than that of only analog circuits and larger than that of mixed analog and digital circuits. It is critical to select the optimum circuits technology to achieve a small size Brain Computer. The second factor is high speed data processing technology with low power consumption.

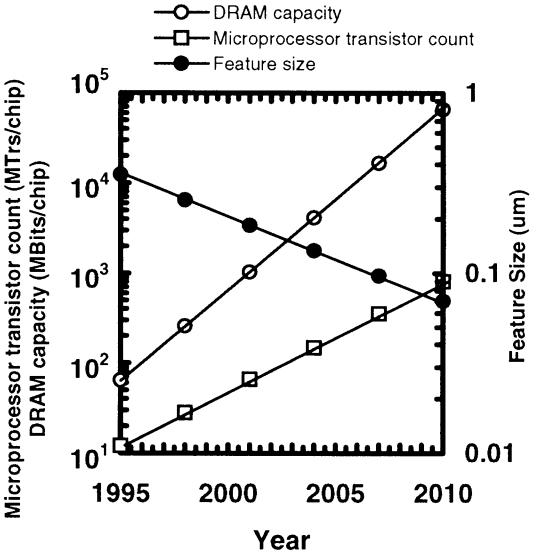


Fig.1 Trend of Integration Capacity and Feature Size in LSIs (SIA Road Map)

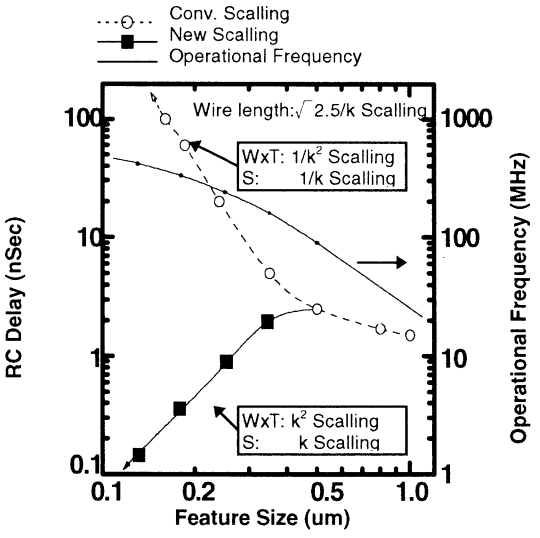


Fig.2 Performance according to scaling rule and dominant factor

Fig.2 shows the trend of the operational frequency and the propagation delay depending on the wire.[4] The performance of the transistor has been improved, but the resistance of the wire

connecting transistors is higher and it is very difficult to achieve high speed operation. It is critical to reduce the number of wires, because large scale neural network uses a large number of wires.

2 The optimum circuits technology

The digital circuits are suitable for stable and high precision parallel processing and learning function in comparison with the analog circuits, because the digital circuits have a wide noise margin,are able to have a wide data bandwidth and are able to include large scale memory. As fabricating Si LSI mixed DRAM and logic circuits is made possible recently, we are able to select an adaptive device technology in some of process variations. However, the sum of products circuit made by the digital technology is larger in order to have a lot of transistors and connection wires. On the other hand the sum of products circuit made by the analog technology needs the countermeasure of noise, but is smaller than that of the digital technology and needs fewer connection wires. The calculation results of these circuits must keep the monotonicity, but does not need high precision. Therfore, we had used the mixed analog and digital technology with A-D converter and D-A converter for the large-scale neural network circuits. Fig.3 shows one example of the layout in chip.

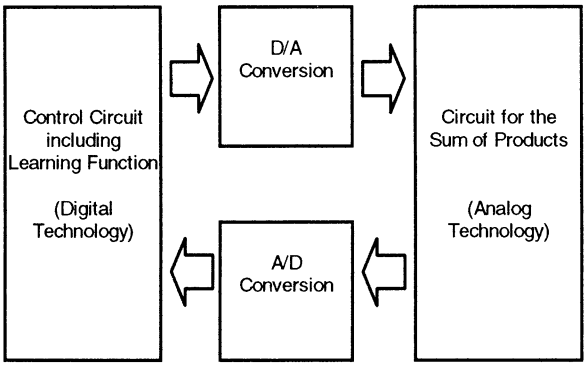


Fig.3 System using Mixed Digital and Analog technology

3 The system assembly technology

It is effective to develop the low power architecture and circuits to make use of the lower supply voltage to solve the power consumption problem. In order to reduce the number of wires, it is effective to use 3 dimensional LSI technology[5] because the electronic wiring technology is better than the optical connection technology for high speed operation when the wire connection is less than 6 mm long [6] and when implementing shared memory technology because the memory for the synaptic efficacy is used in commom. 2 dimensional neuro LSI prototyped by wafer scale integration technology has already been reported [7], but this technology is large-scale, 3 dimensional and high speed operation. Fig.4 shows the 3 dimensional structure using electrical wiring technology. Each layer becomes a chip which consists of the basic controller with learning function, the shared memory and the sum of products circuits. The digital circuits portion between layers are connected by electrical wires, but the analog circuits portion between layers are not connected for the sake of noise reduction. In the case of a long wire length, an optical connection is effective for high speed operation. Neural network using an optical connection for artificial retina has already been investigated [8], but a lot of new optical technologies still need to be realized.

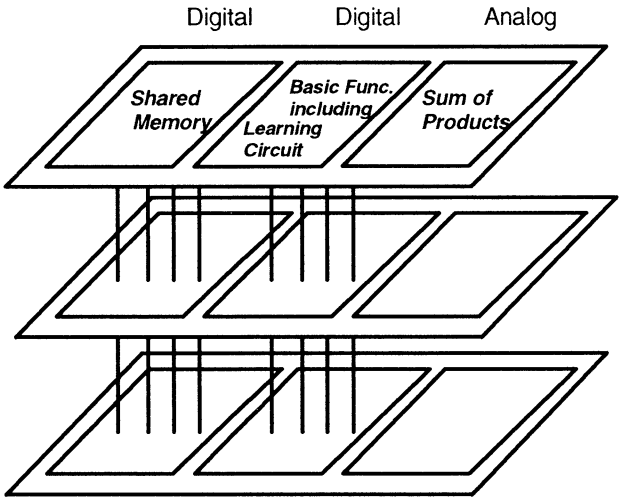


Fig.4 3 Dimention Structure Using Electrical Wiring Technology

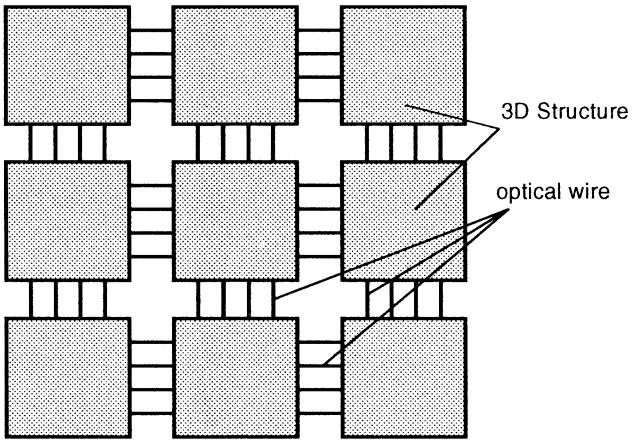


Fig.5 Mixed Electronic Circuit and Optical Wiring Technology

4 Summary

High performance and compact large-scale neural network is realized by using mixed analog and digital circuits for chip elements, and both electronic wiring technology for short length wiring and an optical connection technology for long length wiring in the system assembly. The proposed method is not limited by pins because this method is unnecessary the package assembly technonology is not needed in this method (Fig.4,Fig.5).

Acknowledgements

The authors thank members of the Brainway group of Brain Science Institute of RIKEN for the useful technical discussion about Brain Computer. The authors also would like to thank Mr. A.Miyahara, the president of Fuji Xerox.

References

1)NIKKEI Business Publications,inc.,"The front of Neuro Computer". Nov. 1988
2)Ichikawa, M.,Shigematsu, Y.,Matsumoto, G., "a newly developed micro-processor for Brain Computing," Intenational Workshop on Brainware, pp66-68,Dec.1996.
3)SIA Road maps.
4)Furuyama,T., "Scaling and integration Limits for Sub 0.1 μ m LSI's," Abst. of Second International Research Workshop on Future Information Processing Technologies, Aug. 1997
5)Hirano, K.,Miyakawa, N.,Koyanagi, M.,et al.,"A New Three-Dimensional Multiport Memory for Shared Memory in High Performance Parallel Processor System,"Ext. Abst. Of SSDM,pp824-826,Aug.1996
6)A.Iwata "Optical Interconnection for ULSI Technology Inovation," Optoelectronics-Devices and Technologies(OP-DET), Vol.9, No.1, pp39-54, Mar. 1994
7)Yasunaga, M.,Masuda, N., Yagyu, M.,et al.,"A Self-Learning Digital Neural Network Using Wafer-scale LSI, IEEE Journal of Solid-State Circuits,Vol. 28,No.2,pp.106-114,1993
8)Kyuma, K., " J.IEE Vol.113, No.4, pp279-286,1993.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Temporal Learning Rule and Dynamic Neural Network Model

Yukifumi Shigematsu

Super Molecular Division, Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan

Abstract

The central nerve system is a very dynamic network which is changed by learning process. A new temporal learning rule, revised Hebbian rule with synaptic history, was proposed in order to organize the dynamic associative memory. The learning rule was applied to a pulse-driven neural network model, and a temporal associative memory was self-organized by input temporal signals. This result leads a new concept that the temporal sequences of events is memorized among the asymmetric connections in the network.

It was also shown that dynamic neural networks were effectively organized by using temporal information. Grouping or isolation for the multi-modal information was well done by the temporal learning processing. Those results suggest that the temporal information may do the important factor to organize the information processing circuits in the nerve system as well as the spatial one.

1 Introduction

The central nerve system receives external information by sensory organs, recognizes it and dynamically remakes own neural network into a spatio-temporal associative memory. It is possible by this leaning and memory process to construct high order of intelligence. Neural cell models in the conventional neural networks are fundamentally static, therefore recurrent loop or time delay elements are indispensable to treat time sequence signals. An appropriate learning rule has not been available for temporal associative memory.

Temporal information, however, is very important information for our daily lives. To consider the classical conditioning experiment, an animal can easily predict a late unconditioned stimulus from a preceding conditioned stimulus after training in the experiment. The order of time sequence is important and irreversible in temporal associative memory[1]. The real neural cells must have an essential capability of temporal information processing, and also a temporal learning rule may be inherent in their synapses.

There are many reports of physiological results of the synaptic plasticity in nerve cells for temporal signal processing. The long-term potentiation (LTP) or long-term depression(LTD) is an effect that the synaptic efficacy is strengthened by input stimulus for long time. The activity-dependent changes in synaptic efficacy are a key factor in the formation of neural circuits underlying learning and memory.

In this paper, we propose a new temporal learning rule and introduce a spatio-temporal associative memory. The learning rule is physiologically plausible one, a revised Hebbian rule with synaptic history[2]. It can realize an associative memory with sequence recalling in normal time order, and introduce an united understanding of spatio-temporal coding in the neural networks.

2. Temporal Learning Rule

We attempt to design make a novel learning rule for temporal information processing based on the understanding of the physiological results of synaptic plasticity.

The Hebbian learning rule is a well-known rule, which enhances the connection efficacy of a particular synapse if responses of both input and output cells are simultaneously active. It can generally be formulated by a following equation,

$$\Delta W_{ij} = C_{ij} \cdot X_j(t) \cdot Y_i(t) \qquad \cdots\cdots(1)$$

where W_{ij} is synaptic efficacy from the j-cell to the i-cell; $X_j(t)$ and $Y_i(t)$ are input and output response values of the i-cell, respectively; and C_{ij} is a parameter of learning speed. X_j and Y_i are binary values of 1 or 0 for pulse response.

In order to expand the Hebbian rule to the temporal learning process, a synaptic history of the input signal is introduced [5],

$$H_{ij}(t) = X_j(t) + q \cdot H_{ij}(t-1). \qquad \cdots\cdots(2)$$

where q is the decay time constant ($0<q<1$) for the input history. The value of $H_{ij}(t)$ is a accumulated value of the input signal received from the j-cell at a synaptic site of the dendritic portion in the i-cell. It is analogous to an after-effect such as the intracellular calcium ion concentration.

Two types of the revised Hebbian rule were introduced by using the history. One is an input dependent learning rule which is formulated as the product of the input signal and the synaptic history.

$$\Delta W_{ik} = C_{ik} \cdot \left(H_{ik}(t) - H_0\right) \cdot X_k(t) \qquad \cdots\cdots(3)$$

where H_0 is a learning threshold which distinguishes between enhancement and depression.

The other is an output dependent one in which a particular attention is paid on the output cell. This is formulated as

$$\Delta W_{hj} = C_{hj} \cdot \left(H_{hj}(t) - H_0\right) \cdot Y_h(t). \qquad \cdots(4)$$

where H_0 is the same one as in Eq.3.

These equations were applied to the case of the classical conditioning experiment. The output dependent learning rule was acceptable to explain the conditioning experiment result. Therefore, we came to a conclusion that the output dependent type of the revised Hebbian rule with synaptic history (Eq.4) was the temporal learning rule for the associative networks.

3. Neural Cell Model with Learning Rule

In order to construct the neural networks, we introduce a pulse neural cell model. The cell model is a spike accumulation and delta modulation(SAM) model, and is illustrated in Fig.1. It consists of three parts: (1) a leaky integrator for input pulses, (2) a binary threshold function to generate an output pulse train, and (3) subtraction part to remove a constant given value from internal potential at spike firing. The operating factors are the input pulse signal $X_j(t)$ from the j-cell, output signal $Y_i(t)$, accumulated potential $U_i(t)$ and internal potential $V_i(t)$. The $U_i(t)$ for the i-cell is described as

$$U_i(t) = \sum_j W_{ij} \cdot X_j(t) + a \cdot V_i(t-1). \cdots (5)$$

where 'a' is a decay constant (0<a<1) for the potential $V_i(t)$. Comparing $U_i(t)$ with threshold T, the output $Y_i(t)$ and internal potential $V_i(t)$ are

$$Y_i(t) = g[U_i(t) - T] \quad \cdots\cdots (6)$$

$$V_i(t) = U_i(t) - p \cdot Y_i(t) \quad \cdots\cdots (7)$$

where $g[z] = 1$ if $z >= 0$ and $g[z] = 0$ if $z < 0$, and 'p' is the subtraction constant at pulse firing. We adopt a discrete time interval and parallel operation for calculation. The time interval may include intervals of both action potential and depression period. The remaining value of the internal potential $V_i(t)$ is held for the next step and acts as a charged signal for the next firing.

This pulse neural cell model is easily adopted with the learning operation of the revised Hebbian rule with synaptic history. The input history decreases exponentially to zero after the input pulse, and two threshold levels (H_1, H_2) are set for enhancement and depression of the efficacy as shown in Fig. 2-A. For computation, the learning rule of Eq.4 is simplified into the stepwise relationship function, as shown in Fig.2-B. When the i-cell responds to fire, $Y_i(t) = 1$, the spike signal is feedback to each synapse and new efficacy is calculated by comparing the history and the thresholds.

The learning threshold H_1 is for enhancement, another threshold H_2 is for depression. When the after-effect of an input signal is nearly zero and the history is smaller than H_2, the efficacy is not changed. It means that the memory in the efficacy is not erased by non-correlated signals.

The following are learning procedures.

If $H_{ij}(t) >= H_1$ then

$$\Delta W_{ij} = c_1 \cdot (W_{max} - W_{ij}) \cdot Y_i(t). \quad \cdots (8)$$

If $H_{ij}(t) < H_1$ and $H_{ij}(t) > H_2$, then

$$\Delta W_{ij} = c_2 \cdot (W_{min} - W_{ij}) \cdot Y_i(t). \quad \cdots (9)$$

where, C_1 and C_2 are parameters of learning speed, and W_{max} and W_{min} are maximum and minimum values of the connection efficacy ($W_{max} > W_{ij} > W_{min}$), respectively. The learning calculation is applied to the firing cell only.

The synaptic history $H_{i*}(t)$ is calculated at each synapse(*) of the i-cell. It would otherwise require an elaborate calculation, but it is simplified; a history of output $Y_j(t)$ instead of each input history is calculated by using Eq.2, and deliverd it as each input history $H_{*j}(t)$ to all cells receiving from the j-cell.

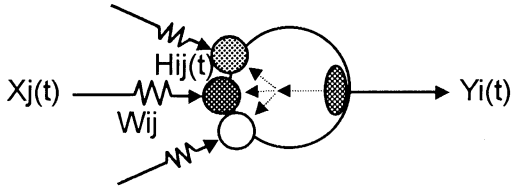


Fig.2 A model of temporal learning

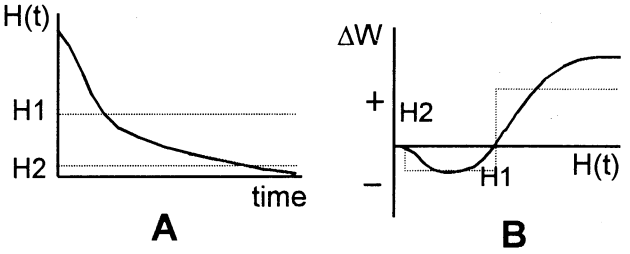


Fig.3 H(t) time sequence and ΔW

4. Temporal Associative Memory

Using the above pulse neural cell model and the revised Hebbian learning rule, a new type of temporal associative memory network is introduced. Each neural cell in this neural network interconnects with all other cells and has input and output terminals. The mutual connection efficacy is initially weak. The connections among the cells are changed by the following learning process.

A particular cell group A shown in Fig.3 is fired by an input signal. This firing output signal is transmitted to every other cell and stored at each synapse in the form of input history. When another cell group B is fired by a later input stimulus, the learning process is done in the fired cells of B. The efficacy of synaptic connection from group A to group B is enhanced by comparison of the values of the input history of B with the learning thresholds of H_1 and H_2. The new firing signal of B is transmitted to every

other cell. Another cell groups, C and D, and also fired one by one after these steps. The series of the temporal firing pattern 'A→ B→C→D' is memorized by applying the stimuli repeatedly to the above networks. After the training, the temporal sequence of input firing pattern is imprinted within the connections of the neural network. The neural network finally becomes a temporal associative memory.

The recall process of the associative memory is as follows. The recalling stimulus is applied to the input terminals of the group A for a short duration to fire the group A. The next group of B is induced by firing the signal from A through the enhanced mutual connection to the group B, and group C and group D can be recalled sequentially. The sequence of recalling is induced by the enhanced mutual connections. The recalling speed is independent of the timing period of the training process, but relates to the intensity of the input firing and the mutual connection efficacies among the sequence. If the network is well trained, predictive recalling is capable for the leaned sequences of the events.

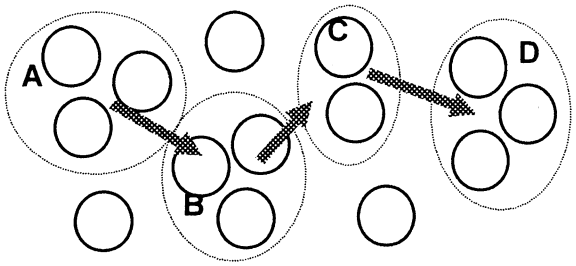


Fig.3 Temporal associative memory

In this temporal associative memory network, the temporal information is used to organize the asymmetrical connections among the neural cells. This network memorizes the sequences of the input signals or events, but does not memorize the duration of these events. This seems logical, based on our memory of the daily life. This learning rule also includes the characteristic of the Hebbian rule. Therefor, it can self-organize the spatial associative memory. The spatio-temporal associative neural network is self-organized by using this learning rule, the symmetrical connection is enhanced for spatial associative memory and the asymmetrical connection is done for temporal one.

5. Interaction among Different Signals

Second trial was effect of interaction between the different input signals for temporal learning process. The neural network was two layered network of input layer and receiving layer shown in Fig.4. The input layer consisted of five cells. Two of them output different random pulse trains, and other two cells output periodic pulse trains of which firing was out of phase to each other. The fifth cell output the same pulse train as one of these four input pulse signals. Averaged pulse interval for each cell was about one shot per 20 time steps.

The receiving layer cells had the weak excitatory mutual connections among four near R-cells. We also consider the effect of noise signal. Many neural cells generate spontaneous spikes without input signal, therefore random noise input is introduced into the neural cell in order to simulate the spontaneous spikes.

First(1-25) quarter of R-cells received random pulse train from the first input cell. Second(26-50), third(51-75) and fourth(76-100) quarter of R-cells received periodic, random and another periodic pulse trains from other three input cells respectively. The fifth input cell sent signal to whole R-cells(1-100). Each R-cell received two inputs from the fifth cell and one of other input ones.

We set signal of the fifth cell as the same periodic pulses of the second input cell, and tested the interaction from several initial conditions of the different connection weight(W1,W2,W3,W4,W5) from five input cells to R-cells. Calculation and learning was done until averaged value of connections within some zone was saturated (Wmax=0.30).

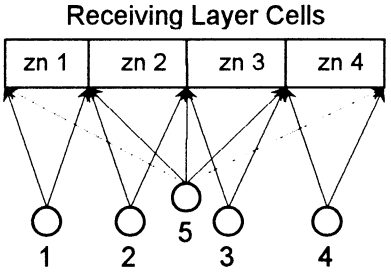


Fig.4 Pulse-driven neural networks with different pulse train inputs

Table 1 Wn by two pulse train inputs(Wmax=0.30)

	Initial	Zone 1	Zone 2	Zone 3	Zone 4
Wn	0.15	0.26	0.29	0.26	0.16
W5	0.15	0.23	0.29	0.20	0.19
Wn	0.03	0.26	0.29	0.28	0.09
W5	0.03	0.21	0.29	0.20	0.22

Fig.5 were a result from an initial condition of strong zone connections (W1-4=0.27) and weak common one (W5=0.03), where Wmax=0.3. Learning of the weak common connection was dominated by strong inputs to each zone cells. Synchronizing pulse signals co-worked to strengthen the connections from input cells to R-cells in 2nd zone, but asynchronizing signals suppressed the connections from 5th cell to R-cells in 4th zone. Random pulse signal accepted to make the moderate connections of the common input in 1st and 3rd zones.

Learning process under other conditions was studied by setting the different initial connection weights such as Wn/W5=0.03/0.27, 0.15/0.15 and 0.03/0.03. There is

similar tendency, as we noted above. If initial connection weights of common input and zone inputs were same value, both input signals competed to get dominant connections to R-cells shown in Table 1.

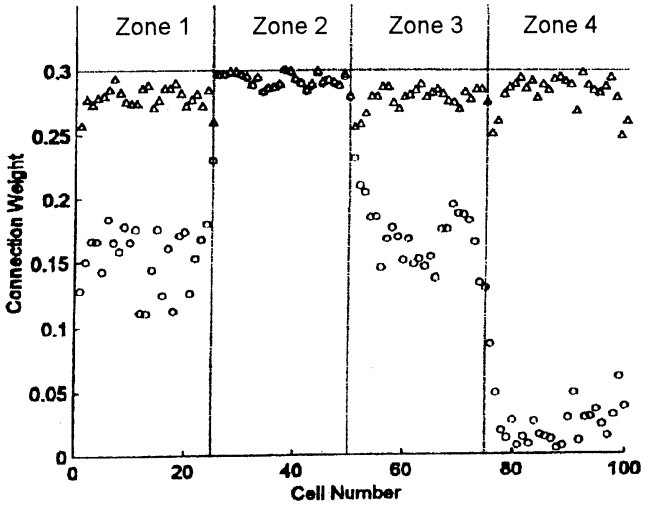


Fig.5 Connection weight of each R-cells from zone input cells (△, Wn) and the 5th cell (○, W5) after learning process.
Initial weight was $W_{1\text{-}4}$=0.27 and W_5=0.03. Quarter of R-cells(1-25) in first zone received random pulses. The cells in 2nd, 3rd and 4th zone received cyclic, random and another cyclic input pulse trains, respectively. The common input of the 5th cell was the same cyclic as 2nd one.

5. Summary and Discussion

The real nervous system receives huge flows of spatial and temporal information from the external world. Input signals to the memory system are complex, for examples simultaneous visual, auditory and motional signals and then the nervous system treats all of them to memorize associatively. In this process, role of the spatial and temporal associative memory is very important. We have shown the new temporal associative memory by means of the very simple networks.

Some important results were shown by the simulation of temporal learning process. Spatial information processing hardly treats the multi-modal or more higher leveled information, because spatial relation is no meaning for these field. Temporal information processing can do it by using the time correlation; The cells which received the correlated pulse input became to fire concurrently and made a group by increasing the connection weights for similar inputs. The pulse neural network could also isolate temporally correlated activity from non-correlated one by the temporal learning rule.

These results propose that temporal information processing is an important factor to self-organize the associative networks in the neural system as well as spatial information one.

We presented a primary function for temporal associative memory which the neural cell model should have. The change in synaptic efficacy may be the result of interference between potentiation and depression. We simplified it to use one component for this learning rule, but take two learning thresholds. Some physiological results suggest more complex mechanisms for example that the learning thresholds should vary as a function of previous synaptic activity.

We take a hypothesis of the feedback signal from the cell body through the dendrite to the synapses in the learning rule. Stuart and Sakmann[3] show that action potentials are initiated first in the axon hillock and then actively propagate back into the dendritic tree. The role of the dendritic action potential is unknown, but it can be a candidate for the feedback signal.

This network model was studied by using one shot pulse calculation, but burst pulses for strong input signal may be more realistic and effective. It is more difficult to make relation between the events for very long time interval by this processing, because time interval for grouping was taken within short time interval. Other function or mechanism should be added to associate the networks for long time relation.

The recall process is different from the conventional associative memory with loop circuits. The speed of recalling is flexible, and it may be possible to predict the subsequent events with an experienced network. This memory model provides the temporal relationship within the connections of the neural cells, and fundamentally does not need special recurrent loop. There are, however, a lots of recurrent loops in the real neural networks, because the recurrent loops can be used to introduce for other functions.

This learning rule depends on only local information of input and output signal of the neural cell. Global information is also necessary to construct a large scaled network system. One of the global information is a value factor such as reward and punishment used to controls the learning process to the suitable object.

For the more complex memory system, it is necessary to refer the functions and structures in the real nerve system such as hippocampus and neocortex. This model may have a possibility to simulate the real complex nerve systems because of the physiologically plausible learning rule and the realistic pulse-driven neural cell model.

References

[1]Herz,A., Sulzer,R. and van Hemmen, J.L. *Biol. Cybern.*, 60, 457-467. 1989. **[5]** Shigematsu,Y., Matsumoto,G. and M.Ichikawa, Brain Processes, Theories and Models, MIT Press 164-172. 1995. **[10]**Stuart, G.J. and Sakmann, B. Nature, 367, 69-72. 1994

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Robust Segmentation of Behaviorally Relevant Events in Sensory Recognition by Synergy of Thalamo-cortical and Cortico-cortical Control

Edgar Koerner
Future Technology Research
HONDA R&D Europe GmbH,
D-63073 Offenbach, Germany

Ursula Koerner
Future Technology Research
HONDA R&D Europe GmbH,
D-63073 Offenbach, Germany

Abstract

In this paper, we propose a large-scale computational theory of how the brain may organise computation in the context of a demanding behavioral problem, namely of how to recognise the sensory situation rapid and reliably despite the high complexity of sensory input. We show that the cortical architecture and the observed dynamics support the assumption of two functionally distinct forward processing pathways in the cortical hierarchy of processing. One selects a rapid but coarse initial hypothesis and, if required, activates temporarily an orthogonal thalamo-cortical control flow of gamma frequency characteristics which then enables the repetitive verification and refinement of the hypothesis utilizing the second pathway. Our model suggests a novel efficient coding scheme for cortical-type processing that results in a rapid and reliable convergence to the optimal solution even in case of highly complex decision problems. Furthermore, the model provides a novel model of cognitive binding in the brain which is both biologically and computationally plausible.

1. Introduction

Understanding the mechanisms that enable the brain to define a globally consistent cognitive event out of the massively parallel, distributed, and basically asynchronous local processing in this large-scale system, known as the problem of cognitive binding, is a key to better understand at all what brain-like processing is about. For several years, dynamic linking by mutual synchronisation of neurons in a network (14) was considered a reasonable model for cognitive binding in neural systems, which has been interpreted with respect to the experimentally observed gamma oscillations in different subsystems of the vertebrate cortex as "active feature linking"(10).
However, the model is neither from computational nor from neurobiological point of view a plausible one. Synchronisation by mutual entrainment via direct interaction between neurons implies an a-priori full connectivity in the distributed system, and a sufficient high activity in the network to establish and keep a synchronous state that, according to this theory, should be the representative of a globally consistent cognitive event in the brain. We know meanwhile that less than 100 neurons form a so-called ensemble encoding a complex sensory event, that the spike rates of cortical neurons are very low, usually less than 40 Hz, and that there is a variety of transmission delays inbetween tentative ensembles of neurons, practically discarding the active feature link model. Moreover, as we never can observe steady synchronous activity in mammalian cortices, but only brief transient epochs of gamma synchronisation, there is still no functional interpretation of this manifest and widespread phenomenon of gamma oscillation in cortex, and no realistic model of cognitive binding, consequently.
To avoid the trap of local computational models, we approach the definition of a globally consistent semantic entity in cortical computation in the context of a demanding behavioural problem: Instead of considering the recall of one pattern out of few stored ones in a homogeneous neural network, we assess the required architecture and dynamics for a rapid and reliable recognition of real-world sensory situations. It is widely agreed on that sensory recognition must be a bi-directional process to escape from the threat of combinatorial explosion of alternative decisions inherent in a local forward filtering. The top-down prediction of an activated global hypothesis reduces the high-dimensional decision problem of sensory recognition to a simple comparison of input vs. expectation (13,5). But before a top-down prediction can do the trick, an a-priori hypothesis is required. In case of a sudden change of sensory context, the brain must have the capability to cope with that novel input too when the existing global hypothesis resulting from the previous recognition processes looses its predictive value and has to be replaced by a new one.
Rapid activation of a reliable initial hypothesis in sensory recognition must be done in a purely forward processing mode, since there is not yet any top-down prediction that can support the selection of the best fitting stored representation of sensory environment in response to a certain input configuration. How to rapidly activate a true initial hypothesis and refine it subsequently is a yet unsolved problem that we address in this paper taking vision as an example.

2. The computational problem

2.1. Activation of an initial hypothesis

Our assumption is as follows: To avoid the problem of combinatorial explosion, the cortical processing should utilise a subjective reduction of sensory complexity to trigger the initial hypothesis by only a limited number of local decisions having the highest internal truth-value. Later, after successful triggering a true hypothesis, the already acquired knowledge could be used to reconstitute the sensory input as much as possible under the constraints of time and existing knowledge (which could be recalled for refinement of the hypothesis).

Neuronal feature detectors do not respond exclusively to the feature they are tuned to, but give a graded response to many different input configurations within its receptive field. Only in case of a maximum response of a lower level detector neuron, the neuron next in the hierarchy after such a feature detector could know for sure that the feature this neuron is tuned to is at its input. The lower the activation of detector neurons in its receptive field, the more the subsequent neuron needs an already established context to properly interpret the description by the neurons in its receptive field. We define the truth value of a neuronal decision as the degree of matching of the feature the respective neuron is tuned to with the input in its receptive field. Hence, to select the initial hypothesis, only perfect matches should be used. The signalling of partial matches should be delayed till the activated initial hypothesis provides the context for a proper interpretation of partial matches.

How to segment perfect from partial matches? Because of the low firing rates of cortical neurons, the truth value of local decisions cannot be defined by firing rate. Already about 100 ms after presentation of a face to a monkey, the first activation of neurons encoding a face in inferotemporal cortex is observed, just few milliseconds more than the sole time needed for transmission of signals through the hierarchy of local feature detection in the visual pathway (9). At each hierarchic level, the decision is taken within 5..10 ms after arriving of the afferent inputs. Discriminating inputs by firing rate, with cortical firing rates being lower than 40 Hz, would require integration intervals of more than 100 ms.

Therefore, the firing rate related PCM-code which is verified for subcortical encoding cannot be a candidate for this aspect of cortical computation. Instead, we propose the following ensemble temporal encoding scheme (relative latency encoding). We assume:

1) Neocortical neurons encode the truth value of their respective decision on the input by the relative latency of its response. That means, neurons signal the presence of its respective feature at the input by sending a spike to the next level of filtering with minimumn latency if there is a perfect match. The less the degree of matching between input and feature the neuron is tuned to, the longer the latency between arriving of the input and generation of a spike, if there is a spike at all. There is experimental support for this asssumption (1).

2) Neocortical neurons are coincidence detectors (4). Provided that a coherent trigger of the sensory input (saccade, or a sudden appearance of a visual object) provides a global time relation within the asynchronous cortical processing, then at any same-level feature detectors the local decisions are ranked in time according to their truth-values. And if there is a sufficient number of levels of local filtering before an object definition is made (which is the case in vision), then, only the best matching features have an impact on decision making at the respective next higher hierarchic level (12). The global recognition process would be started by a very low number of exclusively reliable local decisions (recognized features) on the input. That scheme could explain the capability of visual processing to activate with a high degree of probability a true hypothesis in the forward processing pathway only.

2.2.Refinement and rectification of hypothesis

However, there are two serious flaws inherent in this scheme: 1) Only the very first sweep of afferent spike wave could be used for decision making, since the coherent trigger input provides only once the global timing that is required for relative latency encoding in a basically asynchronous processing system as the brain. However, it is not guaranteed that the first sweep enables a correct recognition indeed, a repetitive evaluation of the input must be possible. Therefore, a decision-process related clocking of cortical processing is required to provide the global coherent time relation.

2) The coding capacity is rather low, and the information encoded in partial matches is lost. It is very unlikely that the neocortical neuron's response duration of up to 200 ms contains simply noise to be deleted. Besides partial matches being needed for fine discrimination, learning of prior unknown objects always starts with partial matches to already acquired object representations. Hence, the system must have the ability to include partial matches into the internal description of sensory input.

As we will show, these conflicting requirements (to pick up only the decision results with highest truth-value to trigger the recall, while utilizing even weak partial matches for refinement of the activated hypothesis in a basically asynchronous processing mode) can be satisfied by a 'cortical-type' homogeneus network composed of complex nodes, the columnar units, which can switch from asynchronous to clocked mode by induced temporary rhythmic control of processing.

3. Implementation of the theory into a model neocortical architecture

The required additional mechanism for both embedding of partial matches into the activated initial hypothesis and repetitive evaluation of afferent input could be related to globally coherent clocking and modulation of neocortical neurons by thalamo-cortical gamma oscillations which are experimentally observed in all cortical areas.

3.1. Induced ILN gamma oscillations enforce coherent decision intervals in cortical processing

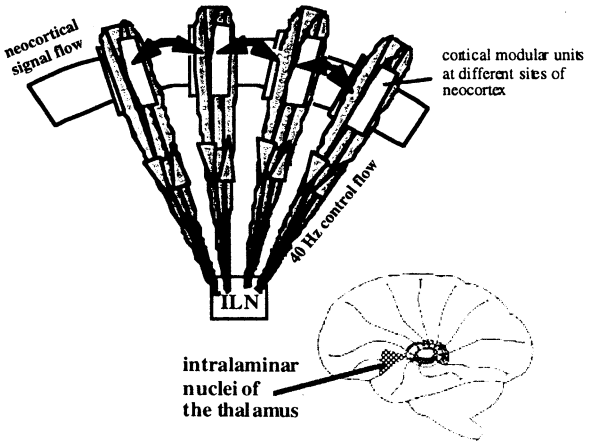


Fig.1: The thalamic intralaminar nuclei (ILN) have a recurrent topographic connectivity to neocortical areas and are involved in the generation of cortical gamma rhythm (7).

Intralaminar thalamic nuclei (ILN) receive convergent inputs from lower layers 5 and 6 of cortical columnar modules, and return a topographic but locally divergent excitation to layer 1 (Fig. 2). During arousal and REM sleep, ILN neurons produce short (≈10 ms) high-frequent bursts of spikes that are repeated in gamma range if they are activated by cortical inputs (11). Because of the central position of ILN (Fig. 1), the transmission delay from and to ILN is the same for any cortical region, and it is very short. Even a relatively small but coherent wave of spikes in the cortical stream could trigger gamma bursting of ILN which in turn will modulate the membrane potential of all principal neurons in cortical columns, except those which are the direct recipients of afferent inputs (Fig.2). The non-specific thalamo-cortical loops via ILN are here proposed to constitute an orthogonal modulatory control of bi-directional cortical processing. Effectively, the gamma modulation would be set to a well defined phase relation with the cortical coherent trigger input. And since the afferent inputs form a preferentially gamma oscillating loop between specific thalamic nuclei and the neurons in columnar neurons that receive direct afferent inputs (8), the repetition of the sparse and specific coherent afferent input in gamma frequency would be followed by the resulting global and non-specific modulation for some epoch with a constant phase delay. We interpret this fixed-delay gamma range repetition of those two qualitatively different activation processes as a short-term clocking of the otherwise asynchronous cortical processing, setting coherent intervals for making a decision overall the cortex (see. ch. 3.3.).

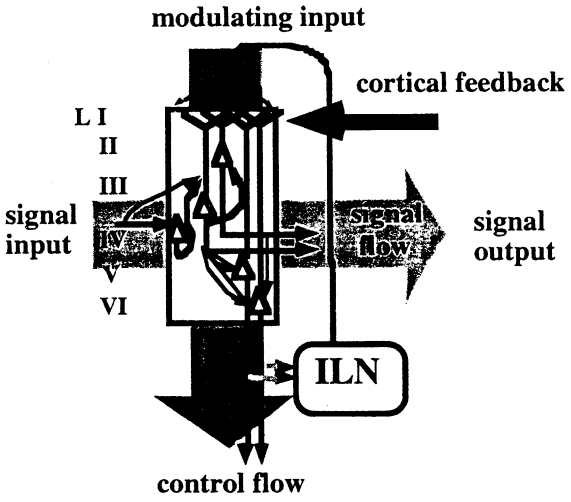


Fig. 2: Principal neurons of all layers extent their apical dendrites to layer I, except those which are the major recipients of afferent inputs (spiny stellate neurons of layer IV - SS4). Layer I is the site of major contacts of cortico-cortical feedback and of ILN modulation. SS4 do not get neither thalamic modulation nor cortical feedback, while the Py of other subsystems do.

3.2. Two forward pathways in cortical object recognition

We abstract the architecture of cortical columnar units into 3 functional subsystems (Fig. 3): The express forward processing subsystem A that is driven by afferent inputs dominantly, and which selects the initial hypothesis at each respective hierarchic level of encoding from the best matches of previous level (local decisions with highest truth-value that arrive as a coherent wave of shortest relative latency, see. ch. 2); subsystem B that has the capability to evaluate partial matches signalled from the previous hierarchic level but only then if it is enabled to do so by cortico-cortical predictive (top-down) feedback and ILN modulation - otherwise it is silent; and subsystem C which evaluates local processing in the respective columnar unit and generates the feedback both to subcortical systems (including thalamus) and lower levels of cortical processing stream. SS4 neurons of

subsystem A evaluate the parallel afferent input by coincidence detection and have a sufficinetly high threshold. Therefore only a highly coherent wave of

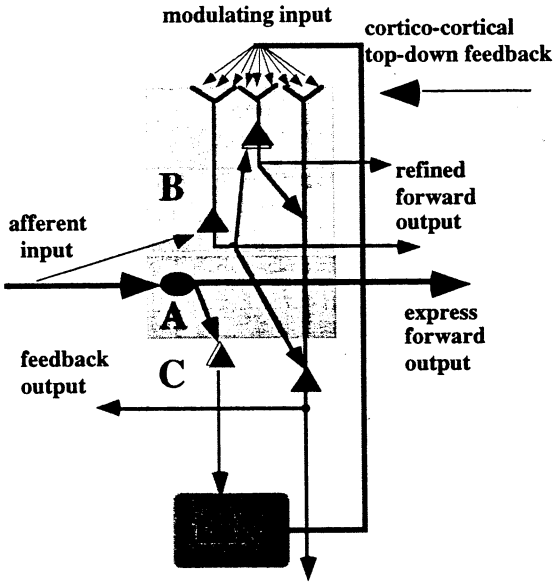


Fig. 3: Simplified model cortical columnar unit: subsystems A of the same hierarchic level activate an initial hypothesis and start ILN modulation, B is only then enabled to refine the hypothesis by using partial matches from lower level output, if ILN modulation and cortical feedback are active, C generates the output of the columnar unit.

spikes from feature detector neurons in the respective receptive field (best matches only) can start an initial hypothesis at A subsystems of columnar units which then are transmitted as express feedforward to the next level. Since the express feedforward output is a coherent wave of spikes travelling upward the cortical parallel processing stream, it can trigger the ILN gamma oscillation even if the ensemble of spikes in this wave is rather small. Because of the fast feedforward inhibition in A mediated by local inhibitory neurons which we set according to cortical

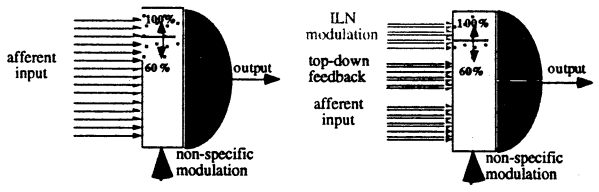


Fig.4: Principal neurons of A subsystem can be activated by its massive afferent inputs only. Neurons of subsystems B and C receive sparse inputs from 3 distinct sources and cannot be activated by afferent input only, but with additional ILN modulation and cortical predictive feedback.

layer 4 architecture, only the very first activation at any local A subsystem is enabled, while any subsequent activation of A by partial matches with their larger relative latency compared to best matches is prohibited for the duration of approximately one gamma period by this long lasting inhibition.

The return of ILN modulation to B and C subsystems which was triggered by A subsystem express output coincides with the return of the first predictive top-down feedback resulting from the same A express output to the next higher level of processing (see Fig.3). Hence, those spots at one hierarchic level benefit most of ILN modulation which are predicted by the initial hypothesis actiavted at the next higher one.

Neurons of B subsystem constitute a tapped delay line like architecture. Afferent inputs and the outputs of A are input to neurons of the lower part of B, while each neuron gives sparse inputs to a large number of neurons subsequent to it towards layer 1 of the columnar unit (Fig. 3). Activation of a columnar unit starts at A, which triggers ILN gamma activity; then with enabling activation of B by the return of ILN modulation, activation moves from the boundary between B and A subsystems upward. The observed delay between activation at A and final activation at the upper part of B is about 12 ms (Douglas & Martin 1991), which is T/2 of gamma. While there is a rather small number of principal neurons in A that may encode sparse prototypes, there are plenty of principal neurons capable of encoding degenerate extensions made of those prototypes.

3.3. Gamma-paced repetition of decision processes

An onset of afferent input will first activate a set of SS4 (maximum one per macrocolumn) that trigger the recall of an initial hypothesis at the subsequent hierarchic level, while the refinement of hypothesis at each level starts with the return of induced ILN modulation (Fig.5).

The constant delay between arrival of the coherent wave of best hits in A subsystems and the beginning of refinement in B subsystems at any level of processing clocked by MP modulation enables the system to interpret partial matches in the context of an already well activated hypothesis - which is provided by the predictive top-down feedback that in turn is the result of the express forward output of A systems at the same level but evaluated already in a more global context at the next higher one. Thus, the predictive feedback enables only those columnar units that are expected to be active if the initial hypothesis activated at the next higher level is true. At those columnar units, predictive feedback and ILN modulation superpose. This definitely sets the context for interpretation of the partial matches

evaluated by subsystem B, by limiting the local units at which the ILN modulation can be most effective.

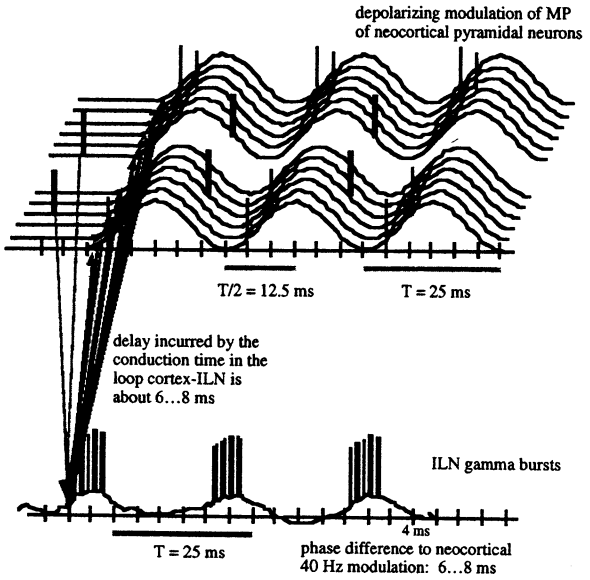


Fig. 5: The upper figure shows a model case for two macrocolumns which respond to a coherent onset of input in their receptive fields. Thick bars refer to spikes signalling best hits, thin bars refer to partial matches.

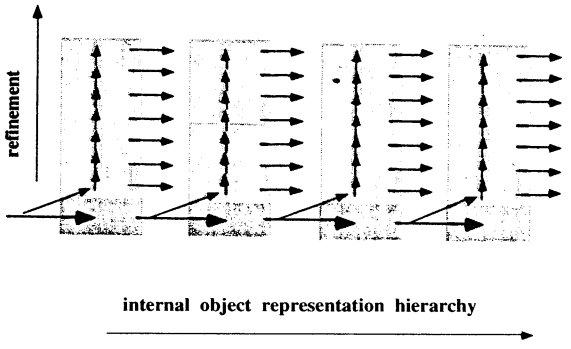


Fig.6: Parallel-sequential mode of processing generated by gamma paced temporal encoding in cortical recognition systems composed of the proposed model columnar units. Feedback is not shown here. See text for explanation.

If our model is correct, then sensory inputs arriving within T/2 gamma should be recognised as only one cognitive event: This is just what has been observed experimentally. As long as the time difference between two tones is larger than 12..15 ms, these tones can be separated by a human observer, if the time distance is shorter, only one event is recognised (3).

We have simulated the fundamental characteristics of the proposed gamma-paced temporal encoding for the case of A subsystems arranged in macrocolumns as it is the case for the neocortical sensory processing areas. Those simulations show the time course of repetitive decision making as outlined above (6).

4. Conclusion

The proposed architecture and the described type of ensemble temporal encoding enables a novel type of fast but robust recognition, that is refined in an analysis-by-synthesis manner [5]. Furthermore, our model provides a novel model for, and a functional interpretation of, neocortical gamma oscillations in a scheme of cognitive binding which is both computationally and biologically plausible.

References:

(1) Celebrini, S, Thorpe, S., Trotter, Y. & Imbert, M. (1993) Dynamics of orientation coding in area V1 of the awake primate. Visual Neuroscience 10: 811-825
(2) Douglas, R.J. & Martin, K.A.C. (1991) A functional microcircuit for cat visual cortex. Journal of Physiology 440 pp.735-769
(3) Joliot, M., Ribary, U. & Llinas, R.: Human oscillatory brain activity near 40 Hz coexists with cognitive temporal binding. Proc. Natl. Acad. Sci. USA 91(1994)pp. 11748-11751
(4) Koenig, P., Engel, A.K. & Singer W. (1996) Integrator or coincidence detector? The role of cortical neuron revisited. TINS 19, pp. 130-136
(5) Koerner, E., Tsujino, H., Masutani, T. (1997) A Cortical-type Modular Neural Network for Hypothetical Reasoning. Neural Networks 10:791-814
(6) Koerner, E., Koerner U, & Nagai, S. (1997) Recurrence of decision making in cortical type neural systems by gamma controlled temporal encoding in spiking neurons. Soc. Neurosci. Abstr. vol.23, p.1304
(7) Llinas, R. & Ribary, U. (1993) Coherent 40-Hz oscillation characterizes dream state in humans. Proc.Natl.Acad.Sci.USA, 90, 2078-2081
(8) Llinas, R. & Pare, D. (1996) The brain as a closed system modulated by the senses. in: The mind brain continuum, R.Llins & P.S. Churchland (eds.),1-18
(9) Oram, M.W. & Perrett, D.I. (1992) Time course of neural responsesdiscriminating different views of the face and the head. J. of Neurophysiology 68: 70-84
(10) Singer, W. (1993) Synchronization of cortical activity and its putative role in information processing and learning. Annu. Rev. Physiol. 55, pp. 349-374
(11) Steriade, M, CurroDossi, R. & Contreras, D. (1993) Electrophysiologicalproperties of intralaminar thalamocortical cells discharging (40 Hz) spike bursts at 1000 Hz during waking and eye movement sleep. Neurosciences 56: 1-9
(12) Thorpe, S. & Gautrais, J. (1997) Rapid visual processing using spike asynchrony. Advances in Neural

Information Processing Systems 9, M. C. Mozer, M. I. Jordan, and T. Petsche (eds.) MIT Press Books 1997
(13) Ullman, S. (1995) Sequence seeking and counter streams: A computational model for bidirectional flow in the visual cortex, Cerebral Cortex, 1, pp. 1-11
(14) v.d.Malsburg, C. & Schneider, W. (1986) A neural cocktail-party processor.
Biol. Cyb. 54, pp.29-40

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Hypothetical Reasoning and Brainware

Hiroshi Tsujino
Fundamental Research Division
Wako Research Center
HONDA R&D Co., Ltd.
Wako-shi, Saitama, 351-01, Japan.

Edgar Koerner
Future Technology Research Division
HONDA R&D Europe GmbH
63073 Offenbach/Main, Germany.

Hiroshi Kondo
Fundamental Research Division
Wako Research Center
HONDA R&D Co., Ltd.
Wako-shi, Saitama, 351-01, Japan.

Abstract

In order to solve real-world problems like visual recognition where incomplete knowledge is used, it is necessary to have a robust computational theory to effectively deal with both internal knowledge and noise. Although considerable research on the problem of hypothetical reasoning has been conducted in the filed of artificial intelligence, the required logical representation of the instantiated hypothesis is a major disadvantage.

The brain constructs a virtual reality as a hypothesis and acts on that rather than on objective sensory input. To perceive the world, building an internal model of the external world and fitting this to sensory input is a general characteristic of cortical processing. The control structure of hypothetical reasoning is the most important part of brainware, for not only the cognitive process but also the self-organization of knowledge in the brain.

We propose a multi-layer neural network architecture that can implement the hypothetical reasoning control, which the neocortex seems to perform. The simulation shows continuous and robust reasoning and representation for real-world visual recognition.

Keywords: Hypothetical Reasoning, Brainware, Visual Cortex, Modular Neural Network

1 Introduction

In order to solve real-world problems, such as visual recognition, where incomplete knowledge is used, it is necessary to have a robust computational theory to effectively deal with both internal knowledge and noise. Although considerable research on the problem of hypothetical reasoning has been conducted in the filed of artificial intelligence (Poole[11], de Kleer[1]), the required logical representation of the instantiated hypothesis is a major disadvantage.

The brain constructs a virtual reality as a hypothesis and acts upon it rather than on objective sensory input. In perceiving the world, building an internal model of the external world and fitting this to sensory input is a general characteristic of cortical processing. Basically, any area in neocortex has bi-directional capabilities. There is the control pathway which can be used for hypothetical reasoning. Additionally, there is much evidence that the brain creates hypotheses in the higher cortical area which modulate hypotheses in the lower cortical area. Picton and Stuss[9] suggested that the human brain generates an internal model

to fit incoming information about the external world. Ramachandran and Cobb[12] demonstrated that metacontrast, known as a visual illusion, could be strongly modulated by top-down influences. Kosslyn et al.[6] reported that the primary visual cortex was activated when subjects closed their eyes and visualized objects. Logothetis et al.[7] found that binocular rivalry involved the higher cortical area and was in the general category of multistable phenomena.

The control structure of hypothetical reasoning is the most important part of brainware for not only the cognitive process but also for the self-organization of knowledge in the brain. There are some related studies which cite that bi-directional control is one of the important elements of brain processing (Fukushima[2], Kawato et al.[3], Mumford[8], Poggio[10], Ullman[14]). However, as yet no one has proposed a hypothetical reasoning model that controls the virtual image as a hypothesis like a brain. We propose a novel hypothetical reasoning scheme based on the basic structure of the neocortex, a columnar module. The columnar module, creating and evaluating hypotheses, acts as an elementary processor in the brain.

2 Basic design

2.1 Macro scheme

A proper global hypothesis will accelerate the formation of a consistent combination of the local hypotheses. However, to obtain the proper global hypothesis is exactly the problem.

We hypothesize that the brain avoids this dilemma by coarse input to the highest cortical area via the amygdala. Amygdala is an old subcortical part of the limbic system relating to emotion. Koerner et al.[4] proposed that phylogenetic top-down development proceeds under the control of the amygdala, and, also, that in the primate neocortex the top-down control would still be dominant in defining the context for understanding sensory input (Fig.1).

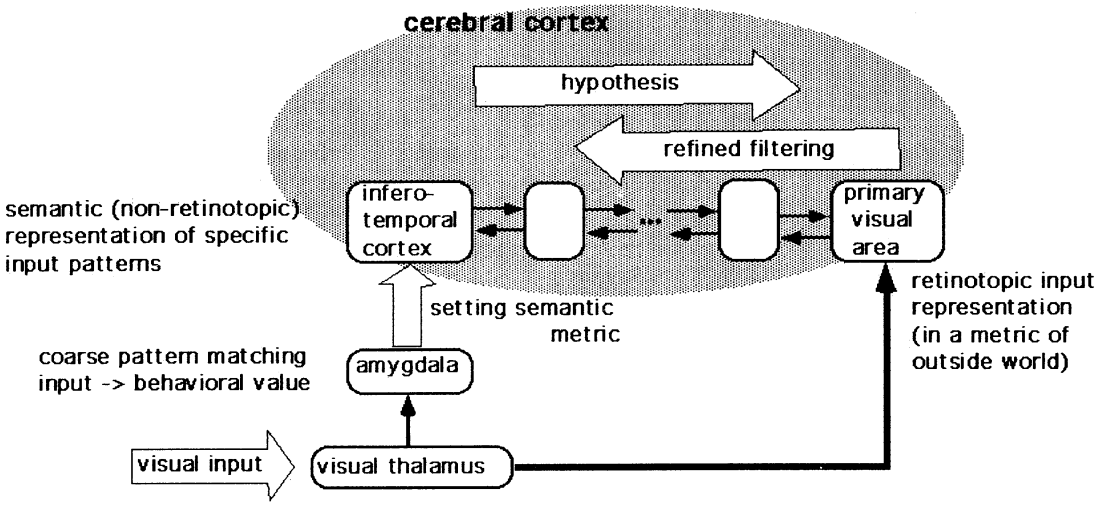


Figure 1. Macro scheme of hypothetical reasoning

2.2 Columnar architecture

Flattening the cortex, we find a horizontal six-layer laminar structure and vertical columnar structure. The basic organization of the column does not depend on the information represented there.

We[5] proposed that this architecture of columnar units represents not the structure of the knowledge stored there, but the control that forces the system to make the representations. Fig. 2 shows correspondence of the proposed columnar unit to the general architecture of a neocortical column. Afferent input is evaluated by neurons in layer 4 (R0) that trigger activation of supragranular neurons (R1) of the same column. A subpopulation of pyramidal neurons in the infragranular layers extends its apical dendrites throughout the column till layer 1. Furthermore, all pyramids of layer 2 and 3 provide collateral input to lower pyramids when passing their feedforward to R0 of columns in the next higher cortical area.

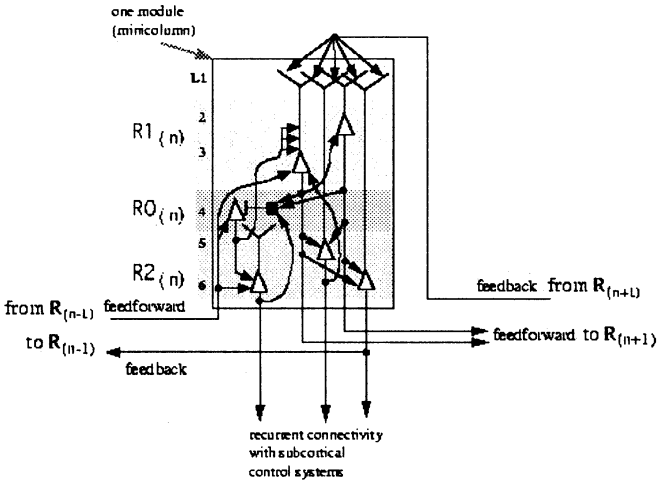


Figure 2. Columnar Model

3 Simulation

3.1 Segmentation by hypothetical control

Tsujino et al.[13] simulated face recognition under a complex background. The system consists of V4, posterior inferotemporal area and anterior inferotemporal area of neocortex. Fig.3 shows the result of a recognition task.

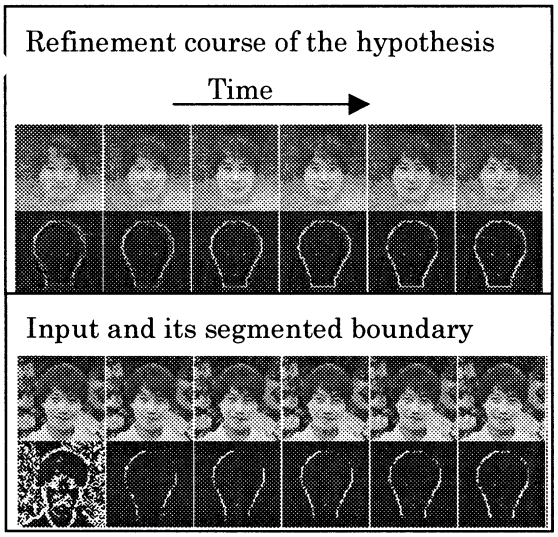


Figure 3. Result of the face recognition

3.2 Representation in the column

We propose that the columnar structure in the cortical area may be the result of the mapping of the information represented in the previous cortical area and that supragranular neurons of each column may represent the information as compactly as possible in order to send it to the next cortical area.

By determining how to represent the information, we gained new insights into the complex cells. Fig.4 shows the simulated result of the highly tuned receptive field of complex cells in the primary

visual area.

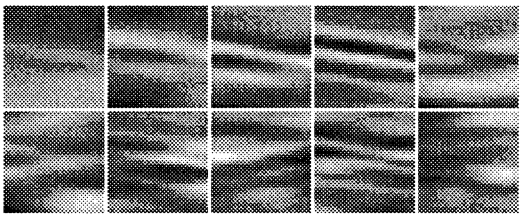


Figure 4. The variety of the complex cells in V1

4 Conclusion

We propose a basic scheme of how hypothetical reasoning is performed in the brain. The scheme not only elegantly illustrates the performance of the recognition task but also suggests a new interpretation of brain computing.

References

1. de Kleer, J., An Assumption-based TMS, Artificial Intelligence, 28:127-162, 1986.
2. Fukushima, K., A neural network model for selective attention in visual pattern recognition, Biological Cybernetics, 55:5-15. 1986.
3. Kawato, M., Hayakawa, H. and Inui, T., A forward-inverse optics model of reciprocal connections between visual cortical areas, Network, 4:415-422, 1993.
4. Koerner, E., Koerner, U. and Mastumoto, G., Top-down Selforganization of Semantic Constraints for Knowledge Representation in Autonomous Systems: A Model on the Role of an Emotional system in Brains, Bulletin of the Electrotechnical laboratory, 60:405-409, 1996.
5. Koerner, E., Tsujino, H. and Masutani, T., A Cortical-type Modular Neural Network for Hypothetical Reasoning, Neural Networks, 10 (5):791-814, 1997.
6. Kosslyn, S.M., Thompson, W.L., Kim, I.J. and Alpert, N.M., Topographical representation of mental images in primary visual cortex, Nature, 378:496-498, 1995.
7. Logothetis, N.K., Leopold, D.A. and Sheinberg, D.L., What is rivaling during binocular rivalry?, Nature, 380:621-624, 1996.
8. Mumford, D., On the computational architecture of the neocortex, II. The role of cortico-cortical loop, Biological Cybernetics, 66:241-251, 1992.
9. Picton, T.W. and Stuss, D.T., Neurobiology of conscious experience, Current Opinion Neurobiology, 4:256-265, 1994.
10. Poggio, T., Torre, V. and Koch, C., Computational vision and regularization theory, Nature, 317:314-319, 1985.
11. Poole, D., A Logical Framework for Default Reasoning, Artificial Intelligence, 36:22-47, 1988.
12. Ramachandran, V.S. and Cobb, S., Visual attention modulates metacontrast masking, Nature, 373:66-68, 1995.
13. Tsujino, H., Koerner, E. and Masutani, T., Multi-Agent-Based Hypothetical Reasoning for Image Recognition (in Japanese), Journal of Japanese Society for Artificial Intelligence, Vol.12 (3):440-447, 1997.
14. Ullman, S., Sequence seeking and counter streams: A computational model for bidirectional information flow in the visual cortex, Cerebral Cortex, 5:1-11, 1995.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Identification of Nonlinear Dynamic Systems Based on Universal Learning Network

Min Han, Kotaro Hirasawa, Masanao Ohbayashi,
Junichi Murata, Jinglu Hu and Masayuki Hashimoto
Graduate School of Information Science and Electrical Engineering,
Kyushu University, higashi-ku, Fukuoka 812-81,Japan
han@scel.ees.kyushu-u.ac.jp

Abstract

In this paper, we study how the generalization ability of a Universal Learning Network(ULN) can be improved by taking advantages of the switching mechanism and the second order derivatives of ULN. The switching mechanism between the nodes is used to control branch deletion and to make the structure of the networks compact. The new criterion function with second order derivative terms to minimize the deviation caused by the external input changes is also considered. All of these approaches may improve the generalization ability of the network.

1 Introduction

Neural networks have been widely studied in recent years. Neural networks can simulate a certain complicated systems by learning. But, current neural networks are composed of fixed nodes and branches and can not be equipped with arbitrary time delays. So it is difficult to apply these kinds of neural networks to the modeling of large scale complicated systems. In order to solve this problem, Universal Learning Network (ULN)[1][2] has been proposed. Learning algorithms of ULN by forward and backward propagation are useful for modeling and controlling large scale complicated systems such as industrial plants, economic, social and life phenomena.

Generally most of the large scale complicated systems can be modeled by the networks which consist of nonlinearly operated nodes and multi-branches that may have arbitrary time delays including zero or minus ones. Therefore, ULN can be applied to many kinds of systems which are difficult to be expressed by ordinary first order difference equations with one sampling time delays.

Generalization ability is known to be an important characteristics of neural networks. In this paper, we study how the generalization ability of the ULN can be improved by taking advantages of the switching mechanism and second order derivatives of ULN. The new points of this paper in terms of improving the generalization ability are that a switching mechanism between the nodes is introduced, and a new criterion function with second order derivative terms are used in order to minimize the deviation caused by the external input changes.

2 Structure of ULN

The structure of ULN with multi-branches and switching mechanisms [4][5] is shown in Fig.1.

In order to make the network compact, every branch has filtering mechanisms, namely, switching mechanisms such as $\alpha_{ij}(p)$ on pth branch from node i to node j . The learning parameter in the switching mechanism should be trained so that $\alpha_{ij}(p)$ becomes 0.0 if a branch from node i to node j is unnecessary and $\alpha_{ij}(p)$ becomes 1.0 if a branch from node i to node j is necessary for the network to have predetermined performance.

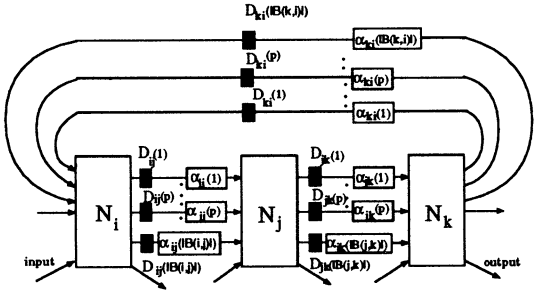


Fig.1 Structure of ULN with multi-branches and switching mechanisms

Basic equation of ULN is represented by Eq.(1),

$$h_j(t) = O_j(\{h_i(t - D_{ij}(p))|i \in JF(j), p \in B(i,j)\}, \{r_n(t)|n \in N(j)\}, \{\lambda_m(t)|m \in M(j)\}) \quad (1)$$

where $h_j(t)$ is the output value of node j at time t, O_j the nonlinear function of node j, $r_n(t)$ the value of n-th external input variable at time t, $\lambda_m(t)$ the value of m-th parameter at time t, $D_{ij}(p)$ the time delay of p-th branch from node i to node j, $B(i,j)$ the set of branches from node i to node j, $JF(j)$ the set of nodes whose outputs are connected to node j, and $N(j)$, $M(j)$ the sets of external inputs and adjustable parameters related to node j, respectively.

As mentioned before there are switching mechanisms between the nodes, that is used to control branch deletion. The switching mechanism $\alpha_{ij}(p)$ is supposed to be

$$\alpha_{ij}(p) = \frac{1}{1 + e^{-\varphi\beta_{ij}(p)}} \tag{2}$$

Parameter $\beta_{ij}(p)$ is also adjusted in the same way as other parameters in order to minimize the extended criterion function including compactness evaluation.

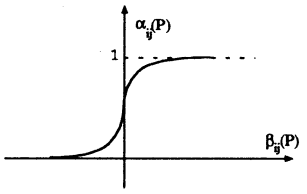


Fig.2 Switching mechanisms

One of the things to be noticed is that φ in Eq.(2) is set to a small value at the beginning of learning, and is scheduled to increase gradually as the learning progresses. By executing the above procedure, upper bound $\alpha_{ij}(p) = 1.0$ or lower bound $\alpha_{ij}(p) = 0.0$ which mean connection and disconnection of the pth branch from node i to node j can be acquired at the end of learning.

The criterion function L in this paper consists of three parts, which is shown by Eq.(3).

$$L = E + E_s + E_d \tag{3}$$

$$E_s = k_s \sum_i \sum_j \sum_p (\alpha_{ij}(p))^2 \tag{4}$$

$$E_d = k_d \sum_{n \in N} \sum_{t'} \left(\frac{\partial^\dagger E}{\partial r_n(t')} \Delta r_n(t') \right)^2 \tag{5}$$

where E is an usual criterion term defined by a function of node's outputs, parameter and time. E_s is an extended term which is related to the structure of network, and a second order derivative term to minimize the deviation caused by the external input change is expressed by E_d. k_s is a coefficient related to the structure of the networks. k_d is one related to the second order derivative.

When k_s is large, unnecessary branches are deleted($\alpha_{ij}(p) = 0.0$) so that the structure of the network becomes compact, that is, the generalization ability is improved. On the other hand, when k_s is small, most branches are connected($\alpha_{ij}(p) = 1.0$) so that the large structure of the network is produced, that is, the generalization ability degrades.

Moreover, if k_d is assigned to a relatively large value, then the deviation of E caused by the change of the external input can be curbed so that the generalization ability is improved.

Depending on the value of weight coefficient k_s and coefficient k_d, the balance between the criterion function of E and the generalization ability of the network may be adjusted.

In the following chapters, a computing method for the second order derivatives of the criterion function E with respect to parameter λ_m and $r_n(t')$ is presented.

3 Learning of Parameters

Let parameter λ_m of ULN be defined as follows,

$$\lambda_m = \{w_{ij}(p), \beta_{ij}(p) | j \in J, i \in JF(j), p \in B(i,j)\} \tag{6}$$

where $w_{ij}(p)$ and $\beta_{ij}(p)$ are the adjustable parameters related to node j.

Training of ULNs is realized by minimizing a criterion function L based on a gradient method

$$\lambda_m \leftarrow \lambda_m - \gamma \frac{\partial^\dagger L}{\partial \lambda_m} \tag{7}$$

where γ is the learning coefficient assigned to a small positive value and $\frac{\partial^\dagger L}{\partial \lambda_m}$ is the ordered derivatives defined by Werbos[3], which means the change of the criterion function L caused by the change of λ_m with other variables being fixed.

$$\frac{\partial^\dagger L}{\partial \lambda_m} = \frac{\partial^\dagger E}{\partial \lambda_m} + \frac{\partial^\dagger E_s}{\partial \lambda_m} + \frac{\partial^\dagger E_d}{\partial \lambda_m} \tag{8}$$

$$\frac{\partial^\dagger E_d}{\partial \lambda_m} = 2k_d \sum_{n \in N} \sum_{t' \in T} \left[\left(\frac{\partial^\dagger E}{\partial r_n(t')} \Delta r_n(t') \right) \times \frac{\partial^{\dagger 2} E}{\partial r_n(t') \partial \lambda_m} \Delta r_n(t') \right] \tag{9}$$

Now, computation of $\frac{\partial^\dagger E}{\partial \lambda_m}$, $\frac{\partial^\dagger E_p}{\partial \lambda_m}$ can be carried out by making use of the first.

In the following subsections , the computation method of $\frac{\partial^\dagger E}{\partial r_n(t')}$, and $\frac{\partial^{\dagger 2} E}{\partial r_n(t')\partial \lambda_m}$ will be explained in detail.

3.1 Computing of $\frac{\partial^\dagger E}{\partial r_n(t')}$

First order derivative of E with respect to external input $r_n(t')$ can be written in the form of Eq.(10) assuming t' to be designated sampling times,

$$\frac{\partial^\dagger E}{\partial r_n(t')} = \sum_{r\in R_o}\sum_{s\in S_o}\Big(\frac{\partial E}{\partial h_r(s)}\frac{\partial^\dagger h_r(s)}{\partial r_n(t')}\Big)+\frac{\partial E}{\partial r_n(t')} \quad (10)$$

As $\frac{\partial E}{\partial h_r(s)}$, and $\frac{\partial E}{\partial r_n(t')}$ can be calculated easily, it is important to calculate $\frac{\partial^\dagger h_r(s)}{\partial r_n(t')}$.

$\frac{\partial^\dagger h_r(s)}{\partial r_n(t')}$ can be generally transformed into Eq(11).

$$\frac{\partial^\dagger h_k(t)}{\partial r_n(t')} = \sum_{j\in JF(k)}\sum_{p\in B(j,k)}\Big(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \times\frac{\partial^\dagger h_j(t-D_{jk}(p))}{\partial r_n(t')}\Big)+\frac{\partial h_k(t)}{\partial r_n(t')} \quad (11)$$

Putting $P_1(k,t,r_n(t')) = \frac{\partial^\dagger h_k(t)}{\partial r_n(t')}$, iterative equation of P_1 by forward propagation can be obtained from Eq(12).

$$P_1(k,t,r_n(t')) = \sum_{j\in JF(k)}\sum_{p\in B(j,k)}\Big[\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \times P_1(j,t-D_{jk}(p),r_n(t'))\Big]+\frac{\partial h_k(t)}{\partial r_n(t')} \quad (12)$$

$$P_1(j,t<t',r_n(t')) = 0 \quad (13)$$

3.2 Computing of $\frac{\partial^{\dagger 2} E}{\partial r_n(t')\partial \lambda_m}$

Second order derivative of E with respect to parameter $r_n(t')$ and λ_m can be obtained as follows,

$$\frac{\partial^{\dagger 2} E}{\partial r_n(t')\partial \lambda_m} = \sum_{r\in R_o}\sum_{s\in S_o}\Big[\frac{\partial^\dagger\Big(\frac{\partial E}{\partial h_r(s)}\Big)}{\partial \lambda_m}\frac{\partial^\dagger h_r(s)}{\partial r_n(t')} + \sum_{r\in R_o}\sum_{s\in S_o}\frac{\partial E}{\partial h_r(s)}\frac{\partial^{\dagger 2} h_r(s)}{\partial r_n(t')\partial \lambda_m}\Big] + \frac{\partial^\dagger\Big(\frac{\partial E}{\partial r_n(t')}\Big)}{\partial \lambda_m} \quad (14)$$

Putting $P_1(k,t,r_n(t')) = \frac{\partial^\dagger h_k(t)}{\partial r_n(t')}$, and $P_2(k,t,r_n(t'),\lambda_m) = \frac{\partial^\dagger h_r(t))}{\partial r_n(t')\partial \lambda_m}$, as in the case of the first order derivatives, iterative equation of P_2 by forward propagation can be obtained by differentiating P_1 with respect to λ_m,

$$\frac{\partial^{\dagger 2} h_k(t)}{\partial r_n(t')\partial \lambda_m} = \sum_{j\in JF(k)}\sum_{p\in B(j,k)}\Big[\frac{\partial^\dagger\Big(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(k))}\Big)}{\partial \lambda_m}\frac{\partial^\dagger h_j(t-D_{jk}(p))}{\partial r_n(t')} + \frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\frac{\partial^{\dagger 2} h_j(t-D_{jk}(p))}{\partial r_n(t')\partial \lambda_m}\Big] + \frac{\partial^\dagger\Big(\frac{\partial h_k(t)}{\partial r_n(t')}\Big)}{\partial \lambda_m} \quad (15)$$

$$P_2(k,t,r_n(t'),\lambda_m) = \sum_{j\in JF(k)}\sum_{p\in B(j,k)}\Big[\frac{\partial^\dagger\Big(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\Big)}{\partial \lambda_m} \times P_1(j,t-D_{jk}(p),r_n(t')) + \frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \times P_2(j,t-D_{jk}(p),r_n(t'),\lambda_m)\Big] + \frac{\partial^\dagger\Big(\frac{\partial h_k(t)}{\partial r_n(t')}\Big)}{\partial \lambda_m} \quad (16)$$

$$P_2(j,t<t',r_n(t'),\lambda_m) = 0 \quad (17)$$

$\frac{\partial^\dagger\Big(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\Big)}{\partial \lambda_m}$, $\frac{\partial^\dagger\Big(\frac{\partial h_k(t)}{\partial r_n(t')}\Big)}{\partial \lambda_m}$, in Eq.(16) can be calculated by the computation of first order derivatives putting $E' = \frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}$,$E'' = \frac{\partial h_k(t)}{\partial r_n(t')}$ respectively.

Substituting $\frac{\partial^\dagger h_r(s)}{\partial r_n(t')}$, and $\frac{\partial^\dagger h_r(s))}{\partial r_n(t')\partial \lambda_m}$ obtained from Eq.(12) and (16) into Eq.(14), $\frac{\partial^{\dagger 2} E}{\partial r_n(t')\partial \lambda_m}$ can be calculated.

4 Example

As an example of investigating the generalization ability, a nonlinear system identification problem was studied, which consists of a cascade connection of nonlinear subsystem $g_N(.)$ and linear subsystem $g_L(.)$ shown in Fig.3.

The cascade connection system of the nonlinear and linear subsystem was modeled by the ULN in Fig.1,

which has 5 nodes and fully recurrent connections with 3 branches between the nodes and one external input. It was supposed that the output of ULN can be obtained from N_5 node.

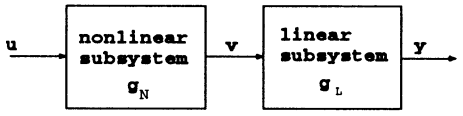


Fig.3 Cascade connection of nonlinear and linear subsystem

Input value $u(k)$ and output value $y(k)$ of the system to be identified are expressed by Eq.(18).

$$y(k+1) = \begin{cases} 1.34y(k) - 0.277y(k-2)) \\ -0.80y(k-4) + 0.01, \quad u(k) \geq 0 \\ 1.34y(k) - 0.277y(k-2) \\ -0.80y(k-4) - 0.01, \quad u(k) < 0 \end{cases} \tag{18}$$

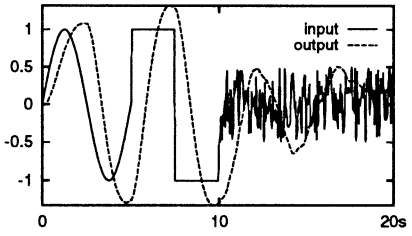


Fig.4 Input and output data for training

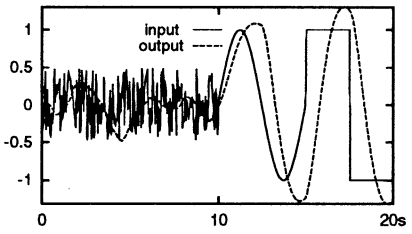


Fig.5 Input and output data for investigating the generalization ability

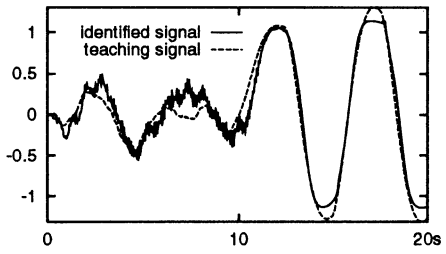


Fig.6 Results of testing the generalization ability ($k_s = 0.1, k_d = 0.0$)

Fig.5 shows input and output data for investigating the generalization ability. In Fig.5, the order of sine function, rectangular function and random number for input $u(k)$ is changed from the order of the training data shown in Fig.4.

Fig.6 shows one of the results when $k_s = 0.1$ and $k_d = 0.0$.

5 Discussion

In this paper, a new method for improving the generalization ability of the networks has been proposed. One of the features of the proposed method is to use the extended criterion function defined by Eq.(5). Therefore, second order derivatives obtained by the Universal Learning Network are utilized effectively and usefully. Future works which should be done are to study quantitative evaluation on how much the generalization ability is improved by appropriately choosing the parameters k_s and k_d.

References

[1] K. Hirasawa, M. Ohbayashi and J. Murata, " Universal Learning Network and Computation of its Higher Order Derivatives," *Proc. of 1995 IEEE International Conference on Neural Networks*,pp 1273-1277, 1995

[2] K. Hirasawa, M. Ohbayashi, M. Koga and M. Harada, " Forward Propagation Universal Learning Network", *Proc. of 1996 IEEE International Conference on Neural Networks*,pp 353-358, 1996

[3] P.Werbos, "Beyond regression: New Tools for Prediction and Analysis in the Behavior Science", *Ph.D. dissertation, Harvard University*, 1974

[4] M. Han, K. Hirasawa, M. Ohbayashi and H. Fujita, " Modeling Dynamic Systems using Universal Learning Network", *Proc. of 1996 IEEE International Conference on Systems, Man and Cybernetics*, pp 1172-1177,1996

[5] M. Han, K. Hirasawa, M. Ohbayashi and H. Fujita, "Generalization Ability of Modeling Dynamic Systems Using Universal Learning Network", *11th IFAC Symposium on System Identification*, pp1243-1248, 1997

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Recognition of Shapes and Shape-Changes in 3D Objects by GRBF (Generalized Radial Basis Function) Network

Noriaki Kinoshita, Takanori Katsuki, Tetsuya Miyazaki,
Miwako Hirakawa, Masahiro Okamoto*

Dept. of Biochemical Engineering and Science,
Kyushu Institute of Technology,
Iizuka-city 820, Japan

*To whom correspondence should be addressed
(Tel) +81-948-29-7821
(Fax) +81-948-29-7801
(e-mail) okahon@bse.kyutech.ac.jp

Abstract

Poggio and Edelman have shown that for each object there exists a smooth mapping from an arbitrary view to its standard view and that the mapping can be learnt from a sparse data set. They have demonstrated that such a mapping function can be well approximated by gaussian GRBF(Genaralized Radial Basis Function) network. In this paper, we have applied this network to the recognition of three kinds of hand shape-changes, such as grasp, storoke and flap. Also during learning stage, we have introduced a structural learning algorithm to GRBF network in order to explore the small-sized and essential network structure.

1 Introduction

Automatic acquisition of 3D models from images is usually a very difficult problem. One way to overcome the need of 3D model is to exploit methods for representing objects by a collection of 2D views (2D projection('snap shots')) rotating 3D object. Poggio and Edelman [1], and Edelman and Poggio [2] have proposed such a view-based object-recognition method named GRBF(Generalized Radial Basis Function) network which relys on multiple 2D views instead of 3D models. This network resembles to conventional artificial neural network, however, each unit in a hidden layer is represented by radial basis function such as gaussian. In this paper, we have applied GRBF network to the recognition of shape-changes of flexible object; three kind of hand shape-changes such as grasp, stroke and flap. We have also designed structural learning algorithm to explore the small-sized and essential GRBF network structure. We show their performance by computer simulation.

2 GRBF network for rigid objects

A gussian GRBF network is shown in *Figure* 1, which consists of three layers (input layer + one hidden layer + output layer). Since 3D object is supposed to be a collection of 2D coordinates (perspective view) of the feature points on the image, each 'snap-shot' 2D view is represented as a 2n vector $X_1, Y_1, X_2, Y_2, ..., X_n, Y_n$ of the coordinates on the image plane of n labelled and visible feature points on the object. We assume here that all features are visible. Each basis unit in hidden layer can be represented by $Eq.(1)$, where T_x and T_y is a center of gaussian function for X-axis and Y-axis, respectively, m is the number of basis units in hidden layer. The W is supported to be binding coefficient (weight) between units in input layer and those in hidden layer. The resulting value $G_k(k = 1, 2, .., m)$ can be regarded as the 'activity' of the unit. The output of GRBF network is the linear superposition of the activities of all the basis unit (G_k) in hidden layer as shown in $Eq.(2)$. Since GRBF network is designed to explore the mapping that transforms each 'snap-shot' 2D view to its standard 2D view of the object, the objective function of the network can be written in $Eq.(3)$, where $(X_{st,i}, Y_{st,i})(i = 1, 2, ..., n)$ is standard coordinate of

$$G_k = exp\left(\sum_{i=1}^{n} -(X_i - T_{x_k})^2 W_{k(2i-1)}^2\right) \times exp\left(\sum_{i=1}^{n} -(Y_i - T_{y_k})^2 W_{k(2i)}^2\right) \quad (k = 1, 2, ..., m) \tag{1}$$

$$f_j = \sum_{k=1}^{m} C_{k,j} G_k \quad (j = 1, 2, ..., 2n) \tag{2}$$

$$E = \sum_{q=1}^{N} \sum_{i=1}^{n} \left((X_{st,i} - f_{2i-1})^2 + (Y_{st,i} - f_{2i})^2\right) \tag{3}$$

$$f_{2n+p+1} = \sum_{k=1}^{m} C_{k,2n+p+1} G_k \quad (p = 0, 1, .., obj - 1) \tag{4}$$

$$H = \sum_{q=1}^{N} \left(\sum_{i=1}^{n} \left((X_{st,i} - f_{2i-1})^2 + (Y_{st,i} - f_{2i})^2\right) + A \sum_{p=0}^{obj-1} (F_{st,p} - f_{2n+p+1})^2\right) \tag{5}$$

the ith object, N is the number of learning data set (examples). In order to be learnt several kinds of 3D objects in one gaussian GRBF network, teacher flags for recognition of object, $F_{st,i}(i = 0, 1, ..., obj - 1)$, were prepared, where obj represents the number of 3D objects to be learnt; for example, supposed three kinds of objects should be learnt, when 2D projection view of the 1st. object is given as an example, $(F_{st,0}, F_{st,1}, F_{st,2})$ is set to be $(1, 0, 0)$, for the case those of the 2nd. object is given, $(F_{st,0}, F_{st,1}, F_{st,2})$ will be $(0, 1, 0)$ and so on. As the same manner as $Eq.(2)$, flag output $(f_{2n+p+1}, (p = 0, 1, .., obj - 1))$ will be calculated as shown in $Eq.(4)$. In this study, considered the mean squared error between flag output and teacher flag, total objective function (H) can be written in $Eq.(5)$, where A is an arbitrary coefficient.

3 GRBF network for flexible objects

If the shape of a flexible object changes smoothly, the mapping from an rbitrary view of the object to standard view is expected to be synthesized by smooth interpolation of shape-specific mappings. We demonstrate that the mapping which gives the standard view of a flexible object can be synthesized by learing in expand GRBF network shown in *Figure* 2, where situation 0 and 1 represent arbitrary feature status before shape-changing and the feature status after shape-changing, respectively. The other symbols and definition are as same as in *Figure* 1. In

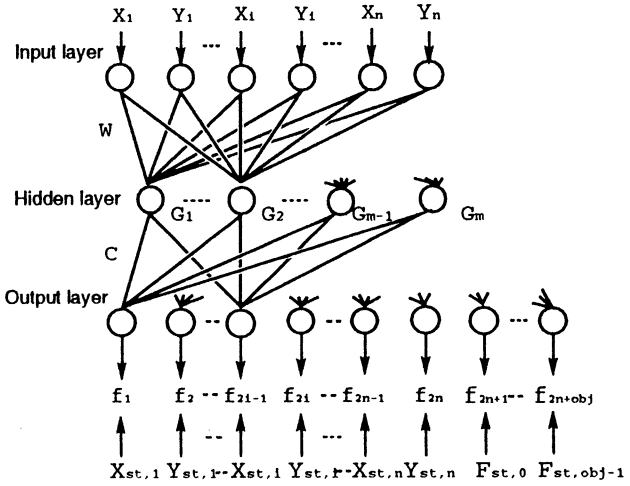


Figure 1: GRBF network structure

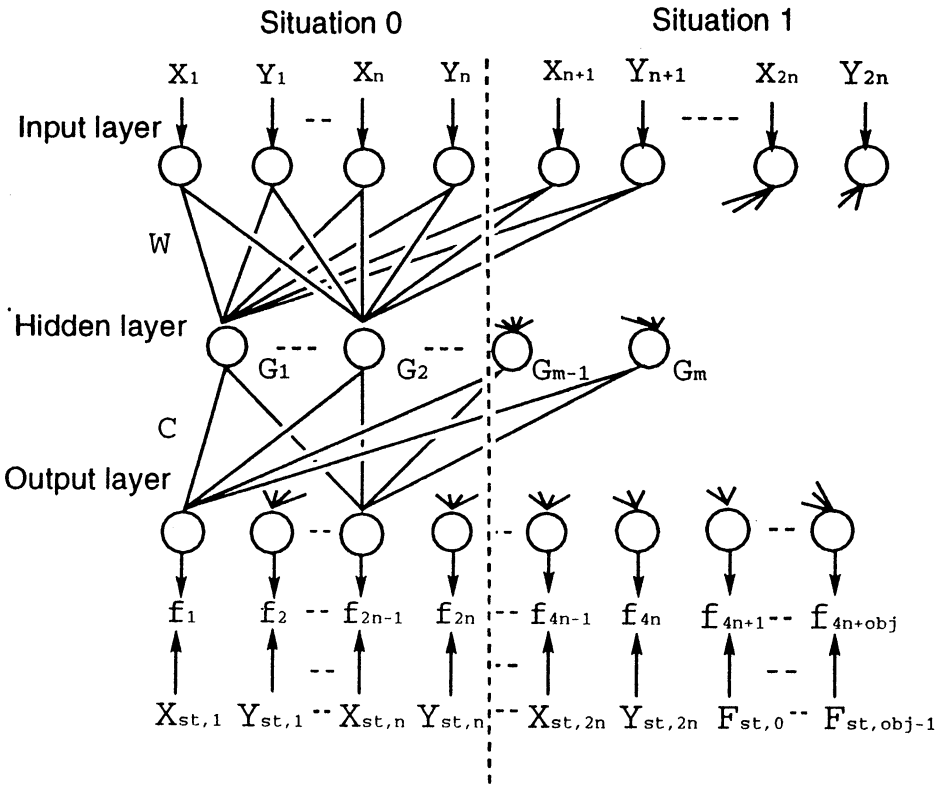


Figure 2: Expand GRBF network structure

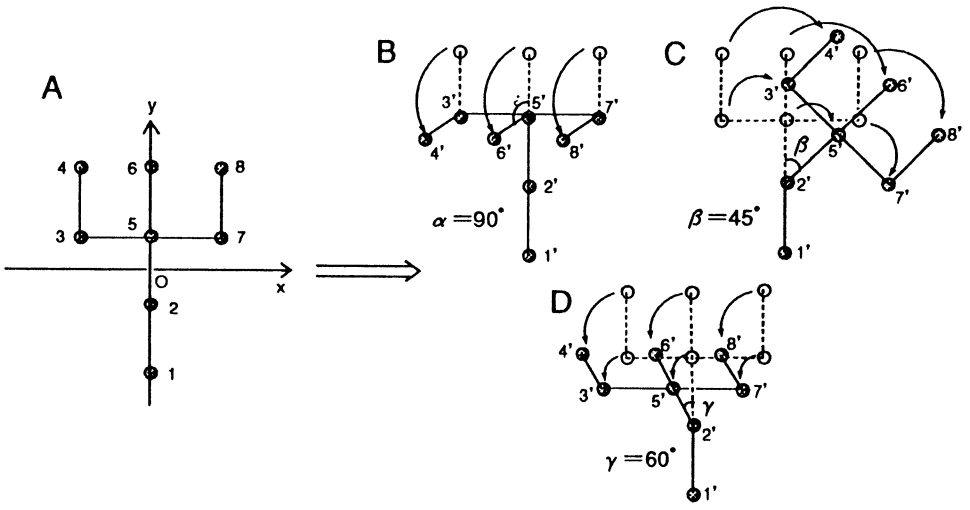


Figure 3: Wire frame models for hand shape-change

Table 1: Evaluation of normalized error for the recognition of hand shape-change (m = 30)

shape -change	ave. of Error* against grasp-st	ave. of Error* against stroke-st	ave. of Error* against flap-st	flag output $(f_{4n+1}, f_{4n+2}, f_{4n+3})$**	teacher flag $(F_{st,0}, F_{st,1}, F_{st,2})$
grasp	0.0174	0.632	0.203	(1.08, 0.0496, -0.0759)	(1, 0, 0)
stroke	0.726	0.00816	0.720	(-0.0158, 1.00, -0.00944)	(0, 1, 0)
flap	0.189	0.627	0.0115	(0.035, -0.0102, 0.960)	(0, 0, 1)

* the value of Eq.(6), ** average value of 100 2D views

this study, we have applied this gaussian GRBF network to the recognition of three kinds of hand shape-changes, such as grasp, stroke and flap; *Figure* 3 shows wire frame models for hand shape-change where situation A represents basic feature status of hand before shape-changing, B, C and D are feature status after shape-changing; grasp, stroke and flap can be regarded as $AtoB$, $A \rightarrow C$ and $A \rightarrow D$, respectively. Thus, in learning process, 2D view of situation A is given to situation 0 at input layer and that of B, C or D is given to situation 1 at input layer. The value of teacher flags for recognition, $(F_{st,0}, F_{st,1}, F_{st,2})$ is $(1,0,0)$ for grasp, $(0,1,0)$ for stroke, and $(0,0,1)$ for flap, respectively. Rotating angle of 3D object around point 0,$\phi = (\phi_x, \phi_y, \phi_z)$, is set from $-30°$ to $30°$ in this study.

4 Recognition of hand shape-change

Prepared 20 2D views for each shape-change (total 20 × 3=60sets) as training examples, the values of parameter set, Tx, Tx, W and C were optimized to minimize the value of objective function written in *Eq.*(5) (change n to $2n$). After learning, 100 unknown 2D views (data not being used in learning process) are provided to the gaussian GRBF network (*Figure*2). The number of basis unit (m) in hidden layer was fixed at 30. The performance of the GRBF network was evaluated by the following normalized error:

$$Error = \frac{1}{N}\sum_{j=1}^{N} \frac{\sqrt{\sum_{i=1}^{2n}\left((X_{st,i} - f_{2i-1})^2 + (Y_{st,i} - f_{2i})^2\right)}}{\sqrt{\sum_{i=1}^{2n}\left((X_{st,i} - g_{xst})^2 + (Y_{st,i} - g_{yst})^2\right)}} \quad (6)$$

where g_{xst} and g_{yst} represent x-coordinate and y-coordinate of the center of gravity in standard 2D view, respectively, and N is the total number of unknown 2D views (training example). *Table* 1 shows the performance of gaussian GRBF network for recognition of hand shape-change. Provided 100 unknown 2D views of grasp to the trained GRBF network, the average normalized errors defined by *Eq.*(6) against standard 2D views of grasp (grasp-st), stroke (stroke-st) and flap (flap-st) are 0.0174, 0.632, 0.203, respectively. The average value of flag output $(f_{4n+1}, f_{4n+2}, f_{4n+3})$ is (1.08, 0.0496,-0.0759), which is very close to the value of teacher flag for grasp ((1, 0, 0)). As the same manner as this, average normalized error against standard 2D view of stroke takes the minimum value (0.00816) when unknown 2D views of stroke are provided. These results show that the gaussian GRBF network defined in *Figure* 2 can recognize three kinds of hand shape-change, such that grasp, stroke and flap. Given 100 unknown arbitrary 2D views to the trained GRBF network one by one, the correct recognition rate was examined only by evaluating the values of flag output; when flag output $(F_{st,0}, F_{st,1}, F_{st,2})$ takes $(1.0 \pm \theta, 0 \pm \theta, 0 \pm \theta)$, unknown arbitrary 2D view is regarded as 'grasp' because of that teacher flag for grasp is supported to be (1, 0, 0). The case of $(0 \pm \theta, 1.0 \pm \theta, 0 \pm \theta)$ and $(0 \pm \theta, 0 \pm \theta, 1.0 \pm \theta)$ are regarded as 'stroke' and 'flap', respectively. The results are shown in *Table* 2 under the conditions of $\theta = 0.1$ and in *Table* 3 for $\theta = 0.35$. The GRBF network showed high recognition capability over 90%.

5 Structural learning of GRBF network for rigid objects

Since the network structure of the gaussian GRBF resembles to that of conventional artificial neural network, the first difficulty is the necessity of a priori specification of the number of basis units (m) in hidden layer; the decision of the number is hardly available in most cases, therefore, learning with trial and error becomes inevitable. The flexibility of GRBF network structure in accordance with the difficulty in recognition of shapes or shape-changes should be required. We shall show here structural learning algorithm which have focused on automatic adjustment of the number of basis units in hidden layer; structural learning method aims at exploring the small-sized and

Table 2: Correct recognition rate for hand shape-change ($\theta = 0.1$)

unknown-shape-change	recognition			correct recognition rate
	grasp	stroke	flap	
grasp	82	0	0	82%
stroke	0	91	0	91%
flap	0	0	93	93%

Table 3: Correct recognition rate for hand shape-change ($\theta = 0.35$)

unknown-shape-change	recognition			correct recognition rate
	grasp	stroke	flap	
grasp	96	0	0	96%
stroke	0	96	0	96%
flap	0	0	100	100%

Table 4: Evaluation of normalized error for the recognition of rigid objects

object	ave. of Error* aginst object A-st	ave. of Error* against object B-st	ave. of Error* agianst object C-st	flag output ($f_{2n+1}, f_{2n+2}, f_{2n+3}$)**	teacher flag ($F_{st,0}, F_{st,1}, F_{st,2}$)
A	0.0770	3.88	4.81	(0.975, 0.00223, 0.0273)	(1, 0, 0)
B	5.39	0.00563	3.40	(-0.0216, 0.991, 0.0350)	(0, 1, 0)
C	5.00	3.33	0.0506	(-0.0210, 0.0105, 1.02)	(0, 0, 1)

* the value of Eq.(6) (change $2n$ to n), ** average value of 100 2D views

Table 5: Correct recognition rate for rigid objects

unknown-object	recognition			correct recognition rate
	object A	object B	object C	
A	95	0	0	95%
B	0	99	0	99%
C	0	0	94	94%

essential network structure enough to recognize 3D shapes or shape-changes. In this section, the recognition of three kinds of rigid objects having 8 apexes shown in *Figure* 4 was examined with using gaussian GRBF network shown in *Figure* 1. The algorithm for structural learning can be represented as follows:

(1) Set up the number of basis units (m) in hidden layer arbitrary.

(2) Start training (learning) with using examples.

(3) If network can succeed in learning, proceed to the module of decreasing m (go to step (4)). Otherwise proceed to the module of increasing m (go to step (6)).

(decreasing module)

(4) After cutting off one basis unit in hidden layer (G_k) and all relative connection lines binding to the G_k, calculate the value of objective function H_k written in *Eq.* (5). Repeat this procedure (4) with changing k with the increment 1 until $k = m$.

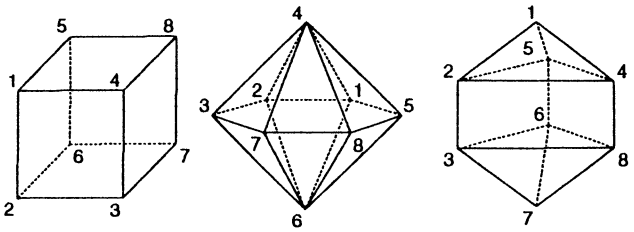
(5) If k is equal to m, search the minimum value among H_i ($i = 1, 2, .., m$), and consider new network with cutting off the ith basis unit (G_i) which provides the minimum value of H. Set m-value to m-1 and back to (2).

(increasing module)

(6) If network has an experience in succeeding of learning, back to the previous trained network structure and terminate; the current m-value is determined to be the final number of basis units in hidden layer. Otherwise proceed to (7).

(7) Increase the value of m with $\alpha\%$ (m: integer value) and back to (2).

Set initial guess of m (the number of basis units in hidden layer) at 21, the efficiency of our algorithm of structural learning was examined. The program could automatically decide the final number of m at 14. Fixed m-value at 14, the performance of the gaussian GRBF network for recognition of rigid objects A, B, and C was calculated with using the normalized error function written in *Eq.*(6) (change *2n* to *n*); the results are shown in *Table* 4. Provided 100 unknown arbitrary 2D views of object A to the GRBF network, the average values of normalized error function against A, B and C are 0.0770, 3.88, 4.81, respectively; the value against A (object A-st) takes the minimun among them. The average value of flag output ($f_{2n+1}, f_{2n+2}, f_{2n+3}$) is (0.975,0.00223,0.0273), which is very close to the value of teacher flag for object A($(F_{st,0}, F_{st,1}, F_{st,2})$=(1, 0, 0)). In the same manner as this, this gaussian GRBF network with having structural learning algorithm showed high performance as shown in *Table* 4. Provided 100 unknown arbitrary 2D views among ojects A, B, C, one by one to the trained GRBF network, the correct recognition rate was examined by evaluating the values of flag output. The results are shown in *Table* 5. The trained GRBF network showed high capability of recognition over 90%.



No. of Apex	A (x,y,z)	B (x,y,z)	C (x,y,z)
1	50,-50, 50	-87, 0, 0	0, 0, 87
2	50,-50,-50	43,-75, 0	-38,-65, 29
3	50, 50,-50	-43,-75, 0	-38,-65,-29
4	50, 50, 50	0, 0, 87	75, 0, 29
5	-50,-50, 50	-43, 75, 0	-38, 65, 29
6	-50,-50,-50	0, 0,-87	-38, 65,-29
7	-50, 50,-50	87, 0, 0	0, 0,-87
8	-50, 50, 50	43, 75, 0	75, 0,-29

Figure 4: Model of rigid objects having 8 apexes. The lower parts represent coordinate (x,y,z) of each apex.

References

[1] Poggio T. and Edelman S., (1990), "A network that learns to recognize 3D objects", *Nature*, **343**, pp.263-266.

[2] Edelman S. and Poggio T., (1990), "Bringing the grandmother back into the picture: a memory-based view of object recognition", *MIT AI Lab. Memo*, No.1181.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Adaptive Control for Autonomous tractor using Gyroscope and DGPS

Keiich INOUE, Kanji OTUKA, Mitsuho SUGIMOTO, Noriyuki MURAKAMI
Dept. of Farm Mechanization, Japan National Agriculture Research Center.
3-1-1 Kannondai ,Tsukuba-shi, Ibaragi-ken ,305 Japan
keich@narc.affrc.go.jp

Abstract

An autonomous tractor of land based vehicles of 75HP was developed using gyroscope and DGPS for field work and turning at a headland. Present position of tractor is estimated by a method of Kalman filter with the data of DGPS and optical fiber gyroscope . We evaluate also a method of steering control to follow a fixed line in the field using a mobile simulation model to estimate position of tractor after some time lag to operate. Some unknown characteristic coefficient of tractor i.e. wheel base on turning ,coefficient of turn radius ,coefficient of hydraulic time lag are estimated by a method of adaptive control. We made experiments of automatic tillage using this automobile tractor equipped a rotary machine and a liquid drug splayed in the field.

Key Words: tractor, autonomous vehicle, adaptive control, sensor fusion, gyro, GPS

1 Introduction

Autonomous ground tractor is very useful for ground farm work, such as plowing, seeding, fertilizing, etc. An operator dose not have to concentrate to track a line . He can concentrate to the operation of farm work. Internal sensors such as optical fiber gyro or speed meter is indispensable for autonomous vehicles to control the vehicles at real time with the data of attitude. However, this internal sensor system is not sufficient owning to its integrated position error. While ,GPS is very useful and easy to measure position. Since the advent of modern GPS receivers, Low-cost sensor is now available for measuring position and velocity of vehicles. We could get a very accurate code –differential DGPS of ±15cm accuracy .By using two sensors, the accuracy of position is improved with a method of filtering and sensor fusion. We made a experiment of automatic tillage using a auto mobile tractor equipped a rotary machine in the field.

2 Method

2.1 Vehicle Hardware and Control system

A 4-wheeel drive tractor of 75 PS was remodeled for automatic control vehicle. Steering, throttle ,tri-link(these are controlled through D/A converter),Go force and back and stop switch ,4 wheel drive and dual speed wheel drive mechanism, crutch, brake ,PTO(these are controlled through digital parallel I/O interface) are controlled by a micro computer. Front wheel was actuated using a modified Electro-hydraulic steering unit equipped parallel to the power steering system. The angle of the wheel was sensed with a potentiometer. The microprocessor converted serial commands from the control computer into pulse-width modulated ring signals which were sent to the power circuits that control the steering valves. Throttle and tri-link are also controlled through potentiometers .Crutch or Brake pedals is pulled by a wire connected to geared DC motor with constant torque keeper clutch mechanism. PTO lever is also pulled by a wire connected to a electro-magnetic crutch. Another switches are controlled by relay systems through parallel I/O interface board.

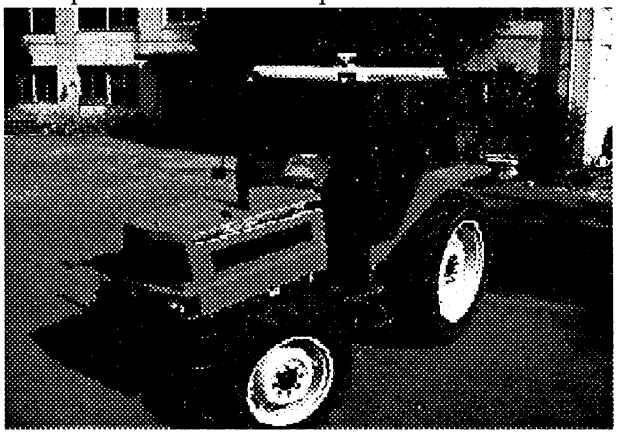


Photo .1 Autonomous tractor (75PS) mounted on DGPS and gyroscope

This tractor mounts an optical fiber gyro and a ultra sonic speed meter and DGPS . The optical fiber gyro is 0.3 degree accuracy and 3 degree drift per hour .The DGPS(Leica MX9400 GPS receiver) is 15cm accuracy and 1 Hz sampling rate. The data of DGPS is transferred through RS-232C interface to a data communication computer.

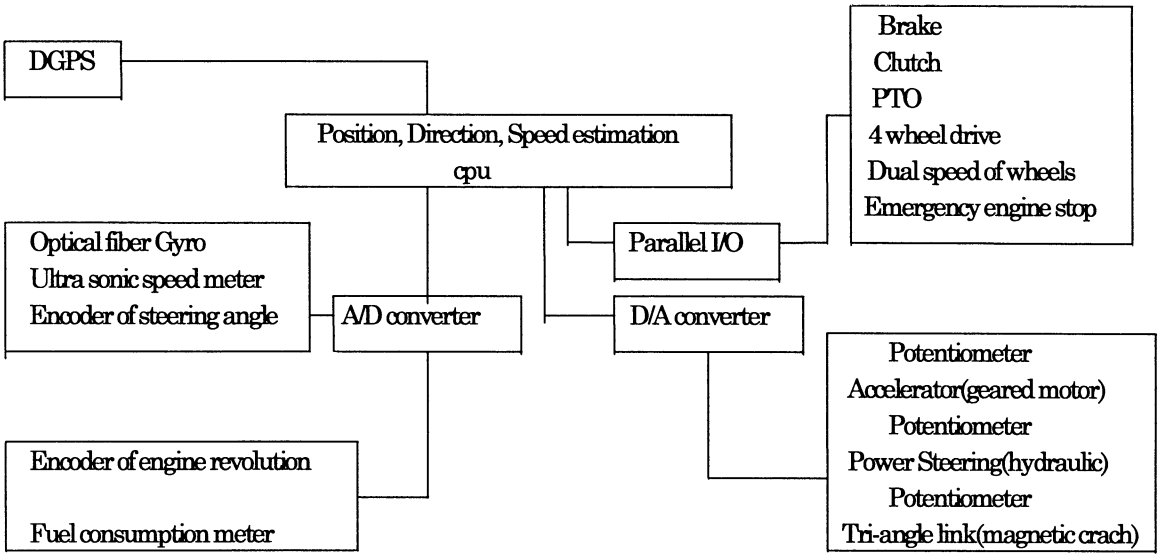


Fig. 2 Schematic diagram of the autonomous tractor

2.2 Vehicle modeling and Kalman filter process

A direction(θ) and center position(x;lateral position, y;nominal path) of tractor at a time $k\cdot\Delta t$ are expressed like these equations(1)-(4) using steering angle(ω) and velocity(v) of a tractor. The time interval Δt means machine cycle or pass time.

$$\theta_k = \theta_{k-1} + (v/r)\cdot\Delta t \qquad \theta_k = \theta'_k + \varepsilon_{\theta k} \tag{1}$$

$$r = (lw/\omega_k)\cdot(1+c\cdot v^2) \quad (\xi\text{; noise or drift, lw; wheel base c; coefficient of turn radius}) \tag{2}$$

$$v_k = v'_k + \varepsilon_{vk}$$

$$x_k = x_{k-1} + v_k\Delta t\sin(\theta_k + (v/r)\cdot\Delta t/2) \fallingdotseq x_{k-1} + v_k\Delta t\sin\theta_k = x_{k-1} + (v'_k + \varepsilon_{vk})\Delta t\sin(\theta'_k + \varepsilon_{\theta k})$$
$$= x_{k-1} + v'_k\Delta t\sin\theta'_k + \varepsilon_{vk}\Delta t\sin\theta'_k + \varepsilon_{\theta k} v'_k\Delta t\cos\theta'_k \tag{3}$$

$$y_k = y_{k-1} + v_k\Delta t\cos\theta_k = y_{k-1} + (v'_k + \varepsilon_{vk})\Delta t\cos(\theta_k + \varepsilon_{\theta k})$$
$$= y_{k-1} + v'_k\Delta t\cos\theta'_k + \varepsilon_{vk}\Delta t\cos\theta'_k - \varepsilon_{\theta k} v'_k\Delta t\sin\theta'_k \tag{4}$$

(3),(4) eq. are shown like these matrix using vector.

$$\begin{vmatrix} x_k \\ y_k \\ \varepsilon_{vk} \\ \varepsilon_{\theta k} \end{vmatrix} = \begin{vmatrix} 1 & 0 & \Delta t\sin\theta'_k & v'_k\Delta t\cos\theta'_k \\ 0 & 1 & \Delta t\cos\theta'_k & -v'_k\Delta t\sin\theta'_k \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} x_{k-1} \\ y_{k-1} \\ \varepsilon_{vk-1} \\ \varepsilon_{\theta k-1} \end{vmatrix} + \begin{vmatrix} v'_k\Delta t\sin\theta'_k \\ v'_k\Delta t\cos\theta'_k \\ 0 \\ 0 \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \\ \xi_{vk} \\ \xi_{\theta k} \end{vmatrix} \tag{5}$$

This is shown as follows

$$x_k = A_k x_{k-1} + b_k + u_k \tag{6}$$

Observation by GPS is shown as follows.

$$x_{gk}(i) = x_k(i-1)_{-d} + L\sin\theta_k = x_k(i-1)_{-d} + L\sin(\theta'_k(i)_{-d} + \varepsilon_{\theta k}(i))$$
$$= x_k(i-1)_{-d} + L\sin\theta'_k(i)_{-d} + \varepsilon_{\theta k}(i)L\cos\theta'_k(i)_{-d} \tag{7}$$

i; sampling number of GPS data L; differnce between center of tractor and GPS d;time delay of GPS.

$$\begin{vmatrix} x_{gk}(i) \\ y_{gk}(i) \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & L\cos\theta'_k(i)_{-d} \\ 0 & 1 & 0 & -L\sin\theta'_k(i)_{-d} \end{vmatrix} \begin{vmatrix} x_k(i-1)_{-d} \\ y_k(i-1)_{-d} \end{vmatrix} + \begin{vmatrix} L\sin\theta'_k(i)_{-d} \\ L\cos\theta'_k(i)_{-d} \end{vmatrix} + \begin{vmatrix} \xi_{gk}(i) \\ \xi_{gk}(i-1) \end{vmatrix} \tag{8}$$

This is shown as follows

$$x_{gk}(i) = H_k(i)_{-d}\, x_k(i)_{-d} + d_k(i)_{-d} + v_k(i) \tag{9}$$

When can get position data from GPS, kalman filter process is as follows.

$$x_k(i)_{-d} = x_k(i)_{-d} + P_i H_k(i)_{-d}T(R_i R_i T)^{-1}(x_{gk}(i) - H_k(i)_{-d}x_k(i)_{-d} - d_k(i)_{-d})$$
$$P_i = M_i - M_i H_k(i)_{-d}T(R_i R_i T + H_k(i)_{-d}TM_i H_k(i)_{-d}T)H_k(i)_{-d}M_i$$
$$M_i = (A_k(i)_{-d}\cdots A_k(i-1)_{-d}+1)\,P_{i-1}(A_k(i)_{-d}\cdots A_k(i-1)_{-d+1})T + \Sigma(A_k(i)_{-d}\cdots A_k(i)_{-d+1})b_k \tag{10}$$
$$x_k = (A_k\cdots A_k(i)_{-d+1})x_k(i)_{-d} + \Sigma(A_k\cdots A_k(i)_{-d+1})b_k$$

M_i; estimate error deviation matrix of sampling number (i-1)

P_i; estimate error deviation matrix of sampling number i

H_k; Hankel matrix

Dead recognizing

$$x_k = A_k x_{k-1} + b_k \quad (11)$$

2.3 Line tracking control

we want to control $\omega(k)$ so as to lead the tractor to follow a line i.e $\theta=0$, x=0 after some control time lag($\nu + |\omega i - \omega(k-1)|/\kappa$, ωi; objective steering angle, κ ;coefficient of steering move velocity, this time lag is mainly caused by hydraulic cylinder) of steering to operate. Generally, objective relations between θ' and x' after time lag τ is expressed as follows using eq.(1), (2) replace Δt to τ .

$$\theta'(k) = f(x'(k), v) \quad (12)$$

$$\omega(k) = \omega(k-1) + \mathrm{sign}(\omega i - \omega(k-1)) \cdot \kappa(\Delta t - \nu) \quad (\Delta t < \nu + |\omega i - \omega(k-1)|/\kappa) \quad (13)$$

$$= \omega i \quad (\Delta t > \nu + |\omega i - \omega(k-1)|/\kappa)$$

Using a simulation model for a tractor mobile trace ,after examining several functions f(x',v) ,determined

$$f(x', v) = -(1+3v) \cdot x' \quad \text{(when strait traveling, } |x'| < \pi/2 \cdot (1+3v)) \quad (14)$$

$$= \mathrm{ATN}(x' \cdot \ln(p)) \quad \text{(when adjusting to next ridge during turning, ax=p^y,p=0.7)} \quad (15)$$

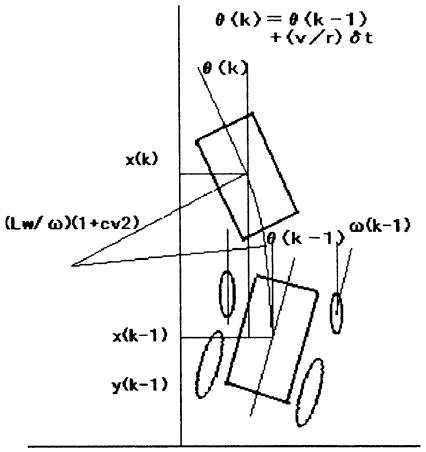


Fig.1 Vehicle model of movement

2.4 U-turns control

At the end of the field after traveling line tracking of fixed length, the tractor was controlled to turn as follows method. At first, tractor turns left at maximum angle of steering with dual speed wheel driving while the direction of the tractor below $-\theta$, and then next turns right at maximum angle also while the direction of the tractor below θ , turns left at maximum angle also while the direction >0. Next ,the tractor drives straight forth collecting direction and position considering the place of rotary same formers trace. We set w as 1.9, 2.2, 3.5, 5 m, and examined the width of result.

$$2r(1-\cos\theta) + 2vt\sin\theta = r - w/2 \quad (w < 2r) \quad (16)$$

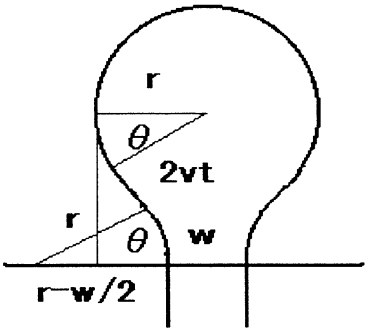


Fig.3 Method of U-turn

2.5 Experiment of rotary tillage by auto mobile tractor in the field

An auto mobile tractor of 75 PS , 4 wheel drive is developed at our laboratory . Using this tractor equipped with a rotary machine (2.2m wide), we made a experiment of automatic tillage of 100×160m field at a rate of 1m/s speed(engine revolution 1500rpm). At the end of the field ,calculate a objective steering angle and turned to enter next ridge(the relation eq.(4) and constants were previously measured). After directing next ridge, measure the position and direction by INS, GPS and geometric gyro and collect s the direction offset of the optical gyro output.

3 Result and Discussion.

3.1 Results of simulation analysis

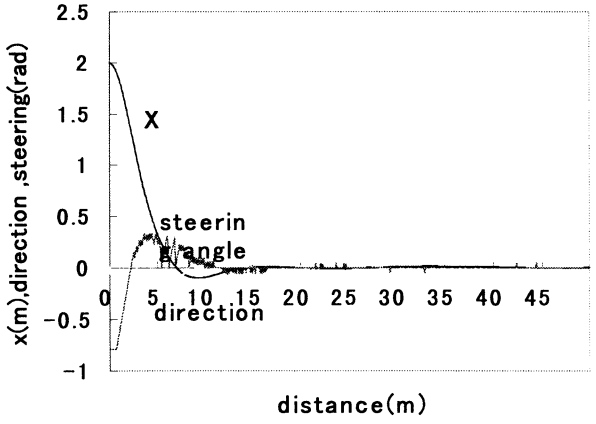


Fig. 4 A simulation of trace X, Steering angle , Direction

Using a simulation model of movement of vehicle, compared the trace of tractor according to equations (14),(15) on the condition of initial difference x=2m, tractor speed 1m/s,initial angle of steering $\pi/8$rad, τ =0.1sec, κ = 1.96 sec/rad . The result of this simulation is like figure 4.. First, steering angle is controlled minus angle to direct nominal line, next adjust plus angle to follow line. The method of equation (14) needs 10 m to focus with in 0.05m while eq.(15) needs 5m. But method of eq.(15) is seemed not adequate for line tracking because of its over response to deviation. This was the same result of experiment. Each trace is stable under the noise level of 4 % to sensors.

Fig. 4 A simulation of trace X, Steering angle , Direction

3.2 Experiment of rotary tillage .

There is no meander movements according to the method of forecasting a position and direction of tractor calculated from steering angle and hydraulic cylinder time lag. Each trace of the edge of the rotary machine was equally parallel and deference from the line are within 0.1m (50m travel)(Fig.5). The accuracy of moving to next ridged was with in 0.12 m(Fig .6, Table 1) . This accuracy is owing to a dual speed wheel drive system on 4 wheel drive on turning.

Photo. 2 Automatic working of rotary.

The position of the tractor which is calculated by the data of internal gyro and speed meter is similar to the data of GPS for sometime about 10 minutes. But, sometimes the direction of the tractor changed after 10 minutes according to the drift of gyro. So,. the sensor fusion algorithm of Kalman filter was effective to the autonomous vehicle. When there is some difference between the estimated position and observed ,then the estimated state of system is collected so as to fit to the observed output according to the Kalman filter process. The traces of the rotary were paralleled for more than 40 minutes (this is experiment time) . The Kalman filter for the sensor output was also effective to improve the accuracy of GPS. The tractor was controlled within 10 cm accuracy with the data of fiber optical gyro and DGPS whose accuracy was 15 cm .

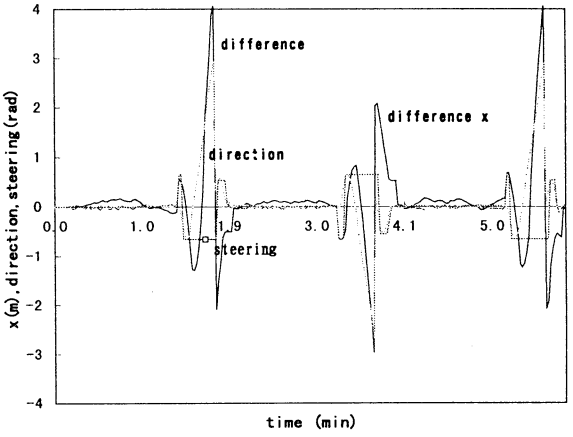


Fig. 5 difference x, direction , steering angle on tillage work..

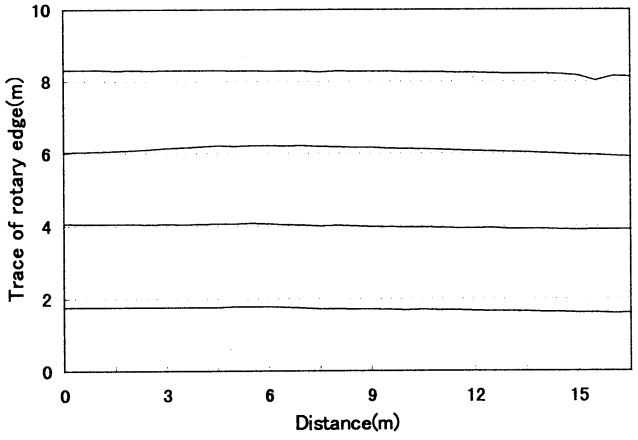


Fig .6 Trace of automatic rotary tillage (2.2 m work width)

Table 1 Accuracy of moving to next ridged by autonomous U-turn at several setting width

Unit: (m)

Setting	result	error(x)	error(y)
1.9	1.82	-0.08	+0.32
2.2	2.38	+0.18	-0.25
3.5	3.67	+0.17	+0.35
5.0	5.25	+0.25	+0.28

4 conclusion

The autonomous tractor equipped with fiber optical gyro and DGPS is developed. There is no meander movements according to the method of forecasting a position and direction of tractor calculated from steering angle and hydraulic cylinder time lag. Each trace of the edge of the rotary machine was equally parallel and deference from the line are within 0.1m (50m travel). The accuracy of moving to next ridged was with in 0.12 m. The sensor fusion algorithm of Kalman filter was effective to the autonomous vehicle. The traces of the rotary were paralleled for more than 40 minutes according to the use of both data of gyro and GPS.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Development of an autonomous tractor for forage production

A.Okado[1], M.Ishida[2], K.Imou[3], H.Takenaga[2], and N.Itokawa[1]

1)National Grassland Research Institute, Nishinasuno, Tochigi, 329-27 Japan
2)Faculty of Agriculture, Utsunomiya University, Utsunomiya, Tochigi, 321 Japan
3)Graduate School of Agricultural and Life Sciences, The University of Tokyo, Bunkyo-ku, Tokyo, 113 Japan

ABSTRACT

An autonomous control system and positioning method for an agricultural tractor are described.
In this article, we discuss an autonomous tractor we developed for forage production. The platform used for testing was a small, radio-controlled(RC) tractor. We modified the RC tractor for autonomous operations.

Our positioning method is based on a dead-reckoning calculation that uses a gyroscope and speed sensor. The gyroscope detects the vehicle direction. Vehicle position is determined by integration of the vehicle direction and speed from the starting point. To get accurate data, we make use of a fiber-optic gyroscope(FOG) and the ultrasonic Doppler speed sensor.
We assume that the trajectory during operation consists of a combination of straight lines. The autonomous tractor moves along a straight line from point to point. The computer controls steering to keep the tractor is within a 10cm width of deviation from the straight line.

This system has performed well in autonomous test operations, such as broadcasting of fertilizer or tedding on grassland. In this positioning system, vehicle position accuracy gets progressively worse with distance from the starting point. Therefore maximum possible distance has a upper limit. A more improved positioning system is needed for applications that require more distance or very precise control.

Keywords. *Autonomous Tractor, Dead-Reckoning, Forage Production Field*

INTRODUCTION

Many forage production fields are very large. And operating an agricultural tractor can be a hard and tedious task. Because speeds are very slow across large fields, control of a tractor requires continuous attention of the operator in a high noise and vibration environment. Dairy farmers often work long hours during the critical time of planting and harvesting. And many operations are usually limited to daylight hours. Therefore, it is necessary to develop an automatic agricultural vehicle.

Guidance system for agricultural vehicles is not a new idea[1-3]. But many of them needed crop rows or markers in the field for guidance. In grassland, there is no marker that can be used for vehicle guidance. It is difficult to detect accurate vehicle position. Therefore, It is necessary to develop another positioning method that does not need any marker.
In the past, some efforts have been made with this aim[4-8].

This paper describes a control system and positioning method for an agricultural tractor. A fiber-optic gyroscope and ultrasonic Doppler speed sensor were used to detect the position of the tractor. A control system using their sensors was implemented on the radio-controlled tractor and tested.

THE TEST PLATFORM

The platform used for testing was a Mitsubishi radio-controlled(RC) tractor RC-1 pictured in Figure 1.
The RC tractor has a 1.47m wheel base, a 14.7kW diesel engine, 12 speed transmission, and 4WD system. The top speed of the tractor is about 2 meters per second.

Figure 1 - Autonomous Tractor

We modified the RC tractor for autonomous operations. Steering, clutch, brake, engine speed... can be controlled by the Personal Computer on the tractor. The computer hardware is NEC PC-9800 architecture (Intel 486 microprocessor) that is popular in Japan.

The maximum steering angle is +/- 36 degrees. The steering angle is sensed from a potentiometer attached to the front wheels. To reduce the computing load, we developed a steering controller that performs as an interface between the computer and actuators. The computer sent the analog steering angle to a steering controller. A steering controller gets the actual steering angle from a potentiometer, compares both angles and controls steering actuators.

THE POSITIONING METHOD

As before, in forage production fields(especially grassland), there is no marker that can be used for vehicle guidance. Therefore, our positioning method is based on a dead-reckoning calculation that uses a gyroscope and speed sensor. The gyroscope detects the vehicle direction. The vehicle position is determined by integration of the vehicle direction and speed from the starting point. The following equations describe the vehicle position:

$$x_i = x_{i-1} + (s_i - s_{i-1})\sin\theta_{i-1}$$
$$y_i = y_{i-1} + (s_i - s_{i-1})\cos\theta_{i-1} \quad (1)$$

where

x, y	x-y coordinates of the vehicle position (m)
s	distance (m)
θ	vehicle direction (rad)

But, errors of measurement greatly influence accuracy of positioning because of its integration. To get accurate data, we make use of the ultrasonic Doppler speed sensor[9] and a fiber-optic gyroscope(FOG). The ultrasonic Doppler speed sensor has higher resolution more than other types of speed sensors. Angle measurement by a gyroscope generally is influenced by the 'drift'. In the case of drift, accuracy of the vehicle direction becomes progressively poorer. FOG resists the drift and has good reliability in a severe environment. Specifications of FOG are shown in Table 1. We decided that this gyroscope has enough accuracy.

One cause of error in angle measurement by a 1-axis gyroscope is tilt of the tractor[10]. But, from the result of experiments, the influence of tilt was very little on tested fields.

Table 1 - Specifications of FOG

type:	JAE JG-35FD
measurement range:	±180°
angular velocity range:	±200° /s
drift:	<±0.5° /h
linearity:	<±0.1%
angle resolution:	0.01°
frequency output:	20Hz
serial interface:	RS232C
operating temperature:	-10 to +50℃

VEHICLE CONTROL

The Personal Computer is used for vehicle guidance and control, as shown in Figure 2. The computer on the tractor gets data from sensors every 50 milliseconds, and calculates vehicle position.

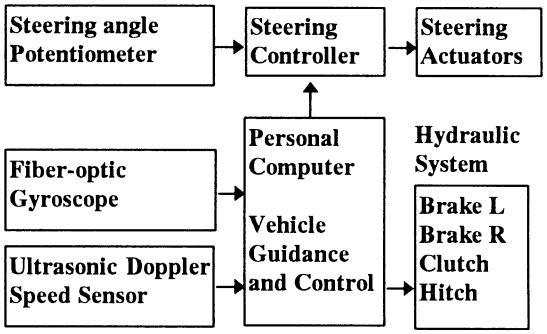


Figure 2- Guidance and Control System

We assume that the trajectory during operation consists of a combination of straight lines and turns. A vehicle control software was developed to run on such trajectory. The operator inputs the points of the path as a sequence of (x,y) points. The autonomous tractor moves along a straight line from point to point. The computer controls steering to keep the tractor within a 10cm width of deviation from the straight line, as shown in Figure 3. When the tractor reaches the end of a straight line, the computer reduces engine speed if necessary, and changes the heading of the tractor to run on the next straight line. In this case, vehicle steering angle is held maximum.

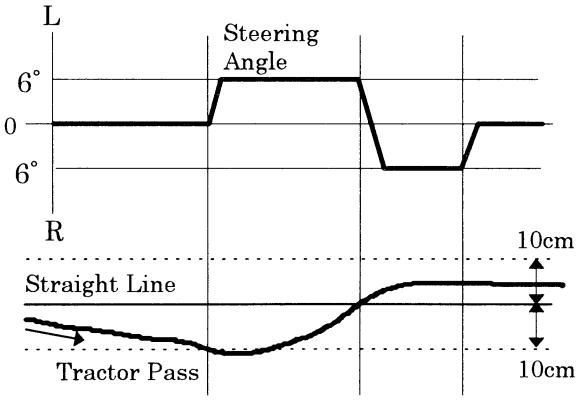
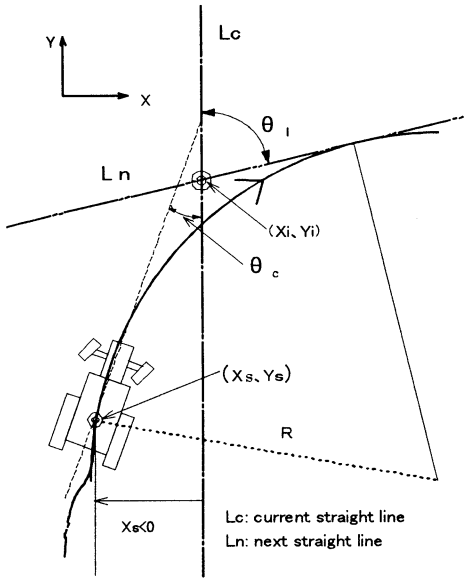


Figure 3 - Steering Control on a Straight Line

Figure 4 shows the algorithm for the decision of a turning point.



$$Y_s = Y_d + X_s \cot\theta_i + R\operatorname{cosec}\theta_i\{\cos(\theta_c - \theta_i) - 1\}$$

$$\left(Y_d = \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2} \right)$$

Figure 4 - Decision Algorithm for Turning Point (Xs,Ys)

FIELD TEST AND RESULTS

Figure 5 shows the result of typical autonomous test operation. This system has performed well in some types of autonomous operations, such as broadcasting of fertilizer or tedding(operation for speeding up drying grass) on grassland. For these operations, a high level of accuracy (centimeter level) is not required. In this positioning system, vehicle position accuracy gets progressively worse with distance from the starting point. The main cause is the error of speed sensing. The ultrasonic Doppler speed sensor is easily influenced by wind or atmospheric temperature. It is difficult to maintain the performance of ultrasonic Doppler speed sensor on grassland all the time.

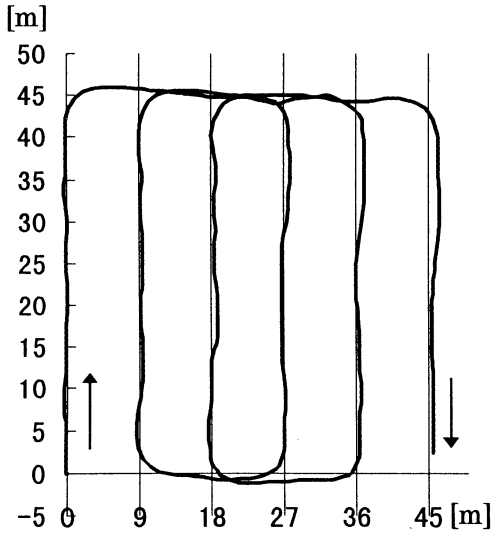


Figure 5 - Typical Autonomous Test Operation

Therefore, maximum possible distance has an upper limit. We decided that possible distance in broadcasting or tedding is about 600m. In 600m, positioning error is within 1-2 m. This is enough accuracy for these operations.

A more improved positioning system is needed for applications that require more distance or very precise control. We think that 10-20cm level of accuracy is required for forage production.

FUTURE DIRECTION

Future direction will be development of a reliable positioning system. Error in the dead-reckoning data needs to be compensated for by external positioning data that provides good absolute accuracy. As for the external positioning system, currently Real-time Kinematic GPS (RTK-GPS) must be a preferred choice. This can achieve centimeter level accuracy under good conditions. Though currently RTK-GPS is still expensive, the price is dropping quickly and is expected to be in realistic range in the near future.

CONCLUSION

(1) To research automatic agricultural operations, an autonomous tractor was developed. This tractor was based on the radio-controlled(RC) tractor. The RC tractor was modified for autonomous operations.

(2) The positioning method is based on a dead-reckoning calculation that uses a gyroscope and speed sensor. To get accurate data, we make use of a fiber-optic gyroscope and an ultrasonic Doppler speed sensor.

(3) We assume that the trajectory during operation consists of a combination of straight lines. The autonomous tractor moves along a straight line from point to point. When the tractor reaches the end of a straight line, the computer changes the heading of the tractor to run on the next straight line.

(4) This system has performed well in some types of autonomous operations. But maximum possible distance has an upper limit. A more improved positioning system is needed for applications that require more distance or very precise control.

ACKNOWLEDGMENTS

The authors would like to thank a group and individuals who made this research possible.
Mitsubishi Nouki LTD. provided the RC tractor for this research.
Shuji Nishimura, Yukinori Shibuya, Katsuhiko Tamaki, and Yoshifumi Honda were all extremely helpful.

REFERENCES

1. N.H.Brown et al: Image Analysis for Vision-Based Agricultural Vehicle Guidance, American Society for Agricultural Engineering Paper No.901653, 1990
2. Mohammad Moozoun et al: Image Automatic Guidance System for Agricultural Equipment, ASAE Paper No.901038, 1990
3. Tillet,N.D.: Automatic Guidance Sensors for Agricultural Field Machines, A Review, Journal of Agricultural Engineering Research, 1991(50), 167-187, 1991
4. Michael O'Connor, Gabriel Elkaim, Dr.Bradford Parkinson: Kinematic GPS for Closed-Loop Control of Farm and Construction Vehicles, Proceeding of ION GPS-95, Part 2 of 2
5. Nieminen,T.,Sampo,M.,Mononen,J: Development of a Control System for Autonomous Agricultural Vehicles, AgEng'96, Madrid, Spain 1996, PAPER 96A-094
6. C.H.Choi, D.C.Erbach, R.J.Smith: Navigational Tractor Guidance System, Transaction of the ASAE 33(3), 699-706, 1990
7. Ishii, K., H.Terao, N.NOGUCHI: Design and Fabrication of Model Vehicle toward Driverless Tractor, Journal of Hokkaido branch of the JSAM vol.36, 11-14, 1995 (in Japanese)
8. S.Nishimura, Y.Honda, K.Imou, H.Takenaga and N.Itokawa: Automatic Tractor Steering System for forage Harvesting, Journal of the Japanese Society of Agricultural Macinery 58(2), 31-38, 1996(in Japanese)
9. S.Hata, M.Takai, and K.Ohmiya: Development of Ground Speed Sensor for Agricultural Tractors, Journal of the JSAM 53(6), 3-10, 1991(in Japanese)
10. K.Imou, M. Ishida, A. Okado, H. Takenaga, Y.Shibuya, Y.Honda, and N.Itokawa: Yaw Angle Measurement on an Autonomous Tractor, Journal of the JSAM 59(4), 79-86, 1997(in Japanese)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB Ⅲ'98),
Oita, Japan, 19-21, January, 1998,

DEVELOPMENT OF WATERMELON HARVESTING ROBOT *'STORK'*

Mikio UMEDA, Susumu KUBOTA, Michihisa IIDA
Laboratory of Farm Machinery
Graduate School of Agriculture, Kyoto University
Sakyou-ku Kyoto 606-1, JAPAN

Abstract

Recently the production of heavy vegetables is decreasing in Japan, because the labor to harvest them is very hard. Thus the robot is required to harvest heavy vegetable easily. We have developed the robot *'STORK'* to harvest the watermelon. The position accuracy and the repeatability of the manipulator, the required vacuum and the allowance of the position error of the vacuum pad were tested in the field in this summer.

Keyword: Counterbalance, Roverval Mechanism, Truss
Vacuum pad

1. Introduction

Recently the production of heavy vegetables like watermelon is decreasing in Japan, because the labor to harvest them is very hard and most of farmers who cultivate heavy vegetable suffer the chronic lumbago. Thus the robot is required to harvest heavy vegetable easily. For this purpose, we developed the robot (the prototype I) with 5 D.O.F. manipulator actuated by the hydraulic cylinders as shown in Fig.1 [1] three years ago. The robot, however, was too heavy and too wide to travel in the field. The total mass of the robot was more than 450kg. Taking the situation of the Japanese agriculture into consideration, the Japanese agricultural machinery is made lighter and smaller than the European or the American one. If agricultural robots are intended to spread in Japan, they should be designed to be light and compact as well as the Japanese agricultural machinery. We, therefore, tried to develop the robot *'STORK'*, which is light and has the wide working space. This paper describes about the design concept and the performance of each element and the experimental result of the robot.

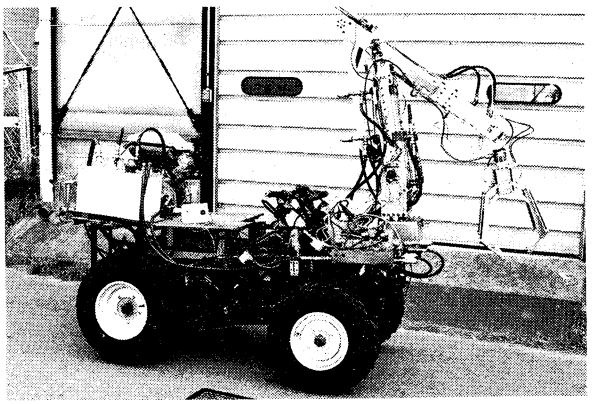


Fig.1 Watermelon harvesting robot (prototype I)

2. Required specification of the robot

Before the design of the robot, the typical cultivating methods of the watermelon were investigated in western Japan. The investigation shows that the robot can harvest most of the fruit if the following conditions are satisfied (1) to be able to travel on the furrow of a width of 0.5 m, (2) to be able to pick up the fruit at a distance of less than 2.8 m from the center of the furrow, (3) to be able to pick up the fruit of maximum 13 kg in mass and of 160 mm to 340 mm in diameter, (4) to be able to horizontally turn round manipulator at the angle of 120 degrees leftwards and rightwards respectively.

3. Design concept of manipulator

The structure of the manipulator is shown in Fig.2. The parallel four-bar linkage consists of Joint *A, B, C* and *E*. The stay including Joint *C* and *E* is fixed, so that Link *AB* is always vertically moved. Linkage *ABD* is compound and it behaves like a rigid body. Therefore, even if the vertical load acts on somewhere of the boom *AD*, the same vertical load acts on Joint *A*. That is to say, nevertheless the position of the fruit on the boom *AD* is varied, the loading point is always at Joint *A*. The load multiplied by arm length is the moment. Therefore in the case of the adoption of the parallel four-bar linkage, the left-handed moment in Fig.2 is changed by only variation of the gravitation of the fruit. We came up to this idea by ourselves. The mechanism, however, was invented by

French Roverval in 1669 in order to exactly measure the mass with a balance in spite of its position on a dish. The counterbalance produces righthanded moment in Fig.2, so that the torque to lift up the fruit may reduce. Further, by adopting the truss, the bending moments are not produced at the end of the links, so that the structural material becomes light.

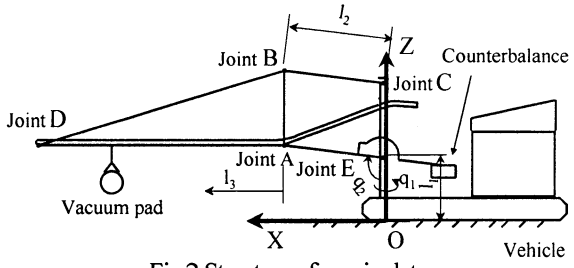


Fig.2 Structure of manipulator

4. Watermelon harvesting robot *'STORK'*

4.1 Manipulator

The developed robot is shown in Fig. 3. The shape of the robot is similar to 'stork' carrying a baby on a nursery tale, so that the robot is named *'STORK'*. The length of Link *AE* and *BC* are 0.8 m. The length of Link *AB* and *CE* are 0.6 m. The length of the boom is 2 m. The parallel linkage can turn upward at 45 degrees and downward at 10 degrees. The link, also, can turn right- and leftwards at 120 degrees respectively. AC servo-motors of the rated power 400 kW, of the rated torque 1.27 Nm and of the maximal speed 3000 rpm with an incremental encoder are used. Rotating speed are reduced by the cogged belts and pulleys. The hand, that is, the fruit on the boom is conveyed by rotating plastic chain. The reduction ratio are 168 for turn of ups and downs, 196 for turn of right and left, and 8.65 for rotation of the plastic chain respectively. Most of structural materials are made of steel for machinery

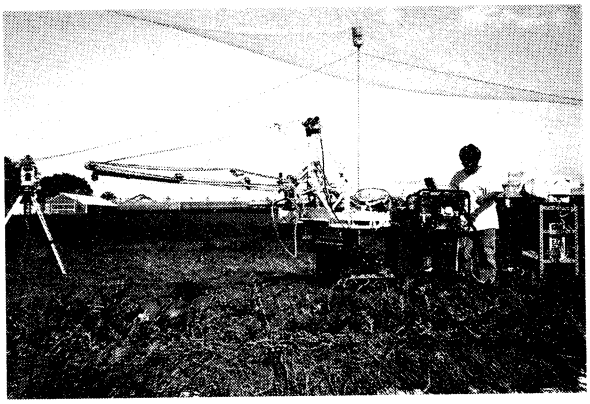


Fig.3 Watermelon harvesting robot *'STORK'*

instead of aluminum. Steel has higher and clear fatigue strength than aluminum, so that the manipulator under not a few fluctuation of stress can be rather made lighter using steel. The mass of the manipulator is less than 100 kg. The working space is 2.8 m. The lifting capacity is 150N.

4.2 Coordinate of the manipulator

The coordinate frame O-XYZ of the manipulator is set as shown in Fig.2. Let the point O be an intersecting point between the axis for the right-left turning of the manipulator and the ground. This point is an origin. The direction of moving forward is x-axis. The direction of vertical is z-axis. The relative position between the fruit and the robot is represented by the coordinates $[x,y,z]$. These are transformed to the parameters $[q_1,q_2,l_3]$, where $q_1$and q_2 are the angles of right-left turning and up-down turning respectively, and l_3 is the distance form Joint *A* to the vacuum pad. The transformation equations from $[x,y,z]$ to $[q_1,q_2,l_3]$ are

$q_1=\tan^{-1}(y/x)$

$q_2=\cos^{-1}((l_1-z)/l_2)$

$l_3=y/\sin q_1-l_2\sin q_2$

where l_1 is the distance between the ground and the axis for up-down turning and l_2 is the length of parallel link *AE*.

4.3 Hand

The robot has the vacuum pad as the hand. The pad has triple-bellows to fit with the shape of the fruit and the special sponge at the contact part with the fruit to prevent the air suction failure. To product the vacuum, ejector type vacuum pump and compressor are used. They can produce the vacuum of -20kPa to -86kPa by using the air pressure of 200kPa to 500kPa.

4.4 Vehicle for moving

As the vehicle for moving, the crawler vehicle, which was commercially sold as the self-propelled sprayer to scatter the chemical liquid in the green house, is used. It has the track layer of the outside width 0.4m. Its mass is 165kg and the maximum of the output power is 4.5kW. The total mass of the robot is less than 300kg.

4.5 Eye

Two small CCD cameras are installed near Joint *B* in Fig.2. To pick up the fruit, the relative position between the fruit and the manipulator is detected using the stereo image

method. Detecting the fruit has been studied by Dr. Tokuda and et. al. of Kobe University[2]. This study will be separately reported later.

5. Material and methods

5.1 Position accuracy and repeatability of the manipulator

The desired position [*x,y,z*] of the vacuum pad were inputted to the computer. After the pad moved, the error between the desired position and the position of the pad was measured in the coordinate frame O-XYZ. The errors of *x* and *y* were measured by a ray beam range finder and that of *z* was measured by a metal scale. The test was repeated four times.

5.2 Performance of the vacuum pad

The required vacuum to pick up the fruit and the allowance of the position error between the vacuum pad and the fruit were measured.

Air pressure supplied to the vacuum pump was gradually decreased using the relief valve, while the pad lifted up the fruit. The vacuum is nearly proportionally decreased to the air pressure. Then the fruit suddenly dropped at a certain vacuum. The vacuum and the pressure at that time were measured with a vacuum sensor and a strain gage type pressure gage respectively. The allowance of the position error was investigated using three fruits and the bowling ball. The ball may be sphere. Further, the effect of the vines and the leaves which were gotten between the pad and the fruit was tested. They made clearance between the pad and fruit. The vacuum is decreased by the inflow of the air through the clearance.

5.3 Field experiment

Field experiment was done at Takatsuki Experimental Farm of Kyoto University in the middle of July 1997. The vine was cut by manual before a fruit was harvested, because manual vine cutting is not so heavier labor comparing with the cost of automated vine cutter. The vehicle was moved near the fruit and stopped there. The position of the fruit was measured by scale or machine vision [2]. The data [*x,y,z*] was given to the computer.

Both angular velocities of right-left and up-down turning were 7 deg/s. The conveying speed was 80mm/s.

6. Results and discussion

6.1 Position accuracy and repeatability

The experimental results are shown in Table 1. The maximal position error was 60 mm. Also the maximal repeatable error was 36 mm. Since the manipulator was handmade, the manufacturing accuracy was not sufficient. Taking this fact into consideration, the position accuracy is not so bad, because repeatability is good. In the case of professional made or adding correction, the accuracy might be improved.

Table.1 Position accuracy and repeatability

	(1500,1500,700)				(1500,500,900)		
	x	y	z		x	y	z
First	20	-20	-2	First	-10	25	-2
Second	0	0	0	Second	-5	25	1
third	0	-5	-2	third	-2	20	0
fourth	0	-10	-1	fourth	0	20	-1
Average	5	-8.75	-1.25	Average	-4.25	22.5	-0.5
	(2000,1000,500)				(0,2600,550)		
	x	y	z		x	y	z
First	-20	-12	-15	First	-35	-35	17
Second	-12	17	5	Second	-45	-35	10
third	-22	17	5	third	-45	-35	15
fourth	-22	20	0	fourth	-45	-30	12
Average	-19	10.5	-1.25	Average	-42.5	-33.75	13.5
	(2000,1000,750)				(500,2000,800)		
	x	y	z		x	y	z
First	-21	30	0	First	0	25	10
Second	-20	28	0	Second	0	25	8
third	-20	25	-1	third	3	22	7
fourth	-20	22	-3	fourth	0	20	12
Average	-20.25	26.25	-1	Average	0.75	23	9.25

6.2 Performance of the vacuum pad

6.2.1 Required vacuum

The relation between the mass of fruit and vacuum is shown in Fig. 4. The relation is expressed by

$P_V = -4.43m + 8.12$

where m [kg] and P_V [kPa] represent the mass and the vacuum respectively. Vacuum of -20 to -86 kPa is produced by air pressure of 200 to 500 kPa. To pick up the fruit of 13kg, the vacuum of -49.5 kPa is required. To make this vacuum

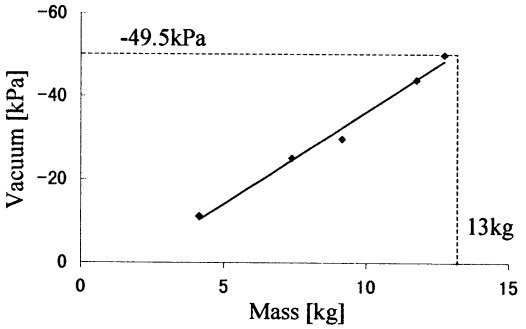


Fig.4 Relation between the mass and vacuum

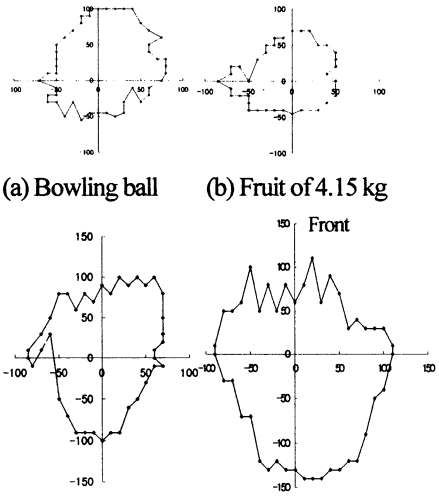
the air pressure of only 290 kPa is required. Thus this vacuum pad is practical.

6.2.2 Effect of vines and leaves

Preliminary experiment was done using a bowling ball of 6.6kg(15 pounds) and silicone rubber. The vacuum pad could lift up the ball, even though silicone at thickness of less than 1.5 mm was gotten between the pad and the ball. Watermelon leaf, however, has veins which is more than 1.5 mm in diameter at middle of leaf, so that, the pad could not lift up the fruit if the leaf was gotten. The vines and the leaves must be removed from the contact part of the pad.

6.2.3 Allowance of Position Error of vacuum pad

Experimental results are shown in Fig.5. In the case of the ball, the allowance of the position error was 80 mm in radius, but the center of allowance was shifted 20 mm, because of the effect of downwards movement of the manipulator. If the position error was less than 80 mm, the vacuum pad could pick up the ball. In the case of watermelon, the pad could not pick up the fruit at the point of –50 mm and –60 mm in x-axis as shown in Fig.5 (b) and (c) respectively. This was the influence of hollow at the calyx. As for Japanese watermelon, the diameter is nearly proportional to the mass, so that heavier fruit has the advantage of the position error. The allowance is 110 mm in radius in mass 11.3 kg as shown in Fig.5 (d).



(a) Bowling ball (b) Fruit of 4.15 kg

(c) Fruit of 8.0 kg (d) Fruit of 11.3kg

Fig.5 Allowance of position error

6.3 Field experiment

In fifteen times of experiments, ten fruits could be harvested but five fruits were failed at first. Then the success rate was 66.7 %. In two fruits, the positions of the vacuum pad were shifted more than 40 mm. In three fruits, the calyx acted as obstacle, so that the vacuum was not sufficient. In case of failure of positioning error, if the height of the pad was reset lower than 100 mm in z-axis, the fruits could be picked up. In the case of the calyx as obstacle, the calyx must be turned not to contact with the pad. Turning the calyx, the fruits could be picked up.

7. Conclusions

The watermelon harvesting robot '*STORK*' has been developed. To design *'STORK'*, lightness has been the top priority. The total mass of the robot is less than 300 kg. Taking the working space of 2.8 m and the lifitng capacity of 150 N into consideration, the light-weight manipulator may be materialized. The position error was maximal 60 mm. As the allowance of position error was, however, about 80 mm in radius, this error might be allowable. If the air pressure is set at 300 kPa, the fruit in mass 13 kg can be picked up. Then this adopting vacuum system is useful. The vines and the leaves over a fruit must be completely removed and the calyx must be turned not to contact with the pad before harvesting. In the case that the pad is shifted from the top of the fruit, the height of the pad is required to be adjusted.

[1] Namikawa, K., M. Umeda, M. Iida and M. Suguri : WATERMELON HARVESTING HYDRAULIC ROBOT, Proceedings of JICA-IPB 5TH Joint Seminar as International Conference on Engineering Application for the Development of Agriculture in the Asia and Pacific Region, pp. B233-B240 (INDONESIA), October,15,1992

[2] Tokuda, M., K. Namikawa, M. Suguri, M. Umeda and M. Iida, Development of Watermelon Harvesting Robot(I) —Machine Vision System for Watermelon Harvesting Robot, Proceedings of International Symposium on Automation and Robotics in Bioproduction and Processing, Vol.2, pp9-16, 1995 (Japan)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB Ⅲ'98),
Oita, Japan, 19-21, January, 1998,

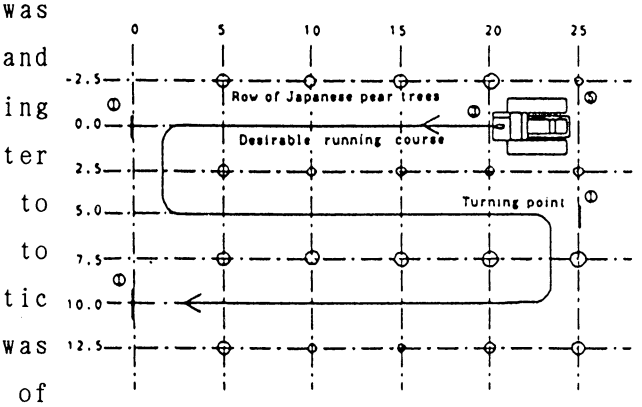
An image processing technique for automatic steering vehicle in Japanese pear orchard

Yoshiaki Misao
Department of Agricultural System Engineering
Tottori University
Tottori City, Tottori 680, Japan

1 Introduction

The 20th century pear is a very important crop in Tottori Prefecture in Japan. To maintain quality the crop is sprayed several times - 12 to 15 times every summer. Because of health effects associated with chemical application and scarcity in labor supply due to decreasing rural population, it is desirable to develop automatic steering sprayers. The author has been trying to develop an automatic steering vehicle for the Japanese pear crop.

In these studiest the video camera was separated from the personal computer which controlled image processing. To protect the computer from noise and vibration, it was located at a remote location and the vehicle was controlled using remote control. A transmitter was located on the vehicle to send the video image signal to the computer. An automatic stopping and starting device was also included in case of loss of signal due to yawing, pitching, or rolling of the vehicle.



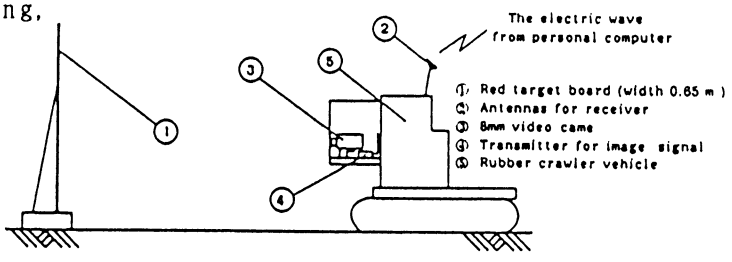


Fig. 1 Schematic diagram of automatic steering vehicle which utilizes image processing.

2 Experiment apparatus

Figure 1 shows the new system which consists of a personal computer with a colored image processing board and a rubber crawler

vichicle. The personal computer remotely controlled the clutch and brake of the vehicle by a radio control device. A red target board of 0.65 m width was used for guidance. This target board was placed 5 m from the row of pear trees. These pear trees were trellis-trained with trelliswires located 1.6 m above the ground surface. The pear trees were planted along rows spaced 5m apart and trees were 5 m apart along the row. The vehicle was steered by independently operating the clutch and brake. If the vehicle swerved the desired course, the personal computer detected the error using the video images and sent a digital proportional radio control signal to correct the error. The vehicle has provision to mount liquid sprayer tank.

This system consisted of automatic vehicle steering system and image processing system. The video camera took pictures which included the red target board. This picture was converted to electronic waves by the transmitter and sent to the monitor for image processing. An antenna on the monitor received the signal and sent it to the image processing board. The image processing board divided this signal into 256 X 256 pixels per one screen and separated the pixels into R(red), G(green), and B(blue) intensities and produced a hexadecimal number. The personal computer (NEC9801UV2) stored the hexadecimal numbers in the memory.

The personal computer detected the red target board by judging the brightness level and estimating the distance between the video camera and the red target board and the offset from the center line.

Finally the personal computer controlled the vehicle based on these information. Note that there was no operator on the vehicle. The operator watched the personal computer monitor to keep an eye on the performance but did not control the vehicle.

The vehicle was automatically started, steered, and stopped.

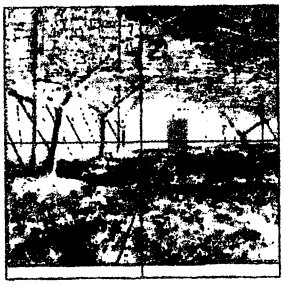


Fig.2 Photograph displaying pixel over 32 (half) brightness of RGB

3 Detecting the position of the red target board

Each pixel was analyzed for its color (Red, Green, and Blue) and was divided into 64 brightness levels as stated before. Figure 2 shows the image classification into three colors

(Red, Green, and Blue) over 32 brightness level. This picture shows the red target board at a distance of 3 m from the turning point in the Japanese pear orchard. The distance from video camera to the red target board is 10 m. The red target board is on the right side of the center on the CRT-screen.

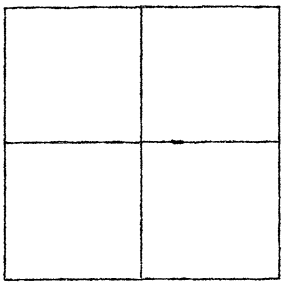


Fig. 3 Photograph displaying pixel 64 (full) brightness of red component.

In this picture, you can see the trunks and the leaves of Japanese pear trees and the weeds. Although weeds were actually green, they were shown in red color because of half the brightness level scale used. As human beings are superior in pattern recognition than computer, they can detect the red target board at low brightness level and other difficult conditions.

The personal computer can not detect the position of the target board because of lots of noise. Figure 3 shows a copy of the picture using 64 (full) brightness levels. Only red component is displayed (no green or blue component) in this figure.

This figure shows only the red target board. All other noises have been eliminated. The human beings can find out clearly the position of target. But the personal computer can not detect it out from only this picture. The author divided screen into 256 divisions vertically and counted red pixels in each column.

The relationship between the distance from video camera to the red target board and width of the red components at 64 (full) brighness levels can be represented by the following empirical equations:

$$D = 128.84\ P^{-0.81}$$

Where D is distance from video

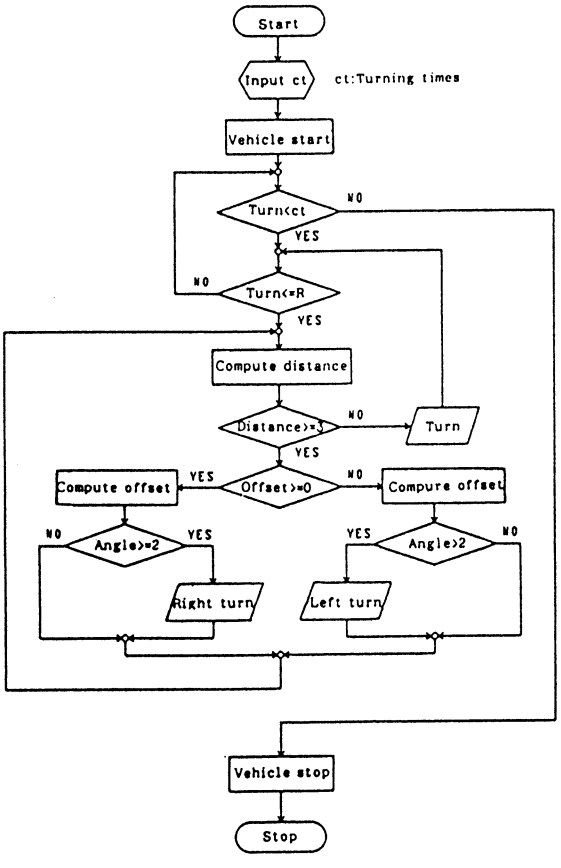


Fig. 4 Flowchart for automatic steering system.

camera to the red target board (m), and P is width of red components (pixel).

4 Results and discussion

Figure 4 shows the flow chart of the control program utilized in this study. This program was written in Turbo-C using MS-DOS. At the begg ing of this program the author initialized image processing and I/O board, and set the number of turning the vehicle had to make. The turning time was initialized by inputting reasonable value. The computer kept track of the number of turning mode. When it arrives near the target board (3 m from it), it automatically steers the vehicle. If it is the first, third, fifth or any odd turn, vehicle turns left. On the other hand, if it is second, forth or any other even turn, it turns right. If the number of the particular turn exceeds the input value, vehicle automatically stop. When the vehicle is away have the red target board, it goes straight until it is three meters from the board. The program computes the distance of the red target board from the width of the red strips, the offset from direction of vehicle to camera center, judges the turning direction from the sign of the offset, and computes the turning angle from offset angle. The program computes the operating times of the brake and clutch on the vehicle. Then the computer turns on the brake and clutch for the computed duration. Upon completing the turn, the program returns to the routine which takes images and the sequence of operation starts all over. As a result, our vehicle can run straight 20 m and turn into the next row without an operator.

5 summary

This paper describes the development of an automatic steering sprayer which utilized an image processing technique for self-guidance on the Japanese pear orchard in Tottori Prefecture, Japan. The vehicle was equipped with a video camera and an image signal transmitter. The computer with an image processing board which detects vehicle position with respect to a target red board was not mounted on the vehicle but remotely located which protects it from vibration and the noise. Our experiments show that vehicle is able to run straight 20 m and turn into the next row without an operator.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Autonomous vehicle guidance system in paddy field

Y. Nagasaka, R. Otani, K. Shigeta, K. Taniwaki
Dept. of Farm Mechanization,
National Agriculture Research Center,
Tsukuba 305, Japan

Abstract

An automated steering system for operation in paddy fields has been developed. The objective of this study is to develop an automated operation system to make operation in paddy fields safer and more comfortable. In this study, RTKGPS and FOG were used to control a rice transplanter. GPS data output was delayed because the GPS rover station had to communicate with the reference station by wireless modem which took about a few seconds. Therefore the real time position of the rice transplanter was estimated by integrated speed and yaw angle. The results of experiments in a paddy field showed the rice transplanter could be controlled well. When the rice transplanter traveled 100m, the deviation from the straight line was less than 8cm. ***Keywords*** : automated operation, paddy fields, RTKGPS, FOG

1 Introduction

Recently in Japan, there is a trend to consolidate and enlarge paddy fields. However, while the number of operators is decreasing, the work of operators is increasing. Automated operations can reduce the working time and release operations from the vibration and noise of vehicles.

In order to realize automated operation, it is necessary to locate the position and direction of the vehicle in the program and to drive straight ahead with high accuracy. In paddy fields, the inter-row spacing of plants is about 30cm, so the vehicles must be driven to maintain a rational accuracy. However, unevenness of the ground and side slip of the wheel distort the vehicle direction. So compared with driving on solid terrain, it is difficult to drive a vehicle straight in paddy fields. The precise position and the direction of the vehicle must be located at short intervals. In recent research, the yaw rate sensor [1], the combination of a fiber optic gyro (FOG) sensor and a velocity sensor [3], an optical wave range finder [2][4] and a laser beam [4] were used to locate the vehicle position. The authors used real time kinematic GPS (RTKGPS) to locate the precise position and FOG to get the posture of the rice transplanter. If one GPS reference station is built in a specific area and rover stations are set on the vehicles, it is easy to make an automated operation system. This paper reports about automated rice transplanting in paddy fields using the RTKGPS and FOG. This research was collaborated with Trimble Japan K. K.

2 Materials

This automated operating system consists of a GPS reference station and a 5-row rice transplamter as a rover station. Figure 1 shows the rice transplanter modified to be controlled by a computer.

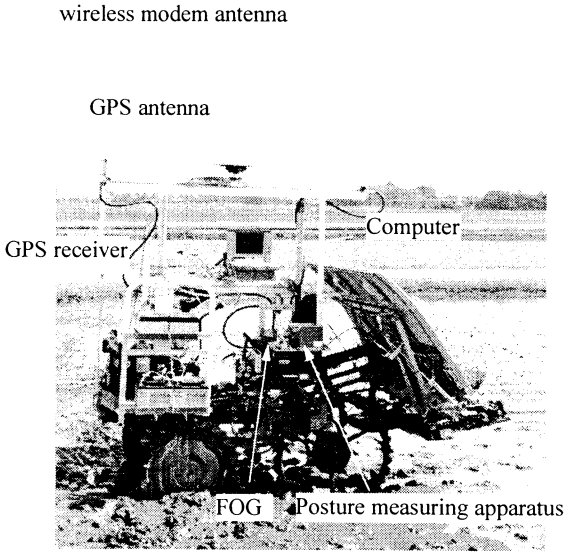


Figure 1 The rice transplanter

A RTKGPS is used to get the position of the rice transplanter. It has ± 1cm precision. For the communications between the reference station and the rover station, 5mW output wireless modems are used. The baud rate was set to 9600bps. A FOG is used to measure the yaw angle and a posture measuring apparatus comprised of 3 FOGs and 3 accelerometers is used to measure the roll and pitch angle of the rice transplanter.

The rice transplanter has a hydraulic pump, so the steering, clutch, brake and operating machine up-down are controlled by hydraulic valves. Figure 2 shows the

hydraulic circuit. The engine throttle, CVT throttle, PTO clutch and direction control gear are controlled by DC-motors.

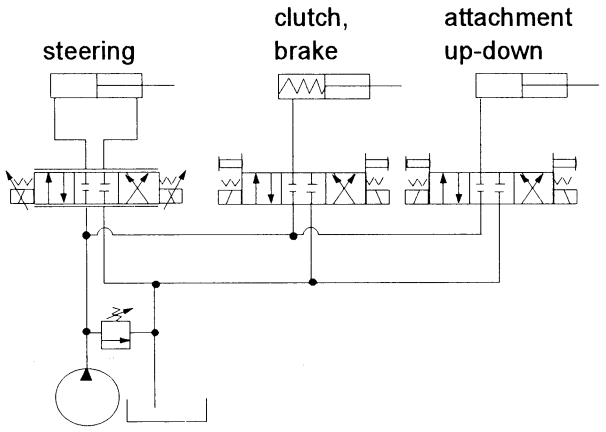


Figure 2 Hydraulic circuit

The GPS data is transferred to a computer through RS232C interface at 1Hz sampling rate while other data are taken by the analog to digital converter board at 25Hz. And control data is transferred to hydraulic valves and DC-motor controller by analog to digital converter board and digital I/O board.

3 Correction of the inclination

When a rice transplanter travels in a paddy field, the roll and pitch angle is about 3 degrees. In this system, the GPS antenna is fixed on the rice transplanter. Assuming the antenna height to be 2m, the horizontal distance between the top of the antenna and the bottom is 10.5cm when the roll angle is 3 degrees. The GPS data is the position of the antenna top, so to locate the precise position of the antenna bottom, which is the center of gravity of the rice transplanter, the influence of the roll and pitch must be corrected.

The space coordinates xyz on the rice transplanter and the plane coordinates XY on which the rice transplanter travels are estimated. The desired path is defined as the straight line drawn between the starting point and the target point on XY. It is expressed as follows.

$$Y=kX+C \qquad (1)$$

The bottom of the GPS antenna O_{xy} is on the origin of xyz. The orthographic projection P_{XY} on XY of the antenna top P_{xyz} on xyz is expressed as $P_{XY}(p_1,q_1)$. $O_{xy}(0,0,0)$ on xy can be expressed as $O_{XY}(p_0,q_0)$ on XY. The GPS antenna height is h. When the roll angle of the rice transplanter is θ and the pitch angle is ϕ, p_0 and q_0 are expressed as follows.

$$p_0 = p_1 - \frac{k}{\sqrt{1+k^2}}\left(h\sin\theta + \frac{1}{k}\sin\varphi\right)$$
$$q_0 = q_1 - \frac{k}{\sqrt{1+k^2}}\left(h\sin\varphi + \frac{1}{k}\sin\theta\right) \qquad (2)$$

By this equation, the precise position of the antenna bottom could be located.

Figure 3 shows the desired path, the yaw angle ϕ on the coordinates xy and its orthographic projection ϕ' on XY. When the roll angle is θ, the pitch angle ϕ, ϕ' is expressed as follows.

$$\cos\psi' = \frac{1}{\sqrt{1+\tan^2\psi\dfrac{\cos^2\theta}{\cos^2\varphi}}} \qquad (3)$$

The linear function (1) is given by the desired straight path and the yaw angle ϕ'. So by measuring the roll, the pitch and the yaw angle and by giving the antenna height and the target point, the precise position of the rice transplanter can be located.

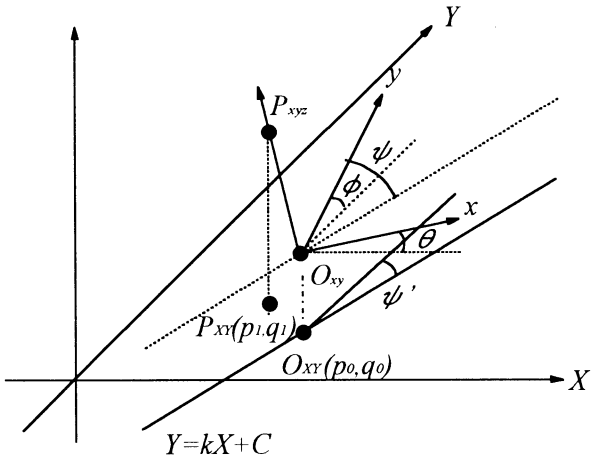


Figure 3 The desired straight line, antenna top P, yaw, roll and pitch angle

4 Estimation of the real time position and a control method

The GPS rover station has to communicate with the reference station to get the reference data over a radio link. In the 1Hz mode, the rover receiver requires synchronized GPS measurement data from the reference station once per second. The solution is with high accuracy but the data buffer capacity of the radio link dictates the transmission delay time. In this case, it is 2.8 seconds. So the real time position has to be estimated.

To control the rice transplanter along the desired straight line, the deviation from the line, d, must be estimated. At n seconds after starting, the deviation given by GPS data is d_{nGPS}. This is the data at 2.8 seconds before. When the starting point is $A_s(a_s, b_s)$, and the target point is $A_t(a_t, b_t)$, the point of the rice transplanter is $P(p_0, q_0)$, d_{nGPS} is expressed as follows.

$$d_{nGPS} = \frac{(b_t - b_s)(a_s - p_0) - (a_t - a_s)(b_s - q_0)}{\sqrt{(a_t - a_s)^2 + (b_t - b_s)^2}} \qquad (4)$$

When the traveling speed is v and the yaw angle is $\phi(t)$, the deviation d_t for 2.8 seconds is given by following equation.

$$d_t = \int_0^{2.8} v \sin\psi\ (t)dt \qquad (5)$$

So the d_n, the deviation n seconds after starting, is expressed as follows.

$$d_n = d_{nGPS} + \int_{n-2.8}^{n} v \sin\psi(t)dt \qquad (6)$$

By the expression (4,4), the real time deviation from the desired straight line is estimated.

The rice transplanter is driven along the target line. When the deviation from the target line is more than 5cm, the steering is controlled to set the yaw angle to 5 degrees to get back close to the line. When it is from 2.5cm to 5cm, the yaw angle is set to 3 degrees. When it is under 2.5cm, the yaw angle is set to 0 degree.

The steering angle δ is given by following expression, in which ϕ_0 is the aiming yaw angle and ϕ is the yaw angle.

$$\delta = K_p(\psi_0 - \psi) \qquad (7)$$

K_p is a constant experimental data.

5 Results and discussions

The experiment was conducted 3 days after puddling of the field.

Figure 4(a) shows the deviation from the desired straight line and the change of the yaw angle when the rice transplanter traveled 80m at 0.4m/s. At this speed, the motion is maintained stability. The yaw angle changed no more than 3 degrees. From 70 seconds to 120 seconds, the deviation was kept about 8cm. It was thought the steering was moved to change the direction, but the paddy terrain prevented from changing. Figure 4(b) shows the difference between estimated real time deviation and measured deviation. In this speed, the estimation has less than 4cm precision.

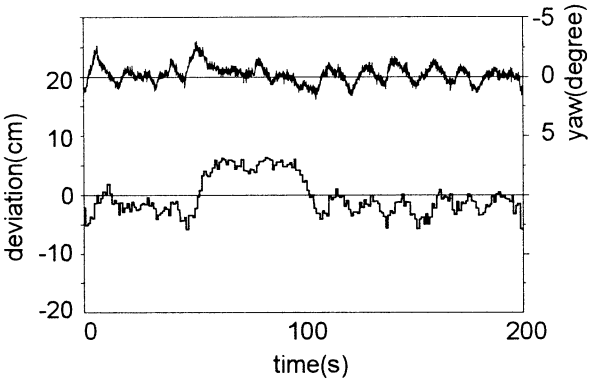


Figure 4(a) The deviation from the desired straight line and change of the yaw angle at 0.4m/s

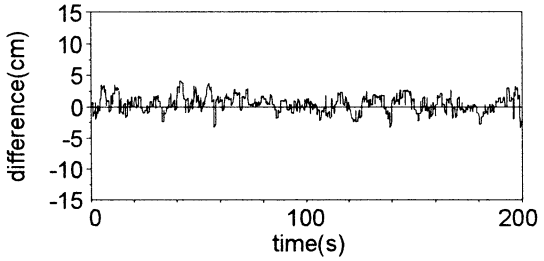


Figure 4(b) The difference between estimated real time deviation and measured deviation at 0.4m/s.

Figure 5(a) shows the deviation and the yaw angle when the rice transplanter traveled 100m at 0.8m/s. The deviation is about 15 cm and rice transplanter traveled oscillatory. It is thought because the motion of steering is delayed. Figure 5(b) shows difference between estimated real time deviation and measured deviation. It is less than 11cm. It was thought the difference caused by the side slip of the rice transplanter. Because a fiber optical gyro sensor sensed only the yaw angle, it could not sense the motion without changing the yaw angle.

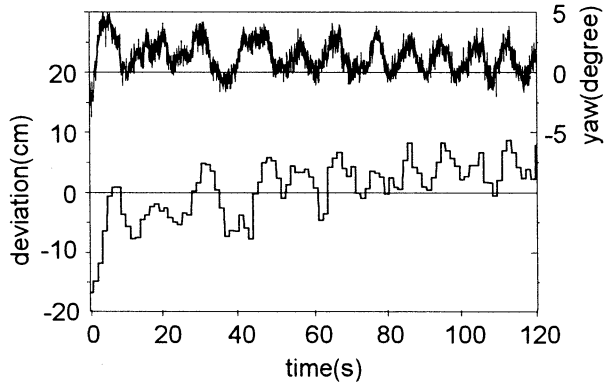


Figure 5(a) The deviation from the desired straight line and change of the yaw angle at 0.8m/s

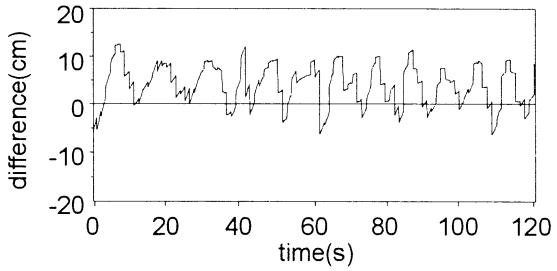


Figure 5(b) The difference between estimated real time deviation and measured deviation at 0.8m/s.

It is thought the speed is higher, the side slip may be larger. And the yaw angle 0 line was not perfectly agreement with the target line. If the speed is higher, the estimated distance is longer, So the error may be larger.

To improve the real time position precision, the delay of GPS data output must be shortened.

5 Conclusion

In this study, a rice transplanter was driven automatically in a paddy field with a fiber optical gyro sensor and real time kinematic GPS. In this experiment, the real time position estimated by a fiber optical gyro sensor and posture measuring apparatus had an error less than 4cm at speed 0.4m/s and less than 11cm at speed 0.8m/s. When the rice transplanter traveled 100m in a paddy field, the deviation from the target line was less than 8cm at 0.4m/s and less than 15cm at 0.8m/s.

References

[1] Nonami K, Komatsu M, Higuchi H, Nakao S Adachi K(1994) Studies on automatic traveling control of riding-type rice transplanter (Part 2).(in Japanese) Journal of the Japanese society of agricultural machinery, 56(3), pp77-84, 1994.

[2] Yukumoto O, Matsuo Y(1995) Research on autonomous land vehicle for agriculture. Proceedings of International Symposium on Automation and Robotics in Bioproduction and Processing., Vol. I , pp41-48, 1995.

[3] Nishimura S, Honda Y, Imou K, Takenaga H, Itokawa, N(1996) Automatic Tractor Steering System for Forage Harvesting.(in Japanese) In Journal of the Japanese society of agricultural machinery, 58(2), pp.31-38,1996.

[4] Yoshida J(1997) A study on the automatic farm machine system for rice. Crop Engineering system(in Japanese) Laboratory, Inc. 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Measurement of the vibration characteristic with image processing
-- Measurement of natural frequency with HSI color space --

Hiroshi TAKEDA, Fac. of Agriculture, Tottori University, Tottori 680 Japan
Kiyoshi NAMIKAWA, Fac. of Agriculture, Kyoto University, Kyoto 606-01,Japan
Mikio UMEDA and Kyoto University, Kyoto 606-01,Japan
Masami IWASAKI Fac. of Agriculture, Tottori University, Tottori 680 Japan

Abstract

It is effective to analyze characteristics of object with many branched system, and it was reported that characteristics of plant could be analyzed with transmit matrix method. But it is necessary to find equivalent damping coefficient by experiment. And characteristics of the plant are differing with the shape of the plant or the size of leaves in the case of the same kind of the plant. So it is effective to find characteristics of plant not only with analysis but also measurement when you harvest by vibration or you train or pruning considering prevention falling by strong wind. In this study, we videotaped vibrating plant by vibrator with CCD camera, tried to extract limb and branch by image processing with HSI color space, and measured displacement of specific point, and found natural frequency. The measured result with the image processing was compared to the one with the laser displacement sensors. And It was proved that the measurement with the image processing is available for measuring the vibration characteristic of the plant.

[Keywords] HSI color space, Image processing, Natural frequency

1 Introduction

It is effective to analyze characteristics of object with many branched system, and it was reported that characteristics of plant could be analyzed with transmit matrix method. But it is necessary to find the equivalent damping coefficient by not only the calculating but also the experiment. And characteristics of the plant are differing with the shape of the plant or the size of leaves in the case of the same kind of the plant. So it is effective to find the characteristics of the plant not only with analysis but also the measurement in the case of the vibration harvesting or training or pruning considering prevention falling by strong wind.

There are many studies to measure the vibrating characteristic, for example, with the strain gages, stroboscopic photography and the laser displacement sensors. But, those methods have some problems. The first problem is that it is important to measure the vibrating characteristic under the noncontact condition. The second one is that it is impossible to the displacements of the plural points of the plants. So we have developed the method of measuring the vibration characteristic of the plant by the image processing with HSI model to extract the limbs and branches.

2. Method and equipment

Figure 1 shows the experimental equipment. This experiment was held in the laboratory in Kyoto University. The steel model or the plant is attached to the hydraulic vibrator. Those objects were vibrated in 20 seconds. The vibrating frequency was 0.1 to 10 Hz frequency. And the displacements of the tip and the bottom of the objects were measured with image processing and the laser displacement sensor (KEYENCE LB-45 for the bottom, LB-1000 for the tip)

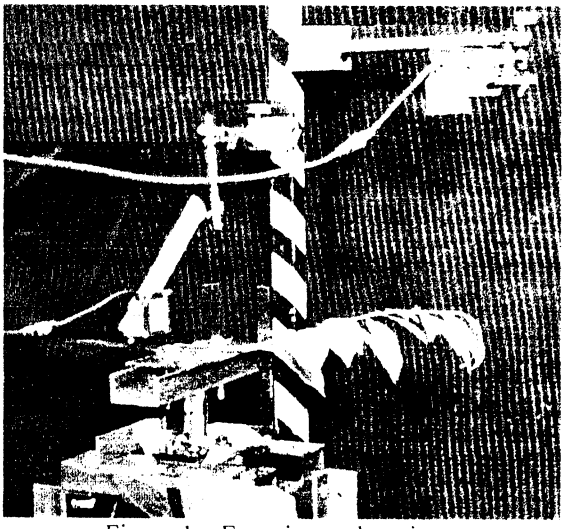


Figure 1 Experimental equipment

The steel model was consisting of the limb and the branch. The steel model was painted in light blue to make easy be extracted with image processing by extracting the brightness pixels. And this model was made of SS400.

The length between the object and the CCD camera (TOSHIBA IK-C40) with 12mm focal lens was 300 mm. So the resolution was 2 mm/pixel.

The object for plant was Ligustrum lucidum Ait.. It is important to measure the natural frequency of branch for measuring the whole of tree.

The 900-g weight was attached to the tip of the steel model for lowering the natural frequency of the object.

The image recorded by the CCD camera was captured to PC (IBM Pentium 133 MHz) with the image capture board (Canopus IMAGEPAQ-AT). And the image processing was done in real-time. The sampling rate was 66 ms in the image processing and 10 ms with the laser displacement sensor.

At first, the vibration characteristic of the steel model was measured and made sure that this method was effective.

After capturing the images of the plant, the displacement of the tip of the plant and that of the bottom were measured. And the gain and the natural frequency of the plant were calculated from those displacements.

3. Image processing for the steel model

Figure 2 shows the algorithm for the image processing.

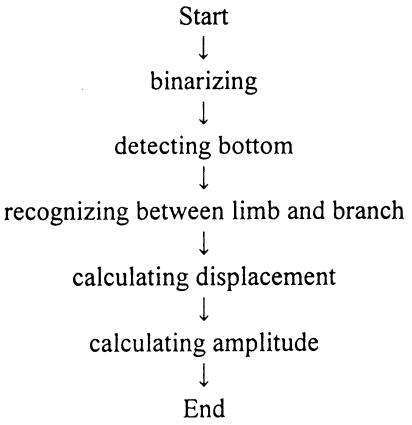


Figure 2 Flowchart of image processing

After capturing the image, the algorithm binarizes the image by extracting the bright pixels, detect the bottom point of the object by finding the rectangular area which is more than 10 in x direction and 3 pixels in y direction from the pixel in the tip of left to that in the bottom of right. At the divergence point, this algorithm compares the angle of two directions and decides which is the limb. This part of process will be held repeatedly until finding the tip of the object. After detecting the tip of the object, the displacement of the tip and bottom will be calculated and the amplitude will be calculated.

Figure 3 shows the result of measuring the displacement of the bottom and the tip of the object in the case of 1.6 Hz vibration.

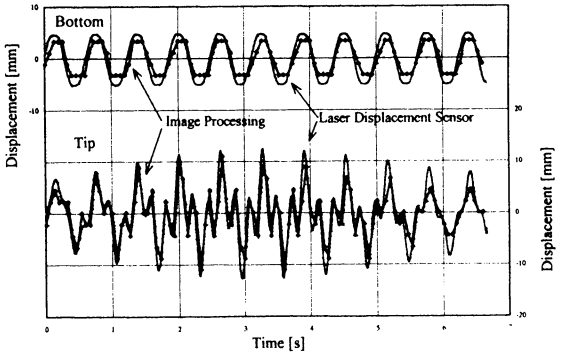


Figure 3 Result for steel model (1.6Hz)

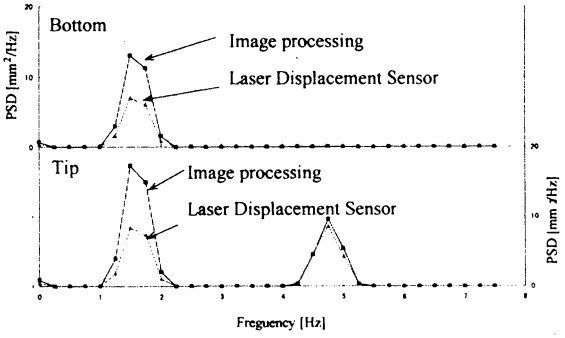


Figure 4 PSD for steel model

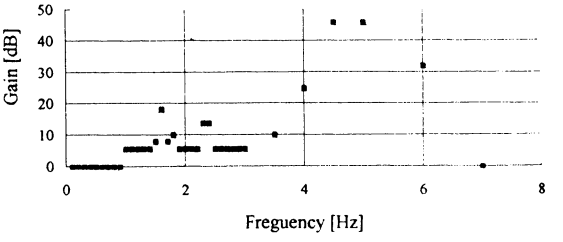


Figure 5 Gain for steel model

Figure 4 shows the power spectral density of the steel model with the image processing or the laser displacement sensors. Figure 5 show the gain measured with the image processing. From these figures, you can see that measuring the displacement by the image

processing is in accordance with measuring the displacement by the laser displacement sensor. Figure 6 shows the gain of the steel model. In this figure, there are two high value points (1.4 and 4.0 Hz), which are the natural frequencies of the steel model.

So, this method is available to measure the vibrating characteristic of the steel model.

4. Image processing for Ligustrum lucidum Ait.

Figure 6 shows the HSI (Hue, Saturation and Intensity) color space. If the intensity is less than threshold, the color of that pixel will be regarded as black, and if the intensity is more than threshold and the saturation is less than the threshold, the color of that pixel will be regarded as white. And, if the intensity is more than threshold and the saturation are more than threshold, the value of the saturation will decide the color of that pixel as red, yellow, green, cyan, blue and magenta.

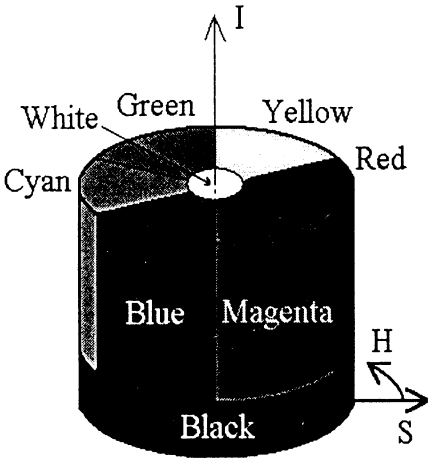


Figure 6 HSI color space

Figure 7 and 8 shows the captured image of the plant and the processed image of that. From these images, you can see that the area of the limb and the branch could be clearly extracted.

Figure 9 shows the result of measuring the displacement of the bottom and the tip of the object in the case of 2.2 Hz vibration.

Figure 10 shows the power spectral density of the plant with the image processing or the laser displacement sensors. Figure 11 shows the gain of the plant. In this figure, there are high value points, which are the natural frequencies of the steel model. So the natural frequency of the plant was 2.25-Hz and 4.5-Hz.

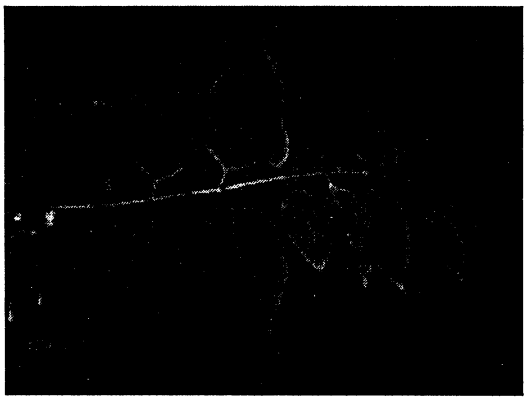


Figure 7 Captured image

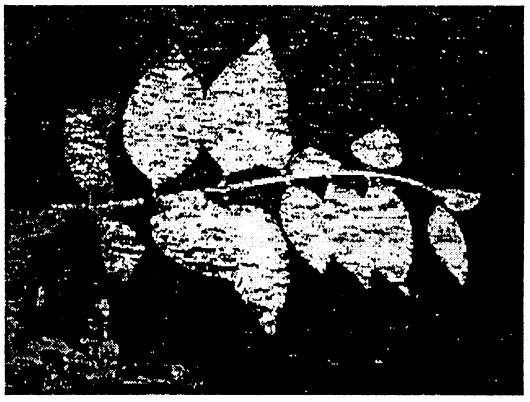


Figure 8 Processed image

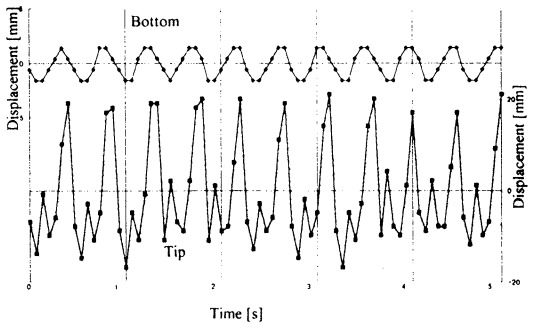


Figure 9 Displacement for plant

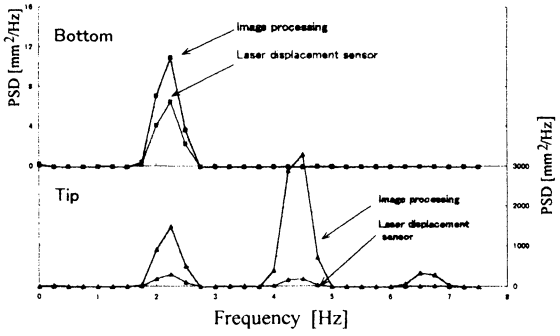


Figure 10 PSD for plant

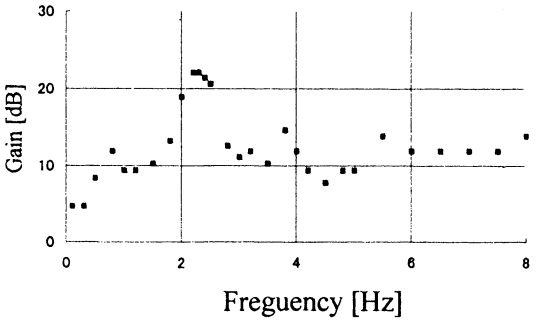


Figure 11 Gain for plant

So, this method is available to measure the vibrating characteristic of the plant.

5. Conclusions

In the experiment of the steel model, the measured natural frequency was 5.0 Hz. The one calculated was 4.8 Hz, so this method is effective to measure the natural frequency. In the experiment of the plant, the measured natural frequency was 2.2 Hz

It took 66 ms to process one image, so 15 captured images could be processed. It means to measure the vibration characteristic of the plant, if the natural frequency of the plant is less 7.5 Hz.

It became possible to measure the vibration characteristic of the plant by the image processing with HSI color space. By this method, it was impossible to measure the vibration characteristic whose natural frequency was over 7.5Hz. But, the natural frequency of the plant is generally less 7.5Hz. So, this method is useful for measuring the vibration characteristic of the plant.

6. References

[1] Mikio Takagi and Haruhisa SHIMODA,Image analysis handbook, The University of Tokyo publishing, 1991, PP485-491

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

BUILDING AN ARTIFICIAL BRAIN USING AN FPGA BASED "CAM-BRAIN MACHINE"

Hugo de GARIS (1), Felix GERS (2), Michael KORKIN (3), Arvin AGAH (4), Norberto NAWA (1)

(1) Brain Builder Group, ATR, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto-fu, JAPAN. tel. (+ 81) (0)7749 5 1079, fax. (+ 81) (0) 7749 5 1008 www.hip.atr.co.jp/~degaris, degaris@hip.atr.co.jp

(2) IDSIA, Corso Elvezia 36, CH-6900, Lugano, SWITZERLAND tel. (+41) (0)91 911 9838, fax. (+ 41) (0)91 911 9839 www.idsia.ch/~felix/ felix@idsia.ch

(3) Genobyte Inc. 1319 Spruce St., Suite 210, Boulder, CO, 80302, USA. tel. (+1) (303) 545 6790, fax. (+1) (303) 545 9667, www.genobyte.com korkin@genobyte.com

(4) Dept. of Electrical and Computer Science, University of Kansas, Lawrence, Kansas, 66045, USA. tel. (+ 1) (785) 864 8821, fax. (+ 1) (785) 864 4971, agah@eecs.ukans.edu

Abstract

This paper reports on recent progress made in ATR's attempt to build a 10,000 evolved neural net module artificial brain to control the behaviors of a life sized robot kitten.

Key Words

10,000 Neural Net Module Artificial Brain (CAM-Brain Project), Evolutionary Engineering, E-Hard (Evolvable Hardware), CAM-Brain Machine (CBM), Robot Kitten (Robokoneko), Brain Architecture.

1. Introduction

This paper presents progress in ATR's Artificial Brain (CAM-Brain) Project. The broad aim of ATR's artificial brain project, is to build/grow/evolve an artificial brain containing a billion artificial neurons by the year 2001. The basic ideas of the CAM-Brain Project are as follows. Use cellular automata (CA) as the foundation upon which to grow and evolve neural network circuits, with user defined functionality. The state of each cellular automata cell can be stored in one or two bytes of RAM. Since nowadays it is possible to have a gigabyte of RAM in one's workstation, there is a huge space in which to store the CA cell states, more than enough to contain an artificial brain. The next consideration in the CAM-Brain Project was how to evolve these neural networks quickly enough for "Brain Building" (i.e. the assemblage of 10,000s and more of such evolved neural network modules into humanly defined artificial brain architectures). We chose to use evolvable hardware techniques that our team invented [de Garis 1993]. Special reconfigurable hardware chips (Xilinx XC6264) have been programmed to grow neural circuits literally in nanoseconds, according to a simplified neural growth and signaling model, that we call "CoDi". This special hardware will update 100 Billion CA cells a second, which is fast enough to complete a full run of a genetic algorithm (e.g. with a population of 100, for hundreds of generations) in about a second. A user specified fitness definition for each module is compiled directly into programmable hardware, so as to automate the grown circuit's fitness measurement at hardware speeds. This hardware is called a CAM-Brain Machine (CBM) and will be built by the spring of 1998.

Our team is currently involved in several parallel tasks. The first is the construction of the CBM. The second is the behavioral specification and design of a life sized robot kitten which a 10,000 evolved neural net module artificial brain will control. The third is the design of the brain itself. The kitten robot, called "Robokoneko" (Japanese for "robot kitten") will be controlled by an on-body radio link to the artificial brain consisting of some 80 Megabytes of RAM, which is updated 2500 times a second by the CBM, which is fast enough for real time control. The year 1998 will be spent in gaining experience with the CBM and evolving a large number of modules. At the same time, the kitten robot will be built and tested, and the brain architecture will be designed. In 1999, the brain and the kitten should be brought together, hopefully in time to show the world, at the turn of the millennium, the world's first functioning artificial brain. The CBM's incredible speed should allow brain building to become practical. Our team hopes that the CBM will give birth to a new research field, that we call simply "Brain Building". If the kitten robot and its brain are successful, then it is likely that industry will get interested. The rest of this paper consists of the following sections. Section 2 introduces the neural net model that we are currently implementing in FPGA (field programmable gate array) hardware in a special piece of apparatus we call a CAM-Brain Machine (CBM) whose description is found in section 3. Section 4 presents some early software simulation results of this machine, which although very preliminary, show the machine and the

CoDi model to be highly evolvable. Section 5 gives some early thoughts on the specifications of the robot kitten, which a 10,000 evolved neural net module artificial brain will control.

2. The CoDi Neural Network Model

This section describes the neural net model implemented in the CAM-Brain Machine. We call it "CoDi" (i.e. Collect and Distribute). CoDi is a simplified CA-based neural network model developed at ATR in the summer of 1996 with two goals in mind. One was to make neural network functioning much simpler compared to the older CAM-Brain model developed in 1993 and 1994 [de Garis 1994], so as to be able to implement the model directly in electronics and thus to evolve neural net modules at electronic speeds. In order to evolve one neural module, a population of 100 modules is run through a genetic algorithm for about 100 generations, resulting in 10,000s of different module evaluations. Each module evaluation consists of growing a new set of axonic and dendritic trees which interconnect about 100 neurons in the 3D cellular automata space of 4096 cells, then running the module to evaluate its performance (fitness). This typically requires 300 update cycles for all the cells in the module.

On the hardware we currently use to update the CA cells, i.e. MIT's CAM-8 machine, it takes one minute to update the twelve billion cells needed to evolve a single neural module. Even in a simple "insect-like" artificial brain, there are a million neurons arranged into ten thousand 100-neuron modules. It would take 7 days (running 24 hours a day) to finish the computations. Due to a very large module chromosome size, an evolution time longer than 100 generations may be needed. Another limitation was very apparent concerning full brain simulation involving thousands of modules interconnected together. For a ten thousand module brain, the CAM-8 is capable of updating every module at the rate of one update cycle five times a second. However, for real time control of a robotic device, an update rate of 50-100 cycles per module, 10-20 times a second is needed. So, the second goal was to have a model which would be portable into hardware to eventually design a machine capable of accelerating both brain evolution and brain simulation by a factor of 100-500 compared to the CAM-8 machine.

The CoDi model [Gers, de Garis, Korkin 1997] operates as a 3D cellular automata. Each cell is a cube which has six neighbor cells, one for each of its faces. By loading a different phenotype code into a cell, it can be reconfigured as a neuron, an axon, or a dendrite. Neurons are configurable on a coarser grid, namely one per block of 2*2*2 CA cells. Cells are interconnected with bi-directional 1-bit buses and assembled into 3D modules of 4096 cells (16*16*16). Modules are further interconnected with 32-bit wide buses to function together as an artificial brain. These inter modular connections are implemented as linked lists in a module interconnection memory (see section 3.5). In a neuron cell, five (of its six) connections are dendritic inputs, and one is an axonic output. A 4-bit accumulator sums incoming signals and fires an output signal when a threshold is exceeded. Each of the inputs can perform an inhibitory or an excitatory function (depending on the neuron's chromosome) and either adds to or subtracts from the accumulator. The neuron cell's output can be oriented in 6 different ways in the 3D space. A dendrite cell also has five inputs and one output, to collect signals from other cells. The incoming signals are passed to the output with an OR function. An axon cell is the opposite of a dendrite. It has 1 input and 5 outputs, and distributes signals to its neighbors. The "collect and distribute" mechanism of this neural model is reflected in its name "CoDi". Blank cells perform no function in an evolved neural network. They are used to grow new sets of dendritic and axonic trees during the evolution mode. Before the growth begins, the module space consists of blank cells. Each cell is seeded with a 5-bit chromosome. The chromosome will guide the local direction of the dendritic and axonic tree growth. 5 bits encode 32 different growth instructions, e.g. grow straight, turn left, split into three branches, block any growth, T-split up and down etc. Before the growth phase starts, a small number of cells (1-3%) are seeded as neurons at random locations.

As the growth starts, each neuron continuously sends growth signals to the surrounding blank cells, alternating between "grow dendrite" (sent to the neuron's dendritic connections) and "grow axon" (sent to the axonic connection). A blank cell which receives a growth signal becomes a dendrite cell, or an axon cell, and further propagates the growth signal, being continuously sent by a neuron, to other blank cells. The direction of the propagation is guided by the 5-bit growth instruction, described above. This growth mechanism allows the growth of a complex 3D system of branching dendritic and axonic trees, with each tree having one neuron cell associated with it. The trees can conduct signals between the neurons to perform complex spatio-temporal functions. The end-product of the growth phase is a phenotype bitstring which encodes the type and spatial orientation of each cell.

3. The CAM-Brain Machine (CBM)

This section describes the hardware implementation of the above CoDi model, allowing a CoDi neural net module to be grown in nanoseconds, and a whole population of modules to be evolved in about a second. The CBM consists of the following five major blocks. 1) Cellular Automata Module, 2) Genotype & Phenotype

Memory, 3) Fitness Evaluation Unit, 4) Genetic Algorithm Unit, 5) Module Interconnection Memory.

Each of these blocks are discussed in detail below. The cellular automata module is the hardware core of the CBM. It is intended to accelerate the speed of brain evolution through a highly parallel execution of cellular state updates. The CA module consists of an array of identical hardware logic circuits or cells arranged in two identical 3D structures of 16*16*16 cells (a total of 8192 cells). Cells forming the top layer of the module are recurrently connected with the cells in the bottom layer. A similar recurrent connection is made between the cells on the north and south, and the east and west vertical surfaces. Thus a fully recurrent toroidal cube is formed. This feature allows a much higher axonic and dendritic growth capacity by effectively doubling each of the three dimensions of the block.

The CA module is implemented with new Xilinx FPGA devices XC6264. These devices are fully and partially reconfigurable, featuring a new co-processor architecture with data and address bus access in addition to user inputs and outputs, and allow the reading and writing of any of the internal flip-flops through the data bus. An XC6264 chip contains 16384 logic function cells, each cell featuring a flip-flop and some Boolean logic capacity, capable of toggling at a 220 MHz rate. Logic cells are interconnected with neighbors at several hierarchical levels, providing identical propagation delay for any length of connection. This feature is very well suited for a 3D CA space configuration. Additionally, clock routing is optimized for equal propagation time, and power distribution is implemented in a redundant manner. To implement the CA module, a 3D block of identical logic cells is configured inside each XC6264 device, with CoDi specified 1-bit signal buses interconnecting the cells. Given the FPGA internal routing capabilities and the logic capacity needed to implement each cell, the optimal arrangement for a XC6264 is 8*8*4 (256 cells). Four external surfaces of each module cube have 8 cells reserved to be used as external inputs to the module (32 signals total), and two other surfaces have 8 cells reserved as external outputs from the module (16 signals). To assemble two identical 4096-cell modules, 32 XC6264 chips are needed in a BGA560 package, laid out as 2*2 chips on 8 boards.

There are two modes of CBM operation, namely evolution mode and run mode. In the evolution mode, memory space is used to store the chromosome bitstrings of the evolving population of modules (module genotype). For a module of 4096 cells there are 20480 bits of memory needed. For each module the genotype memory also stores information concerning a maximum of 128 neurons locations and orientations inside the module. This includes, X, Y, Z coordinates (12 bits), gating code (3 bits), input functions (excitatory/inhibitory) (5 bits), giving a total of 20 bits per neuron. Thus, the total chromosome memory requirement for one module is 20480+20*128 = 2880 bytes. The genotype/phenotype memory size for 10,000 modules is 32 Mbytes. The host computer memory can be used when needed to reload this space with more data.

In run mode, memory is used as phenotype memory for the evolved modules. The phenotype data describes grown axonic and dendritic trees and their respective neurons for each module. For phenotype storage there are 6 bits required per cell, i.e. for gating (3 bits), cell type (2 bits), and signal value (1 bit). Neuron cells additionally require 4 bits for the accumulator value stored. Phenotype data is loaded into the CA module to configure it according to the evolved function. Genotype/phenotype memory is used to store and rapidly reconfigure (reload) the FPGA hardware CA module. Reconfiguration can be performed in parallel with running the module, due to a dual pipelined phenotype/genotype register provided in each cell. This guarantees the continuous running of fast FPGA hardware at full speed with no interruptions for reloading in both evolution and run modes. 32 Mbytes of memory can store over 10 thousand modules at a time. This amount is sufficient to evolve 10 thousand modules at high speed, or to run a simulated brain with one million neurons. A large memory will be based in the main memory of the host computer (Pentium-Pro) connected to the CBM through a PCI bus, capable of transferring data at 132 Mbytes/s. Genotype/phenotype memory is connected to the hardware CA module and is used for rapid reconfiguration of the neural module by loading new chromosome data (or phenotype data) into the hardware registers of each cell through the XC6264 data bus access. Thus the fast hardware for the CA module is time-multiplexed between multiple neural modules in a large brain.

When a useful module is being evolved, each instance of a module must be evaluated in terms of it's fitness for a targeted task. During the signaling phase, each module generates a sequence (an array) of 16 output signals, or vectors, which is compared with a target array in order to guide the evolutionary process. This comparison gives a measure of performance, or fitness, of the module. Fitness evaluation is supported by a hardware unit which consists of an input vector array stack, a target vector array stack, and a fitness comparator. The input stack is 32-bit wide, and the target vector stack is 16-byte wide SRAMs (FIFOs) storing up to 2048 input and target vectors each to support the signaling phase of up to 2048 cycles. During each clock cycle an input vector is read from its stack and fed into the module's inputs. At the same time, a target vector is read from its stack to be compared with the current module output vector by the fitness evaluation unit. The fitness comparator computes a Hemming distance between each output vector and a corresponding target vector, and accumulates the result for the whole duration of the signaling phase.

At the end of the signaling phase, a final measure of the module's fitness is instantly available. Multiple target and input arrays are stored in the host computer memory. To evolve a module, a population of 100 modules is evaluated by computing every module's fitness measure, as described above. The ten best modules are then selected for further reproduction. A hardware support will be provided to store and update a "current best ten" list. This list is an array of the current ten best module numbers. Each number is an address of the module chromosome in the chromosome memory. After each generation of modules, the ten best are mated and mutated to produce 100 offspring modules to become the next generation. Mating and mutation is performed by the host computer software on the chromosome memory. Because this process can be performed in parallel with the module evaluation, and only takes place once in a generation, it is expected that there will be no significant slowdown in the evolutionary process. In order to support the run mode of operation, which requires a large number of evolved modules to function as an artificial brain, a module interconnection memory is provided. It consists of an output vector array stack, similar to the input array and target stack, and a 64 Mbyte memory for inter modular pathway storage. When each module of a large brain is configured in the CA hardware core (by loading its phenotype), an input stack is loaded with an array of input vectors. These vectors are previously stored output vectors recorded during the signaling phase of other modules, connected to this module. A large module interconnection memory will store the connection map and the current state of signals "traveling" between modules. There will be software and hardware support provided for combining output arrays from up to 8 modules, to be used as inputs for one module. In addition, an externally connected set of inputs and outputs is provided in hardware, which will allow the reception of signals from external sensors and the sending of signals to external effectors for robotic control. When running a simple million-neuron brain which contains ten thousand 100-neuron modules, the CBM is capable of updating each module for 100 cycles at the rate of about 25 times per second, allowing real-time control of robotic devices.

4. Some CBM Simulation Results

This section presents some early CBM software simulation results. At the time of writing (Oct. 1997), we are very much in the thick of this simulation, so we are unable to present extensive results. However, judging from the results we already have, it is clear that the CBM will be highly evolvable, which is very encouraging. We give here a taste of what we have done. It concerns a digital "timer" module, which will be fairly typical of the type of module needed to be evolved when the time comes to specify the sensor, control and behavior modules to be evolved for the artificial brain. We wanted to see if a CoDi 1-bit module (consisting of 4K 3D CA cells, with about 150 artificial neurons in the 4K space, could evolve a "timer", i.e. where constantly firing inputs generate at a single cell output (placed elsewhere in the CA space) a string of 0s during the first 30 clocks, then a string of 1s during the next 20 clocks, and finally a string of 0s in the last 20 clocks. This is quite a demanding evolutionary task we thought and would be a useful test vehicle for the early evolutionary trials. The fitness definition was pretty simple. If a 0 appeared in the first (0) block, score one point. If a 1 appeared in the second (1) block, score 3 points. If a 0 appeared in the third block (0) score 2 points. Hence a perfect score would be 30*1 + 20*3 + 20*2 = 130 points. The population size was 24, with no crossover. The CoDi-1bit simulation evolved with a fitness of 100% in about 150 generations. Since a perfect score gives no indication of the limits of the evolvability of the CBM, we chose a much more demanding task, similar to the above. Instead of 3 regions, 5 were chosen, shown below, including the best evolved output.

Target Output :-
00000000000000000000000000000
11111111111111111111
00000000000000000000000 11111111111111111
00000000000000000000

Best Evolved Output :-
00000000000000000000000000000
00011111111111111111
10000000000000000000000 01111111111111111
10000000000000000000

which obtained a fitness score of 96%, where the fitness was defined similarly to that above, with point weightings for the five regions being 12, 7, 3, 2, 1. Since 96% was not a perfect score, it showed the limits of the CBM evolvability. Presumably with more neurons and CA space, greater evolvability will be achievable. Such hypotheses we need to test in the near future. However, the above is rather impressive. It shows that the CBM and the CoDi-1bit model is capable of evolving modules with quite demanding behavioral characteristics. In the coming months we will be writing papers giving more detailed descriptions of these ongoing CBM simulation experiments. However, we are conscious that to truly test the capabilities of a CBM, we need a CBM!

5. Robokoneko, the Robot Kitten

An artificial brain without something to control is rather pointless, so we decided to have the brain control the behaviors of a life sized robot kitten. This section discusses the behaviors that the authors would like the robot kitten ("Robokoneko") to have, which will require roughly 10,000 modules to implement. It seems to the authors that there are two broad approaches to mapping a set of behaviors to a given number of modules and vice versa. One

approach is empirical, i.e. just keep adding behaviors and capabilities to the brain, until the number of modules concerned is reached, or just add more modules until a given behavioral repertoire is reached. The other approach, is more theoretical and does not exist yet. It is called here "Behavioral Set Analysis (BSA)", which attempts to provide brain architects with principles with which to estimate the number of modules needed for a given behavioral set, or for a given number of modules, how extensive (how intelligent?) could the behaviors of such a brain be. Since brain building is entirely new, at least on the scale envisioned in this paper, BSA does not yet exist. There is not enough experience yet to create such a body of theory, but as more artificial brains are built (as CBMs and similar devices become more commonplace), this theory should be easier to construct. Since the attempt to build a 10,000 module artificial brain is the first of its kind ever, there is zero experience to fall back on. It is possible that a behavioral set suggested here may need a lot less than 10,000 modules. de Garis implemented an artificial creature with less than 100 modules which performed "intelligent" behaviors such as stalking prey and mates, and avoiding predators, so imagine what one could do with 10,000 modules (possible in 1998), or 10,000,000 modules (which may be only a few years away). We begin now with some suggestions as to what Robokoneko could do with the 10,000 modules that the CBM will be able to evolve and control in real time in 1998. Note, that this paper only presents the early thoughts of the authors. More detailed plans must be created by Xmas of 1997 in anticipation of the delivery of the CBM in the spring of 1998. The behavioral set of the kitten robot is partly determined by "political" considerations. Building an artificial brain is a pioneering effort, and so our first task is to show the world that such a thing is possible. ATR's Brain Builder Group needs "proof of concept". For this reason, it is important to make the brain as media friendly as possible to maximize publicity. When millions of people see a cute media friendly robot kitten displaying the many behaviors made possible by a 10,000 module artificial brain, they will be impressed, and hopefully also will Japan's top scientific research policy makers who control the purse strings to pay for a 2000 evolutionary engineer (EE) 5 year J-Brain Project (2000-2005). Hence, the broad aim is to create an artificial brain which will be capable of making a kitten sized robot as life like as possible. Since 10,000 modules is a lot, it will be possible to give this robot many behaviors and capabilities. de Garis and Agah brainstormed for 2 days and came up with the following desiderata for the robot. (A few more aspects were added later). The body of the robot kitten should be real kitten size, with of course 4 legs, a tail, whiskers, a head, with 2 color camera eyes, 2 microphone ears, and maybe (if it is not too heavy) some soft furry covering over its metal springs and frame etc, to make it cute. The kitten should be controlled by wireless modem and be battery powered to be active for about 30 minutes before needing a recharge. The robot could probably be built within a year, especially if grad students get involved. For functionality, we thought of the following. For behaviors, we wanted the kitten to walk on its 4 legs, stand on 3 legs, manipulate with its front leg(s), jump on the floor, jump onto objects, curl, lie down, sleep, move its tail, move its head, wiggle its ears, react to whisker touching, urinate (water), avoid obstacles, purr when stroked, show hostility when its tail is pulled etc. This is only an initial list which will probably be extensively added to, considering we have 10,000 neural net modules at our disposal to put into the kitten's brain. For sound generation and detection, we want a speaker with a ROM "meow" expressing a range of emotions, e.g. attention seeking, anger, sadness, purring etc. The sound detector would be 2 microphone ears and 2 signal processing chips, so that one could call the kitten "Here, kitty kitty kitty", "Scram", "Woof" (fear reaction), etc.

We wanted heat sensors (to detect people in the neighborhood), touch sensors (on the back, whiskers, tail etc) and feet force sensors. For vision, we wanted 2 color cameras for the eyes, with accompanying signal processing chips on the robot. The whole thing should weigh about 1 to 2 Kgms, so that it can jump and move at real kitten speed. It will be made as light as possible and with minimal on-board processing. Most of the intelligence of the kitten will be generated off-body by the artificial brain, whose decisions will be transmitted by wireless modem to (and from) the kitten. A certain minimum weight will be necessary due to the weight of the modem (about 0.5 Kg) and the battery (0.5 Kg). The binary I/O of the modem could interface directly with the 1-bit signaling of the "CoDi model" of ATR's CAM-Brain design [Gers, de Garis, Korkin 1997].

References

[de Garis 1993] "Evolvable Hardware : The Genetic Programming of Darwin Machines", Hugo de Garis, Int. Conf. on Artificial Neural Networks and Genetic Algorithms, 1993, Innsbruck, 1993,

[de Garis 1994] "An Artificial Brain : ATR's CAM-Brain Project Aims to Build/Evolve an Artificial Brain with a Million Neural Net Modules inside a Trillion Cell Cellular Automata Machine", Hugo de Garis, New Generation Computing Journal, Vol. 12, No. 2, 1994, Ohmsha & Springer Verlag.

[Gers et al 1997] "CoDi-1bit : A Simplified Cellular Automata Based Neuron Model", Felix Gers, Hugo de Garis, Michael Korkin, Artificial Evolution Conf, Oct 1997, Nimes, France.

[Korkin & de Garis 1997] "CBM (CAM-Brain Machine) : A Hardware Tool which Evolves a Neural Net Module in a Fraction of a Second and Runs a Million Neuron Artificial Brain in Real Time", Michael Korkin & Hugo de Garis, Genetic Programming Conference, July, 1997, Stanford, USA.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

"CoDi-1Bit" A CELLULAR AUTOMATA BASED NEURAL NET MODEL SIMPLE ENOUGH TO BE IMPLEMENTED IN EVOLVABLE HARDWARE

Felix GERS (1), Hugo de GARIS (2), Michael KORKIN (3)

(1) IDSIA
Corso Elvezia 36
CH-6900, Lugano,
SWITZERLAND
tel. (+41) (0)91 911 9838
fax. (+ 41) (0)91 911 9839
http://www.idsia.ch/~felix/
felix@idsia.ch

(2) Brain Builder Group
ATR, 2-2 Hikaridai, Seika-cho
Soraku-gun, Kyoto-fu,
JAPAN.
tel. (+ 81) (0)7749 5 1079,
fax. (+ 81) (0) 7749 5 1008
http://www.hip.atr.co.jp/~degaris
degaris@hip.atr.co.jp

(3) Genobyte Inc.
1319 Spruce St., Suite 210,
Boulder, CO, 80302,
USA.
tel. + 1 303 545 6790,
fax. (+1) 303 545 9667,
http://www.genobyte.com
korkin@genobte.com

Abstract

This paper presents details of a cellular automata based neural network model which is simple enough to be implementable in evolvable hardware, allowing the growth of neural nets in nanoseconds, and the evolution of neural nets in seconds. This model and the hardware tool based upon it, should enable the birth of a new research field, called "brain building" because it will become practical to assemble tens of thousands of such evolved neural net modules into humanly architected artificial brains.

1. Introduction

This paper gives details on the cellular automata based neural net model which is simple enough to be implementable in evolvable hardware. This model, called CoDi-1Bit (from Collect and Distribute, with 1 bit signaling) is one of the cornerstones in ATR's efforts to build an artificial brain. Before giving details of the CoDi-1Bit model, some background knowledge of ATR's "CAM-Brain Project" needs to be given. The broad aim of ATR's artificial brain ("CAM-Brain") project, is to build/grow/evolve an artificial brain containing a billion artificial neurons by the year 2001. The basic ideas of the CAM-Brain Project are as follows. Use cellular automata (CA) as the foundation upon which to grow and evolve neural network circuits, with user defined functionality. The state of each cellular automata cell can be stored in one or two bytes of RAM. Since nowadays it is possible to have a gigabyte of RAM in one's workstation, there is a huge space in which to store the CA cell states, more than enough to contain an artificial brain. The next consideration in the CAM-Brain Project was how to evolve these neural networks quickly enough for "Brain Building" (i.e. the assemblage of 10,000s and more of such evolved neural network modules into humanly defined artificial brain architectures). We chose to use evolvable hardware techniques that our team invented [de Garis 1993]. Special reconfigurable hardware chips (Xilinx XC6264) have been programmed to grow neural circuits literally in nanoseconds, according to a simplified neural growth and signaling model, that we call "CoDi". This special hardware will update 100 Billion CA cells a second, which is fast enough to complete a full run of a genetic algorithm (e.g. with a population of 100, for hundreds of generations) in about a second. A user specified fitness definition for each module is compiled directly into programmable hardware, so as to automate the grown circuit's fitness measurement at hardware speeds. This hardware is called a CAM-Brain Machine (CBM) and will be built by the spring of 1998. Our team is currently involved in several parallel tasks. The first is the construction of the CBM. The second is the behavioral specification and design of a life sized robot kitten which a 10,000 evolved neural net module artificial brain will control. The third is the design of the brain itself. The kitten robot, called "Robokoneko" (Japanese for "robot kitten") will be controlled by an on-body radio link to the artificial brain consisting of some 80 Megabytes of RAM, which is updated 2500 times a second by the CBM, which is fast enough for real time control. The year 1998 will be spent in gaining experience with the CBM and evolving a large number of modules. At the same time, the kitten robot will be built and tested, and the brain architecture will be designed. In 1999, the brain and the kitten should be brought together, hopefully in time to show the world, at the turn of the millennium, the world's first functioning artificial brain. The CBM's incredible speed should allow brain building to become practical. Our team hopes that the CBM will give birth to a new research field, that we call simply "Brain Building". If the kitten robot and its brain are successful, then it is likely that industry will get interested. The rest of this

paper consists of the following sections. Section 2 introduces the neural net model that we are currently implementing in FPGA (field programmable gate array) hardware in a special piece of apparatus we call a CAM-Brain Machine (CBM). Section 3 elaborates upon section 2. Section 4 gives some simulation results and section 5 concludes.

2. The CoDi Neural Network Model

This section gives an overview of the "CoDi" neural net model implemented in the CAM-Brain Machine hardware. (The following section gives further details). We call this model "CoDi" due to its " Collect and Distribute" nature of its neural signals. CoDi is a simplified CA-based neural network model developed at ATR in the summer of 1996 with two goals in mind. One was to make neural network functioning much simpler compared to the older CAM-Brain model developed in 1993 and 1994 [de Garis 1993, 1994], so as to be able to implement the model directly in electronics and thus to evolve neural net modules at electronic speeds. In order to evolve one neural module, a population of 100 modules is run through a genetic algorithm for several 100 generations, resulting in 10,000s of different module evaluations. Each module evaluation consists of growing a new set of axonic and dendritic trees which interconnect about 100 neurons in the 3D cellular automata space of 9K cells, then running the module to evaluate its performance (fitness). This typically requires 300 update clock cycles for all the cells in the module.

On the hardware we currently use to update the CA cells, i.e. MIT's CAM-8 machine, it takes one minute to update the twelve billion cells needed to evolve a single neural module. Even in a simple "insect-like" artificial brain, there are a million neurons arranged into ten thousand 100-neuron modules. It would take 7 days (running 24 hours a day) to finish the computations. Due to a very large module chromosome size, an evolution time longer than several 100 generations may be needed. Another limitation was very apparent concerning full brain simulation involving thousands of modules interconnected together. For a ten thousand module brain, the CAM-8 is capable of updating every module at the rate of one update cycle five times a second. However, for real time control of a robotic device, an update rate of 50-100 cycles per module, 10-20 times a second is needed. So, the second goal was to have a model which would be portable into hardware to eventually design a machine capable of accelerating both brain evolution and brain simulation by a factor of 100-500 compared to the CAM-8 machine. The CoDi model [Gers & de Garis 1996] operates as a 3D cellular automata. Each cell is a cube which has six neighbor cells, one for each of its faces. By loading a different phenotype code into a cell, it can be reconfigured as a neuron, an axon, or a dendrite. Neurons are configurable on a coarser grid, namely one per block of 2*2*2 CA cells. Cells are interconnected with bi-directional 1-bit buses and assembled into 3D modules of 9K cells (24*24*16). Modules are further interconnected with 32-bit wide buses to function together as an artificial brain. These inter modular connections are implemented as linked lists in a module interconnection memory. In a neuron cell, five (of its six) connections are dendritic inputs, and one is an axonic output. A 4-bit accumulator sums incoming signals and fires an output signal when a threshold is exceeded. Each of the inputs can perform an inhibitory or an excitatory function (depending on the neuron's chromosome) and either adds to or subtracts from the accumulator. The neuron cell's output can be oriented in 6 different ways in the 3D space. A dendrite cell also has five inputs and one output, to collect signals from other cells. The incoming signals are passed to the output with an OR function. An axon cell is the opposite of a dendrite. It has 1 input and 5 outputs, and distributes signals to its neighbors. The "Collect and Distribute" mechanism of this neural model is reflected in its name "CoDi". Blank cells perform no function in an evolved neural network. They are used to grow new sets of dendritic and axonic trees during the growth phase. Before the growth begins, the module space consists of blank cells. In the earlier version of the Codi model, each cell is seeded with a 5-bit chromosome. The chromosome guides the local direction of the dendritic and axonic tree growth. 5 bits encode 32 different growth instructions, e.g. grow straight, turn left, split into three branches, block any growth, T-split up and down etc. Before the growth phase starts, a small number of cells (1-3%) are seeded as neurons at random locations.

As the growth starts, each neuron continuously sends growth signals to the surrounding blank cells, alternating between "grow dendrite" (sent to the neuron's dendritic connections) and "grow axon" (sent to the axonic connection). A blank cell which receives a growth signal becomes a dendrite cell, or an axon cell, and further propagates the growth signal, being continuously sent by a neuron, to other blank cells. The direction of the propagation is guided by the 5-bit growth instruction, described above. This growth mechanism allows the growth of a complex 3D system of branching dendritic and axonic trees, with each tree having one neuron cell associated with it. The trees can conduct signals between the neurons to perform complex spatio-temporal functions. The end-product of the growth phase is a phenotype bitstring which encodes the type and spatial orientation of each cell.

3. Details of the CoDi-1Bit Model

This section elaborates upon the previous section, by providing a point by point description of the more modern 1Bit version of the CoDi model, i.e. "CoDi-1Bit".

3.1 Growth Phase

a) About 100 neurons are randomly seeded in the cellular space, and growth instructions (chromosomes) are loaded (6 bits per CA cell) into all the remaining cells. 6 bits are needed, one for each face of the CA cell cube direction (N, S, W, E, Top, Bottom). A 1 indicates growth into that direction. All zeros blocks growth. Each seeded neuron has 5 sides which will grow dendritic trees, and a 6th side which will grow an axonic tree. Neurons can be seeded with any of the 6 possible spatial orientations.

b) Each neuron then emits 2 types of growth signals, "axon growth signal" and "dendrite growth signal". The dendrite signals are sent in the directions of the 5 neighbors of the neuron (to grow 5 dendritic trees), the axon growth signals are sent in the direction of the 6th neighbor (to grow an axonic tree). Because only 1-bit signals are allowed in the CoDi-1Bit model (in order to simplify its electronic implementation), the axon and dendrite growth signals are alternated in time (time multiplexed). First extend an axon tree for one clock, then extend a dendrite tree for one clock, then repeat, etc. A blank cell knows from the time multiplexing whether it is an axon or a dendrite signal that is hitting it.

c) When a blank cell is hit on one of its sides by an incoming growth signal, it decodes which side it comes from and "knows" whether it is in an axonic or dendritic signal time slot. It thus "becomes" a dendritic or an axonic cell, which means that it creates a phenotype code to be used later during the signaling phase. This code includes the orientation of the cell (which is determined by the direction of the received growth signal, and the cell type (axon or dendrite). If more than one neighbor sends growth signals to the cell, the cell remains blank (illegal condition).

d) The cell then propagates the same growth signal which it received (axonic or dendritic) in the direction of its neighbors according to those bits which are set high in its preloaded growth instruction (chromosome). The direction from where the original growth signal came is ignored (no signals are sent) even if the corresponding bit was set high in the chromosome.

e) After a blank cell receives a growth signal and becomes a non blank cell, it cannot receive any more growth signals.

3.2 Neural Signaling Phase

a) Each neuron receives signals from 5 neighbors through its 5 dendritic inputs, and sends an axonic output into the 6th direction. Each of the 5 dendritic inputs has a positive or negative sign (equivalent to an excitatory or inhibitory synapse in biology). This 6 bit combination of signs is evolvable. So each neuron's chromosome combines code for spatial orientation and code for synaptic combination.

b) A neuron sums the incoming 1-bit signals accounting for their signs specified by the synaptic code. The resulting sum can be positive or negative in the range (-5 +5), depending on the synaptic code and the signal activity. If all of the synapses are positive, and each of the dendrites receives a signal in one clock cycle, the sum in this clock cycle would be 5. If, for example, there are 2 positive and 3 negative synapses, and each of them receives a signal, the sum is (-1). If only one of the positive dendrites and only one of the negative dendrites receives a signal, the sum would be (0).

c) The sum of all dendritic signals in each cycle is added to a 4 bit accumulator, which sums the results over many clock cycles. So it can go positive and negative. When the accumulator reaches (or goes over) +2, it fires a signal into to axonic output direction, and resets to zero (i.e. discharges). If the accumulator goes below -7 it resets itself to zero without firing. The actual accumulator range is (-8 .. +7), i.e. 16 values.

d) An axon cell receives signals from one side and propagates them to the remaining 5 sites. A dendrite cell receives 5 signals from its 5 sides, and if one of them is present (high), it sends a signal to its 6th side. So it is an OR function. (Actually, in our simulation studies, we find we get better evolvability, if the signal is sent on only if the number of incoming signals is one. Two or more incoming signals cause no output).

e) In both the growth and signaling phases, the neurons are placed on a coarser grid of every other cell in each of the 3 dimensions. (This is done for hardware implementation reasons). So there is one neuron possible in a block of 2*2*2 cells, and the closest possible neurons are always separated by a 1-cell thick layer of cells.

4. CoDi-1Bit Simulation Results

This section presents some early CoDi-1Bit Model software simulation results. At the time of writing (Oct. 1997), we are very much in the thick of this simulation, so we are unable to present extensive results. However, judging from the results we already have, it is clear that the CBM will be highly evolvable, which is very encouraging. We give here a taste of what we have done. It concerns a digital "timer" module, which will be fairly typical of the type of module needed to be evolved when the time comes to specify the sensor, control and behavior modules to be evolved for the artificial brain. We wanted to see if a CoDi 1-bit module (consisting of 4K 3D CA cells, with about 150 artificial neurons in the 4K space, could evolve a "timer", i.e. where constantly firing inputs generate at a single cell output (placed elsewhere in the CA

space) a string of 0s during the first 30 clocks, then a string of 1s during the next 20 clocks, and finally a string of 0s in the last 20 clocks. This is quite a demanding evolutionary task we thought and would be a useful test vehicle for the early evolutionary trials. The fitness definition was pretty simple. If a 0 appeared in the first (0) block, score one point. If a 1 appeared in the second (1) block, score 3 points. If a 0 appeared in the third block (0) score 2 points. Hence a perfect score would be 30*1 + 20*3 + 20*2 = 130 points. The population size was 24, with no crossover. The CoDi-1bit simulation evolved with a fitness of 100% in about 150 generations. Since a perfect score gives no indication of the limits of the evolvability of the CBM, we chose a much more demanding task, similar to the above. Instead of 3 regions, 5 were chosen, shown below, including the best evolved output.

Target Output :-
00000000000000000000000000000000
11111111111111111111
00000000000000000000000000 1111111111111111
0000000000000000000

Best Evolved Output :-
00000000000000000000000000000000
00011111111111111111
10000000000000000000000000 0111111111111111
1000000000000000000

which obtained a fitness score of 96%, where the fitness was defined similarly to that above, with point weightings for the five regions being 12, 7, 3, 2, 1. Since 96% was not a perfect score, it showed the limits of the CBM evolvability. Presumably with more neurons and CA space, greater evolvability will be achievable. Such hypotheses we need to test in the near future. However, the above is rather impressive. It shows that the CBM and the CoDi-1bit model is capable of evolving modules with quite demanding behavioral characteristics. In the coming months we will be writing papers giving more detailed descriptions of these ongoing CBM simulation experiments. However, we are conscious that to truly test the capabilities of a CBM, we need a CBM!

5. Conclusions

The CoDi-1Bit neural net cellular automata based model is a simplified version of an earlier 2 bit model. The simplification was needed so as to allow it to be implemented in state-of-the -art FPGA (field programmable gate array) based electronics (Xilinx's XC6264 chips). Thus, starting in 1998, it will be possible to evolve neural net modules at electronic speeds, with CA cell update speeds of 100 billion a second, allowing a complete run of a genetic algorithm with tens of thousands of circuit growths and fitness evaluations in about 1 second. These incredible speeds will make the new field of brain building practical. At the time of writing (Oct. 1997), software simulation experiments on the CoDi-1Bit model are continuing. Results so far look very encouraging. It appears that the CoDi-1Bit model is highly evolvable.

References

Note : Any paper containing the name de Garis, can be downloaded from www.hip.atr.co.jp/~degaris

[de Garis 1993] "Evolvable Hardware : The Genetic Programming of Darwin Machines", Hugo de Garis, Int. Conf. on Artificial Neural Networks and Genetic Algorithms, 1993, Innsbruck, 1993,

[de Garis 1994] "An Artificial Brain : ATR's CAM-Brain Project Aims to Build/Evolve an Artificial Brain with a Million Neural Net Modules inside a Trillion Cell Cellular Automata Machine", Hugo de Garis, New Generation Computing Journal, Vol. 12, No. 2, 1994, Ohmsha & Springer Verlag.

[de Garis 1995] "The CAM-Brain Project : The Genetic Programming of a Billion Neuron Artificial Brain by 2001 which Grows/Evolves at Electronic Speeds inside a Cellular Automata Machine", Hugo de Garis, Int. Conf. on Artificial Neural Networks and Genetic Algorithms, April 1995, Ales France.

[de Garis 1996] "CAM-Brain : ATR's Billion Neuron Artificial Brain Project : A Three Year Progress Report", Hugo de Garis, Int. Conf. Evolutionary Computation, May 1996, Nagoya, Japan.

[de Garis 1997] "One Chip Evolvable Hardware : 1C-EHW", Hugo de Garis, Int. Conf. on Artificial Neural Networks and Genetic Algorithms, April 1997, Norwich, UK.

[de Garis et al 1997] "Designing an Artificial Brain with 10,000 Evolved Neural Network Modules : Initial Thoughts, Hugo de Garis, Sung-Bae Cho, Michael Korkin, Arvin Agah, IJCAI (Int. Joint Conf. on Artificial Intelligence) Workshop on Evolvable Systems, August 1997, Nagoya, Japan.

[deGaris et al 1997] "Million Module Neural Systems Evolution : The Next Step in ATR's Billion Neuron Artificial Brain (CAM-Brain) Project", Hugo de Garis, Lishan Kang, Qiming He, Zhengjun Pan, Masahiro Ootani, Edmund Roland, Artificial Evolution Conf, Oct 1997, Nimes, France.

[Gers & de Garis 1996] "CAM-Brain : A New Model for ATR's Cellular Automata Based Artificial Brain Project", Felix Gers & Hugo de Garis, Int. Conf. on Evolvable Systems, October 1996, Tsukuba, Japan.

[Gers et al 1997] "CoDi-1bit : A Simplified Cellular Automata Based Neuron Model", Felix Gers, Hugo de Garis, Michael Korkin, Artificial Evolution Conf, Oct 1997, Nimes, France.

[Korkin & de Garis 1997] "CBM (CAM-Brain Machine) : A Hardware Tool which Evolves a Neural Net Module in a Fraction of a Second and Runs a Million Neuron Artificial Brain in Real Time", Michael Korkin & Hugo de Garis, Genetic Programming Conference, July, 1997, Stanford, USA.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Learning and relocation capabilities of CAM-brain machine

Hitoshi Hemmi, Kazuhiko Shinozawa, Tomofumi Hikage and Katsunori Shimohara
Evolutionary systems group
NTT Human Interface Research Laboratories
1 - 1 Hikarinooka Yokosuka, 239 Japan
{hemmi, hikage, katsu }@nttcvg.hil.ntt.co.jp
shino@aether.hil.ntt.co.jp

Abstract

Two techniques to enhance capabilities of CAM-brain machine are proposed. The first one is learning capability which is realized by providing "decay register" in each neuron cell. The second one is neural network relocation capability which make it possible to compact the evolved neural network and make room for further evolution. Both techniques are operated extrinsic manner, and considered supplement to intrinsic evolutionary capability of CAM-brain.

1 Introduction

CAM (Cellular Automata Machine)-brain ([1], [2]) is a project to construct a huge neural network in cellular automata (CA). In the CAM-brain system, the axons and dendrites of the network are implemented as trajectory-like patterns in the CA space, and signal transition in them are realized as propagation of CA state along with the pattern. So the neural network in the CAM-brain has very high flexibility for its structure. The issue arising here is how to determine the structure of the neural network. The original idea of CAM-brain is to use evolutionary engineering techniques; The neural network is evolved according to the criterion of how well its operation fits the problem applied. This strategy is reasonable, because there are not any other known technique that can make huge neural network to adapt to particular problem efficiently. Although, in generous, evolutionary engineering techniques may works well, it is not a good idea only to rely on these.

One reason to this claim is that one of the most important aspect of neural network is its learning capability. The original idea of CAM-brain do nothing about this direction. In this paper, we propose a technique which bring learning capability to CAM-brain.

Another reason to the previous claim is that the evolutionary techniques cannot discriminate functionary equivalent networks. Especially, it seems to be very difficult to eliminate redundant portion of neural network by evolutionary techniques. The existence of such useless portion is not good for electric power consumption and efficiency of utilization of CA space. So we propose another technique that relocate neural network with preserving the functionality of the network.

The next section explain the learning capability of CAM-brain. Discussion of relocation capability follows.

2 Learning capability

In neural network study, learning process are often performed according to the activity of neurons. In this section, we discuss how to observe the activity of neurons in CAM-brain. In CBM (CAM-Brain Machine [3]), each neuron has a counter to implement firing mechanism.

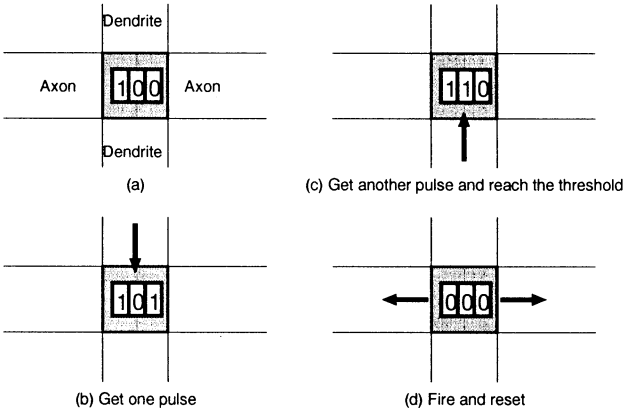


Figure 1: Firing mechanism of CAM-brain

The operation of this counter is like this (Fig. 1):

1) Each time the neuron get pulse along with dendrite, the counter counts up one.
2) When the counter reach the predefined threshold, the neuron fires and send out a pulse along with axons.
3) Then, the counter get reset.

It seems possible to measure the activity of the neuron by observing this counter .

One problem here is how to observe this counter. It is not good idea to add CBM design an wire for each neuron to observe counter value, because such wire from deep inside of large CA space cause not so small impact to chip area utilization efficiency. Instead of provisioning of wires, we can use sampling technique. The FPGA chips (XC6200 series) on which the CBM are implemented allow control/observation of inside flip flop values. This operations are achieved by borrowing the wires for chip configuration, so there are few penalty to use them. Of course number of flip flops which simultaneously can be handled are limited. However, sampling the value of each counter periodically, it is possible to determine average activity of the neuron.

Another problem for measurement of neuron activity by this technique is like this: if the counter of one neuron was in just below the threshold, and have not get any pulse for long time, then the activity of the neuron is mistakenly determined very high.

This difficulty can be solved by following technique (Fig. 2):
First, number of bits of counters are increased by adding supplement flip flops to its LSB side. For example, if original counter consists of 3 bits, by adding 5 bits to LSB side, total bits of counter will be 8 and the original LSB will be the fifth bit from current LSB.
Second, if the neuron gets a pulse, increment occur in the original LSB position (i.e in previous example, add 100000 (= 32 in decimal) in current counter).
Third, synchronizing the system clock, the counter decrement in current LSB position.

Using this technique, if the neuron did not get any pulse for a while, the counter decreased gradually. The number of bits of additional counter determines the decaying speed. We call this counter "decay counter".

Using techniques described above, we can determine the activity of the neuron in reasonable reliance. Determination of activity of neuron and learning processes are performed outside of CA space in independent with evolutionary process.

Learning strategy itself is another issue, and we will not discuss about it here. However, it can be said that

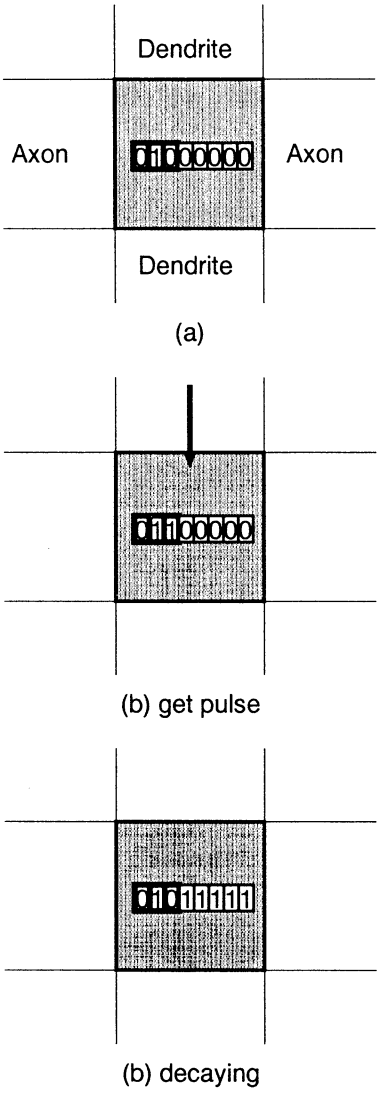


Figure 2: Operation of decaying register

using similar technique as next section, we can change connection weight between two neurons according to the activity.

3 Relocation capability

If we use evolutionary engineering techniques to construct a neural network in CAM-brain, resulting network usually has many redundant portion. In this section, we discuss a technique to avoid the redundancy and to utilize the CA space efficiently.

Figure 3 shows example neural network in CAM-brain which is actually constructed by evolutionary process. As can be seen, there are some branches of

axons and dendrites that do not form a synapse with other dendrites or axons. Such branches are not concerned in functionality of the neural network, while they consume the electric power when operate as well as functioning network portion. No doubt, it is also a waste of CA space. So it is good for system to remove such branches.

It may not be a good idea to prune such redundant branches intrinsically by CAM-brain operation; Implementation of such operation requires extra circuit in each CA cell, and this may deny the expected merits with this operation. On the contrary, it is easy and useful to detect and prune such branches by extrinsic operation. Figure 4 shows the result of such operation to neural network in Figure 3. Of course this operation is time consuming one and is not suitable to apply every individual networks in evolving stage. It is suitable to apply to resulting matured network by evolutionary process.

Once pruning is done to a neural network, there come to be a free space here and there. It is also possible to move neurons and synapses, shrink axons and dendrites to make compact network almost equivalent in functioning with original network. Figure 5 shows the result of this operation applied to the network in figure 4. In figure 5, there appear a collected free area in CA space, and this space can be used to allocate some neurons and to evolve additional network. (Fig. 6) Thus we can implement more functionality in the same CA space; I.e. we can utilize CA space more efficiently.

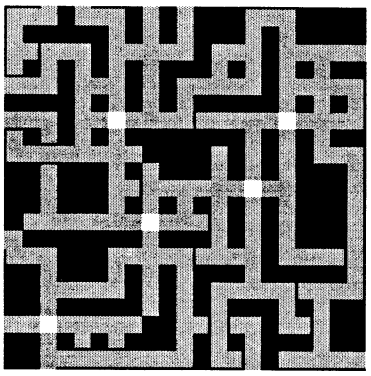


Figure 3: Sample neural network in CAM-brain

4 Summary

This paper propose two enhancement techniques for CAM-brain system. Using decay register, CAM-brain can get learning capability. Using network relocation technique, the electric power can be saved, or CA space can be utilize more efficiently.

References

[1] de Garis, H. "CAM-BRAIN: The Evolutionary Engineering of a Billion Neuron Artificial Brain by 2001 which Grows/Evolves at Electronic Speed Inside a Cellular Automata Machine (CAM)", in *"Towards Evolvable Hardware"*, Springer, Berlin Heidelberg, NY, 1996.

[2] Gers, F. & de Garis, H. "CAM-Brain: A New Model for ATR's Cellular Automata Based Artificial Brain Project", in *"Evolvable Systems: From Biology to Hardware"*, Springer, Berlin Heidelberg, NY, 1997.

[3] Gers, F., de Garis, H., & Korkin, M. "CoDi-1bit: a simplified cellular automata based neuron model", (submitted)

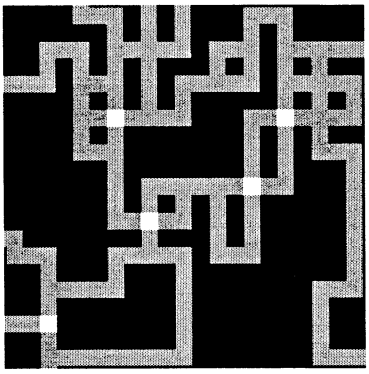


Figure 4: Pruned network

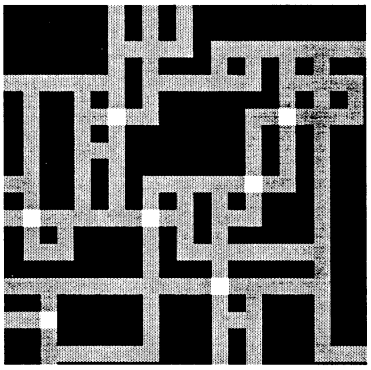


Figure 6: Enhanced network

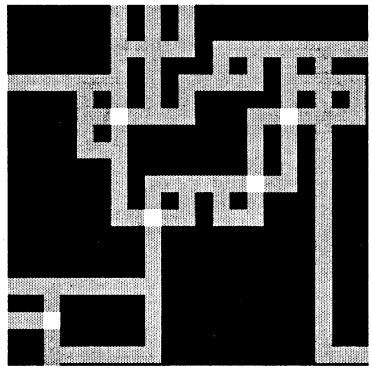


Figure 5: Relocated network

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Evolving CAM-Brain to Control a Mobile Robot*

Sung-Bae Cho, Geum-Beom Song, Jee-Haeng Lee and Seung-Ik Lee
Department of Computer Science, Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea

Abstracts

CAM-Brain is the model to create neural networks based on cellular automata (CA), and finally aims at developing an artificial brain. In particular, this model can rapidly evolve the neural networks composed of several thousand modules with special purpose computers such as CAM-8 at MIT and CBM at ATR. This paper attempts to evolve a module of CAM-Brain for a problem to control a mobile robot, especially Khepera, which might show the feasibility of evolutionary engineering to develop an artificial brain. A thorough experiment reveals a couple of problems that have to be solved when CAM-brain evolves to control a mobile robot, and the remedy is also proposed to solve them.

Keywords

Artificial Brain, Cellular Automata, Evolutionary Engineering, Mobile Robot Control.

1. Introduction

Recently, neuroscientists have put a lot of efforts into trying to understand and reconstruct the functions of brain. Most of the research in this field take bottom up approach, in which slices of visual and auditory cortex are extracted and investigated for constructing their possible functionality in a larger context. The basic idea is that the many little insights obtained from investigating each part would reveal the functionality of larger and more complex parts, finally giving us a broad understanding on brain.

However, it is sometimes impossible to extract a component and assign a certain function because the behaviors may result from a sophisticated interaction between many components. Since the brain has evolved to complex organization for a long time, evolutionary approach might provide a bigger possibility to the problem. For an instance, a brain builder group at ATR Human Information Processing Research Laboratories in Japan has attempted to develop artificial brain called CAM-Brain [1, 2].

This system is based on Cellular Automata (CA) and controls growth and signaling of neuron. In particular, due to the features of CA it is possible to evolve very quickly on parallel hardware. It is already proved that it can be implemented on MIT's CAM-8 machine [3, 4].

It is expected that several features of biological brain can be found in the CAM-Brain. This is because it would be much easier to investigate artificial brains than biological brains. This engineering solutions might give us another insight on working mechanisms of the evolved system as a whole [3].

Evolutionary engineering (EE) is an approach to combine neural network modules that has evolved with particular functions to develop artificial brain. Recently it has been extensively exploited to apply each neural network module to a given problem. As a possible application, this paper attempts to apply the CAM-Brain to control a mobile robot and reports a preliminary result.

2. Khepera : Mobile robot

Khepera robot (see Fig. 1) contains 8 infrared sensors to detect by reflection the proximity of objects in front of it, behind it, and to the right and the left sides of it, and to measure the level of ambient light all around the robot. Also, the robot has two motors to control left and right wheels. Khepera simulator (see Fig. 2) also features the ability to drive a real Khepera robot, so that we can very easily download our simulation results to the real robot.

Each sensor of Khepera simulator returns a value ranging between 0 and 1023. 0 means that no object is perceived, while 1023 means that an object is very close

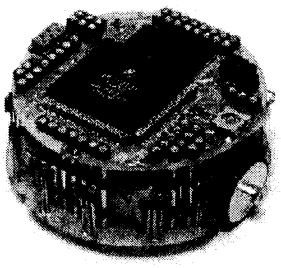


Fig. 1 Khepera robot.

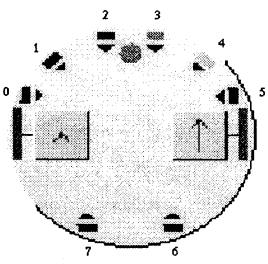


Fig. 2 Simulator.

* This work was supported in part by a grant no. SC-13 from the Ministry of Science and Technology in Korea.

to the sensor. Intermediate values give an approximate idea of the distance between the sensor and the object. Each motor can take a speed value 5, 0 and -5. We construct an environment in order to observe the behavior of the robot. Initially, the robot is located below the central point and the problem that robot must workout is to go round this world without bumping against walls [5, 6].

3. Evolved CA-based Networks

One neural network module is defined by CAM-Brain model based on CA, and formed by its own chromosome that has information about CA-cell structure. One chromosome is mapped to exactly one neural network module. Therefore, with genetic algorithm working on this chromosome, it is possible to evolve and adapt the structure of the neural network to a specific task. It is basic idea of "CAM-Brain Project" that brain-like system can be made by combining many neural network modules that have various functions [1]. This section illustrates a design of CA-space for developing a neural network module.

3.1 Applying CA-Space to CAM-Brain

Neural structure of CAM-Brain is grown inside 2-D or 3-D CA-space, and two different CAs are used for the growth phase and the signaling phase [7]. The growth phase makes the neural network structure inside CA-space and signaling phase transmits the signal in CA-space. Each cell has one of four types (Neuron, Axon, Dendrite, Blank), individual chromosome information, and cell state interface to the neighbors.

3.2 Formation of Chromosome

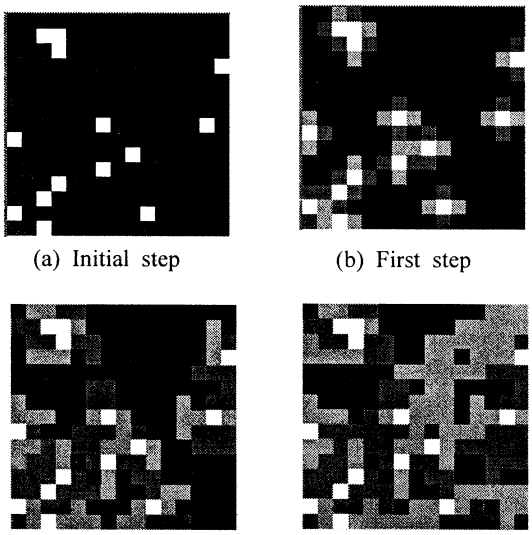
Chromosome is formed with the set of materials about each cell. Each cell propagates the growing signals to neighborhood decided by individual chromosome information. Also the starting positions of the seed cells are specified by chromosome. One chromosome makes the unique neural network module because the starting point and the signal direction are specified by chromosome. Therefore chromosome plays a role of the map of neural network, and the operations such as mutation and crossover effect a partial change of the neural structure. Initial chromosome is formed randomly and evolves by genetic algorithm.

3.3 Growth Phase

The growth phase organizes neural structure and makes the signal trails among neurons. Fig. 4 shows growing inside 2-D CA-space. Initially, all cells are set to blank type, and some cells are decided as neuron-seed cells by chromosome information. Neuron cells are made only a period of initial state as shown in Fig. 4(a). Neuron cells send two kinds of growth signals to their neighbors, either "grow a dendrite", or "grow an axon".

(a)Neuron (b) Dendrite (c)Axon (d)Blank

Fig. 3 Cell state.



(a) Initial step (b) First step

(c) Intermediate step (d) Final step

Fig. 4 Growth of neural structure.

The blank neighbors, which receive a neural growth signal, turn into either an axon cell or a dendrite cell. Fig. 4(b) shows the growing axon and dendrite by growth signals. The grown cells never change it's type and send the signal to it's neighbor blank cell. Fig. 4(c) shows the progress of growing axon and dendrite, and Fig. 4(d) shows the completion of one neural network module. Each axon and dendrite cell belongs to exactly one neuron cell. The weighting is expressed indirectly via the length of the dendritic trail that a signal has to pass through.

3.4 Signaling Phase

Signaling Phase transmits the signal from input cells to output cells continuously. The trails of signaling are performed with already made structure on the growth phase. Dendritic cells collect data and eventually pass it to the neuron body. Axonic cells distribute data originating from the neuron body. Input cells and output cells can be arbitrarily decided, so chromosome adapts to deal with effective signaling from input to output cells. During signaling phase, fitness evaluation is executed. Due to targeted task, various methods are used, such as the number of activated cells, hamming distance of target and output vector, or function to evaluate fitness. This fitness is used for evolving chromosomes.

4. Simulation

Khepera robot simulator is programmed with C++, and the experiment has been performed on Sun SparcStation 10. The population size is 100, and the maximum steps for sensor sampling is 30000 steps to find the fittest one.

4.1 Problems

There are a couple of problems to apply the model to controlling robot. One is that CAM-Brain does not perfectly utilize activation values of robot sensors. After a sensor value (scaled between 1 to 32) that is greater than threshold, enters CAM-Brain, the corresponding neuron sends 1 and -1 to axons. This can be thought of activation value of robot sensor as scaled between -1 and 1 so that it cannot reflect various situations. The other problem is that delay time is needed until CAM-Brain makes output value. After sensor values are inputted to input cells some steps are needed until this value arrives at output cells. It hinders the robot from reacting promptly.

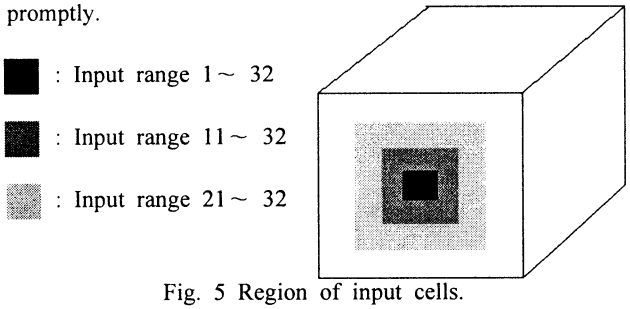


Fig. 5 Region of input cells.

4.2 Solutions

The first problem has been solved by dividing input range further as shown in Fig. 5. The number of reacting input cells varies according to the size of input value. When input value is greater than 21, the region of input cells gets to be the largest. On the other hand, when input value is less than 11, the region of input cells is the smallest. The other problem is solved by executing dummy signaling phase for some duration until signals started from input cells arrive at output cells. This enables the timely reaction of robot according to situation.

4.3 Environment

We use 7X7X7 CA-space that solves the problem. Only four sensors are used in this simulation and each input cells are in center region of four face of hexahedrons of CA-space (see Fig. 5). Output cells are in top and bottom faces of hexahedrons. Output cells of top and bottom faces produce the speed of left and right motors, respectively. One robot completes the growth phase with a chromosome and then starts receiving inputs. Only neuron cells can take inputs. A signaling phase is performed for some steps per one sensor sampling time unit.

We make use of a simple method for fitness evaluation. Let the center of simulator space be O, robot's starting point be S, and robot's present location be N, then fitness is given like this.

$$Fitness = \frac{\angle SON}{2\pi}$$

This fitness leads robot's clockwise rotation. Once robot goes round the simulation space, then fitness becomes 1.0 and robot stops moving. If robot crashes to the wall then stop movement, and evaluates fitness on that position. Even if we do not give any knowledge, such as "turn right", "turn left" and "avoid bumping," CAM-Brain guides the evolution naturally to solve the problems.

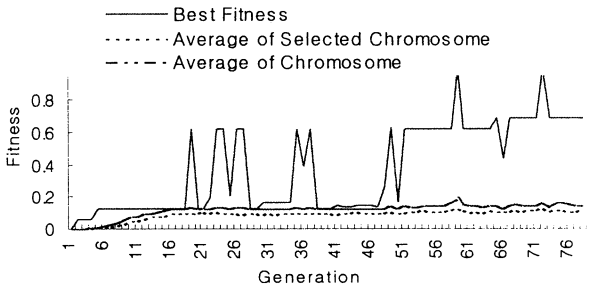


Fig. 6 Change of fitness.

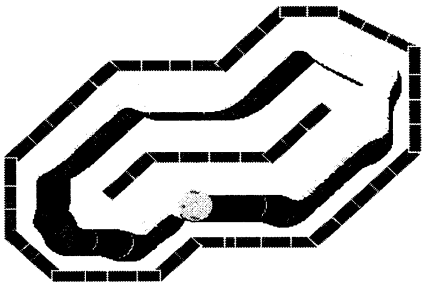


Fig. 7 Reaching target spot.

4.4 Result and Analysis

Fig. 6 shows the change of fitness with increased generations. At the beginning, the fitness does not change, but it increases radically after the 18th generation. At the 59th generation, we obtain the robot which moves around the environment without bumping against the walls (Fig. 7) and we get another robot at the 72nd generation.

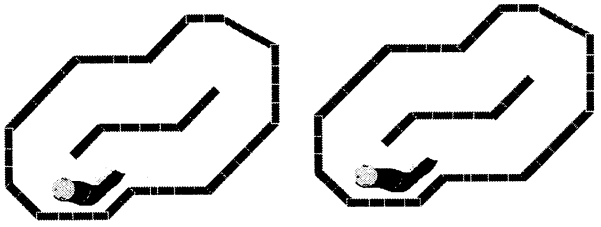
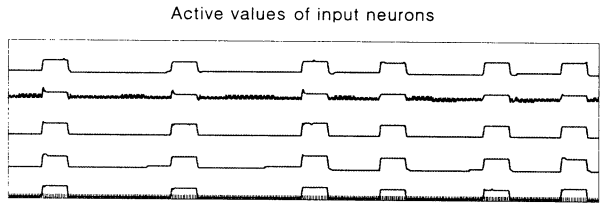
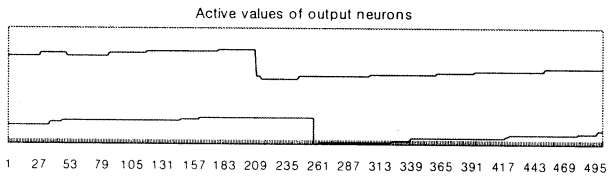


Fig. 8 Move straight.



(a) Input neurons.



(b) Output neurons.

Fig. 9 Active values of neurons when the robot moves straight.

Fig. 8 shows trajectory of the robot which moves straight. Fig. 9(a) shows the active values of input neurons from sensor 0, because it works mainly to robot movement. Fig. 9(b) shows active values of output neurons. Each neuron produces similar values, so that the left and right motors yield similar velocity. This is makes the robot to move straight.

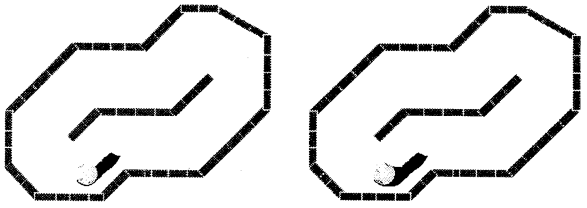
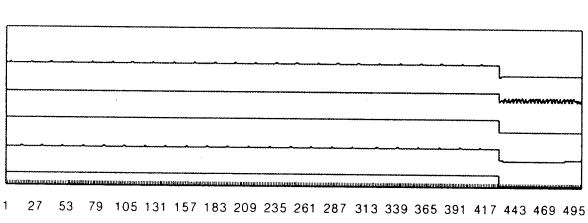
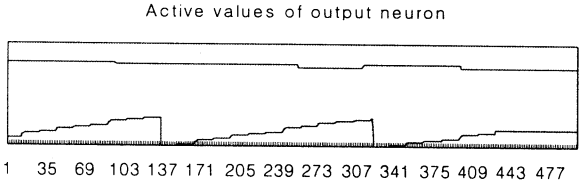


Fig. 10 Avoid bumping.



(a) Input neurons



(b) Output neurons

Fig. 11 Active values of neurons when the robot avoid bumping.

Fig. 10 shows trajectory of the robot which avoid bumping. Fig. 11(a) shows active values of input neurons from sensor 0. As the robot gets close to wall, all input values become high. Fig. 11(b) shows active values of output neurons. The two output neurons produce different values. The velocity of left wheel is almost 5 because active value of below output neuron (left wheel) gradually increases. While the velocity of right wheel is almost 0 or -5 because active value of upper output neuron (right wheel) is unchanged or decreased. This makes the robot turn right to avoid bumping.

The trajectory of the robot which reaches the target spot is shown in Fig. 7. Analysis of the fittest robot's neural structure reveals that two output cells become neuron cells to be able to produce outputs. This result has come from that most of the robot's movements consist of turning right and going straight. Even though the fitness does not directly reflect robot's speed and efficiency of movement, Kephera controller evolved by CAM-Brain solves the problem successfully.

5. Concluding Remarks

This paper presents the evolved neural network based on CA, and applies it to mobile robot control. Also, we attempts to observe evolving processes of neural network and behavior of the robot. CAM-Brain model has organized the neural structure to solve the problem. We expect to be able to solve more complex problems. To achieve this goal, we have to investigate possible mechanisms for learning the CAM-Brain model. Furthermore, we should devise a method to integrate many neural network modules evolved.

References

[1] H. de Garis, "CAM-Brain : ATR's billion neuron artificial brain project : A three year progress report," Proc. *Int. Conf. on Evolutionary Computation*, Nagoya, Japan, May 1996.

[2] M. Korkin, H. Garis, F. Gers and H. Hemmi, "CBM (CAM-Brain Machine) : A hardware tool which evolves a neural net module in a fraction of a second and runs a million neuron artificial brain in real time," *Proc. Genetic Programming Conference*, Stanford, USA, July 1997.

[3] F. Gers and H. de Garis, "Porting a cellular automata based artificial brain to MIT's cellular automata machine 'CAM-8'," *Proc. SEAL '96*, Taejeon, Korea, November 1996.

[4] N. Margolus, "CAM-8 : A computer architecture based on cellular automata," MIT December 1993.

[5] O. Michel, "Khepera simulator version 1.0 user manual," 1995.

[6] S-I. Lee and S-B. Cho, "Evolutionary learning of fuzzy controller for a mobile robot," *Proc. Int. Conf. Soft Computing*, Iizuka, Japan, pp. 745-748, 1996.

[7] F. Gers, H. de Garis, "CAM-Brain : A new model for ATR's cellular automata based artificial brain project," *Proc. Int. Conf. on Evolvable Systems*, Tsukuba, Japan, October, 1996.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Artificial Brain for a Mobile Vehicle

Masanori Sugisaka and Xin Wang

Department of Electrical and Electronic Engineering
Oita University, 700 Dannoharu, Oita 870-11 Japan
Tel: +81-975-547831, Fax: +81-547841, E-mail: msugi@cc.oita-u.ac.jp, wangxin@cc.oita-u.ac.jp

Abstract

In this paper, we present a primitive artificial brain which consists of a 32-bit Von Neumman type microcomputer NEC PC-9821Ap/U7, a *RICOH* neurocomputer *RN*-2000, two PIO 32/32 interfaces, the software for constructing a neural network and learning, and the software for steering two front wheels of a mobile vehicle. The primitive artificial brain developed in our laboratory which was used in the mobile vehicle, is presented. Details are given about how to use the primitive artificial brain to control the steering action of the mobile vehicle.

Keywords: Neurocomputer, Artificial Brain, Control, Mobile vehicle

1 Introduction

Modern medical science has shown us that the brain of human being is a mysterious complex system[1]. To build a so-called artificial brain is an attractive project as well as a difficult one. It is a better way to make the artificial brain by learning the real brain of human being[2]. We have developed a primitive artificial brain for the purpose of steering two front wheels of a mobile vehicle called neural mobile vehicle (denoted NMV). The configuration of the NMV is shown in Fig.1. In this configuration, the part enclosed by the solid line constitutes the primitive artificial brain. As shown in Fig.1, the primitive artificial brain consists of the 32-bit Von Neumman type microcomputer NEC PC-9821Ap/U7, the RICOH neurocomputer RN-2000 (called Yamato), two PIO 32/32 interfaces, the software for constructing a neural network and learning (called Nadeshiko), and the software for steering two front wheels. The original idea how to construct the primitive artificial brain was developed[3] for tracking a moving object in the system for the recognition and tracking[3,4].

In this paper, we discuss a primitive artificial brain developed in our laboratory for the NMV. It is used to control the steering angle of two front wheels in the NMV. We will show the neural network which is used in the artificial brain for steering two front wheels of the NMV, image processing method, and control strategy in this paper.

2 Neural network for the steering control

Let $\Psi=\{x_1(i),\cdots,x_m(i)\}\in R^m$ be a vector containing m measurements representing the states at time index i and $\Omega=\{y_1(i),\ldots,y_n(i)\}\in R^n$ be a output vector which responses those inputs. The control system in the primitive artificial brain used is shown in Fig.2. The problems are how to design a training pattern, how to select the layers of the neural network, how to decide the neurons in each and how those inputs after image processing are transferred into the neurons in the input layer. It is necessary to select the parameters of the neural network in the *RICOH* neurocomputer *RN*-2000. These parameters are numbers of the neurons in the input layer, numbers of the neurons in the output layers, intermediate layers in the neural network, numbers of the neurons in each intermediate layer, learning rate of the back propagation method used in the RN-2000[5,6]. The RICOH neurocomputer RN-2000 has four layers, one input layer, two hidden layers and one output layer. Though that available neurons in each layer can be up to 16, and input patterns are 64 are designed. By our experiences with the neurocomputer RN-2000, we used 8 neurons in the input layer, and one output neuron. The umbers of the neurons in two intermediate layers are 16 respectively, this selection is for the purpose of utilizing the capability of the hardware as possible as we can. The possible learning rate, which is used to adjust the weights in the neural network, is from 0.1 to 1.0. Based on the experiments, we found no big difference when the learning rate was changed, so 1.0 was selected. The initial weights of the neural network in the neurocomputer RN-2000 is set randomly.

The neurocomputer RN-2000 required special formats of teaching patterns. The neurons in the input layer can only accept one of two values 0 or 127 and the output of the neurons in the output layer range from 0 to 127. That means the hardware of neurocomputer RN-2000 required that the values inputted to the neurons in the input layer must be a value of 0 or 127. So the data

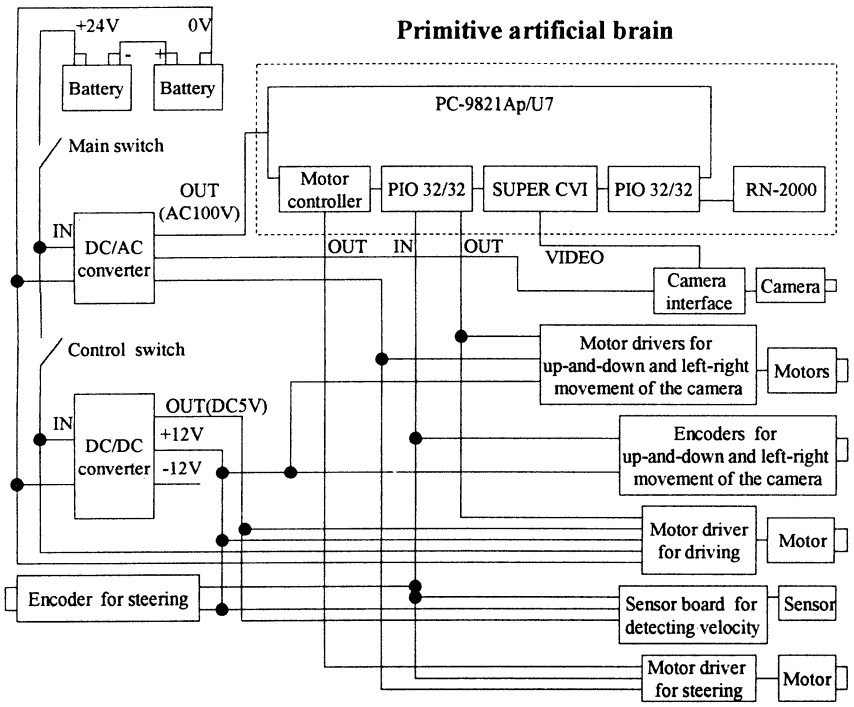


Figure 1: The configuration of the neural mobile vehicle (NMV)

Table 1: The data of the neurocomputer: $\alpha_T(t)$; is teaching signal; $\alpha_L(t)$; is learned signal

$\beta(t)$	$\beta_n(t)$	Input pattern	$\alpha_T(t)$	$\alpha_L(t)$
0∼ 6.9°	0	00000001	10	10
7∼ 13.9°	1	00000010	20	19
14∼ 20.9°	2	00000100	30	31
21∼ 27.9°	3	00001000	40	38
28∼ 34.9°	4	00010000	50	53
35∼ 41.9°	5	00100000	60	57
42∼ 48.9°	6	01000000	70	68
49∼ 55.9°	7	10000000	80	76
56∼ 62.9°	8	00000011	90	92
63∼ 69.9°	9	00000110	100	100
70∼ 77.9°	10	00001100	110	109
78°≤	11	00011000	120	115

after image processing have to be changed in order to form 0 or 127 input patterns.

3 Experiments and discussion

The mechanical configuration of the NMV is shown in Fig.3. The environmental information after imaging processing is concentrated into an image scene shown in Fig.4. The image of an object is divided into three parts, concerning the centers of gravity caught in each part respectively. This method of image processing is good to use more image information. There are three cases to decide the final center of gravity of an object.

1. Three centers of gravity of the object are caught. That is $\xi_p(t) \neq 0$, $\eta_p(t) \neq 0$ (p=1, 2, 3),

$$\xi(t) = \frac{1}{2}\{\frac{1}{2}[\xi_1(t) + \xi_2(t)] + \frac{1}{2}[\xi_2(t) + \xi_3(t)]\} = \frac{1}{4}[\xi_1(t) + 2\xi_2(t) + \xi_3(t)] \quad (1)$$

$$\eta(t) = \frac{1}{2}\{\frac{1}{2}[\eta_1(t) + \eta_2(t)] + \frac{1}{2}[\eta_2(t) + \eta_3(t)]\} = \frac{1}{4}[\eta_1(t) + 2\eta_2(t) + \eta_3(t)] \quad (2)$$

2. One and only one center of gravity of the object does not exist. That is $\xi_p(t) = 0$, $\eta_p(t) = 0$ (p=1 or 2 or 3),

$$\xi(t) = \frac{1}{2}\sum_{p=1}^{3}\xi_p(t) \quad (3)$$

$$\eta(t) = \frac{1}{2}\sum_{p=1}^{3}\eta_p(t) \quad (4)$$

3. One and only one center of gravity of the object exist. That is $\xi_p(t) \neq 0$, $\eta_p(t) \neq 0$ (p=1 or 2 or 3).

$$\xi(t) = \xi_p(t) \quad (5)$$

$$\eta(t) = \eta_p(t) \quad (6)$$

Therefore, the final center of gravity of the object [$\xi(t)$, $\eta(t)$] are decided.

The image scene is divided into 24 fan sections. The fan sections are symmetrical about center line. Therfore, we discuss only half a image scene on the left. The fan angle in each fan section is about 90°/12=7.5°.

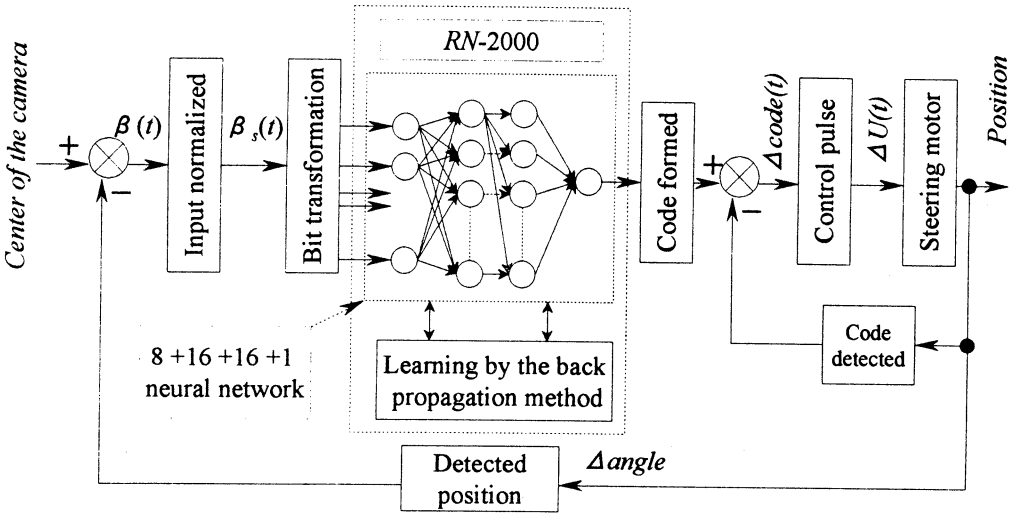


Figure 2: The control system of an artificial brain for the NMV

Let the angle $\beta(t)=tan^{-1}[319-\xi(t)]/[400-\eta(t)]$. If we normalized the $\beta(t)$ as $\beta_n(t)=tan^{-1}[319-\xi(t)]/[400-\eta(t)]/7$, then $\beta_n(t)$ is the input variable and is ranged from 0 to 11. So 24 steering patterns correspond to 12 teaching patterns in the RICOH neurocomputer RN-2000. One of requirements of the neurocomputer RN-2000 is that all of the variables to the neurons in the input layer must be 0 or 127. Therefore, $\beta_n(t)$ has to be changed into formal pattern i.e. input pattern. The output of the neurocomputer RN-2000, $code(t)$, is the steering code of $(\xi(t),\eta(t))$ at time index t. Since the reference steering code is the center code of the steering wheels, so the real steering code $code(t)$ is

$$code(t) = CENT + P_n \times \frac{\alpha_T(t) - 10}{2} \qquad (7)$$

where $CENT$=115 is center code of the steering wheels, P_n is decided by

$$P_n = \begin{cases} -1 & 0 \le \xi \le 319 \\ 1 & 319 < \xi \le 638 \end{cases} \qquad (8)$$

If we want to steer the neural mobile vehicle, it is necessary for us to detect the code of the steering wheels at the time index $t-1$, which is presented as $code(t-1)$. $code(t-1)$ is detected by means of the encoder of the steering motor. The control code $\Delta code(t)$ is the difference between the two codes,

$$\Delta code(t) = code(t) - code(t-1) \qquad (9)$$

In the controller, the mathematical symbol of $\Delta code(t)$ presents the steering directions. If $\Delta code(t) = 0$, no direction is changed, and if $\Delta code(t) > 0$, steer left else right. The steering motor in the mobile vehicle is stepping motor. From Fig.5, the relationship between the steering pulses to the motor $\Delta U(t)$ and steering code $\Delta code(t)$ is established.

$$\Delta U(t) = RATE \times \Delta code(t) \qquad (10)$$

where $RATE$=20 is decided by the experiments. This is a nonlinear proportional system, and the contro' 'eatures are shown in the Fig.5.

5 Conclusions

This paper showed the neural network configuration of the primitive artificial brain. A description of how to use the developed artificial brain to control the steering of the NMV is discussed. Compared with other controller, the neural network controller in the artificial brain has no obvious difference with other controllers such as PID controller, fuzzy controller etc., but has property of on-line learning capability and then facilitate advanced intelligent control. The results proved that the primitive artificial brain is successful.

References

[1] Sato M(1996) Brain, neural system and action (in Japanese). Iwanamisyoten, Tokyo, pp 25-85

[2] Sugisaka M(1997) "Neural networks for control in an artificial brain of recognition and tracking system", *Proc. of Int. Symposium on Artificial Life and Robotics*, Beppu, Oita Japan, Feb.18-20, pp. 99-102

[3] Sugisaka M(1997) Neurocomputer control in an Artificial brain for teaching moving object, Artificial Life and Robotics. 1(1):47-51

[4] Sugisaka M(1992) Recognition and tracking sysytem of moving objects. Patent Tohkaihe: 5-2333062 (in Japanese)

[5] Sugisaka M, Wang X, Lee JJ(1997) "Intelligent control strategy for a mobile vehicle with neurocomputer" *Proc. the* 12^{th} *Korea Automatic Control Conference*, Seoul Korea, Oct. 17-18, pp. 815-818

[6] Sugisaka M, Wang X, Lee JJ(1997) "Intelligent control strategy for smooth running a mobile vehicle with neurocomputer" *Proc. the 12th Int. Conference of System Engineering*, Coventry UK, Sept. 9-11, 2:664-669

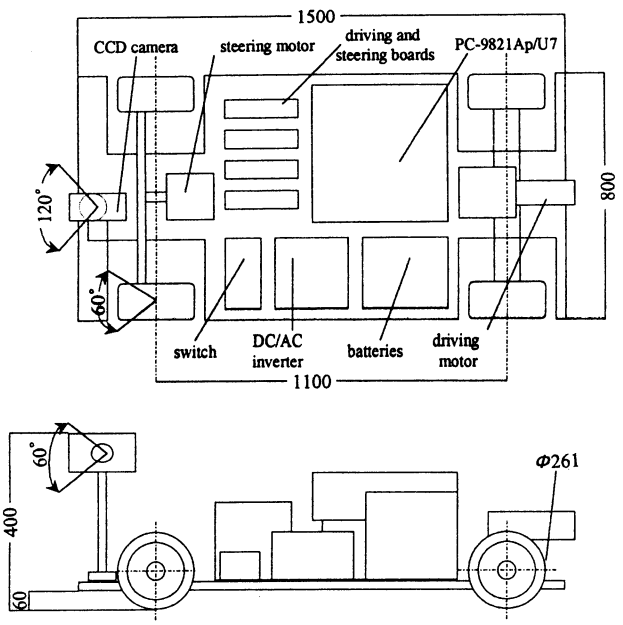


Fig. 3: The mechanical configuration of the NMV

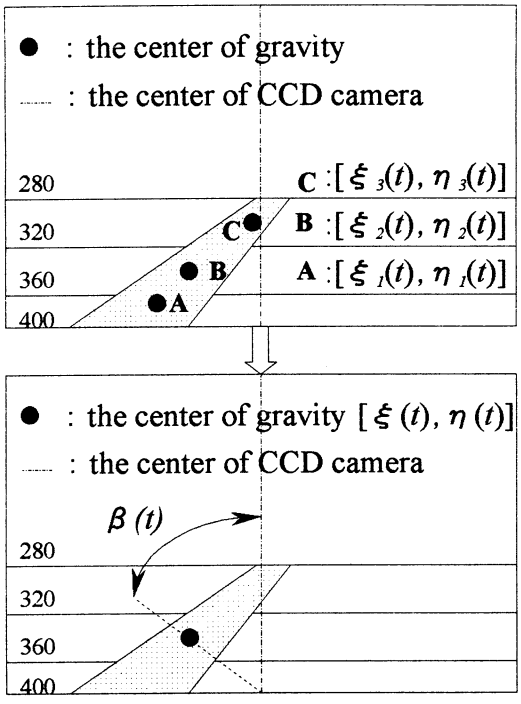


Fig. 4: Image scene in the NMV: left-up point is location (0, 0), and right-down point is (640, 400). The numeral values are the locations in vertical direction.

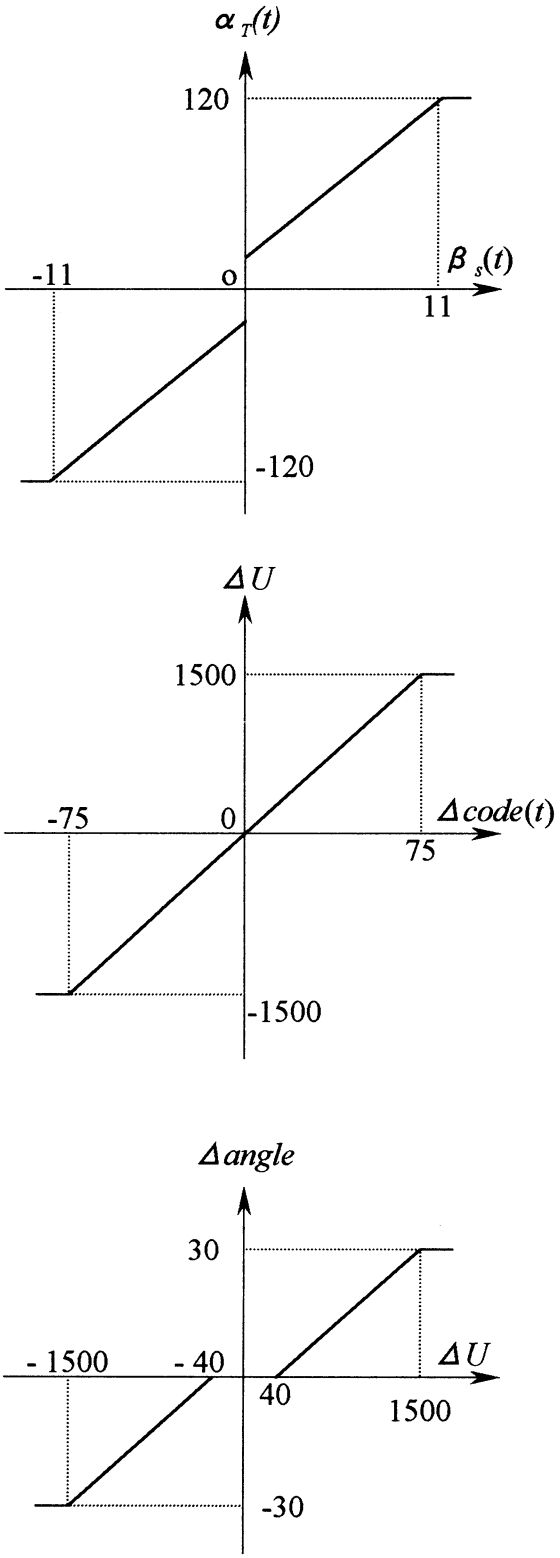


Fig. 5: Control features of the artificial brain for the NMV: $\Delta angle$ is the real angle of the steering wheels

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japa- 19-21, January, 1998,

Genetic Programming with Loops*

Qi Yuesheng Wang Baozhong Kang Lishan
(State Key Laboratory of Software Engineering, Wuhan University, China, 430072)

Abstract: Loop function in genetic programming is proposed , and the problems of how to present loops and how to avoid too many layers of loops in genetic programming are solved. The advantage of loop function in genetic programming is shown by the example of the mower problem.

Keywords: Genetic Programming, Loop Function, Mower Problem

Genetic Programming is an important branch of evolutionary computation. The idea of GP is mainly the same as that of GA. The difference between them is that the chromosome (or individual) in GP is a tree, other than a linear structure. This paper will add loop structure in GP. By introducing loop function, we have successfully solved the mower problem which appeared first in Koza[1]. We also compared the computational results with that of GP using no loop.

This article has four parts: part 1 is a brief introduction to GP, part 2 discusses the loop function fully, on its representation and the problems it is likely to cause and the method to deal with them, part 3 gives the computational results of the mower problem, part 4 is a plan for future work.

1 genetic programming

There is a famous formula: program=data structure+algorithm. When we try to solve a problem, we need not only the data but also a definite algorithm. Without an algorithm, that is to say, if we don't tell the computer every step in detail, we can not make the computer to work for us. Sometimes it is not easy to give a good algorithm for some problems. Can we manage to make the computer find a good algorithm itself? An algorithm can be represented as a tree, for example, cos(3+y)-a can be shown as parent1 in Fig. 1. On the other hand, when we have a tree, we have an algorithm. Using evolutionary method to make the computer obtain itself a satisfactory tree, a good algorithm, is just what genetic programming does. GP always take the following steps:

1. Determine the function set Fset. Fset is also the set of operators. All the nodes in a tree, except those leaf nodes, are elements of Fset, as cos, +,- in the above example. Every element (function) of Fset has one or more arguments. (We regard those functions having no arguments as elements of terminal set Tset.) These arguments can be shown as branches below this function node. The number of argument is also the degree of the corresponding node. The argument is also called operand, and the result of the function called the function value.

2. Determine the terminal set Tset. Tset is also the set of operands. All the leaf nodes are elements of Tset. As 3,a,y in the above example in Fig.1. To guarantee the tree's validity, any value of any function can be served as argument of any function of the Fset. This is the requirement of closeness of operation in GP.

The following steps are similar to GA[2].

3. Determine the evolution parameters, including maximum generation MaxGeneration, population of every generation PopulationNum, copy rate Crate and crossover rate Xrate, (mutate operation is not often used in GP). In GP , the maximum length of a tree, MaxLength, is needed to control the complexity of the tree. We also should determine the fitness function before going next.

4. Generate randomly PopulationNum individuals as the initial generation (Generation:=0),.

5. Evaluate every tree by the fitness function, if the

* The paper is supported by the National Natural Science Foundation and 863 High Technology Project of China..

best one satisfies the need, or Generation equals to MaxGeneration, stop the evolution: if not, go to 6.

6. By a certain selection strategy and copy and crossover operations, generate a new generation. Selection strategy should make the better individual have more chance to be selected. Copy operation adds the selected individual (parent) from the current generation into the new generation. Crossover operation works a little more complicated. Select two individuals from the current generation (these two can be same as each other) as parent1 and parent2. Randomly select one node in parent1 and get subtree1, randomly select one node in parent2 and get subtree2, exchange subtree1 and subtree2 and get two new individuals offspring1, offspring2, see Fig.1.

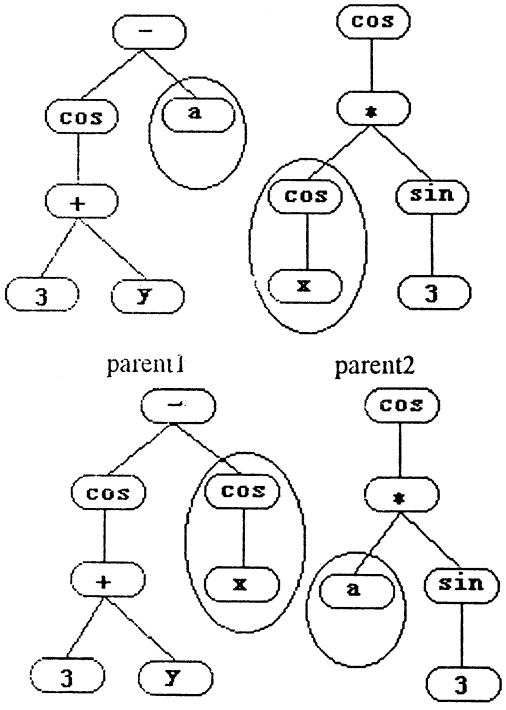


Fig.1

Add these two offspring into the new generation. Sometimes, one of the two offspring may be too long, have more than MaxLength nodes. When this happens, simply shorten the corresponding subtree. Repeat copy and crossover operation until the new generation has PopulationNum individuals.

7. Replace the current generation with the new generation. Generation increases by 1, return to 5.

2 Loop function

It has been proved that any program can be constructed by three simple structures: Sequence, branch and loop structure. Generally speaking, a tree in GP is a sequence structure with depth preference. ADF(Automatically Defined Function) in Koza[1] makes use of the technique of subroutine in programming. IF_ELSE function makes use of the branch structure. The importance of loop in programming is obvious. So we must make use of loop structure in GP if we want GP to result in good programs.

Anyhow, loop will cause two problems in GP:

1. How to represent loop?. A program is often represented as a tree. Loop is likely to be represented as a graph with cycles. Crossover between graphs with cycles must be very complicated.

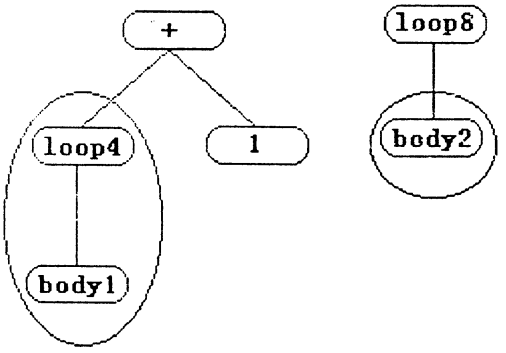
2. How to avoid too much layers of loop? Even if programs with loop can be represented as a tree, two programs with one layer loop can produce a program with nested loops by crossover . Too many layers of loops will surely cost much time and the efficiency must be very low.

Fig.2

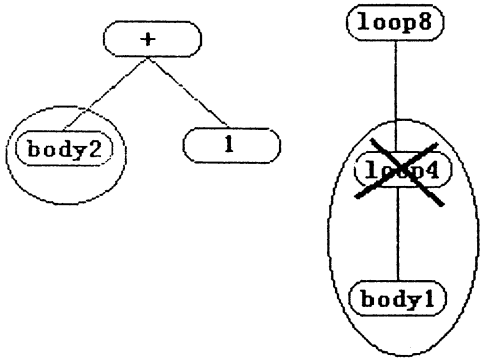
To solve these two problems, we designed loop function. If a loop body should run 8 times, we can use one of the following two functions : 1). fixed loop function Loop8; 2): variable loop function Loop(8), see Fig.2. So a tree can represent programs with loop.

How to solve the second problem, the nested loop problem? To make things simple, we forbid any nested loop. In the initial generation, we take such action: if a node is a loop function, we don't select any loop function as part of the loop body. Later, when crossover operation makes new individual for the new generation, we take such action: if one subtree contains loop function, we denote this subtree as type A, if one subtree is part of a

loop body, we denote this subtree as type B, before exchanging type-A subtree and type-B subtree, delete loop function of type-A subtree, see Fig.4. More subtle classification can limit programs have at most 2 or 3 lavers of loop.



parent1 parent2



offspring1 offspring2

Fig.3

3 Example of Application

Mower Problem appeared in Koza[1]. There is a mower on a 8 by 8 lawn. Its initial position is (0,0), facing north. The mower has two basic actions: Left and Mow. Left makes the mower turn left 90°, without changing its position; Mow makes the mower move one square forward and mow the grass in the square it just arrived if there is any grass in the square. We are asked to control the mower so that it can mow all the grass in the whole lawn, that is , to find a tree so that the mower can pass all the squares of the lawn.(passing one square two or more times is permitted). Suppose the lawn is toroid. As in Fig.5, when the mower moves north from square A(3,7), one of the most northern squares, it arrives at square B(3,0), the corresponding one of the most southern squares.

For such problem in our experiments, we define:

Function Set Fset={Loop4,Loop8,Prog};

Terminal Set Tset={Left, Mow, Right};

Loop4 is a loop function which makes the loop body run 4 times, degree(Loop4)=1;

Loop8 is a loop function which makes the loop body run 8 times, degree(Loop8)=1;

Prog is a function with two arguments, and executes the two arguments sequentially;

In this example, function values have no meaning to us, we ignore them.

Left ,Right and Mow is easy to understand.

We take other parameters as:

MaxGeneration=40;

PopulationNum=20;

MaxLength=250;

copy rate Crate=0.1;

crossover rate Xrate=0.9;

fitness=number of squares the mower passes.

We did six experiments. Table 1 gives the results of five experiments;

One of the six experiments failed. The best fitness was no more than 8 from generation 6, the reason was that all trees lacked Left and Right nodes from generation 6.

No.	generation when stops	best fitness	length of best individual	average fitness	average length
1	13	64	88	37	48.3
2	6	64	26	15.45	12.05
3	8	64	22	15.25	13.8
4	14	64	91	35.85	59.3
5	12	64	33	21.95	25.75

Table.1

Fig.4 is the structure of the best solution of the fifth experiment. Fig.5 shows the route of the mower

controlled by this program.

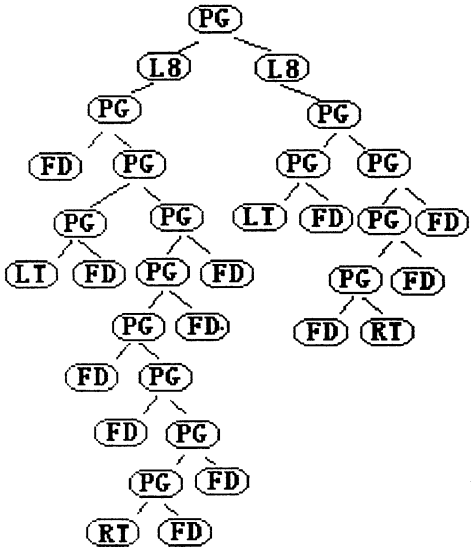


Fig.4

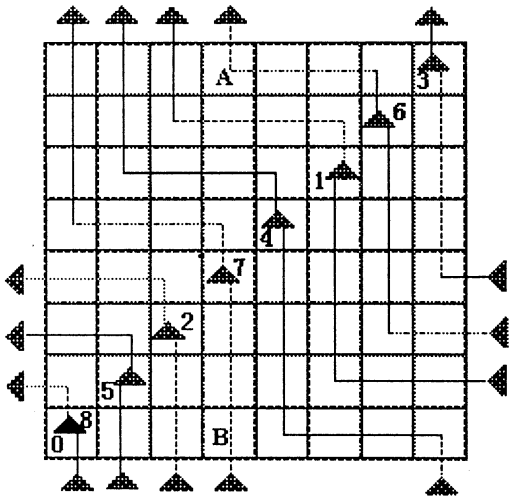


Fig.5

PG, L4, L8, LT, FD, RT are Prog, Loop4, Loop8, Left, Mow, Right respectively. This program has two branches, each is a loop function. It is interesting that the first branch can complete the mowing task. The digits in Fig.5 shows the end position of every time the loop body completes. .

Other experiments show that, when we delete Loop8 and Loop4 from Fset, we can not get an individual with fitness of 64, even if MaxLength increase from 250 to 450, MaxGeneration increases from 40 to 80 in our ten times trying. This demonstrates that loop function can improve the convergence rate and individual structure very much.

4 Future work

Loop plays a very important role in programming. It must appear in GP sooner or later this or that way. Now we propose the simplest form of loop, and hope this paper can interest other researchers in this area. In fact , much more work needs to be done on this subject.

1. How to determine the number of loop body running? The above example is simple, so we can determine the number by experience and intuition. Can we also evolve the number?

2.How to use the loop control variable in the loop body?

3. How to make good use of such loops like until or while, ?

4. How about combining loop function, ADF (automatically Defined Function) and IF_ELSE function?. We believe this will do much good to GP.

References:

[1]. Koza J. R., Genetic Programming II, Automatic Discovery of Reusable Programs. Cambridge, MA:MIT Press,1994.

[2]. Liu Yong, Kang Lishan, Chen Yuping, Nonnumerical Parallel Algorithm(Book II) Genetic Algorithm, Science Publishing House, 1995.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

A Fast TSP Solver Using GA on JAVA

Hiroaki SENGOKU
Systems Development Laboratory,
Hitachi, Ltd.
Kawasaki, 215 JAPAN

Ikuo YOSHIHARA
Graduate School of Engineering,
Tohoku University
Sendai, 980-77 JAPAN

Abstract

A hybrid algorithm using GA and heuristics for rapid solution of Travelling Salesperson Problems (TSP) is presented. We developed a JAVA program based on this algorithm. Visiting our web pages, everyone can easily try our TSP solver. Since JAVA programs are executable on many platforms, any one who design a new TSP algorithm can compare his own solver and ours on the same machine. Using our program as the criterion, TSP researchers can evaluate their algorithms objectively.

1 Introduction

The Genetic Algorithm (GA)[1] is an optimizing algorithm that is modeled after the evolution of organisms. Some of the GA's merits are that it can be easily developed because it does not require detailed knowledge about the problem, it can search globally, and it can adapt to the changing conditions in the problem.

Despite of these merits, GA is often slower than conventional methods such as heuristic searches. This is because GA does not utilize explicitly the knowledge of how to search for the solutions. Therefore, hybrid methods that combine GA with other techniques have been attempted.

The TSP solver we presented[2] is one of the hybrid methods. It uses GA and the 2opt method (section 2.1). Owing to the 2opt, it is much faster than other TSP solvers based on GA alone.

Generally, it is difficult to compare two approximate algorithms objectively. Some algorithms have the advantage of high speed, but have the disadvantage of a low success rate in finding the optimum solution.

The best way to compare two methods is by running them both on the same machine. But there are few such programs in public domain[4], and we have not found a program that can solve the problems listed in TSPLIB[5].

Consequently, we wrote a JAVA[1] program based on our proposed method, and made it public on our page on the World Wide Web (WWW). JAVA programs are executable on many platforms, so people who have developed a TSP solver can compare their programs and ours by running both programs on the same machine under equal conditions.

We also developed a graphical user interface. People can freely situate towns by using a mouse, and solve the problem they constructed. It's easy and fun. It's helpful for students studying the basics of computer algorithms.

In section 2, we explain our method. In section 3, we show the WWW page in which we present our TSP program.

[1] JAVA is a trademark of Sun Microsystems, Inc.

2 A TSP solver: 2optGA

We present a hybrid method[2] that uses GA and the 2opt method. In our GA, the 2opt method provides mutation, while the crossover operator provides the capability of jumping out from the local minima, where the solution often falls where only 2opt is used.

The algorithm consists of the following steps.

Initialization: Generation of M individuals randomly.

Natural Selection: Eliminate p_e% individuals. The population decreases by $M \times p_e/100$.

Multiplication: Choose $M \times p_e/100$ pairs of individuals randomly and produce an offspring from each pair of individuals. The population reverts to the initial population M.

Mutation by 2opt: Choose p_i% of individuals randomly and improve them by the 2opt method. The elite individual, or the individual that has the best fitness value in the population, is always chosen. If the individual is already improved, do nothing, because it cannot be further improved by 2opt.

We now describe the detail of each step.

2.1 Mutation by 2opt

The 2opt method is one of the most well-known local search algorithms among TSP solving algorithms. It improves the tour edge by edge and reverses the order of the subtour. For example, imagine a tour as shown in the upper part of Fig. 1. Remove the two edges $\overline{ab}$ and $\overline{cd}$, and reverse the order of the subtour (from b to c), and add the two edges $\overline{ac}$ and $\overline{bd}$. This gives us a tour as shown in the lower part of Fig. 1. The lower tour is shorter than the upper one because $\overline{ab} + \overline{cd} > \overline{ac} + \overline{bd}$.

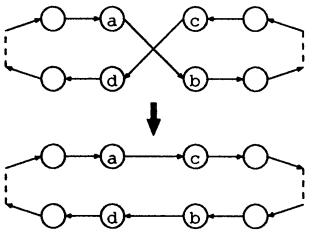


Figure 1: The 2opt method

We check every pair of edges, for example, $\overline{ab}$ and $\overline{cd}$. If $\overline{ab} + \overline{cd} > \overline{ac} + \overline{bd}$ holds, we improve them in the same way as shown in Fig. 1. Actually, if both $\overline{ac} > \overline{ab}$ and $\overline{bd} > \overline{cd}$ hold, then it is not necessary to check the edges. Therefore we can skip the pairs whose edges are far away from each other.

We repeat the procedures described above until no further improvement can be made.

2.2 Crossover in Multiplication

When we apply the 2opt method to a solution, the solution often falls into a local minimum. Then it cannot be improved further by the 2opt method. Consider two solutions that have fallen into different local minima. Potentially, each solution may have the best subtour for a different part of the tour. Then, we can make a better solution if we combine those best subtours appropriately. We cannot say, of course, which of the subtours are good, but after many trials, we can expect offspring solutions to be located in the valley of the global minimum as shown in Fig. 2.

We propose a new crossover operator that acquires the longest possible sequence of parents' subtours. We named it 'Greedy Subtour Crossover (GSX)'. We showed[3] by experiments that using the GSX, the solution can pop up from local minima more effectively than by using simulated annealing (SA) methods.

In the GSX, we use the path representation for a genetic coding. For example, the chromosome

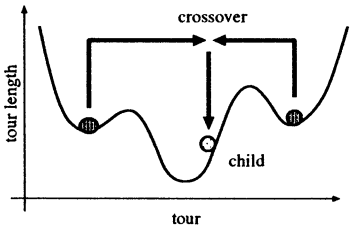


Figure 2: Pop-up from local minima.

$g = (D, H, B, A, C, F, G, E)$ means that the salesperson visits towns D, H, B, A, ..., E, successively, and returns to town D.

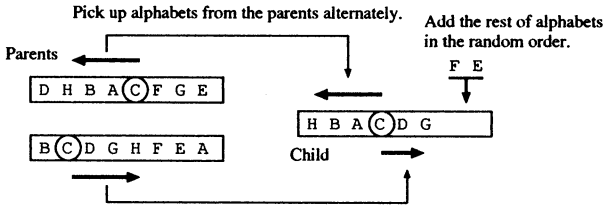


Figure 3: Greedy Subtour Crossover

Algorithm: Greedy Subtour Crossover
Inputs: Chromosomes $g_a = (a_0, a_1, \ldots, a_{n-1})$ and $g_b = (b_0, b_1, \ldots, b_{n-1})$.
Outputs: The offspring chromosome g.

```
procedure crossover(g_a,g_b) {
    f_a ← true
    f_b ← true
    choose town t randomly
    choose x, where a_x = t
    choose y, where b_y = t
    g ← t
    do {
        x ← x − 1  (mod n),
        y ← y + 1  (mod n).
        if f_a = true then {
            if a_x ∉ g then {
                g ← a_x · g,
            } else {
                f_a ← false.
            }
        }
        if f_b = true then {
            if b_y ∉ g then {
                g ← g · b_y,
            } else {
```

```
                    f_b ← false.
                }
            }
        } while f_a = true or f_b = true
        if |g| < |g_a| then {
            add the rest of towns
            to g in the random order
        }
        return g
    }
```

Note that "$\cdot$" in "$g \leftarrow a_x \cdot g$" is the concatenation operator, and that sentence means to add a_x before the chromosome g.

An example is shown in Fig. 3. Suppose that chromosomes of parents are $g_a = (D, H, B, A, C, F, G, E)$ and $g_b = (B, C, D, G, H, F, E, A)$. First, choose one town at random. In this example, town C is chosen. Then, $x = 4$ and $y = 1$ because $a_4 = C$ and $b_1 = C$ respectively. Now the child g is (C).

Next, pick up towns from the parents alternately. Begin with a_3 (town A) because $x \leftarrow 4 - 1 = 3$, and next is b_2 (town D) because $y \leftarrow 1 + 1 = 2$. The child becomes $g = (A, C, D)$.

In the same way, add a_2 (town B), b_3 (town G), a_1 (town H), and the child becomes $g = (H, B, A, C, D, G)$. Now the next town is $b_4 = H$ and town H has already appeared in the child (remember the salesperson may not visit the same town twice), so we can't add any more towns from parent g_b.

Therefore we add towns from parent g_a. The next town is $a_0 = D$, but D is already used. Thus we can't add towns from parent g_a, either.

Then, we add the rest of the towns, i.e., E and F, to the child in the random order. Finally the child is $g = (H, B, A, C, D, G, F, E)$.

2.3 Natural Selection

Eliminate $R = M \times p_e/100$ individuals. In the so-called simple GA, the possibility of survival is proportional to the fitness value, but we pay more attention to the diversity of the population. We eliminate similar individuals to maintain the diversity in order to avoid the immature convergence that is one of the well-known problems in GA.

First, sort the individuals in fitness-value order. Compare the fitness value of adjoining individuals. If the difference is less than ε (a small positive real number), eliminate preceding individual while the number of eliminated individuals is less than R. Let r be the number of eliminated individuals.

Next, if $r < R$, eliminate $R - r$ individuals in the order of lowest fitness value.

3 JAVA applet

3.1 Applet on WWW

We developed a JAVA applet based on the proposed method, and put it on our WWW page:

```
http://www.hitachi.co.jp
/Div/sdl/e-naiyo/e-seika22/TSP.html
```

It is easy to use. First, using a mouse, put "towns" in the rectangular field. Then, push the "start" button. The applet solves the problem you've just made and displays the tour and its length (Fig. 4). The parameters of the 2optGA method, i.e., "Population" (M), "Selection" (p_e), "2opt" (p_i), are adjustable by filling out each field on the applet.

We announced the web page on the Net News in Japanese (fj.* newsgroups) only once in July 1996. Since then, the page has been accessed more than 200 times a month.

One person who sent us comments had mistakenly thought that GA was a tool only for experiments, and that GA would be so slow that it could not be applied to practical problems. But after accessing our web pages, he could understand why GA is usable.

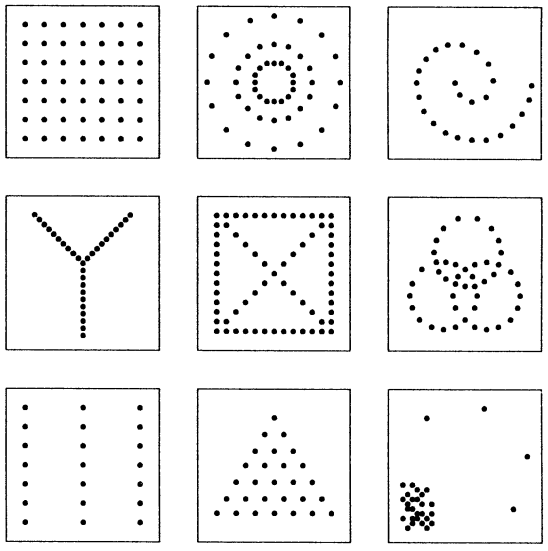
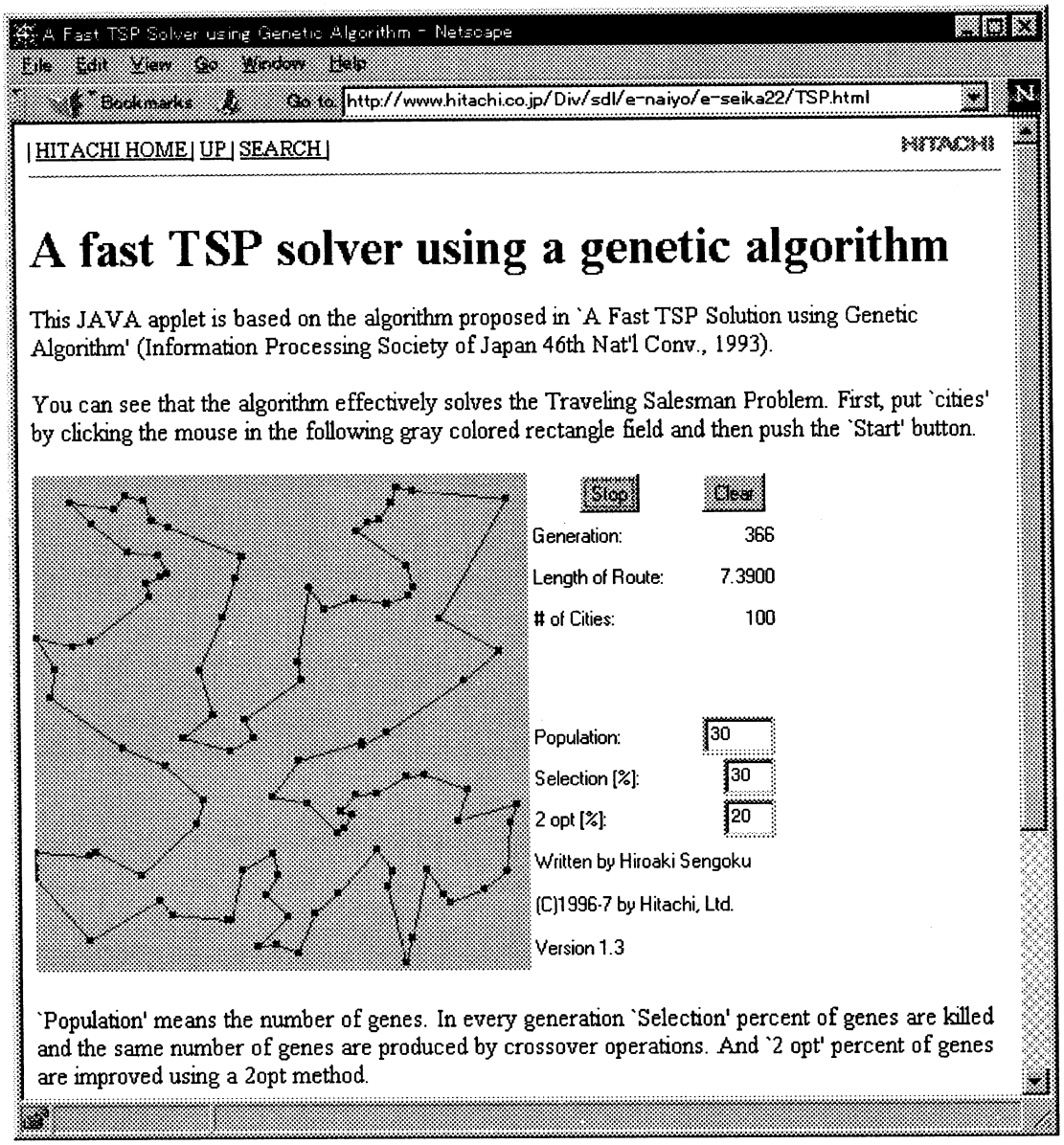


Figure 5: Examples of the problems the students made.

We utilized the web page for education. We introduced the page to the students majoring in computer science at the University of Tokyo and to the audience at a Hitachi employee seminar. They enjoyed mak-



Netscape is a trademark of Netscape Communications Corporation.

Figure 4: applet on WWW

ing new problems one after another and executing the applet. Some of those problems are shown in Fig. 5.

Some students added and removed a few towns in their problems and watched the transformation of the solution. Some tried to alter the parameters of the 2optGA and observe what would happen. Through many trials, they got the feeling of the difficulty of the TSP, and gained a concrete understanding of the optimization problems.

A summary of comments from the students follows:

- Good Points
  - It's easy and fun.
  - I can see the progress of solving the problem simultaneously.
  - I had thought that web pages were only for reading, so I was surprised to see your page for solving the TSP.

- Bad Points
  - No grid. The coordinates of the mouse are not displayed, so I cannot put towns accurately into the places I want.
  - I can't try pre-defined problems. I can't save problems. (Note: the *current* version of the applet *can* read the TSP file that contains the coordinates of towns.)
  - I want to see the convergence graph. I wish to check not only the elite but also the other individuals in the population. It would be convenient if the CPU time consumed to solve the problem was displayed.

We also present the following sample problems in our WWW pages:

- randomly located 100 towns
- double circle 192 towns ("C"-type)
- double circle 192 towns ("O"-type)

Double circle problems were used as benchmarks by Yamamura et al.[6] The minimum solutions of the problems are known and they are either "C"-type or "O"-type.

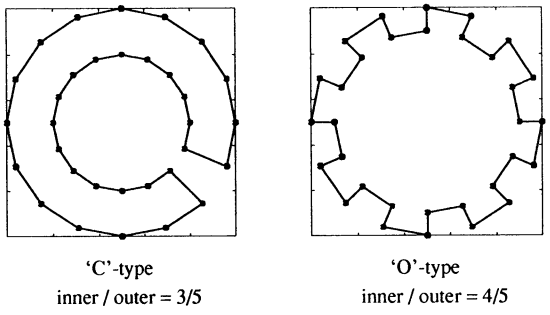


Figure 6: the minimum solutions of the double circle problems

As shown in Fig. 6, the type of solution that will yield the minimum, depends on the number of towns and the ratio of the radii of the inner and outer circles.

3.2 Stand-alone Application

The applet can also be used as a stand-alone application. First, download class files listed in Table 1 and sample problems listed in Table 2 from the URL:

```
http://www.hitachi.co.jp
/Div/sdl/e-naiyo/e-seika22/
```

Table 1: Class files to download.

File name	Size
Cities.class	3k
City.class	1k
GA.class	3k
Gene.class	3k
Sort.class	1k
Sortable.class	1k
TSP.class	8k
TspRoute.class	4k

Table 2: Sample problems.

File name	Problem
gr96.data	Africa-Subproblem of 666-city TSP
gr202.data	Europe-Subproblem of 666-city TSP
test100.dat	Randomly located 100 towns
c192-4682.dat	Double circle 192 towns ("C"-type)
c192-4684.dat	Double circle 192 towns ("O"-type)

"Africa-Subproblem of 666-city TSP" and "Europe-Subproblem of 666-city TSP" are contained in TSPLIB[5].

"TSP.class" contains the *main* method, so execute that file as a JAVA application with arguments 'filename' and 'generation'. The first argument, 'filename', is the name of the TSP file that contains the location of towns of the TSP to be solved. The second arguments, 'generation', is the maximum generation to be evolved. Optional flags listed in Table 3 can be used.

Each line in the TSP file represents the coordinates of a town and consists of two fields separated by a comma. The first field means the X coordinate of the town and the second field means the Y coordinate. The cost between two towns is measured as the Euclidean distance between the coordinates of each town.

If the TSP file contains the line

```
EDGE_WEIGHT_TYPE: GEO
```

the cost is defined by the geographical distance of two towns, i.e., the distance on the earth, instead of

Table 3: Optional flags of TSP.class

Option	Meaning	Default
-p M	The population	100
-e p_e	Eliminate p_e% individuals	30
-R p_i	Improve p_i% individuals by 2opt	20
-v	Increment verbosity level	

the Euclidean distance. The X and Y coordinates described above mean the latitude and the longitude respectively.

If the TSP file contains the line

```
Min:  n
```

where n is the positive real number, the program will stop when the elite, whose tour length is n, is found.

On Windows[2]95 with the Java Development Kit (JDK)[7], for example, type the following command in the "MS-DOS[2] Prompt" window.

```
java TSP -v -p 200 gr96.data 300
```

That command means to solve the TSP stored in the file 'gr96.data' (Africa-Subproblem of 666-city TSP in TSPLIB[5]), where the population is 200, and the maximum generation is 300.

The output from the command follows (partially omitted):

```
C:\>java TSP -v -p 200 gr96.data 300
File:  gr96.data
Population:  200
Selection:  30 %
2 opt:  20 %
1:3[93 94 92 91 76 ..... 27 64 95]57284
2:253[94 93 95 64 65 ..... 76 91 92]55840
3:119[37 34 33 32 10 ..... 31 35 36]55481
19:1251[49 51 52 54 50 ..... 47 46 45]55332
44:2646[93 94 92 91 76 ..... 65 64 95]55322
120:7132[41 40 39 38 78 ..... 49 48 42]55291
165:9780[49 45 46 47 53 ..... 54 52 51]55259
180:10692[52 51 49 45 46 ..... 48 50 54]55209
```

The program outputs the elite, when a new elite is found, in the following format:

$$g : i\ [\ tour\ ]\ \ell$$

where g, i, $tour$, and ℓ mean the generation, the ID of the individual, the tour (a list of towns in the visiting order) that corresponds to the individual, and the length of the tour.

The last line, for example, says that a new elite is found at 180th generation, it is the 10692nd individual, and the length of the tour is 55209. According to TSPLIB, it is the minimum solution.

[2]MS-DOS and Windows are registered trademarks of Microsoft Corporation

4 Conclusion

We presented an algorithm for rapid solution of the TSP that combines the GA and the 2opt method. We published a program based on the algorithm on our web pages. Because the program is written in JAVA language and can be executed on many platforms, anyone can verify the efficiency of our algorithm. Anyone who designed a new algorithm can compare his own TSP solver with ours by running both programs on the same machine. Our TSP solver is useful as a criterion for evaluating the performance of TSP solvers.

References

[1] Holland, J.H.: *Adaptation in Natural and Artificial Systems,* Univ. of Michigan Press (1975)

[2] Sengoku, H., Yoshihara, I.: *A Fast TSP Solution using Genetic Algorithm (Japanese),* Information Processing Society of Japan 46th Nat'l Conv. (1993)

[3] Sengoku, H., Yoshihara, I.: *An Evaluation of Optimizing Capability of Genetic Algorithm — GA vs SA — (Japanese),* Information Processing Society of Japan 47th Nat'l Conv. (1993)

[4] Moscato, P.: *TSPBIB,* http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html

[5] Reinelt, G.: *TSPLIB,* ftp://softlib.rice.edu/pub/tsplib/tsplib.tar

[6] Yamamura, M., Ono, T., Kobayashi, S.: *Character-Preserving Genetic Algorithms for Traveling Salesman Problem (Japanese),* Journal of Japanese Society for Artificial Intelligence Vol.7, No.6 (1992)

[7] Cornell, G., Horstmann, C.S.: *core JAVA,* SunSoft Press (1996)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Automatic Programming of Autonomous Robots using a Genetic Algorithm

Kiyoharu Tagawa, Shunsuke Kawaguchi, Hajime Iwamoto,
Katsumi Inoue and Hiromasa Haneda
Department of Electrical and Electronics Engineering,
Faculty of Engineering, Kobe University
Rokkodai 1-1, Nada-Ku, Kobe City, Japan. 657

Abstract

Programming autonomous robots determining both necessary sensing operations and reactive actions is a difficult and tedious work. This paper presents an automatic programming technique for autonomous robots by using the methodology of Genetic Algorithm (GA). The program of the autonomous robots is described as a finite state machine (FSM). A drawback of using GA to generate such a FSM is that it take a large number of generations to evolve the puerile FSM created randomly into an optimal formation. In order to reduce the number of generations taken, the Lamarckian probability is used in the genetic operations. The proposed strategy of GA based on the Lamarckism is evaluated by comparing with the conventional strategy based on the Darwinism which mimics natural selection and random mutation.

1 Introduction

Autonomous robot will meet with various situations while it moves around the real world. The program of such an autonomous robot is expected to decide an appropriate robot's action in each of the situations. Although we can use exclusive programming tools and languages[1],[2], it might be difficult for us to describe a complete program for an autonomous robot estimating all of the situations. Consequently, the evolutionary approaches, such as Genetic Algorithm (GA), have been advocated to generate some kind of programs for autonomous robots automatically. A large number of researches have been made on the automatic programming by means of simulated evolutionary approaches. These conventional approaches are detailed in the comprehensive survey papers[3],[4].

In this paper, we present an efficient automatic programming technique for autonomous robots by using a GA based on the Lamarckism. As physical autonomous robots, we suppose not only the behavior-based mobile robots but also the intelligent robots which execute complex task under unpredictable environments. Therefore, we adopt a finite state machine (FSM) as the formation of robot program. The FSM has both the generality and the capability to describe high-level robot competencies[2].

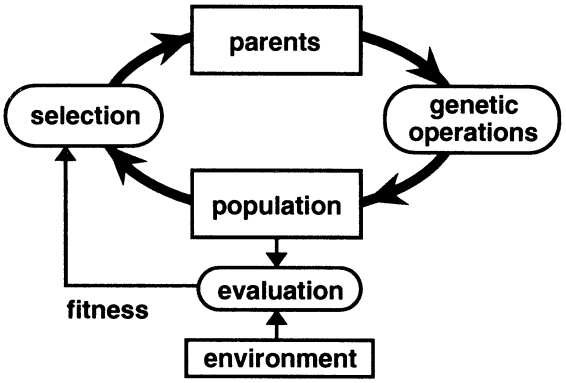


Fig.1 The GA based on the Darwinism

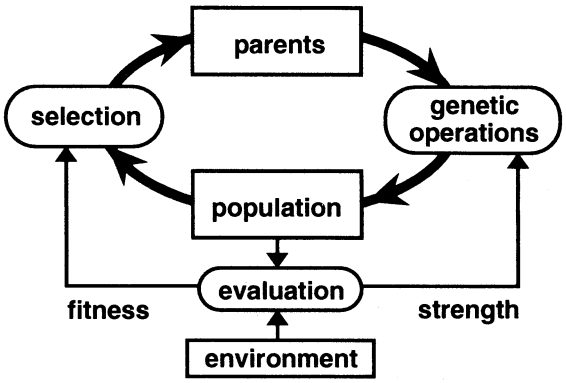


Fig.2 The GA based on the Lamarckism

The evolutionary strategy of conventional GA is based on the natural selection and random mutation advanced by C. Darwin[5]. The framework of the conventional GA based on the Darwinism is illustrated in Fig.1. Selection acts to choose some individuals in the proportion to their relative fitness and makes a group of parents. However, neither the fitness nor another characteristics of individuals affects the genetic operations applied to them.

In order to evolve the robot program, or FSM more effectively, we propose a new evolutionary strategy of GA based on the Lamarckism. The Lamarckism is an alternative evolutionary theory advanced by J. B. Lamarck, in which an useful organ must evolve preferentially in a body[6]. In the proposed GA, the characteristics of individuals is considered in the genetic operations applied to them. We introduce a strength function which denotes the contribution of each state to the performance of FSM and use the strength in the mutation operators which change the qualities of the state. The framework of the proposed GA based on the Lamarckism is illustrated in Fig.2.

2 Robot Program

We have the separated consideration on the control of robot's body which have inertia or disturbance, and on the logical execution of program. The body of autonomous robot has a low-level servo controller and appropriate sensors in order to control itself according to the command given by high-level program. We assume that the control of the physical object with inertia is done by the low-level controller autonomously.

The brain of autonomous robot is the high-level program described as FSM. Essentially, the FSM is an automaton which is defined by a sequence $FSM = (Q,\ C,\ I,\ \delta_c,\ \delta_t,\ q_0)$ with a set of state Q, a set of atomic command C, a set of input $I = \{0,\ 1\}$, an

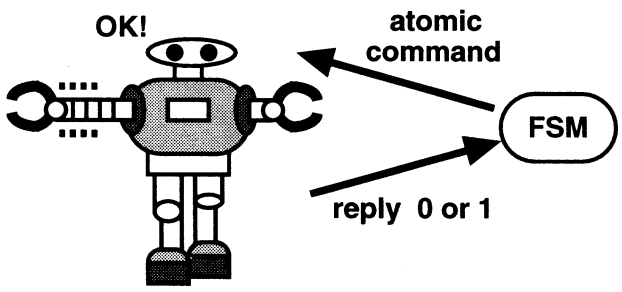


Fig.3 The robot controlled by FSM

		0	1
q_0	c_1	q_1	q_3
q_1	c_0	q_2	q_1
q_2	c_2	q_1	q_2
q_3	c_2	q_0	q_1
q_4	c_1	q_3	q_2
q_5	c_1	q_5	q_5

Fig.4 The FSM in tabular form

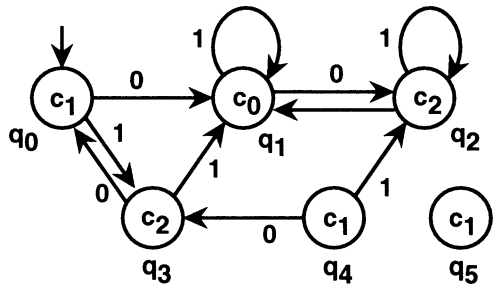


Fig.5 The diagram of FSM shown in Fig.4

command function $\delta_c : Q \rightarrow C$, a transition function $\delta_t : Q \times I \rightarrow Q$, and an initial state $q_0 \in Q$.

First of all, the command function $\delta_c : Q \rightarrow C$ is evaluated at the initial state $q_0 \in Q$. An atomic command $c_k \in C$ orders the robot body one single simple motion, probably of limited extent, such as move jaws right one unit length. If the robot body can execute the atomic command and complete its action, it returns the symbol $0 \in I$ to FSM. On the other hand, if the robot body can not possibly do the atomic command, it returns the symbol 1. Then, according to the robot's reply, namely, the input symbol 0 or 1, the transition function $\delta_t : Q \times I \rightarrow Q$ is evaluated to obtain the next state. Our image of an autonomous robot controlled by FSM is depicted in Fig.3.

The FSM can be summarized by a table as shown in Fig.4. The table lists the states $q_j \in Q$ coupled with an atomic command $c_k \in C$ down the side and inputs $0,\ 1 \in I$ across the top. The column under each input gives the values of the transition function $\delta_t : Q \times I \rightarrow Q$ corresponding to that input, at each state shown on the left.

In the proposed GA, the FSM in tabular form is regarded as a genotype. The plural of genotype is created randomly and modified by genetic operations. However, such a genotype usually has redundancy for the representation of FSM. For example, the FSM described in Fig.4 can be also summarized by a diagram shown in Fig.5 with edges labeled by input symbols

appropriately. Note that some states, namely q_4 and q_5, included in the FSM will not be arrived from the initial state $q_0 \in Q$. Thus, these states do not contribute to the performance of FSM.

3 Genetic Algorithm

In this section, we describe the implementation of our GA briefly. We have shown the framework of the proposed GA based on the Lamarckism in Fig.2.

Step 1: As an initial population, the plural of FSM is created randomly. Initially, all of the FSMs have the same number of states. However, the number of states will be changed individually in each FSM by applying the genetic operations.

Step 2: We measure the performance of each FSM in the population. The computational time of FSM is defined by the total number of the transitions of state which are executed until the robot completes its task. Then, we decide the fitness of each FSM so that it is inversely proportional to the computational time. The fitness function should be defined properly considering the characteristic of the objective problems.

We show a fitness function of FSM in (1). In the function $fit(FSM)$ defined in (1), $time(FSM)$ denotes the computational time spent by FSM. T_{max} and T_{min} denote the maximum and the minimum computational times in the population respectively.

$$fit(FSM) = (T_{max} - time(FSM) + T_{min})^2 \quad (1)$$

Step 3: We assign the strength to each state $q_j \in Q$ of FSM, which denotes the contribution toward the performance of FSM. We define the strength function of state $q_j \in Q$ as shown in (2). In the function $strgh(q_j)$ in (2), $Sum0$ and $Sum1$ denote the total number of inputs 0 and 1 replied to the state $q_j \in Q$ respectively. $BASE$, TAX and $FINE$ are scaling parameters which satisfy $BASE > 0$ and $FINE > TAX > 0$.

$$\begin{cases} strgh(q_j) = BASE + TAX * Sum0 \\ \qquad\qquad + FINE * Sum1, \\ \qquad\qquad (Sum0 + Sum1 > 0) \\ strgh(q_j) = 0, \quad (Sum0 = Sum1 = 0) \end{cases} \quad (2)$$

Step 4: Selection of parental FSMs is carried out by a roulette wheel algorithm in light of the fitness of them. In addition, an elitist selection is employed in which the best FSM in the population always survives to the next generation.

Step 5: Genetic operations, namely reproduction, crossover and mutation, are applied to the selected parents to create new population. The reproduction operation simply copies the selected FSM into the next generation. Crossover operation exchanges some states between two FSMs at random. Note that each state of FSM contains an atomic command and transition function as shown in Fig.4.

There are five possible mutation operations for FSM: 1) change an atomic command; 2) change a state transition; 3) duplicate a state; 4) delete a state; and 5) change the initial state of FSM. The deletion of state and the changing of initial state are only allowed when the FSM has more than one state.

The five mutation operations for FSM have been also adopted in the Evolutionary Programming[7]. The states modified by the mutations are used to be chosen according to the uniform probability. However, as we have shown in Fig.5, a genotype usually contain redundant states which do not contribute to the performance of FSM. Therefore, instead of the conventional uniform probability, we use the Lamarckian probability[6] for the mutations. Each state $q_j \in Q$ of FSM is given its own rate of mutation $\rho(q_j)$ decided by using the strength evaluated in **Step 3**.

$$\rho(q_j) = \frac{strgh(q_j)}{\sum_{q_i \in Q} strgh(q_i)} \quad (3)$$

where, $1 \geq \rho(q_j) \geq 0, \quad (q_j \in Q)$.

From (2) and (3), we can confirm that not a neutral mutation which does not change the performance of FSM occurs with the Lamarckian probability.

Step 6: Unless a sufficiently excellent FSM is created nor the generation reaches a prearranged number, the processes from **Step 2** to **Step 6** are repeated.

4 Experiments

In order to assess the performance of the proposed GA for all sorts of robots in various environments, we use a simulator instead of physical robots. To model the real world of robots, objects and obstacles, we employ the state space model represented by a graph[8]. With the state space model, we can regard that an autonomous robot moves on the graph from one node to another node if the robot executes an atomic command ordered by FSM successfully.

We have conducted experiments on a "carrying sofa problem"[8]. A robot carries a long sofa through a crowded room. Five atomic commands are allowed:

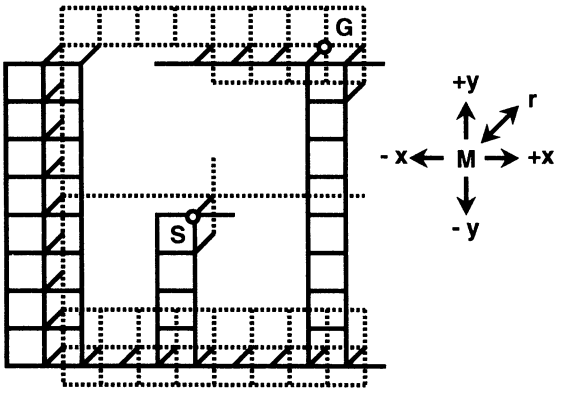


Fig.6 The graph of carrying sofa problem

1), 2) Move $\pm x$ direction one unit length; 3), 4) Move $\pm y$ direction one unit length; 5) Rotate $r = 90°$. The state space model for the problem is given by a graph shown in Fig.6 with a start node (**S**) and a goal node (**G**). FSM is required to lead an imaginary robot from **S** to **G** on the graph as soon as possible.

Figure 7 shows the computational time of the best FSM in each population (size: 50) at generation t (averaged over 40 runs). In Fig.7, we have compared the proposed GA based on the Lamarckism (**lamk**) to the conventional GA based on the Darwinism (**darw**). The parameters in (2) were set at $BASE = 1.0$, $TAX = 1.0$ and $FINE = 3.0$ respectively. From the result shown in Fig.7, we can say that the proposed GA is superior to the conventional GA in the quality of FSM and the convergence performance.

5 Conclusions

In this paper, we have shown that the proposed GA based on the Lamarckism creates excellent robot programs more effectively rather than the conventional GA based on the Darwinism. Since the robot program is described as a finite state machine (FSM) which is able to realize a variety of robot competencies, we believe that our technique is applicable to the programming for many intelligent robots.

The present work can be extended in several directions. First, we want to apply the proposed technique to the development of the control programs for actual robots. Secondly, in order to evolve a large hierarchical program effectively, we have to develop a new technique to discover the reusable parts in program and preserve them automatically.

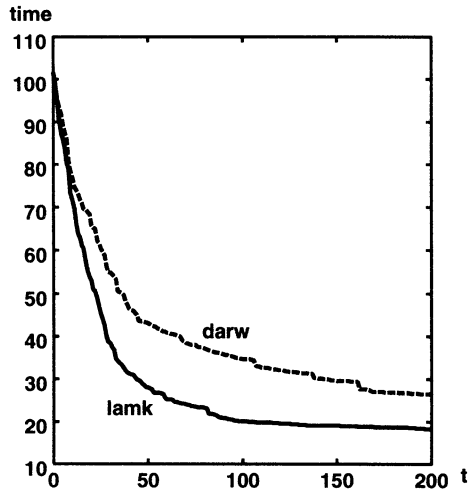


Fig.7 The computational time of FSM

References

[1] C. Laugier and J. P. Troccaz, "SHARP: A system for automatic programming of manipulation robots," *Robotics Research The 3rd International Symposium*, pp.125–132, MIT Press, 1986.

[2] S. Suzuki, S. Yuta and J. Iijima, "A consideration on the programming method of sensor-driven robot behavior based on the action mode representation," *Proc. of 20th ISIR*, pp.127–134, 1989.

[3] K. D. Jong, "On using genetic algorithms to search program spaces," *Proc. of 2nd Int. Conference on Genetic Algorithms*, pp.210–216, 1987.

[4] T. Gomi and A. Griffith, "Evolutionary robotics - an overview," *Proc. of IEEE Int. Conference on Evolutionary Computation*, pp.40–49, 1996.

[5] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1992.

[6] Y. Davidor, *Genetic Algorithms and Robotics - A Heuristic Strategy for Optimization*, World Scientific Publishing, 1991.

[7] L. J. Fogel, A. J. Owens and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, 1966.

[8] D. E. Whitney, "State space models of remote manipulation tasks," *IEEE Trans. on Automatic Control*, Vol.AC-14, No.6, pp.617–623, 1969.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A Novel Genetic Algorithm Based on the Theory of Virus Evolution

Naoshi Nakaya Akinori Kanasugi Kunio Kondo
Faculty of Engineering
Saitama University
255, Shimo-okubo, Urawa 338, Japan

Abstract

Genetic Algorithm is a method to search optimum solution based on Darwinism. On the other hand the theory of evolution is not only Darwinism but also a lot of theories. The theory of virus evolution is one of these theories. This theory has the idea which a virus evolves life. This paper describes a novel genetic algorithm based on the theory of virus evolution.

Until now, the methods based on the theory of virus evolution have been proposed. A virus is not life in our new method, while a virus is life in these methods. Therefore a virus doesn't make a population, doesn't have fitness and doesn't mutate. Namely we think that a virus is the organ to carry genes.

A population in our algorithm has same meaning in a genetic algorithm. A crossover operator does not use in our algorithm, but a population evolves by infection. An infection operator selects two points randomly on the chromosome that is selected according to fitness. These genes between the points make a virus and the virus infects other chromosomes according to some probability. That is to say, some genes spread out a population and the population changes not only well but also badly. Therefore this algorithm can search optimum solution without falling into local minima. A mutation operator behaves infection error, namely this operator replaces some gene in a virus with another gene in infection phase. After repeats of these operations, we can get optimum solution.

The procedure of proposed algorithm is summarized as follows: 1. Generate a population randomly. 2. Select one chromosome according to fitness. 3. Perform an infection operator with performing a mutation operator. 4. Repeat Step 2 and 3. The advantage of our algorithm is that the calculation time is shorter than the time of genetic algorithm because our algorithm has light operations.

We have made computer experiments. In these experiments, we have compared a genetic algorithm with the proposed algorithm under same problem. The results of knapsack problem show that the calculation time of proposed algorithm is 48.0% shorter than the time of genetic algorithm and the probabilities of to get optimum solution by proposed algorithm is higher than that of genetic algorithm.

1 Introduction

A genetic algorithm is a method to search optimum solution. This method has the advantages that are independent of the initial conditions and avoid local minima. But the calculation time to get optimum solution is very long. We think the shortcoming is caused by this method based on Darwinism. The theory of evolution is not only Darwinism but also a lot of theories. Therefore, we propose a novel algorithm based on the theory of virus evolution. The theory of virus evolution has the idea which a virus evolves life. Namely a piece of gene spreads out a population by infection with a virus, and the population evolves quickly. We give attention to this point and propose a novel algorithm, which has short calculation time.

Until now, the methods based on the theory of virus evolution have been proposed [2, 3]. But in these methods, infection with a virus means local search, that is, a virus population is the piece of better solutions. On the other hand, infection with a virus is one and only search operator in our proposed method. Accordingly, a virus is the piece of one solution. In our method, a virus doesn't make a population, doesn't have fitness and doesn't mutate. Namely we think that a virus is not life but the organ to carry genes.

2 Proposed Algorithm

In this section, we propose a novel algorithm based on the theory of virus evolution. A *population* in our algorithm has same meaning in a genetic algorithm. Therefore, a population is constructed randomly which population size is M. A *selection* operator selects one chromosome according to fitness. This

chromosome becomes the origin of virus (figure 1). A crossover operator does not use in our algorithm, but a population is evolved by *infection*. An infection operator selects two points randomly, which the points are named p1 and p2, on the chromosome that is selected in a selection operation. These genes between the points make a virus, if p2 is smaller than p1 then a virus is made two pieces which are constructed by both from the top point to p2 and from p1 to the end point (figure 2). And the virus infects other chromosomes according to some probability that is named a infection rate p_i (figure 3). The infection changes a original gene into the gene of virus at each point. That is to say, some genes spread out a population and the population changes not only well but also badly, because the virus is made randomly. Therefore this algorithm can search optimum solution without falling into local minima. A *mutation* operator behaves infection error, namely this operator replaces some gene in a virus with another gene according to some probability that is named a mutation rate p_{mv} (figure 4). And we defined performing these operations as *cycle*. After many cycles, we can get optimum solution.

The procedure of proposed algorithm is summarized as follows:

1. Generate an initial population randomly.
2. Select one chromosome according to fitness.
3. Perform an infection operator.
   (a) Select two points randomly on the chromosome in Step 2.
   (b) Make a virus from a piece of gene between the points.
   (c) Infect other chromosomes with the virus.
   (d) Mutate genes which are changed in Step (c).
4. Repeat Step 2 and 3.

The advantage of our algorithm is that the calculation time is shorter than the time of genetic algorithm because our algorithm has light operations. That is, the average number of genes that is changed in our algorithm is $\frac{N+1}{2}M$ per cycle, where N is a chromosome length. On the other hand, the number in genetic algorithm is NM per generation. Therefore, the calculation time becomes short in our algorithm.

3 Computer Experiments

We have made computer experiments. In these experiments, we have compared a genetic algorithm with the proposed algorithm under same knapsack problem. The problem is composed of N objects which are numbered $(1, \cdots, N)$, each object i has weight w_i and value v_i. The capacity of knapsack is c. We introduce

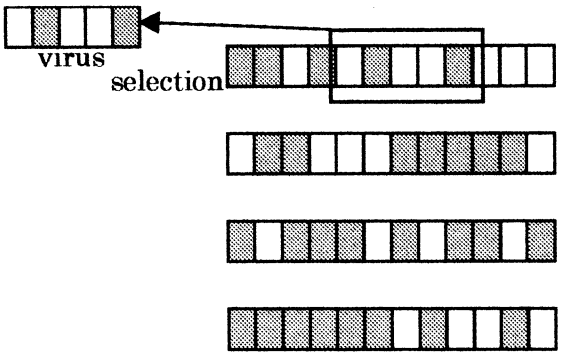


Figure 1: Example of a selection.

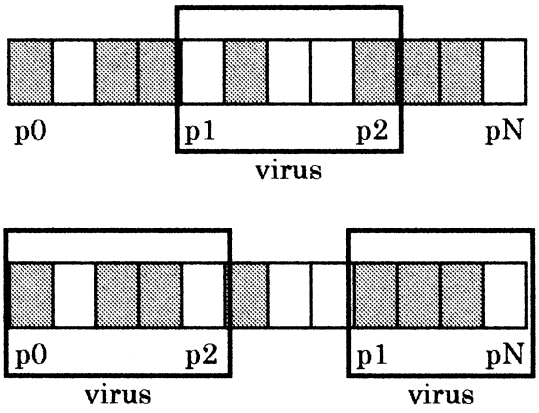


Figure 2: Example of making a virus.

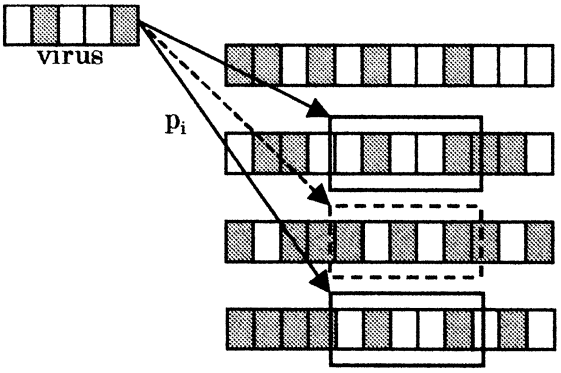


Figure 3: Example of infections.

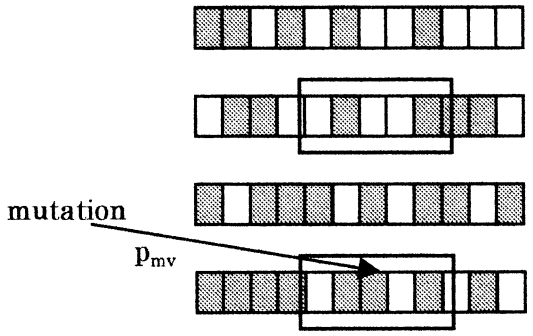


Figure 4: Example of mutations.

x_i that if object i is packed into the knapsack then $x_i = 1$, otherwise, $x_i = 0$.

3.1 Experimental Conditions

We describe experimental conditions. First, the common conditions of both genetic algorithm and proposed algorithm are as follows:

Coding: The i-th gene in a chromosome is defined as a value x_i.

Fitness: Fitness f is defined by next equation.

$$f = \max\left[0, \sum_{i=1}^{N} v_i x_i - \alpha \max\left\{0, \sum_{i=1}^{N} w_i x_i - c\right\}\right] \quad (1)$$

Population: A initial population is constructed randomly which population size is M.

Next, the conditions of genetic algorithm are as follows:

Selection: Roulette wheel selection and elitism. Besides, size of elitism is 1.

Crossover: One-point crossover. Crossover rate is p_c.

Mutation: Mutation rate is p_{mg}.

And last, the conditions of proposed algorithm are as follows:

Selection: Select the one chromosome which has best fitness.

Infection: Infection rate is p_i.

Mutation: Mutation rate is p_{mv}.

We compare a genetic algorithm with the proposed algorithm using above conditions.

Table 1: Experimental parameters

Common		
Population size	M	50
Chromosome length	N	24
Genetic algorithm		
Generations	T_g	500
Crossover rate	p_c	0.95
Mutation rate	p_{mg}	0.05
Proposed algorithm		
Cycles	T_c	500
Infection rate	p_i	0.95
Mutation rate	p_{mv}	0.1

3.2 Experimental Results

Table 1 shows experimental parameters. And we decide that $\alpha = 100$ in the equation (1).

Table 2 gives experimental results. Each algorithm is experimented 10 times and we average those results. Furthermore Table 2 gives final results, thus the best fitness and the fitness average are results of 500th generation or cycle. The CPU time is for a workstation HP-9000/755.

The CPU time of proposed algorithm is shorter than the one of genetic algorithm. The decrease of CPU time is

$$\frac{9.44 - 4.91}{9.44} * 100 = 48.0\ (\%).$$

Figure 5 shows the relationships between CPU time and fitness. It is easily recognized that the proposed algorithm is more effective than genetic algorithm.

Besides, the optimal fitness is 191 in these experiments. The genetic algorithm could get optimal solution two times. On the other hand, the proposed algorithm could get it eight times. Namely, the probabilities of to get optimum solution by proposed algorithm is higher than that of genetic algorithm.

4 Conclusions

We proposed an efficient algorithm based on the theory of virus evolution. In this algorithm, we think that a virus is the organ to carry genes and we define infection with a virus as one and only search operator. A infection operator means the piece of genes spread out a population quickly, therefore the calculation time of our proposed algorithm is shorter than the time of genetic algorithm.

Table 2: Experimental results

	Genetic algorithm			Proposed algorithm		
	Best fitness	Fitness average	CPU time (s)	Best fitness	Fitness average	CPU time (s)
1st	191	128.3	–	191	136.0	–
2nd	189	127.9	–	191	133.7	–
3rd	190	144.3	–	189	120.5	–
4th	190	133.3	–	189	118.5	–
5th	191	141.6	–	191	130.5	–
6th	189	121.6	–	191	151.7	–
7th	190	127.1	–	191	144.0	–
8th	190	124.7	–	191	125.9	–
9th	189	147.5	–	191	108.3	–
10th	189	134.2	–	191	123.1	–
Average	189.8	133.1	9.44	190.6	129.3	4.91

We have made computer experiments using proposed algorithm. In these experiments, we have compared a genetic algorithm with the proposed algorithm by knapsack problem. The results of computer experiments show the efficiencies of proposed algorithm. Namely, the calculation time of proposed algorithm is 48.0% shorter than the time of genetic algorithm and the probabilities of to get optimum solution by proposed algorithm is higher than that of genetic algorithm.

References

[1] Lawrence D (1991) Handbook of Genetic Algorithms, Van Nostrand Reinhold.

[2] Hasegawa K, Matumoto M, Kanoh H, Nishihara S (1996) Solving Constraint Satisfaction Problems Based on Virus Theory of Evolution (in Japanese), Proceedings of IPSJ 52st, 2E-07.

[3] Kubota N, Fukuda T, Shimojima K (1996) Virus-Evolutionary Genetic Algorithm for Self-Organizing Manufacturing System, Computer & Industrial Engineering Journal, 30-4, 1015-1026.

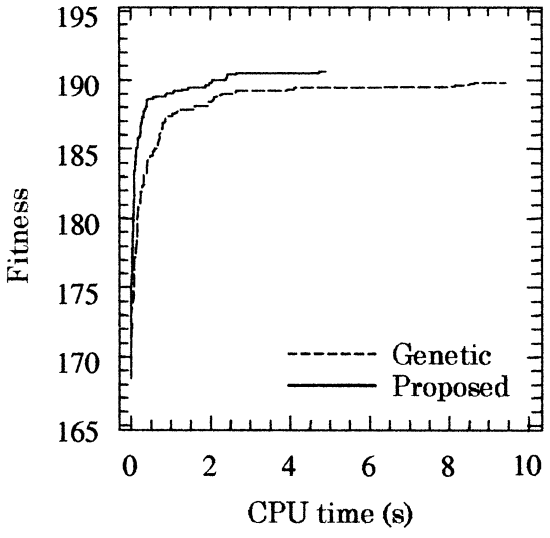


Figure 5: Relationships between CPU time and fitness.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

Learning of Static Evaluation Function in Tsume-shogi with Genetic Programming

Tomoyuki SAWADA † Keiji SUZUKI † Azuma OHUCHI †

† Research Group of Complex Systems Engineering,
Graduated School of Engineering,
Hokkaido University,
Kita 13 Nishi 8, Sapporo 060, Japan
{sawada, suziki, ohuchi}@complex.eng.hokudai.ac.jp

Abstract

In this paper, we apply the Genetic Programming as machine learning method to constructing the static evaluation function for solving the tsume-shogi. The game, "tsume-shogi", consists of the collection of the limited states in "shogi" game for the exercise of it. Since this game rule is same as shogi's rule, the tsume-shogi is also a kind of zero-sum complete information games. Even if the game only teats the limited states, it is still complicate to solve it. To solve this type of games with computer programs, it is required, in general, the static evaluation function to evaluate the possible states for selecting the moves of the pieces. Since the realization of the function for the tsume-shogi is difficult, we expect to construct of it automatically with machine leaning methods. Therefore, we try to apply the Genetic Programming to costructing the static evaluation function for solving the tsume-shogi. Throughout the experiments, we will show the effectiveness of the proposed method.

1 Introduction

The games have been treated in main research field in the artificial intelligence. The reasons are that the implementations to computers are easy and the performance of the programs can be easily evaluated by matches of them. So many games, such as tic-tac-toe, checker, Othello, chess, have been researched vigorously because the games of them can be treated with the game trees in their implementations. Recently, these computer programs become as strong as world champions of the games. The implementations of these programs are equipped with the improved search methods, the databases toward the end of games and the fast information retrieval methods to defeat the human champions. Basically, the strategies of these programs are based on the fast calculations of powerful computers.

While, concerning to more complex games, such as "shogi" and "go", the performances of these programs remain still insufficient. One of the reason is the search spaces in their game trees are huge even if the powerful computers are used. In addition, since the states in the games tend to generate the large possible states with a few moves, it is difficult to evaluate the possible states for selecting the efficient moves from the current states. Therefore, to increase the performance of the programs for these complex games, we have to construct the efficient evaluation methods.

As the evaluation methods, in general, the static evaluation functions are used in the game trees.

Since the realizations of the functions for these games are difficult, we expect to construct of them automatically with machine leaning methods. Then, in this paper, we try to apply the Genetic Programming (GP) as a machine learning method to costructing the function for solving the tsume-shogi. The game, "tsume-shogi", consists of the collection of the limited states in "shogi" game for the exercise of it. Since the game rule is same as shogi's rule, the tsume-shogi is also a kind of zero-sum complete information games. Even if this game only teats the limited states, it is still complicate to solve it. In the following, we propose the method for implementing the GP to this problem and show the results of its experiments.

2 Static Evaluation Function

In the game theory, the best moves of the players in zero-sum complete infomation two-persons games can be searched from the game trees in general. The common game tree in such games is represented in Fig.1. The root node is the positions in present phase.

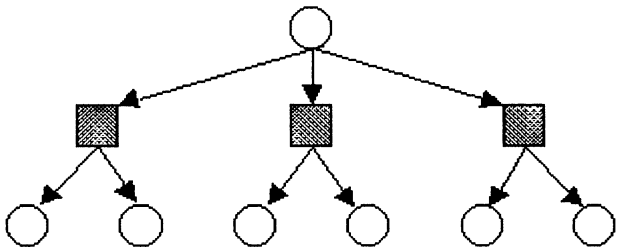


Figure 1: Game Tree

Three arrows from the root node mean the first possible moves from the present position. The next nodes are the positions in the opponent's phase after the first moves,and further arows mean the opponent's possible moves on that occasion.

In Fig.1,we can read two moves ahead from the game tree.

In the simple games, such as tic-tac-toe, we can make and explore the game trees completely. Then, we will get the best moves from the complete game trees in the case of the games being simple. While, concerning to more complicated games, such as checker, Othello, chess, shogi and so on, the size of the complete game trees will become extremely huge. In such cases, we have to control the search processes in the game trees to stop exploring the states under the useless leaves. To control the search process efficiently, it is required the static evaluation functions for comparing the leaves whether it will include the efficient moves or not. Namely, we have to obtain the efficient moves from the incomplete trees. Therefore, the role of the static evaluation functions for solving the games is important. However, according as the complexities of games are increased, the implementations of the static evaluation functions for the games become difficult. Therefore, the studies for the games often need the support from the strong human players to constructing the efficient functions.

3 Tsume-shogi

Shogi is similar game to chess. The objective of the player is to capture the opponent's piece of king by the movements of own pieces. Shogi can be thought as more complicate than chess, because the search space of shogi, about 10^{220}, is lager than the one of chess, about 10^{120}. In addition, even if the states are closed to the end of the game, the states in shogi tend to generate the large possible states with a few moves. This means that the successful strategies in chess based on the powerful computers equipped with the databases are useless for solving the shogi. Therefore, to increase the performance of the programs for shogi, we have to construct the efficient static evaluation functions. However, only to get the effective value from the evaluation function, it is considered that we have to estimate a hundred moves at least.

While, "tsume-shogi" is the one-person game for the exercise of shogi. The game consists of the collection of the final states in shogi. The objective of the player is to continue to put his opponent in check, called "Oute" in shogi, until the states become the checkmate, called "Tsumi" in shogi.

For solving this game, about six or seven moves is required in average. Although the length is enough shorter than that of shogi, it is still complicated for solving the game. For example, the gain of each piece will change non-monotonically. Namely, we should often select the moves causing to lose own pieces intentionally. Therefore, without the support of the strong human players, it is difficult to construct the static evaluation function for tsume-shogi.

In this paper, we expect to produce the function automatically with machine leaning methods. Then, we try to apply the Genetic Programming (GP) as a machine learning method to generating the function for solving the tsume-shogi.

4 Learning of Evaluation with GP

In this study, we apply the Genetic Programming (GP) as an innovative learning method to generate the static evaluation function for tsume-shogi. The framework of the learning process is illustrated in Fig.2.

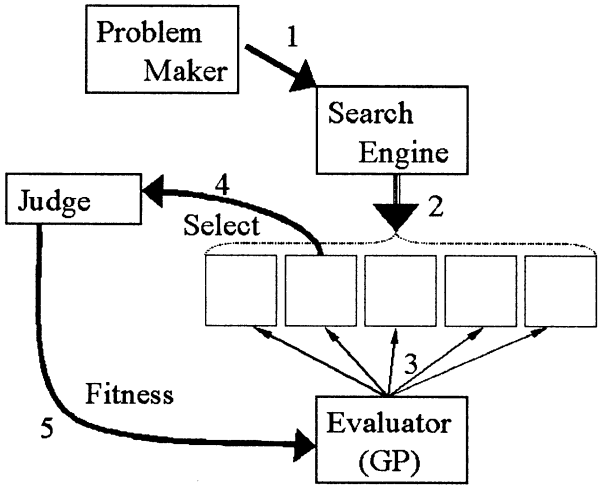


Figure 2: Framework of Learning Process with GP

The problem maker supplies the initial state to the search engine. The search engine explores the possible states in one depth. The evaluator evaluates each of the explored states and selects one state which can be considered to include the check. To make the evaluation and the selection of the evaluator, it uses the static evaluation function which is generated by the individual in GP. The individuals receive the rewards if the selected state correctly includes the check. Otherwise, they receive the penalties (Fig.3). By repeating the process, we expect GP to generate the effective static evaluation function for solving the tsume-shogi. In the following subsection, we introduce the applied methods for learning the tsume-shogi problem with GP.

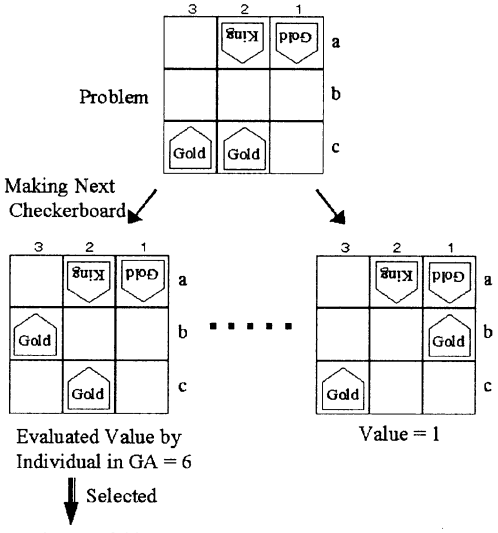


Figure 3: Example of learning process

0	1	...	7	8
10	11	...	17	18
⋮	...	⋱	...	⋮
70	71	...	77	78
80	81	...	87	88

Table1:Indices of checkerboard

	pawn	lance	...	bishop	rook
capturer's	0	1	...	5	6
opponent's	7	8	...	12	13

Table3:Indices of captured pieces

4.1 Data Structures of States

The states of tsume-shogi consist of the set of the available pieces and their positions on the checkerboard of shogi. The checkerboard of shogi has the 10×9 array. Then, each position on the checkerboard is indexed like as Table 1. The kind of the pieces on the checkerboard is represented by the positive integer from 1 to 17 for the player's pieces and the negative values for the opponent's pieces (Table 2). In shogi, the captured pieces which have been removed from the checkerboard are available to reuse as own pieces. Thus, the kind of the captured pieces is indexed from 0 to 13 for the pieces of both players (Table 3).

4.2 Functions of GP

The individual in GP should work as the static evaluation function for tsume-shogi. Thus, it is required that the individuals can receive the current states and must return the values as the result of the evaluations. To generate the evaluation function by each individual in GP, we prepare the two functions, POSI and KING, to refer the kinds of the piece in a position and the position of opponent's king respectively. The states of the captured pieces are referred by the function, PIEC. Thus, the terminal nodes in GP are consisted of constants and the function of KING as listed in table 5. Other nodes are consisted of the functions, POSI and PIEC, and the mathematical operators as listed in table 4.

5 Experiments

In order to confirm whether the static evaluation function will be learned or not with the GP, we use six easy problems to capturing the king by one move (Fig.4). If all six problems are solved, the fitness of the individual will be six.

	pawn	lance	knight	silver	gold	bishop	rook	king
capturer's	1	2	3	4	5	6	7	(8)
opponent's	-1	-2	-3	-4	-5	-6	-7	-8
	promoted pawn	promoted lance	promoted knight	promoted silver		promoted bishop	promoted rook	
capturer's	11	12	13	14		16	17	
opponent's	-11	-12	-13	-14		-16	-17	

Table2:Indices of pieces on checkerboard

indication	number of arguments	return value
+	2	$a_1 + a_2$
-	2	$a_1 - a_2$
*	2	$a_1 * a_2$
%	2	a_1/a_2
IFLTE	4	if $a_1 < a_2$ then a_3,else a_4
IFEQ	4	if $a_1 = a_2$ then a_3,else a_4
POSI	1	value of a piece on position a_1
PIEC	1	number of a_1th captured pieces

Table 4:Functional node

indication	return value
KING	position of king
(constant)	random integer from -17 to $+17$

Table 5:Terminal node

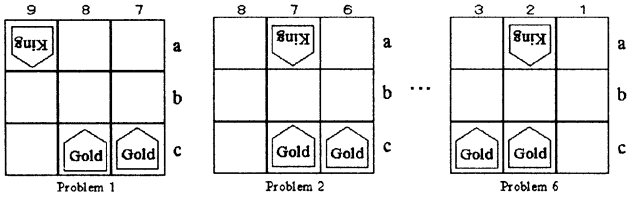


Figure 4: Learning Problems

5.1 Results and Discussion

The experimental results on the average of ten executions are shown in Fig.5. In this experiment, the number of the individuals was 1000 and the generations were 20. We got the various correct solutions 8 times out of the ten executions. The shortest solution are shown as follows.

```
(POSI
  (IFEQ
     -1
     (+ (+ KING KING) -1)
     KING
     (- KING -10 )
  )
)
```

The obtained evaluation function judges, by 'IFEQ', whether King is in position of 0 (Problem1). The return value of the function will be high if king is not in the position of 0 and Gold in the front of King.

In the experiments it can be seen that the static evaluation functions were learned successfully. In addition, the GP generated the correct evaluation functions in the multiple runs. However, the generated functions tend to include the many redundant expressions relative to the required ability. In addition, even if the size of individuals is allowed to increase, the possibility to get the solutions doesn't so increase. Therefore, the learning ability of the introduced GP isn't so effective.

In future, it will be necessary to solve more complex problems that the approach will be improved, for example:

- Improvement of the table and contents of functions in GP.
- Increasing the checkerboard information so as to refer more precise.
- Introducing the Automatically Defined Functions (ADF).

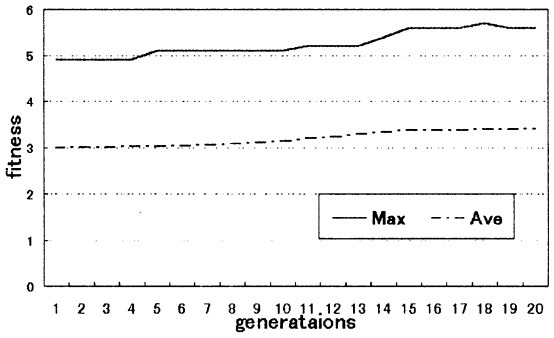


Figure 5: result

6 Conclusion

In this study, we try to make the static evaluation function by learning with Genetic Programming. It is confirmed that simple evaluation function for simple problems could be learned successfully. Future work will be devoted to improve the learning of the evaluation function for more complex problems.

References

[1] Matsubara H,Takeuchi I(eds.)(1997) Game Programming, Kyoritsu press(in japanese)

[2] Richards N. et al(1997) Evolving Neural Networks to Play Go, pp.768-775,proceedings of the seventh international conference on genetic algorithms.

[3] Koza,J.R.(1992)Genetic Programming,The MIT Press

[4] Koza,J.R.(1994)Genetic Programming II,The MIT Press

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Modeling and Complexity: Control Engineering Point of View

Hidenori Kimura
Department of Mathematical Engineering and Information Physics
The University of Tokyo
Hongo, Bunkyo-ku, Tokyo 113, JAPAN
kimura@crux.t.u-tokyo.ac.jp

Abstract

This paper presents a perspective of complexity from control engineering point of view. Implication of complexity in control system design is clarified as a source of difficulty in modeling, which leads us to the notion of physical complexity.

1 Introduction

Recently, the word "complexity", or "complex system" is quite popular in various fields e.g., nonlinear dynamics, economics, physics and biology, as well as scientific journalism. Actually, complexity theory has been a well-established field of research in computer science for at least thirty years. It is still an active and even growing research area that has created various notions of complexity such as computational complexity, combinatorial complexity, algorithmic complexity and information-based complexity. These two notions of complexity, one is old and the other is relatively new, are not directly related, but may be related indirectly at the deepest level of their logical foundations.

Unfortunately, however, these two kinds of complexity do not seem to represent fully the common feature of the word complexity in its original sense and in our daily use. *Computational complexity* [1] measures the complexity in computing a solution of a mathematical problem. Thus, a problem is complex if its computation is lengthy. *Algorithmic complexity* [2] represents the length of an algorithm to describe the rule for generating a numerical sequence. Thus, a sequence is complex if its rule to generate is complex. The complete random sequence is the most complex sequence. This fact is a target of criticisms [3]. *Complexity of nonlinear dynamics* [4] is related to unpredictability of the behaviors associated with a given dynamical systems. Thus, a dynamical system is complex if its behavior is irregular beyond our prediction, though the equation describing its behavior is simple. These notions of complexity represent our common feeling of complexity partially, but the complexity is a much richer notion. There are many other aspects of complexity beyond the complexity in computer science and in nonlinear dynamics. These aspects of complexity are exactly what control engineering has been concerned with and struggling against. We have not yet been very successful for dealing with them. It can be said that we have been unsuccessful even in identifying them. We now began to realize that these aspects of complexity are not described purely mathematically. They are regarded as physical complexity that are described in the context of physics involved rather than mathematics, and emerge as various kinds of difficulties in describing them as models of the real world. Thus, the *third notion of complexity* must be embedded in physical structure of the real world and it emerges when we construct a model of the real world.

In this paper we discuss the characterization of this type of complexity and its consequence. The paper is purely expositly and contains more problems than solutions. In Section 2, some issues of modeling is stated and the implication of physical complexity is stated. In Section 3, physical complexity is identified with the source of uncertainty in modeling. Section 4 proposes a method of representing explicitly the uncertainty included in the model.

2 Modeling Issues

Our notion of complexity characterizes a feature of the real world itself. We think that the complexity of the real world becomes an issue of serious research when we construct its model, i.e., an abstract description of the real world in a specific framework. The complexity of the real world is something to do with a difficulty in its modeling.

One of the most salient feature of modern technol-

ogy is the increasing use of computer simulations for the analysis and synthesis of complex systems. Rapid progress in computer technology enabled us to simulate time-space behaviors of a large-scale complex systems such as devices with millions of elements, of a factory with hundreds of production lines, and of a dynamic system with thousands of interconnected subsystems. It is an important fact that every simulation needs a model of the real system to be simulated. It is the first step of simulation to construct a model, which is usually not an easy task. If the model is not properly constructed, the results of simulation using this model is not realistic. In such a case, simulation is not only useless but also might be harmful, because the simulation may justify misleading conclusions. In this sense, we cannot overemphasize the importance of the model in simulations. Once an appropriate model becomes available, simulation is reduced to computation which is usually simpler than modeling. Unless an appropriate model is available, we cannot use computer for the analysis and design of real systems. We can say that the model is an interface between the real world and theory.

In order to construct a model of the real world, we must have certain amount of knowledge to identify its mechanism that generates the observed behaviors of the real world in the form of mathematical equations or algorithms. Since the real system to be modeled is not an abstract object but a concrete one in front of modelers, we must know its specific characteristics through experiments or special investigations. Finally, we must describe the model in a consistent way to our simulation tool. To sum up, the model is a structured representation of our knowledge and information about the real system in a given framework. Thus, modeling is a highly sophisticated task of mapping the real world to an abstract framework of description.

Now, the complexity of the real world restrict our knowledge and information to construct a model. It is at this point where the complexity of the real world becomes a real issue of engineering. We list up some key words which seem to represent our feeling of complexity, in comparison with their antonyms that express simplicity.

The key words of complexity in Table 1 represent physical nature of the real world. They cannot be described only by mathematical notions associated with dynamical behaviors of the model. Their descriptions entail representations of physical entity involved. For instance, heterogeneity represents coexistence and mutual coupling among many physical processes of different energy domains, e.g., mechanical, electrical, thermal, chemical, acoustical and fluid mechanical processes. On the other hand, homogeneity implies that only one kind of energy domain is involved in the process. It is rather a difficult task to represent heterogeneity mathematically, because energy domain is identified only physically by means of its dimension. Thus, the third notion of complexity must be something that reflects physical properties of the real world. We may call it *physical complexity*, and it is exactly what engineering has been coping with.

Complex	Simple
Nonlinear	Linear
Time-Variable	Time-invariant
Distributed	Lumped
Open	Closed
Irreversible	Reversible
Heterogeneous	Homogeneous
Multiphase	Uniphase
Multicomponents	Single component
Irregular Connections	Regular Connections

Table 1 Key words Representing Complexity

3 Complexity as a Source of Model Uncertainty

Physical complexity characterized through some keywords in Table 1 causes difficulties in modeling of real systems. The problem of dealing with this complexity is a very fundamental issue of models and modeling. There have been two different views towards models and modeling. The one is the optimistic view which believes that the real world is governed by some regular laws of the nature, and rationally built model can represent the real world as accurately as we want. This view does not admit the existence of the fundamental gap between the real world and its descriptions. The other is the pessimistic view for models and modeling asserting that the model of the open system cannot be verified, and it is impossible to establish the fidelity of models. It admits a deep gap between the real world and its descriptions that cannot be surpassed by any method of describing the physical complexity.

The above two views of models and modeling are concerned mainly with the evaluation of our knowledge and information that are to be used in modeling. The optimistic view thinks that our knowledge and information can be increased to the extent that we

can describe physically complex systems as precisely as possible, while the pessimistic view thinks that our knowledge and information are fragmental, scarce and subject to specific conditions, and hence, we cannot claim validity of our model for general cases [5]. Therefore, it is extremely important to evaluate properly the amount of our knowledge and information about the real world, in relation to the description of the physical complexity.

One of the significant contributions of the recent advances in complexity theory lies in the elimination of the belief that the complexity is the mask for the underlying simplicity. The complexity must be accepted as the first principle, not only in biology but also in physics. Actually, engineering has been accustomed to this view for long time. The most typical situations where engineers feel complexity of the real world are as follows:

(1) There are many parameters that are involved in the phenomenon of the process concerned,

(2) The behaviors of the process are sensitive against slight changes of the parameters,

(3) It is difficult to conduct the observations and/or experiments due to the lack of reproducibility.

Of course, there are many other difficult occasions caused by the physical complexity. Each specific physical complexity may attract the interest of specialists in the field. Sometimes, however, it is not advisable to get into the details of the physical complexity from modeling point of view. Modeling is always motivated by its own purpose. The question of necessary preciseness at the cost of complexity of its description is always subject to the purpose of modeling.

Our proposal is to embed the physical complexity in the explicit description of the *uncertainty* of the model. This is almost exactly what control engineering has been doing in designing robust control. In modeling the plant to be controlled, it is sometimes impossible to get through the physical complexity. In these cases, we reduce it to the uncertainty of the parameters and/or the structures of the plant quantitatively. We try to measure the range of parameter variations of the plant or identify the effects of the missing factors of dynamics in the model. It is easier to evaluate the uncertainty of the model than to analize physical complexity in detail. Thus, our remedy against physical complexity is to regard it as a source of uncertainty in modeling. Then, the problem is to evaluate the effect of the model uncertainty on the fidelity of simulations that are made based on the model. Figure 1 depicts the relationship between complexity (physical) and uncertainty.

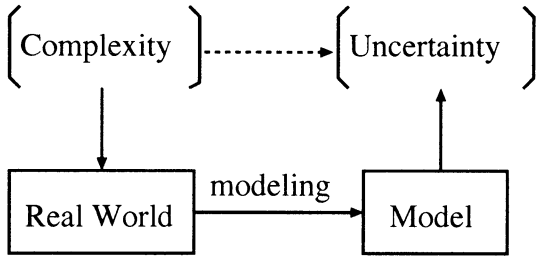


Figure 1 Complexity and Modeling

4 Formal Theory of Models and Modeling

In the preceding section, we regarded complexity as a source of model uncertainty. Significance of models increases when the real system to be modeled is complex. At the same time, uncertainty of the model increases as the real system to be modeled is more complex. Thus, the complexity is of supreme importance in constructing, evaluating and verifying models.

Now, we discuss how to represent uncertainty of the model explicitly to embed the complexity of the real world. Model-based design theory of control systems must be aware of uncertainty of the model based on which control system is designed. It has developed theory of robust control that tolerates the model uncertainty. The major tool of robust control theory is to represent the real system as a set of models rather than a single model. The set of models is usually described as a set of perturbations around a known *nominal model.* The purpose of robust control is to design a control system that achieves desired control performances uniformly for any model that is in the set of models. In this sense, the target of robust control is to design a controller for a set of models rather than a single model. Actually, this was a great paradigm shift in control engineering.

A standard description of a linear model set in robust control is given in Fig.2, where the so-called linear fractional transformation (LFT) is used. In Fig.2, $H(s)$ represents the transfer function of the nominal plant which is known completely. $\Delta(s)$ represents the perturbation which is unknown but its magnitude bound is assumed to be known in some way, e.g., as a perturbation with bounded norm.

$$\|\Delta\| \leq 1$$

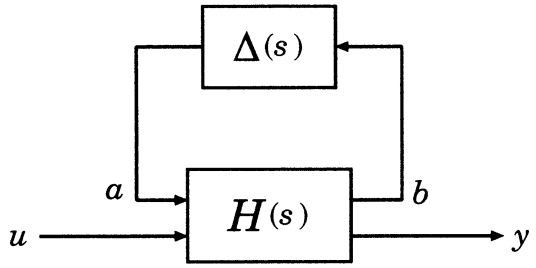


Figure 2 Model Set given by LFT

The transfer function from u to y is given by

$$H_\Delta(s) = H_{22}(s)+H_{21}(s)\Delta(s)(I-H_{11}(s)\Delta(s))^{-1}H_{12}(s). \quad (1)$$

This representation is versatile in describing various kinds of model set. If we take

$$H(s) = \begin{bmatrix} 0 & U(s) \\ I & G(s) \end{bmatrix}$$

then (1) becomes

$$H_\Delta(s) = G(s) + \Delta(s)U(s),$$

which represents an *additively perturbed* model set, and if we take

$$H(s) = \begin{bmatrix} 0 & V(s)G(s) \\ I & G(s) \end{bmatrix},$$

(1) becomes

$$H_\Delta(s) = (I + \Delta(s)V(s))G(s),$$

which represents a *multiplicatively perturbed* model set.

Robust control theory based on the representation of model set based on Fig.2 has achieved significant success that makes sophisticated model-based design method of control systems realistic and useful for practical design. It is worth noting that the appropriateness of the model is severely checked when the control law designed based on the model is implemented to the real system. Thus, control engineering is the area where the model is crucially treated. Our ambition is to extend the success to other field in which models play crucial roles [6].

Perhaps, we need to establish the system theory of model set in order to create a proper framework to deal with the physical complexity of the real world. There are many issues to be addressed in the system theory of model set, a part of which are listed up below:

(1) Representation
There are many ways to represent the same model set. The equivalence relation of the two sets leads us to the notion of canonical representation of the model set and the notion of equivalence class.

(2) Topology
Topological structure of the model set will be an attractive research problem which is of practical importance. The *diameter*, *radius* and *center* of the model set will give measures of uncertainty and the idea of nominal model.

(3) Connections
Various kinds of connections of model sets create new model sets which have different structure from the original ones. It is of great importance to clarify the outcomes of connections of models from the viewpoint of model set. Connection is an important source of complexity.

(4) Reduction
Model set can contain simpler models than the nominal model. This fact suggests the possibility of representing the given model set in a simpler way with some approximations. This is the problem of model set reductions which is of practical importance.

(5) Identification
To identify the model set of a real system is indeed the most important problem of model set system theory. We have some a priori knowledge about the real system to be identified. This gives you a priori model set M_a which includes all the models consistent with our a priori knowledge. Now, we obtain some a posteriori information about the real system through some observations and/or experiments which generate the data E. Then, our task is to find the set of all models m which are included in M_a and consistent with E, i.e.,

$$M = \{m : \ m \in M_a, \ m \sim E\}, \quad (2)$$

where $m \sim E$ implies that the model m generates the data E. To represent M in an appropriate way is the major issue of model set identification.

The model set identification is regarded a method of induction utilizing fully the power of computers, if the model embodies a law. The set M represents all the laws that are considered to be reasonable subject to the priori restriction represented by M_a. The structure of model set identification represented by (2) is quite similar to Bayesian method of statistical inference.

5 Conclusion

Modeling is a highly intellectual, sophisticated and creative activity of human being which maps the real world to a specified framework of its description. Knowledge and information about the real world must be collected, structured, integrated and represented in conformably with a given format of description, Modeling thus reflects the quality of our knowledge and information which are usually quite limited. Complexity of the real world is the major factor limiting our knowledge and information used for modeling. The incompleteness of our knowledge and information on the real world naturally generates uncertainties in modeling, which leads us to the notion of model set.

The implication of models varies from one field to another. Sometimes, the word *model* is used very differently in different fields. This means that the systematic study of models and modeling from general point of view is at a very premature stage. This is the main season why we have totally different and sometimes contradictory views of models.

What we need is a formal theory of models and modeling which gives us a proper view of models and rational way of using models [7]. It is paradoxically true that the more formal and systematic the theory is, the more useful it is for practical applications. System theory of model set will hopefully give us a framework of the formal theory of models and modeling.

References

[1] J. E. Savage, *The Complexity of Computing,* John Wiley and Sons, 1976.

[2] M. Li and P. Vitányi, *An Introduction to Kolmogonov Complexity and its Applications,* Springer, 1994.

[3] M. Gell-Mann, "Remarks on simplicity and complexity," *Complexity*, Vol.1, pp.16-19, 1995.

[4] A. Lasota and M. C. Mackey, *Chaos, Fractals and Noise: Stochastic Aspects of Dynamics,* Springer, 1994.

[5] N. Oreskes, K. S. Frechett and K. Belitz, "Verification, validation and confirmation of numerical models in the earth science," *Science*, Vol.263, February 1974.

[6] J. Doyle, "Theory in modeling and simulation," *Reprint*, 1997.

[7] H. Kimura, "Non-uniqueness, uncertainty and complexity in modeling," to appear in *Computational and Applied Control and Signal Processing*, 1998.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

Deterministic and Probabilistic Approaches to Model Set Validation

Tong Zhou
Dept. of Computational Intelligence
and Systems Science
Tokyo Institute of Technology
Nagatsuta, Yokohama, 226

Shinji Hara
Dept. of Computational Intelligence
and Systems Science
Tokyo Institute of Technology
Nagatsuta, Yokohama, 226

Abstract

Model set validation problem is investigated in this paper. Results of both deterministic and probabilistic approaches are provided. Simulation results show that the probabilistic approach evaluates more appropriately the suitability of a model set in robust controller design.

1 Introduction

It is necessary to have a plant nominal model and its error bound in order to successfully apply robust control theories to engineering problems. In traditional identification theory, however, the problem of simultaneously identifying a plant nominal model and its error bound has not been directly dealt with[2]. The essential demand for robustness of control systems and the development of robust control theory have stimulated the birth of a new research field, which is now widely known as robust control oriented identification. One of the most important topics in this field is model set validation, in which the suitability of a provided model set in robust controller design is evaluated[3][4][5].

More precisely, a model set validation problem can be stated as following.

Given a plant nominal model $G_0(z)$, a noise set $\mathcal{V}$, a modelling error set $\mathbf{\Delta}$, a positive number γ, and a series of identification experiment data $(u_i,\ y_i)|_{i=0}^{n-1}$, verify whether there is an error model $\Delta(z)$ in $\mathbf{\Delta}$, a noise series v in $\mathcal{V}$, such that

$$y = H(G_0(z),\ \Delta(z),\ u, v),\ \ ||\Delta(z)||_\infty \leq \gamma$$

Here, $H(*,*,*,*)$ represents a known relation among the plant input/output, nominal model, and so on. If the answer is negative, then, the nominal model and its error bound is not appropriate for robust controller design. Otherwise, provide an estimation of the probability for the existence of modelling errors and noises which belong to the corresponding prescribed model error and noise sets and are consistent with identification experiment data.

To answer the question of whether or not a model set is falsified by experiment data, a deterministic approach is natural. However, a probabilistic approach seems more reasonable in estimating the possibility of a model set being able to explain the provided experiment data. Therefore, both deterministic and probabilistic approaches are applied in this paper to investigate the formulated model set validation problem, and their appropriateness in evaluating the suitability of a model set in robust controller synthesis is compared through a numerical example.

While the structures of nominal model errors utilized in robust control theories are so abundant, in this paper, our attention is restricted to unstructured ones which perturb a plant nominal model in an additive form. Moreover, the noises in identification experiment data are assumed to be additive and affect the plant output only. Furthermore, only SISO LTI systems and time domain model set validation problems are dealt with. These are mainly for the conciseness of presentation.

Notation. $Tz\{a_i|_{i=0}^{n-1}\}$ is used to represent the lower Toeplitz matrix with its first column elements as a_0, $\cdots$, a_{n-1}, while $\mathcal{P}r\{*\}$ stands for the probability that phenomenon $*$ occurs. Moreover, $f \sim \mathcal{N}(m, \sigma^2)$ means that random variable f is normally distributed, and its expectation and variance are respectively m and σ^2.

2 Deterministic Approach

When a deterministic approach is taken in model set validation, the question of whether or not a prescribed model set is consistent with the provided identification experiment data, can be answered.

It is now well recognized that the classical Cara-

thedory-Fejer extrapolation theory and its extension to matrix valued functions play an essential role in model set validation[3][5]. The following input-output extrapolation-minimization theorem is an immediate result of this extrapolation theory and the properties of Toeplitz matrices.

Lemma 1 *There exists a transfer function which is consistent with input-output pairs* $(\bar{u}_i, \bar{y}_i)|_{i=0}^{n-1}$ *with* $\bar{u}_0 \neq 0$, *and has* $\mathcal{H}_\infty$ *norm not greater than* γ, *if and only if*

$$\bar{Y}^T\bar{Y} \leq \gamma^2\bar{U}^T\bar{U} \tag{1}$$

where $\bar{U} = Tz\{\bar{u}_i|_{i=0}^{n-1}\}$, $\bar{Y} = Tz\{\bar{y}_i|_{i=0}^{n-1}\}$.

Based on Lemma 1, a necessary and sufficient condition for a model set not to be invalidated is established[3][5].

Theorem 1 *A model set* $\mathcal{G}$ *described by*

$$\mathcal{G} = \{G(z)|G(z)=G_0(z) + W(z)\Delta(z), \quad \|\Delta(z)\|_\infty \leq \gamma\} \tag{2}$$

$$G_0(z) = \sum_{i=0}^{\infty} g_i z^i, \qquad W(z)=\sum_{i=0}^{\infty} w_i z^i \tag{3}$$

is not falsified by $(u_i, y_i)|_{i=0}^{n-1}$ *with* $u_0 \neq 0$, *if and only if*

$$\min_{v_i|_{i=0}^{n-1}\in\mathcal{V}} \bar{\sigma}((UW)^{-1}[Y - G_0U - V]) \leq \gamma \tag{4}$$

Here, $U = Tz\{u_i|_{i=0}^{n-1}\}$, $Y = Tz\{y_i|_{i=0}^{n-1}\}$, $V = Tz\{v_i|_{i=0}^{n-1}\}$, $G_0=Tz\{g_i|_{i=0}^{n-1}\}$, $W=Tz\{w_i|_{i=0}^{n-1}\}$.

It is worthwhile to note that $\bar{\sigma}((UW)^{-1}[Y-G_0U-V])$ is a convex function of $v_i|_{i=0}^{n-1}$. Therefore, the condition of Inequality (4) can be verified through convex optimization or linear matrix inequalities, provided that the noise set $\mathcal{V}$ is convex.

3 Probabilistic Approach

While the deterministic approach provides an answer to the question whether or not a model set is falsified by identification experiment data, the possibility that a model set can explain the identification experiment data is still not so clear when the model set is not invalidated. To deal with this problem, it seems natural to introduce a probabilistic framework.

Assumptions on the probability distribution of noises have to be introduced, in order to obtain an estimation of the model set unfalsification probability (MSUP). To make this probability estimation problem computationally attractive, in this paper, assumptions on the frequency domain characteristics of the noises are applied, instead of their time domain characteristics.*

Specifically, the MSUP estimation problem is formulated as following.

Assume that at frequencies $\omega_i = \frac{i\pi}{2n}$, *the real and imaginary parts of the discrete Fourier transformation of noise* $v_i|_{i=0}^{n-1}$ *have independent distribution function* $\mathcal{N}(0, \sigma_{r,i}^2)$, $\mathcal{N}(0, \sigma_{j,i}^2)$, *respectively,* $i = 0, \cdots, 2n$. *When plant time domain input/output pairs* $(u_i, y_i)|_{i=0}^{n-1}$ *with* $u_0 \neq 0$, *are provided, estimate the probability of model set* $\mathcal{G}$ *to be consistent with* $(u_i, y_i)|_{i=0}^{n-1}$.

Remark 1. It is well known that if $v_i|_{i=0}^{n-1}$ is the output of a stable LTI system under the stimulation of a moment limited zero expectation white noise, then the real and imaginary parts of its discrete Fourier transformation asymptotically have a complex normal distribution[1]. Their expectations are zero, while their variances can be computed from the time domain variances of the input signals and the frequency characteristics of the system. Therefore, the assumptions on the noises in the above problem formulation are quite natural and general.

Remark 2. In identification, experiment data length is usually great. This implies that a model set validation problem deals with a more general noise set when the noise frequency domain stochastic characteristics are assumed, compared with the time domain ones.

The next lemma is firstly introduced to investigate this problem. It establishes a relation between a Toeplitz matrix and the discrete Fourier transformation of its first column elements[7].

Lemma 2 *For arbitrary real numbers* ε_i, $i = 0, \cdots, n-1$, *define*

$$E = Tz\{\varepsilon_i|_{i=0}^{n-1}\}, \quad \hat{E}(z) = \frac{z^n}{\sqrt{n}}\sum_{i=0}^{n-1}\varepsilon_i z^i$$

$$d_i(\varepsilon) = \Re\left\{\hat{E}\left(e^{j\frac{i\pi}{2n}}\right)\right\},\ i = 0, \cdots, 2n; \quad \varepsilon = \varepsilon_i|_{i=0}^{n-1}$$

$$D(\varepsilon) = diag\{d_0(\varepsilon),\ diag\{d_i(\varepsilon)I_2|_{i=1}^{2n-1}\},\ d_{2n}(\varepsilon)\}$$

Then, we have

$$\begin{bmatrix} 0 & E^T \\ E & 0 \end{bmatrix} = X^T D(\varepsilon) X, \quad X^T X = 2\sqrt{n}I \tag{5}$$

*Similar results have also been established on model set unfalsification probability when the noise probabilistic characteristics are described in the time domain[6]. In this case, however, the conclusions are correct only when the number of experiment data approaches infinity.

where

$$X = [x_0^T \ x_1^T \ \cdots \ x_{2n}^T]^T, \ x_0 = \frac{1}{\sqrt[4]{4n}} \left[\underbrace{1 \ 1 \ \cdots \ 1 \ 1}_{2n \ columns} \right]$$

$$x_{2n} = \frac{1}{\sqrt[4]{4n}} \left[\underbrace{-1 \ 1 \ \cdots \ -1 \ 1}_{2n \ columns} \right]$$

$$x_i = \frac{1}{\sqrt[4]{n}} \begin{bmatrix} cos\frac{i\pi}{2n} & cos\frac{i\pi}{n} & \cdots & (-1)^i \\ sin\frac{i\pi}{2n} & sin\frac{i\pi}{n} & \cdots & 0 \end{bmatrix}, \ i \in [1, 2n-1]$$

It is worthwhile to point out that X is independent of $\varepsilon_i|_{i=0}^{n-1}$, while $D(\varepsilon)$ is diagonal and its diagonal elements, say, $d_i(\varepsilon)$, are actually the real or imaginary part of the discrete Fourier transformation of $\varepsilon_i|_{i=0}^{n-1}$.

The next theorem provides a necessary and sufficient condition for MSUP to be greater than 0.

Theorem 2 *MSUP is greater than 0 if and only if there exists a noise series $v_i^0|_{i=0}^{n-1} \in \mathcal{V}$, such that matrix P defined as*

$$P := \begin{bmatrix} \gamma M^2 & (Y - G_0 U - V_0)^T \\ (Y - G_0 U - V_0) & \gamma I \end{bmatrix} \quad (6)$$

is positive definite, where, $M = \sqrt{W^T U^T U W}$, $V_0 = Tz\{v_i^0|_{i=0}^{n-1}\}$, $U = Tz\{u_i|_{i=0}^{n-1}\}$, $Y = Tz\{y_i|_{i=0}^{n-1}\}$.

The next theorem gives a lower bound of the probability for a model set not to be invalidated.

Theorem 3 *Assume that the matrix P in Theorem 2 is positive definite and denote its minimal eigenvalue by $\lambda_{min}(P)$. Define σ and α as*

$$\sigma = \max_{0 \le i \le 2n} |\Re\{z_i^n(\sigma_{r,i} + j\sigma_{j,i})|_{z_i = j\frac{i\pi}{2n}}\}|$$

$$\alpha = \min_{0 \le i \le 2n} \Re\left\{ \frac{z^n}{\sqrt{n}} \sum_{k=0}^{n-1} v_k^0 z_i^k |_{z_i = j\frac{i\pi}{2n}} \right\} + \frac{1}{2\sqrt{n}} \lambda_{min}(P)$$

Then,

$$MSUP \ge \mathcal{P}_r\{f \le \alpha\} \quad (7)$$

where $f \sim \mathcal{N}(0, \sigma_f^2)$ and σ_f is given by

$$\sigma_f = \begin{cases} \sigma & \alpha \ge 0 \\ \frac{1}{2\sqrt{n}}\sigma & \alpha < 0 \end{cases}$$

Remark 3. To obtain a good lower bound of MSUP, it is preferable to search a noise series $v_i|_{i=0}^{n-1}$ in the noise set $\mathcal{V}$, which maximizes α, subject to the condition that the matrix P is positive definite. This optimization problem can be proved to be a convex minimization problem.

An upper bound of MSUP is given in the next theorem.

Theorem 4 *Assume that the matrix P in Theorem 2 is positive definite and λ_i, z_i are respectively its eigenvalue and normalized eigenvector, $i = 1, \cdots, 2n$. Then,*

$$MSUP \le \min_{1 \le i \le 2n} \mathcal{P}r\{f_i \le \lambda_i + \mu_i\} \quad (8)$$

Here, $f_i \sim \mathcal{N}(0, \sigma_i^2)$, $i = 1, \cdots, 2n$. Moreover,

$$\sigma_i^2 = w_{i1}^4 \sigma_{r,0}^2 + w_{i,4n}^4 \sigma_{r,2n}^2 + \frac{1}{2} \sum_{k=1}^{n-1} \{(w_{i,4k}^2 + w_{i,4k+1}^2)^2 \sigma_{j,2k}^2 + (w_{i,4k+2}^2 + w_{i,4k+3}^2)^2 \sigma_{r,2k+1}^2\}$$

$$\mu_i = w_{i1}^2 \bar{d}_0 + w_{i,4n}^2 \bar{d}_{2n} + \sum_{j=1}^{2n-1} (w_{i,2j}^2 + w_{i,2j+1}^2) \bar{d}_j$$

$$\bar{d}_i = \Re\left\{ \frac{z_i^n}{\sqrt{n}} \sum_{k=0}^{n-1} v_k^0 z_i^k \Big|_{z_i = e^{j\frac{i\pi}{2n}}} \right\}, \ i = 0, 1, \cdots, 2n$$

$$[w_{i1} \ w_{i2} \ \cdots \ w_{i,4n}] = z_i^T X^T$$

It is obvious from Theorems 5 and 6 that a greater ratio of the minimum eigenvalue of matrix M to its maximum one is preferable in correctly estimating MSUP. This can be achieved by choosing an appropriate input signal series $u_i|_{i=0}^{n-1}$. A physical explanation of this fact seems to be that in order to have a fair evaluation of the appropriateness of a prescribed model set in robust controller design, it is necessary to provide informative input/output data.

4 Numerical Simulation

In this section, a simulation example is provided to illustrate the established results.

Assume that the dynamics of a plant is represented by transfer function $G(s)$

$$G(s) = \frac{10000 e^{-0.001s}}{s^2 + 100s + 10000} \quad (9)$$

Moreover, the model set used to represent the plant dynamics is as that of Equation (2) with

$$G_0(z) = 6.3338 \times 10^{-3} \frac{z^2 + 3.8001z + 0.9048}{z^2 - 1.7826z + 0.8187} \quad (10)$$

$$W(z) = 0.0185 \frac{z^2 - 0.9935z + 0.0130}{z^2 - 1.7826z + 0.8187} \quad (11)$$

By comparing the magnitude of $W(e^{j\omega})$ and $G(j\omega) - G_0(e^{j\omega})$, it is easy to see that $r \ge 1$ is necessary and sufficient for the existence of a transfer function in $\mathcal{G}$ which has the same frequency response as that of $G(s)$.

Now, we investigate the relation between γ and the probability that model set $\mathcal{G}$ is unfalsified.

The plant input is chosen to be the inverse $\mathcal{Z}$-transformation of $W^{-1}(z)$ to have a greater ratio of the minimum eigenvalue of the matrix M to its maximum one. A white noise with distribution function being $\mathcal{N}(0, 0.02^2)$ is added in the plant output to represent measurement errors, etc. The plant is sampled with a sampling period of $0.002s$ and 100 input/output pairs are used in the following computations.

First, the deterministic approach is applied to obtain the minimal γ such that the model set $\mathcal{G}$ is consistent with the simulated plant input output data. In this example, $|v_i| \leq 0.06$ is assumed, $i = 0, 1, \cdots, 99$. By the deep-cut ellipsoid algorithm of convex optimization, it is found that the minimal γ equals 0.3302.

Now, we investigate the MSUP of $\mathcal{G}$ when γ varies from 0 to 3. The results are illustrated in Figure 1, where

$$H(\gamma) = \begin{cases} 1 & \mathcal{G} \ is \ unfalsified \\ 0 & \mathcal{G} \ is \ falsified \end{cases} \tag{12}$$

It is interesting to note that if the model set G is required to be consistent with the input/output data in a probability greater than 0.8, it is necessary that γ is not smaller than 0.911. On the other hand, if MSUP being 0.8 is great enough for robust controller design, then, γ should not be greater than 1.13.

While $\gamma = 0.3302$ provides the most powerful unfalsified model set which must be stabilized by a controller, in this simulation, the probabilistic approach gives a more appropriate evaluation on the suitability of the model set in the forthcoming control system design.

5 Concluding Remarks

In this paper, model set validation problem is investigated under both deterministic and probabilistic framework. It is proved that the verification of whether or not a model set is consistent with time domain identification experiment data can be transformed to a convex minimization problem. Moreover, an upper bound and a lower bound have been obtained for the probability of which a model set is not falsified by time domain identification experiment data. A numerical example is provided to illustrate the deterministic and probabilistic approaches.

Simulation results show that while the deterministic approach is able to supply a model set which should be stabilized by a controller in the forthcoming robust control system synthesis, the probabilistic approach evaluates more appropriately the suitability of a model set in robust controller design. Therefore, the necessity of the probabilistic approach in model set validation has been confirmed.

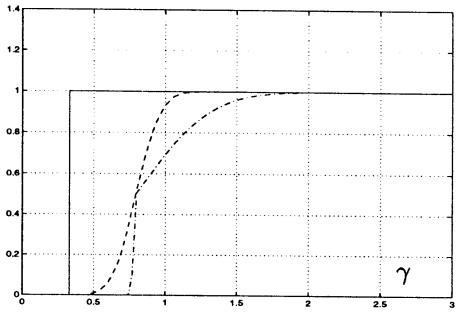


Figure 1: Model Set Unfalsification Probability. $- -$: upper bound; $- \cdot -$: lower bound; —: $H(\gamma)$.

Many important issues, however, are still remaining to be dealt with. The improvement of the upper and lower bounds of model set unfalsification probability, an appropriate probing signal design, etc., are examples of them.

References

[1] D.R.Brillinger. *Times series–data analysis and theory.* Holden-Day, San Francisco, 1981.

[2] L.Ljung. *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, 1987.

[3] K.Poolla, et al. A time domain approach to model validation. *IEEE AC,* Vol.39, No.5, pp.951~959, 1994.

[4] R.Smith and J.Doyle. Model validation–a connection between robust control and identification. *IEEE AC,* Vol.37, No.7, pp.942~952, 1992.

[5] T.Zhou and H.Kimura. Input–output extrapolation-minimization theorem and its application to model validation and robust identification. *Lecture Notes in Control and Information Sciences,* Vol.192, Springer, London, pp.127~137, 1994.

[6] T.Zhou, T.Miyazato and S.Hara. Probability estimation for a model set being unfalsified. *Proc. the 20th SICE Symposium on Dynamical System Theory,* pp.313~316, 1997. (In Japanese)

[7] T.Zhou. A reinvestigation on unfalsified model parametrization. *preprint.*

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Statistical Mechanics of Learning and Generalization

Y Kabashima
Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
Yokohama 226, Japan

abstract

Statistical Mechanics (SM) is a powerful analytical tool which is applicable not only to objectives traditionally studied in physics, but also to general complex systems. In this paper, we review recent application of SM to theoretical analysis of learning in neural networks with presenting several basic results.

1 Introduction

One of the most important properties of neural networks is their capability of learning. Neural networks can adopt their synaptic weights following a set of examples which reflect a target relation in order to make predictions on the output for an novel input.

In order to express such "learning" mathematically, let us formulate it as follows. To simplify the problem, we here focus on learning of feed-forward networks depicted in figure 1. Such a network is considered as a nonlinear function

$$\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x}; \boldsymbol{w}), \tag{1}$$

which is parameterized by synaptic weights $\boldsymbol{w} = (w_{ij})$. This function can represent various input-output relations by changing adjustable parameters $\boldsymbol{w}$.

With the expression of eq. (1), learning of the neural network is generally formulated as the following inverse problem.

Learning of Neural Networks
Suppose that a set of examples

$$\xi^P = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_P, \boldsymbol{y}_P)\}, \tag{2}$$

which reflects some unknown input-output relation, is given. For a fixed network eq. (1), find a set of parameters $\boldsymbol{w}$ which exactly (or approximately) reproduces the relation (2).

The meaning of this inverse problem is easily understood by considering some corresponding real-world problems. For instance, if we replace $\boldsymbol{x}$ and $\boldsymbol{y}$ with "hand-written character" and "alphabet", respectively, this is nothing but a mathematical expression of the recognition problem of hand-written characters. Not only this example but also most of "pattern recognition" by neural networks are mathematically formulated as this type of inverse problems.

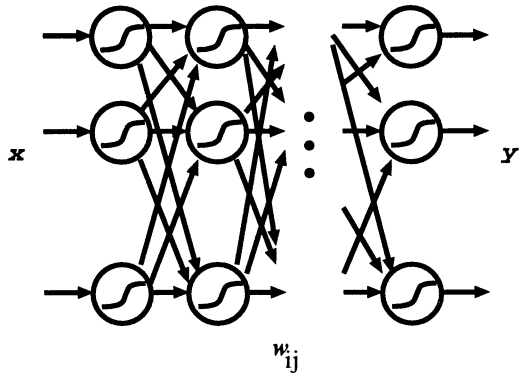


Figure 1: feed-forward network

2 Generalization and Learning Curve

Problems of learning are classified into the following two categories[1].

The first one is on its algorithmic aspect, namely, the problem of *how to find parameters in a finite time.* This problem has a long history since the "perceptron" is proposed in 1950's[2]. Since learning is an inverse problem on complex nonlinear functions, it is almost impossible to design an almighty algorithm that find out the *optimal* solution in a finite time for an arbitrary network. However, it became not so difficult to obtain a *practical* solution due to the proposal of the "Error-Back Propagation Algorithm" in 1980'[3].

Then, what is the second one? It is the problem of *generalization*, namely, the question *how good a predictive ability the trained network has.* It is not so long

since this problem appeared in the theory of learning and this is now actively investigated. Therefore, we will focus on this subject in the remainder of this paper.

The problem of generalization is mathematically expressed in the following manner. Although the student does not know what it is, the target relation should be represented by some input-output relation. For simplicity, we assume this is a deterministic function

$$\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x}). \tag{3}$$

The student network learns this relation following a set of examples

$$\xi^P = \{(\boldsymbol{x}_1, \boldsymbol{G}(\boldsymbol{x}_1)), (\boldsymbol{x}_2, \boldsymbol{G}(\boldsymbol{x}_2)), \cdots, (\boldsymbol{x}_P, \boldsymbol{G}(\boldsymbol{x}_P))\}. \tag{4}$$

We mentioned nothing about the learning method so far. One practical strategy is minimization of the squared error on the examples (4)

$$E(\boldsymbol{w}|\xi^P) = \sum_{\mu=1}^{P} (\boldsymbol{F}(\boldsymbol{x}_\mu; \boldsymbol{w}) - \boldsymbol{G}(\boldsymbol{x}_\mu))^2. \tag{5}$$

In general, the predictive ability after learning strongly depends on a realization of the examples (4). In order to avoid such arbitrariness, we here assume that examples are independently drawn from an identical distribution

$$D(\boldsymbol{x}). \tag{6}$$

How to realize the minimization of eq. (5) is the very problem on algorithms that is mentioned above. We here assume that this problem is resolved in order to concentrate on the subject of generalization.

It is naturally expected that the network exhibits a good predictive ability for examples which appeared during learning. Then, how good a predictive ability does the network have for novel examples? This ability, which is called *generalization ability*, is evaluated by *generalization error* which is the expectation of predictive error for a test example that is independently drawn from the same distribution (6) as in the training phase

$$\varepsilon(\boldsymbol{w}(\xi^P)) = \int (\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{w}(\xi^P)) - \boldsymbol{G}(\boldsymbol{x}))^2 D(\boldsymbol{x}) d\boldsymbol{x}. \tag{7}$$

Here, $\boldsymbol{w}(\xi^P)$ denotes the parameter which is obtained by learning of examples ξ^P. It is expected that the generalization error (7) becomes small as the number of examples P increases. If so, how many number of examples is required to reduce the generalization error to some tolerance level? How complex network should be used? To answer such questions is the main topics of the theory of generalization.

Since the eq. (7) still depends on each realization of examples ξ^P, it is not a proper quantity for evaluation of learning. Instead, its expectation over ξ^P

$$\varepsilon \equiv < \varepsilon(\boldsymbol{w}(\xi^P)) >_{\xi^P} \tag{8}$$

is often used for the measure of generalization. This depends only on the number of examples P and its relation is called *learning curve*. Unfortunately, exact calculation of (8) is generally difficult even for simple networks because of nonlinearity. Nevertheless, it was recently shown that application of statistical mechanics (SM) enables us to calculate learning curves exactly under some simplified conditions[4, 5]. We explain this method briefly in the next section.

3 Statistical Mechanics of Learning

Learning of neural networks is considered as minimization of training error (5). In SM, this is represented by Gibbs-Boltzmann distribution for synaptic parameter $\boldsymbol{w}$

$$\rho(\boldsymbol{w}|\xi^P) = \exp[-E(\boldsymbol{w}|\xi^P)/T]/Z(T|\xi^P). \tag{9}$$

Here, "temperature" T is a parameter which modulates shape of distribution (9) and the normalization factor $Z(T|\xi^P)$ is defined as

$$Z(T|\xi^P) \equiv \int \exp[-E(\boldsymbol{w}|\xi^P)/T] d\boldsymbol{w} \tag{10}$$

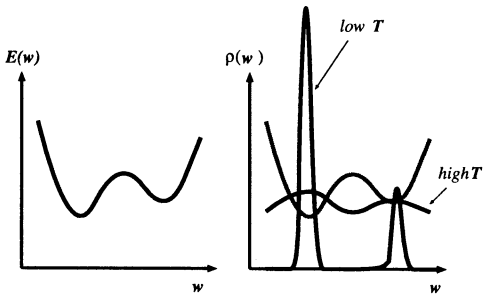


Figure 2: Energy $E(\boldsymbol{w})$ and Gibbs-Boltzmann distribution $\rho(\boldsymbol{w})$

As is shown in figure 2, $\rho(\boldsymbol{w}|\xi^P)$ is maximized at the minimum point of $E(\boldsymbol{w}|\xi^P)$ and its shape becomes steep as T is reduced. In the limit $T \to \infty$, this becomes the "δ-distribution" at the minimum point of

$E(\boldsymbol{w}|\xi^P)$. In this sense, this formulation is a generalization of learning which is mentioned in the previous section.

In SM, properties of thermodynamic systems, composed of many sub-units, are obtained via extremization of "free energy"

$$F(T|\xi^P) = -T\ln Z(T|\xi^P), \tag{11}$$

with respect to several macroscopic variables called "order parameters". However, since our system includes random variables ξ^P, another averaging operation over ξ^P corresponding to (8) is required after the extremization of free energy, which is usually difficult to evaluate analytically.

One way to avoid such difficulty is to average eq. (11) over ξ^P first. Although the direct calculation of average of logarithm of random variables $\ln Z(T|\xi^P)$ is also difficult, it is possible evaluate it by using an identity $\ln a = \lim_{n\to 0}(a^n - 1)/n$ as

$$< \ln Z(T|\xi^P) >_{\xi^P} = \lim_{n\to 0} \frac{< Z(T|\xi^P)^n >_{\xi^P} - 1}{n}. \tag{12}$$

For integer n,

$$Z(T|\xi^P)^n = \int \exp[-\sum_{a=1}^{n} E(\boldsymbol{w}_a|\xi^P)/T] \prod_{a=1}^{n} d\boldsymbol{w}_a, \tag{13}$$

has a physical meaning of the simultaneous partition function for n replicated systems with the identical randomness ξ^P, which can be analytically evaluated. Although the calculation is not possible for non-integer cases, limiting operation in the r.h.s. of (12) is executed with analytical connection of the result for integer n.

This method, which is developed in spin glass theory in physics, is called *replica method* and is now actively applied to calculation of learning curves of neural networks. In the following, we will show several basic results obtained by this method.

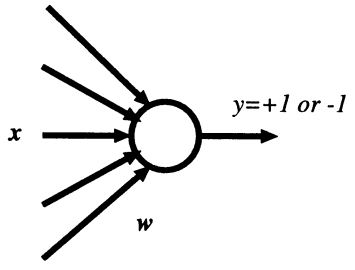


Figure 3: Simple perceptron

Example 1: Learning curve of simple perceptrons (realizable case) [6]

Simple perceptron is a simplest neural network which is composed of N input units and one binary output unit (figure 3). For input $vecx$, the perceptron with synaptic weight $\boldsymbol{w}$ returns a binary output

$$y = \mathrm{sign}[\boldsymbol{w}\cdot\boldsymbol{x}]. \tag{14}$$

As the simplest case, we here consider the case that a perceptron learns from a teacher perceptron of the same structure with synaptic weight $\boldsymbol{w}_0$.

For further simplicity, let us assume that the distribution of input (6) is a uniform distribution on an N-dimensional hyper-sphere centered at the origin. Due to this simplicity, the system is characterized with the following two order parameters

$$R \equiv \frac{\boldsymbol{w}\cdot\boldsymbol{w}_0}{|\boldsymbol{w}||\boldsymbol{w}_0|}, \tag{15}$$

$$q \equiv \frac{\boldsymbol{w}_a\cdot\boldsymbol{w}_b}{|\boldsymbol{w}_a||\boldsymbol{w}_b|}, \tag{16}$$

and free energy is evaluated as

$$\frac{< \ln Z(0|\xi^P) >_{\xi^P}}{N} = \mathrm{ext}_{\{q,R\}}\{\frac{2P}{N}\int DtH[\frac{Rt}{\sqrt{q-R^2}}] \times \ln H[\sqrt{\frac{q}{1-q}}t] + \frac{1}{2}\ln[1-q] + \frac{q-R^2}{2(1-q)}\}, \tag{17}$$

for $T = 0$ and $N, P \to \infty$. Here, $Dt \equiv dt\exp[-t^2/2]/\sqrt{2\pi}$ and $H(x) \equiv \int_x^{+\infty} Dt$.

Since output of the perceptron is a binary value $y = \pm 1$, generalization error is represented by probability of false predictions

$$\varepsilon = \frac{\cos^{-1}(R)}{\pi}. \tag{18}$$

By substituting R which is numerically solved from eq. (17) into this, one can obtain a learning curve. In particular, asymptotic learning curve

$$\varepsilon \sim 0.624\frac{N}{P}, \tag{19}$$

is analytically obtained for $P/N \to \infty$.

Example 2: Learning curve of simple perceptrons (unrealizable case) [7]

In the previous example, we assumed that the structure of teacher network is the same as that of student. However, such assumption is not realistic in practical situations. What type of learning curve is

obtained when teacher's rule is unrealizable by student network?

When student network is a simple perceptron, simplest unrealizability of teacher network is effectively represented by a perceptron of which input is corrupted by uncorrelated Gaussian noise η

$$y = \mathrm{sign}[\boldsymbol{w}_0 \cdot (\boldsymbol{x} + \eta)]. \tag{20}$$

Learning curve of this model obtained by the replica method is found to exhibit a nontrivial power law

$$\varepsilon - \varepsilon_{min} \sim (\frac{N}{P})^{2/3}. \tag{21}$$

This implies that results obtained for realizable situations is not directly applicable for realizable situations which are more practical.

Example 3: Learning curve of Ising perceptrons[8]

In some cases, adjustable machine parameters are not continuous but discrete. Perceptron, of which synaptic weights $\boldsymbol{w} = (w_i)$ $(i = 1, 2, \cdots, N)$ are constrained to binary values $w_i = \pm 1$, is a simple example of such learning machines. This type of perceptron is called *Ising perceptron* from an analogy to Ising model which is a theoretical model of magnet in SM. What type of learning curve is obtained for such machines?

Learning curve obtained by the replica method exhibits quite different behavior from that of machines with continuous parameters. Generalization error does not decrease gradually but suddenly vanishes from a finite value to zero when the number of examples P becomes a critical value which is the same order of the number of input units N (figure 4).

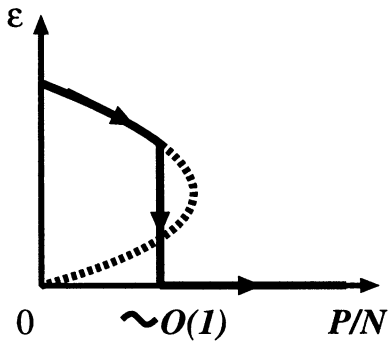


Figure 4: Learning curve of Ising perceptrons

4 Summary

In this paper, we reviewed theoretical framework of learning and generalization in neural networks and application of statistical mechanics to analysis of learning curves. Statistical mechanical analysis taken up here are based on the following policies.

Simplification: Make a model as simple as possible by assuming symmetry, uniformity and so on, with keeping essential points of the system.

Description by Order Parameters: By using the central limit theorem, describe the system with several order parameters.

Such policies are also effective for analysis of other complex systems than neural networks[9]. SM will be applied to much more problems in future.

Acknowledgment

This work was partially supported by the program "Research For The Future (RFTF)" of the Japanese Society for the Promotion of Science.

References

[1] Valiant LG 1984 *Comm ACM* **27** 1134

[2] Minsky ML and Papert SA 1969 *Perceptrons* (MIT, Cambridge, MA)

[3] Rumelhart DE and McClelland 1986 Eds *Parallel Distributed Processing* (MIT, Cambridge, MA)

[4] Gardner E 1988 *J Phys A* **21** 257

[5] Levin EN, Tishby N and Solla SA 1989 in *Proceedings of the 2nd Workshop on Computational Learning Theory* (Morgan Kaufmann, San Mateo)

[6] Györgyi G and Tishby N 1990 in *Neural Networks and Spin Glasses* (World Scientific, Singapore)

[7] Uezu T and Kabashima Y 1996 *J Phys A* **29** L55; Uezu T, Kabashima Y, Nokura K and Nakamura N 1996 *J Phys Soc of Japan* **65** 3797

[8] Györgyi G 1990 *Phys Rev A* **41** 7097

[9] Kabashima Y and Saad D 1997 *"Statistical Mechanics of Error-Correcting Codes"* submitted to *Phys Rev Lett*

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A Canceling Method Against Equi-Status Noises for Rotary Optical Sensor

K. Mine, N. Kubota, F. Morimoto, K. Fujii and S. Yamaguchi
Kyushu Kyoritsu University
Kitakyushu city, Yahata-Nishiku, Jiyugaoka1-8, Japan, 807

Abstract

In this paper we have proposed a removal method of equi-status normal mode noise not only on transmission line but also on sensor line, for optical speed sensor similar to optical encoder.

1. Introduction

Accident investigation for robotics is influenced by noise. In Japan, the accident ratio from 1982 to 1987, is found as 1/2,700 robots, even if technical progress has been so fast [1]. Especially removal method of equi-status noise is one of the important but difficult techniques, not only on robotics but also in technical diagnostics as well [2]. Random noise is generated by the vibration of electrons in the sensor in the instrumentation circuit [3]. Wiener [4], Kalman and Bucy [5] have presented methods for countermeasuring random noise. If the signal and the noise are separated in the frequency domain, the noise can be reduced by filtering or by using a cepstrum [6]. These methods effectively reduce both additive and convoluted noises. By acquiring the reference noise, the adaptive noise canceller cancels the noise [7]. The noise within the signal is reduced by finding the specific noise [8], even if neither the signal nor the noise could be separated in the frequency domain. The switching noise is canceled by using a sensor and a sensor-dummy [9], but this method cannot cancel noise on a signal line.

We have preciously proposed methods of Alternating Noise Canceling (ANC) as removing equi-status noises in instrumentation using strain gauge [10] and others [11], [12], thermocouple, resistance thermometer sensor, photo sensor or potential transformer etc. As an example, we made two signals by alternating the switching between a sensor and a dummy resistance output signals. The sensor output includes true signal and noise, the dummy resistance output includes only noise from electro-magnetic environment. The data processing of the two signals made a reliable signal without any noise on the signal line. In this case, "any noise" means normal mode noise (NMN) under a quarter frequency of switching.

Rotary optical sensors such as encoder and speed sensors are used in robotics, artifical machine and others. We have previously proposed a removal method of equi-status NMN on transmission line for optical speed sensor. But the above method can not remove equi-status NMN on sensor line [13].

In this paper we have proposed a removal method of equi-status NMN not only on transmission line but also on sensor line, for optical speed sensor similar to optical encoder. We used a photo-interrupter as a sensor using slitt disk and another photo-interrupter as a sensor-dummy without using the disk. If the same value equi-status NMNs are added on sensor line and on sensor-dummy line respectively, these NMNs can be removed using a differential amplifier.

The differential amplifier is connected to an analog switch. If output of the

analog switch makes alternative signals such as S and $-S$, equi-status NMN will be added on the transmission line such as $S + n_2$ and $-S + n_2$ sequentially. The transmission line is connected to a hybrid-IC with holder, low-pass filter, subtracter etc. The output signal of the device, 2S should be without noise. The signal is not only without ordinary NMN but also without equi-status NMN using interpolation by low-pass filter and subtracting by subtracter.

The accuracy of this method has been verified with experiments.

2. Principle

Fig. 1 Shows schematic diagram of this system. Photo-sensor is constructed of photo - interrupter with LED and photo-sensor. Dummy sensor is also constructed similarly as the above photo-sensor but the light is shielded by black tape.

In Fig. 1, S is the sensor output signal. D is zero signal as dummy output. n_1' and the n_1'' are NMNs on sensor line and dummy sensor line respectively. n_2 is NMN on the transmission line.

Removing of n_1' is shown by the following equation.

$$(S + n_1') - (D + n_1'') = \widehat{(S)} \fallingdotseq (S) \qquad (1)$$

where $n_1' \fallingdotseq n_1'' \fallingdotseq n_1$

Removing of n_2 is shown by the following equation.

$$(Va)_{i-1} = (S+n_2)_{i-1} = (S)_{i-1} + (n_2)_{i-1}$$

at $t = i-1$ (2)

$$\widehat{(Va)}_{i-1} = \{(-S+n_2)_i + (-S+n_2)_{i-2}\}/2$$

$$\fallingdotseq \widehat{(-S)}_{i-1} + \widehat{(n_2)}_{i-1}$$

at $t = i-1_1$ (3)

$$(Va)_{i-1} - \widehat{(Va)}_{i-1} = (S)_{i-1} - \widehat{(-S)}_{i-1}$$

$$+ (n_2)_{i-1} - \widehat{(n_2)}_{i-1}$$

$$= \widehat{(2S)}_{i-1} \fallingdotseq (2S)$$

at $t = i-1$ (4)

where $(S)_{i-1} = -\widehat{(-S)}_{i-1}$

and $(n_2) \fallingdotseq \widehat{(n_2)}_{i-1}$

3. Experiment

Fig. 2 Shows a circuit diagram of this system.

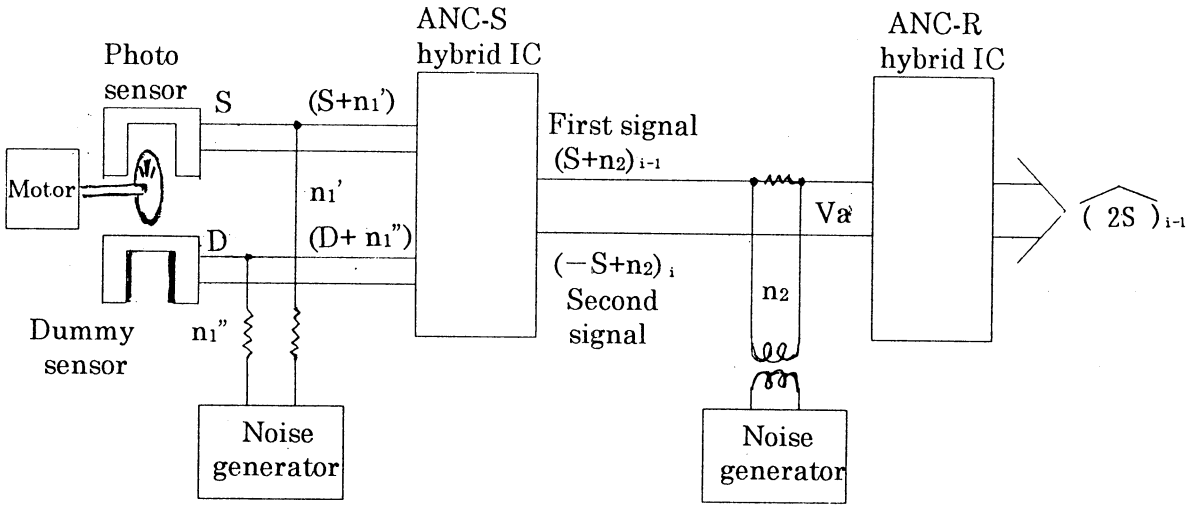


Fig.1 Schematic diagram of a canceling method against equi - status NMN for optical sensor.

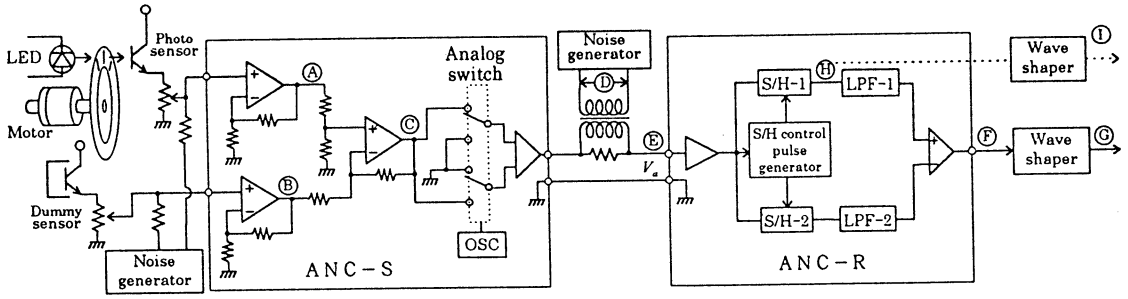


Fig. 2　A circuit diagram of this system.

Fig. 3 Shows the result of the first experiment. The circled capital letters correspond to the same symbols of each location in Fig. 2.
Signal with noise (S + n_1') at Ⓐ on sensor line, rectangular NMN n_1" at Ⓑ on dummy sensor line and sensor output signal S at Ⓒ are shown in Fig. 3.

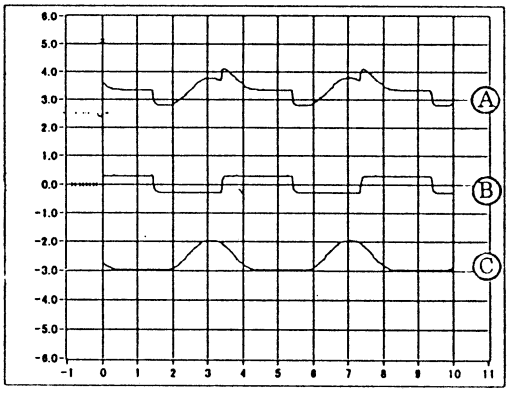


Fig. 3

Fig. 4. Shows the result of the second experiment. Sensor output signal S at Ⓒ, rectangular NMN n_2 at Ⓓ on transmission line, input signals (S + n_2)$_{i-1}$, (-S + n_2)$_i$ as upper envelope and lower envelope respectively, ANC - R hybrid IC Va at Ⓔ and output signal $\widehat{(\ 2S\ )}_{i-1}$ without rectangular NMN n_2 at Ⓕ are shown in Fig. 4.

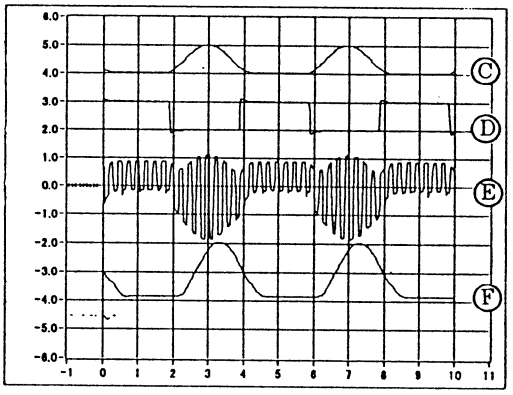


Fig. 4

Fig. 5 Shows the result of the third experiment. Output signal $\widehat{(\ 2S\ )}_{i-1}$ without rectangular NMN n_2 at Ⓕ, output signal of S / H -1, at Ⓗ, output of wave shaper at Ⓖ and output of another wave shaper at imagined Ⓘ in Fig. 2 that is connected to Ⓗ of S / H - 1 by a dotted line. Its wave form is irregular influenced by rectangular NMN n_2, as shown in Fig. 5. Where, signal S is 2.6 kHz. Rectangular NMNs, n_1 and n_2 are 2.5 kHz respectively. Analog switching frequency is 40 kHz.

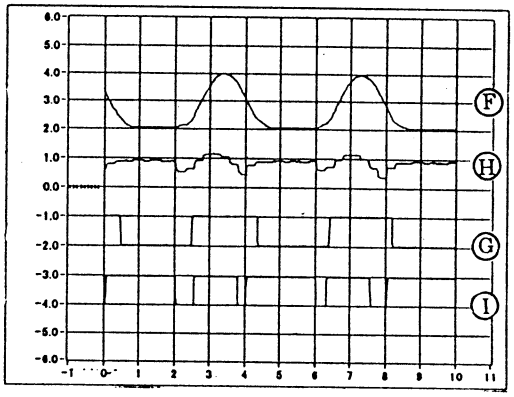


Fig. 5

4. Summary

(1) We have developed an ANC method for optical measurement with photo-interrupter, LED, photo-sensor and rotary slitt disk, as shown in Fig. 1 and Fig. 2.
(2) The method allows us to get reliable signal not only without NMN but also without equi - status NMN even in the absence of reference noise or any specific noise.
(3) The accuracy of this method has been verified with experiments as shown in Fig. 3, Fig. 4, and Fig. 5.

Acknowledgement

The authors wish to express their sincere appreciation to Mr. T. Fukuhara, the chairman of the board of director, Fukuhara Gakuen University Consortium, and Mr. R. Fukamachi, the president of Asahi Techneion Co. Ltd., for their financial support.

REFERENCES

[1] S. Kumegawa , "Mis-operation by Electromagnetic Noises and Safety countermeasure for Industrial Robotics and Others", MEC, vol.18, pp. 14-20, 1989.(In Japanese).
[2] L. Anzai and T. Endo, "On-site indication of damage under complex loading", International Journal of Fatigue, Vol. 1, pp.49-57, 1979.
[3] L. Jones and A. F .Chin, Electronic Instruments and Measurements, New York : Wiely, 1983.
[4] N. Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series, MA : MIT Press, 1949.
[5] R. E. Kalman and R. S. Bucy, "New Results in linear Filtering and Prediction Theory", Trans. ASME, Basic Eng., pp. 95-108, 1961.
[6] R. C. Kemerait and D. G. Chiders, "Signal detection and extraction by Cepstrum techniques", IEEE, Trans. Inform. Theory, Vol. IT-18, No. 6, pp. 745-759, 1972.
[7] B. Widrow and R. Winter, "Neural Nets for Adaptive Filtering and Adaptive pattern Recognition", IEEE Comput., Vol.21, No. 3, pp. 25-39, 1988.
[8] K. Mike , K. Ogawa, M. Yamada, Y . Morimoto, and Y. Jinnouch, "A Method for the Removal of the Specific Noise", Proc. IMEKOXI, Applications, (Houston), pp. 55-62, 1988.
[9] H . Ozaki and K. Taniguchi, Sensor and Signal Processing, Tokyo : Kyoritsu, 1988. (In Japanese).
[10] K . Mine and Y. Morimoto, "Method of Alternating Noise Canceling for an Instrumentation Using Strain Gages", IEEE. Trans Industrial Electronics, Vol. 37, No. 3, pp. 250-252, 1990.
[11] K . Mine, N . Kubota and F. Morimoto, "An Optical Instrumentation Using a Sensor with a Sensor dummy Against Noises on Sensor and Transmission Line", IEEE. Trans EMC, Vol. 36, No. 3, pp. 253-255, 1994.
[12] K . Mine, S . Yamaguchi, F . Morimoto, N . Kubota and S. Ogata, "Classification of Methods of Alternating Noise Canceling Against Equi-status Noise on Instrumentation and a Method Against Common Mode Noise", Proc. IMEKO XIV, Vol. IVA (Tampere), pp. 225-230, 1997.
[13] Y . Morimoto, K . Mine and A. Matsumoto, "Noise Reduces Instrumentation by Creating an Intelligent Optical Encoder", Proc. AIMac, IMEKO, TC7, (Kyoto), pp. 237-241, 1991.

EMC : Electromagnetic Compatibility
IMEKO : International Measurement Confederation
AIMaC : Artificial Intelligence based Measurement and Control

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Sequential Stereo Vision and Phase Transition

Y.Yoshitomi, T.Kitazoe, J.Tomiyama and Y.Tatebe

Department of Computer Science and Systems Engineering,
Faculty of Engineering, Miyazaki University
1-1 Gakuen Kibanadai Nishi Miyazaki 889-21 Japan
yoshi@cs.miyazaki-u.ac.jp

Keywords:neural network, binocular neuron, stereo vision, moving picture, phase transition

Abstract

By using the modified pattern recognition equations describing a dynamic self-organizing process with competition and cooperation, a mechanism of sequential stereo vision has been investigated, on the condition that the input information was introduced in the equations through the factor expressing the exponential growth of the activity of binocular neuron. The process where the new winner beat the former winner in the sequential stereoscopic depth perception had basically two stages. The first stage was that the activities of binocular neurons of former winners decreased by themselves. The second stage was that the activities of binocular neurons of new winners increased stepwise. These two processes were caused through nonequilibrium phase transition. Moreover, the nonequilibrium phase transition was taken place for each binocular neuron with some time-shift.

1 Introduction

From the physiological point of view, it has been suggested that the stereoscopic perception is mainly carried out by binocular neurons in the second visual area (V2) in brain, while the moving perception is mainly carried out in the Middle Temporal visual area (MT)[1]. Moreover, the integration of information from V2 and MT has been investigated[1]. Since the systematic physiological study on brain is a hard work, for further understanding of the process of sequential stereo vision, it is effective that the sequential stereoscopic depth perception is simulated with a simple macroscopic model.

In stereoscopic vision, a three-dimensional scene is imaged from two different points, and the reconstruction of the acquired two-dimensional images follows in order to measure the disparity and estimate the depth. Even if the object is relatively moving, we can recognize the sequential stereo vision, spending some amount of time to process the information coming from left and right eyes. When the sequential stereo vision accompanies the depth change of an object, the process means in terms of binocular neurons activities that a winner neuron corresponding to a specific depth of an object becomes a loser with handing over its role to a new winner neuron which corresponds to another depth. Our brains are able to carry out the complicated procedure with a kind of neural network.

It has been shown that it is possible for human to extract a depth map using the disparity of corresponding image features[2]. Though various algorithms for solving the problem of stereo vision have been reported[3, 4, 5, 6, 7, 8], the process of sequential stereoscopic depth perception has not been clarified yet.

For simulating the process of sequential stereoscopic depth perception, we choose the recent modified pattern recognition equations[8], because the input and the activities of binocular neurons can be independently treated in the equations, where a variable, called an activity of binocular neuron, is assigned to each possible disparity, and some procedures such as competition, cooperation and self-organizing process are also used for simulating neural network. Furthermore, the algorithm includes a potential model and a simulated annealing process, resulting in being applicable to any initial condition of binocular neurons through finding the global minimum of energy or potential function.

In the present study, we investigate the process of sequential stereoscopic depth perception through changing the input during the simulation with the recent modified pattern recognition equations[8].

2 Algorithm for Stereo Vision

With a time-dependent activity of binocular neuron, $\xi_{ab}^{uv}(t)$, the modified Coupled Pattern Recognition (CPR) equations, described below, have been recently developed[8].

$$\dot{\xi}_{ab}^{uv}(t) = -\frac{\partial U}{\partial \xi_{ab}^{uv}(t)} \tag{1}$$

$$U(\xi_{ab}^{uv}(t)) = \frac{\alpha}{2}\xi_{ab}^{uv}(t)^2 - \frac{E}{3}\xi_{ab}^{uv}(t)^3 + \frac{C}{4}\xi_{ab}^{uv}(t)^4 \tag{2}$$

$$\alpha = -\lambda_{ab}^{uv} + (B+C)\sum_{a'b'}{}' \xi_{a'b'}^{uv}(t)^2 - D\sum_{u'v'}{}' \xi_{ab}^{u'v'}(t)^2 \tag{3}$$

,where $U(\xi_{ab}^{uv}(t))$ is the non-vanishing part of U under the derivative in equation (1). The first term in α describes an exponential growth of the amplitude $\xi_{ab}^{uv}(t)$, the second term means a competition effect between all amplitude, the third term expresses a cooperative coupling between the different pattern recognition processes. Moreover, the second term in the potential expressed in equations (2) is specially added to break the left-right symmetry in the potential, the third term in the potential restricts the growth of an amplitude. B, C, D and E are positive constants which have to be chosen appropriately. In $\sum'_{a'b'}$ the summation indices $a'b'$ run over the disparity search area (DSA) defined as $-a_s \leq a \leq a_s$, $-b_s \leq b \leq b_s$ with the restriction $a' \neq a$ and $b' \neq b$[7]. In $\sum'_{u'v'}$ the summation indices u', v' run over the cooperation area (CA) defined as $u-l \leq u' \leq u+l$, $v-l \leq v' \leq v+l$ with the restriction $u' \neq u$ and $v' \neq v$[7].

Here, a deviation of the similarity function from its mean value in the DSA is used as a value of λ_{ab}^{uv}[7]. To solve the correspondence problem of sequential stereo vision, the modified CPR equations are used. In this form of potential function, it is the most marked that the coefficient element for $\xi_{ab}^{uv}(t)^2$, α, depends on λ_{ab}^{uv},$\xi_{a'b'}^{uv}(t)$ and $\xi_{ab}^{u'v'}(t)$. Through λ_{ab}^{uv}, α depends on the input to not only the binocular neuron itself but also other binocular neurons in the DSA. Moreover, α is time-dependent through $\xi_{a'b'}^{uv}(t)$ and $\xi_{ab}^{u'v'}(t)$.

The similarity function $S(u,v,a,b)$ between the feature L of an image point (u,v) in the left image and R of $(u+a, v+b)$ in the right image is given as follows:

$$S(u,v,a,b) = E - \frac{\iint |L(x,y,u,v) - R(x+a, y+b, u+a, v+b)| dxdy}{\iint f(x,y) dxdy} \tag{4}$$

where

$$L(x,y,u,v) = f(x-u, y-v) g_L(x,y) - \overline{g}_L(u,v) \tag{5}$$

$R(x,y,u,v)$ is defined correspondingly. E is a constant. $g_L(x,y)$ and $g_R(x,y)$ are the left and right stereo input data at the point (x,y). The area which defines the features is given by the window function $f(x,y)$. $\overline{g}_L(u,v)$ and $\overline{g}_R(u,v)$ are the local mean gray values of the features corresponding to the image points (u,v). Since the disparity can vary only between certain limits, the search has to be performed in between. In the definition of $S(u,v,a,b)$, a,b are limited in DSA.

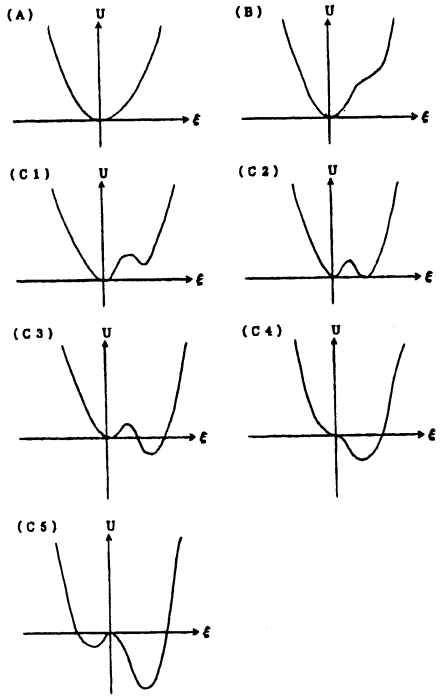


Figure 1: Schematic diagram of the potential function.

The potential function is classified into the following seven patterns, as shown in Figure 1, where the categories A ,B and C come from the conditions that $E^2 - 4C\alpha$ is negative, zero and positive, respectively. Moreover, the category C is divided into five pieces. First, C1, C2 and C3 are from the condition with positive α. Then, C1, C2 and C3 link to the conditions that $E^2 - 4.5C\alpha$ is negative, zero and positive, respectively. Next, C4, C5 are from the conditions that α is zero, negative, respectively.

Since λ_{ab}^{uv} is a deviation of the similarity measure from its mean value in the DSA, α depends not only on the input to the binocular neuron itself, but also on those to other binocular neurons in the DSA. Moreover, α is time-dependent through $\xi_{a'b'}^{uv}(t)$ and $\xi_{ab}^{u'v'}(t)$ of binocular neurons in the DSA and CA, respectively. From the consideration on equilibrium and nonequilibrium, the potential function is time-dependent and can have local minimum, local maximum and saddle point. In addition, a nonequilibrium phase transition[9] may happen in the CPR processes. Therefore, the equilibrium may not be realized if the initial condition or the kinetics of $\xi_{ab}^{uv}(t)$ is inappropriate. However, the initial stage or the state kinetics of the binocular neurons in our brains is considered to have some variety. Therefore, the model for stereo vision should have some kind of robustness to both the initial condition and the kinetics. In this sense, noise is introduced to the model, with the so-called simulated annealing method.

3 Numerical Results and Discussion

In the present study, for simplicity, one-dimensional stereo vision was simulated. Accordingly, $\xi_{ab}^{uv}(t)$ was simplified as $\xi_a^u(t)$. Moreover, the DSA and the CA were also one-dimensional. The dynamic input from the moving object to left and right eyes were simplified as two sequential input, the first and the second, where the second input was presented after the first stereoscopic depth perception had been achieved.

The random 50 dots written below were used as the first input.

Left(1st to 50th)
01010000111011000111010011000110110011100111100101

Right(1st to 50th)
01010000111011000111010100110001110011100111100101

In these dots, the displaced right dots were produced by planting the left dots from 21st to 30th in the right-dot-places from 23rd to 32nd in order.

As the second input, the same random dots to left eye were also used. Then, the second displaced right dots were produced by planting the left dots from 21st to 30th in the right-dot-places from 19th to 28th in order.

In the present study, a was chosen to be from -3 to 3 which corresponded to the DSA. The window function was defined in the range of a from -3 to 3. The value l of the CA was chosen to be 3. The used values of constants, B, C, D, E, in the modified CPR equations (1), (2) and (3) were 0.000001, 0.00025, 0.0001 and 0.01, respectively. These constant values were selected by trial and error in order to get the successful simulation results with the modified CPR equations. The constant E in the similarity function (4) was 4, which presented the successful simulation results for simple stereo vision with the previous CPR equations [7].

To solve the problem of sequential stereo vision, the modified CPR equations were used, where the values of similarity function $S(u,a)$, produced from the first input, were to be chosen as the initial values of the corresponding binocular neurons $\xi_a^u(t)$, through the sequential process.

In the first competing processes, a deviation of the similarity function $S(u,a)$ from its mean value in the DSA was used as a value of λ_a^u. Thus, the value of λ_a^u was produced from the first input. On the other hand, in the second competing processes, the value of λ_a^u produced from the second input was used in the modified CPR equations. Therefore, the influence of input on the CPR processes was given through the value of λ_a^u.

On the condition with the first input, the CPR processes to the equilibrium were successfully simulated, resulting in that the binocular neuron with $a = 2$ became a winner in the case of $u = 21$ to 30. Thus, the depth perception was achieved.

Figure 2 shows the second competing processes of binocular neurons with $u = 26$, where the second input was given after the first competing processes had finished. The CPR processes to the equilibrium were also successfully simulated, resulting in that the binocular neuron with $a = -2$ became a new winner, while the former winner with $a = 2$ became a loser. Thus, the sequential depth perception was successfully simulated by the modified CPR equations; a single bar from 21st to 30th appeared behind the background landscape (1st to 20th, 31st to 50th) and, then, soon, it moved forward by 4 steps, resulting in appearing in front of the background. Through the first and second competing processes, the winner binocular neuron for $u = 26$ was switched from that with $a = 2$ to $a = -2$, because of input change.

As shown in Figure 2, as an example, the second competing processes had basically two stages. The first stage was that the activity of binocular neuron of the former winner decreased by itself. The second characteristic stage was that the activity of binocular neuron of new winner increased stepwise.

Then, in order to clarify the mechanism of the se-

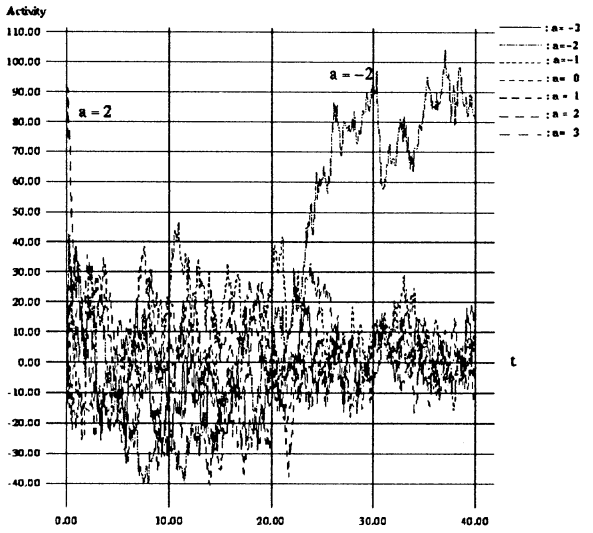


Figure 2: Second competing processes.

quential stereoscopic depth perception, the process of it has been investigated with the special reference to the kinetics of α and the potential function in the modified CPR equations.

Hereinafter, the first, the second and the third terms in α written in equations (3) are referred as the input, the competitive and the cooperative terms, respectively. By changing the input from the first to the second one, most input term corresponding to the binocular neuron of former winner was changed from the negative value to the positive one. On the contrary, by the input change, most input term corresponding to the binocular neuron of new winner was changed from the positive value to the negative one.

Figure 3 shows the potential function change caused through the α value change for the former winner binocular neuron with $a = 2$, in the second competing processes. In the first stage of the second competing processes, the value of α for the former winner increased owing to the rapid increase of the value of cooperative term. In the second stage, which was, in some cases, followed after the incubation period, the value of α for the former winner increased mainly through the stepwise increase of the value of competitive term. In Figure 3, the time values in the upper figure correspond to the characteristic points related to α value change. Through the first and second stages, the potential function for the former winner changed from the category C5 to A. Then, the shape of A type potential function for the former winner became sharper

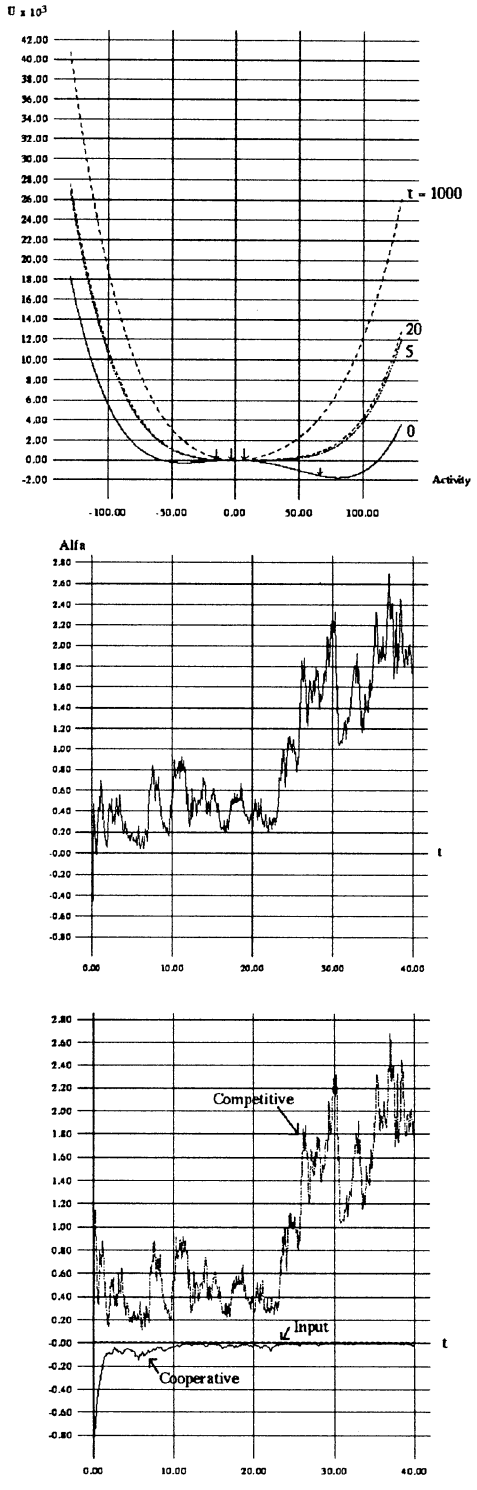


Figure 3: Potential function change caused through the α value change for the former winner during the second competing processes. The upper figure:potential function, the middle figure:α, the lower figure:the input, the competitive and the cooperative terms in α.

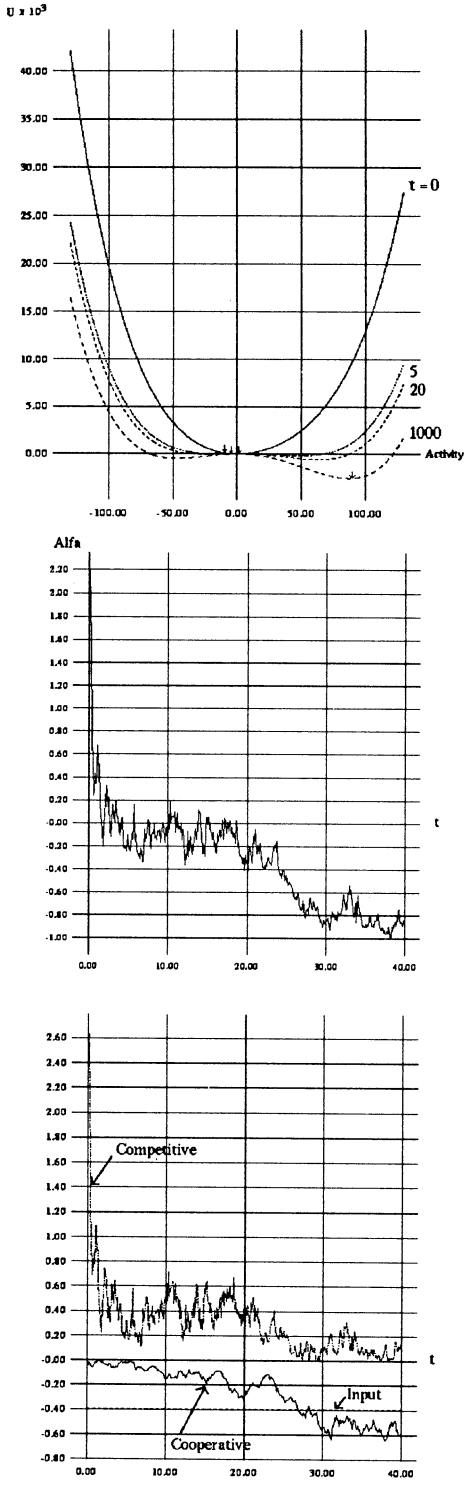


Figure 4: Potential function change caused through the α value change for the new winner during the second competing processes. The upper figure:potential function, the middle figure:α, the lower figure:the input, the competitive and the cooperative terms in α.

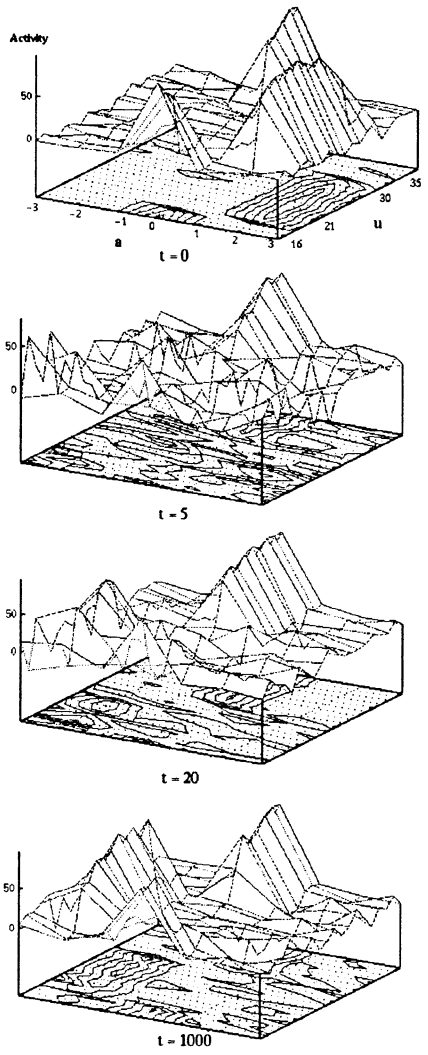


Figure 5: Activities change during the second competing processes. The contour lines of activities are written in each bottom plane.

and sharper. The arrows in the upper figure of Figure 3 indicate the values of $\xi_2^{26}(t)$ at each time.

Next, Figure 4 shows the potential function change caused through the α value change for the new winner binocular neuron with $a = -2$, in the second competing processes. In the first stage of the second competing processes, the positive value of α for the new winner decreased mainly through the rapid decrease of the value of competitive term. In the second stage, the value of α for the new winner decreased through the decrease of the values of competitive and cooperative terms. In Figure 4, the time values in the upper figure correspond to the characteristic points related to α value change. Through the first and second stages, the potential function for the new winner changed from the category A to C5. Then, the well depth of potential function for the new winner became deeper and deeper. The arrows in the upper figure of Figure 4 indicate the values of $\xi_{-2}^{26}(t)$ at each time.

As you can see from Figures 3 and 4, the sequential stereoscopic depth perception was caused through the nonequilibrium phase transition, corresponding to the shape change of the activity potential of binocular neuron. Next, the inhomogeneous depth-perception-process for stereo vision sequence is demonstrated in Figure 5. As suggested in Figure 5, the nonequilibrium phase transition was taken place for each binocular neuron with some time-shift. The observed category-change of potential function during the sequential stereoscopic depth perception was rather simple, namely, either A to C5 or reverse. However, between the potential function categories of A and C5, there are theoretically five categories of B, C1, C2, C3, C4, corresponding to the nonnegative value range of α . The detail investigation on the category transition of potential function will be presented in the forthcoming paper.

4 Conclusion

By using the recent modified pattern recognition equations describing a dynamic self-organizing process with competition and cooperation, the mechanism of sequential stereo vision has been investigated. The sequential stereoscopic depth perception was successfully simulated with the equations, where the effects of exponential growth, competition and cooperation in the coupled pattern recognition processes are expressed by one parameter α. The input information is introduced in the equations through the factor expressing the exponential growth of the activity of binocular neuron. The process where the new winner beat the former winner in the sequential stereoscopic depth perception had basically two stages. The first stage was that the activities of binocular neurons of former winners decreased by themselves. The second stage was that the activities of binocular neurons of new winners increased stepwise. These two processes were caused through nonequilibrium phase transition, corresponding to the shape change of the activity potential of binocular neuron. Moreover, the nonequilibrium phase transition was taken place for each binocular neuron with some time-shift.

References

[1] D.C.van Essen and J.H.R.Maunsell, " Hierarchical organization and functional streams in the visual contex", *Trends in Neuroscience*, pp.370–377, 1983.

[2] B. Julesz, *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, 1971.

[3] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images " , *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.PAMI-6(6),pp.721–741, 1984.

[4] D. Terzopoulos, A. Witkin and M. Kass, " Stereo Matching as Constrained Optimization Using Scale Continuation Methods, Optical and Digital Pattern Recognition " , *SPIE*, Vol.754, pp.92–99, 1987.

[5] H. G. E. Hentschel and A. Fine, " Statistical Mechanics of Stereoscopic Vision " , *Phys. Rev. A*, Vol.40, pp.3983–3997, 1989.

[6] A. Yuille, D. Geiger and H. H. Bulthoff, " Stereo Integration, Mean Field Theory and Psychophysics " , *Network*, Vol.2, pp.423–442, 1991

[7] D. Reimann and H. Harken, " Stereo Vision by Self-Organization " , *Biol. Cybern.*, Vol.71, pp.17-26, 1994.

[8] Y. Yoshitomi, T. Kanda and T. Kitazoe, " Neural Nets Pattern Recognition Equation for Stereo Vision " , *Proceedings of Annual Conference of Japanese Neural Network Society*, pp.295–296, 1996.

[9] H. Harken, *Synergetic Computers and Cognition*, Springer, Berlin Heidelberg, New York, 1991.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A New Sensorless AC Survo Motor and Variable-Speed Control System Driven by VSI

K.Fujii , K.Mine , N.Kubota , S.Yamaguchi , *S.Nonaka
Kyushu Kyoritsu University, *Kinki University in Kyushu
1-8 Jiyugaoka,Yahata-Nishiku,Kitakyushu City,807,Japan
E-mail : fujiikun @ Kyukyo-u.ac.jp
Phone : 093-693-3049
Fax : 093-603-8186

Key word : AC Survo Motor, Actuator, Brushless Synchronus Motor

Abstract

In this paper, the variable-speed drive of a new brushless self-excited type half-speed synchronous motor by the sensorless control is described. The motor can eliminate brush,slip-ring and DC-exciter for minimization of maintenance requirement. The motor has a series excitation characteristic owing to the half wave rectifying of the stator load current, a good damping effect appears at the transient state. Even in an open-loop control system, the speed-control can be attained more stably than the separately excited synchronous motor.

1. Introduction

A new brushless self-excited type half-speed synchronous motor combined with voltage source transistor inverter is presented. This motor can eliminate brush, slip ring and DC-exciter for minimization of maintenance requirement.

It is well known that a wound-rotor type induction motor with a short-circuited single-phase rotor winding runs at nearly half of synchronous speed, according to Görges phenomenon. By superimposing a stationary magnetic field on the stator winding, the motor is compelled to run at exactly half of synchronous speed, i.e., slip s=0.5. In this motor, the difference between direct- and quadrature-axis reactances produces a torque like a reluctance motor, the effective q-axis reactance can be decreased by inserting a q-axis squirrel-cage damper winding on the salient poles.

The type of the half-speed synchronous motor was investigated by Schenfer in 1926[1]and Russel and Norsworthy discussed its feasibility in 1957[2]. However, an AC voltage is induced in the field winding attached to the stator winding. In 1958, Nonaka proposed inserting a diode to one phase of three-phase stator windings to produce a stationary magnetic field[3]. In case that the stator resistance is very small in comparison with the stator reactance, the off-state duration of the diode is so short that the DC component of the stator current is almost proportional to the AC stator current. When this motor is driven by a voltage source inverter, DC components of the stator current circulates through the inverter and feed-back diodes and does not flow into the power source transformer of the converter. Therefore power source transformer core is not biased by the DC-component of the stator current [4].

The steady-state characteristics of the motor are analyzed by the state-variable methods[4] and close agreement between the analytical and the experimental results is obtained. A larger synchronous torque can be obtained when the motor is driven by a square-wave voltage source inverter than when driven by a sinusoidal-wave[5]. The maximum torque of the motor(cylindrical rotor with single-phase winding)is much larger than that of a reluctance motor and is equal to or larger than maximum torque of the three-phase induction motor having the same size[6]. The motor produces a small pulsation torque[5]but can be driven on smooth operation because of a inertia effect except at an excessively low speed drive. In 1987,Fujii et al. proposed a practical optimum rotor configuration[7]-[8]of the motor. The maximum torque of the rotor by putting a q-axis squirrel-cage damper winding on the salient-pole is much larger than that of a cylindrical rotor with single-phase winding.

In this paper, the induction motor and synchronous motor starting methods driven by a new inverter system are proposed. The motor exhibits stable operation with good damping effects for sudden change of load and speed, although the test motor is not equipped with specific damper winding. The rapid convergence is due mainly to the damping effect by the rotor current. Therefore, even in an open loop system of a variable speed control can be

attained more stably than separately excited synchronous motors. The variable speed drive of PWM VSI-fed with an open loop control system by microcomputer control has excellent performances.

2. Construction of proposed motor

A. Construction of motor

The schematic circuit diagram of a new proposed motor is shown in Fig.1. As a test motor, a conventional four-pole, 1.0kW, 60Hz, 100V, Y-connected three-phase induction motor was used. The motor has a stator of the same structure as that of ordinary three-phase synchronous motors. The test motor uses a wound-rotor induction motor(slot number:24), with the salient-pole rotor modified as core in Fig.2.

A flat type copper wire is used for the damper winding on the rotor. The coil end of the q-axis damper winding are insulated from each other with the insulating tape. The solid pole type rotor is not favorable because the d-axis flux is reduced by the eddy current.

B. Damper effect of motor

In this motor, the performances of transient response exhibits suitable damper although the tested motor is not equipped with specific damper windings. The rapid convergence is due mainly to the damping effect of the rotor current. The DC components of the stator current are almost proportional to the AC stator current. Therefore, even in an open loop speed control can be attained more stably than separately excited synchronous motor.

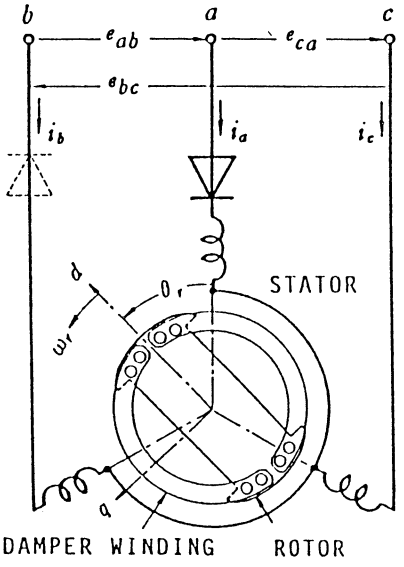


Fig.1. Fundamental circuit of proposed AC survo motor.

3. Inverter drive system of motor

If diodes are inserted into one or two phase of stator windings as shown in Fig.1, then no current flows through main power transistor Tr2 or Tr4. Therefore, it is possible to construct a completely new inverter circuit as shown Fig.3. If GTO thyristors or power transistors are used in the inverter system,then circuit can be simplified considerably and the circuit reliability can be improved. The voltage source type asymmetrical inverter is also effective for producing and controlling a stationary magnetic field. Its operating characteristics have been reported already[8].

Fundamental control functions of the inverter drive system are constructed from voltage frequency amplitude(VFA), start frequency time(SFT) and voltage frequency bias(VFB) respectively.

4. Experimental results and discussion

When the proposed motor is driven in the same condition as the rated voltage(100V) and the rated

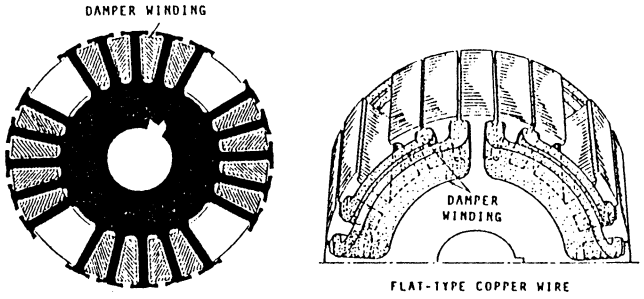


Fig.2. Structure of proposed rotor.

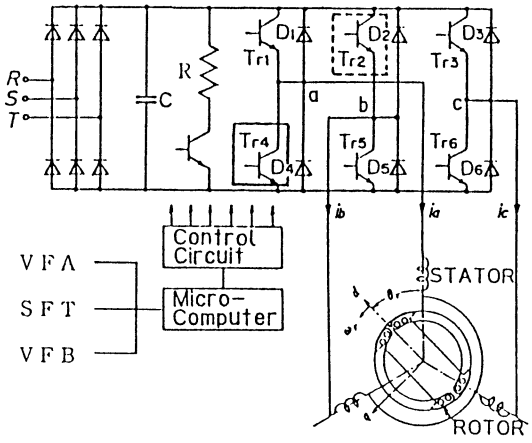
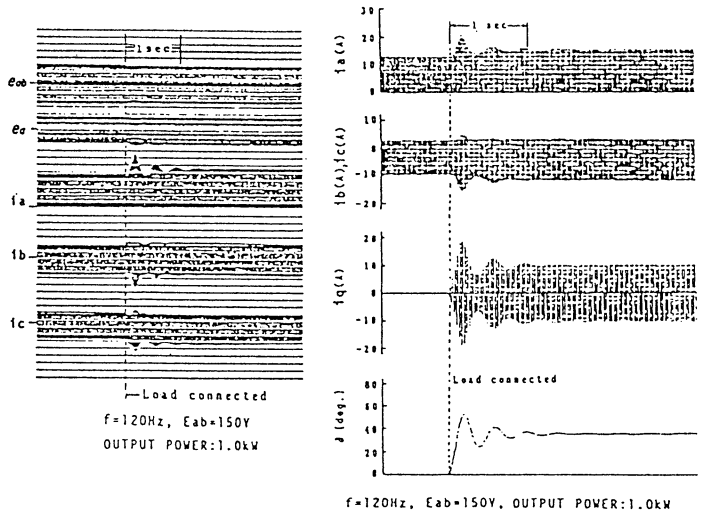
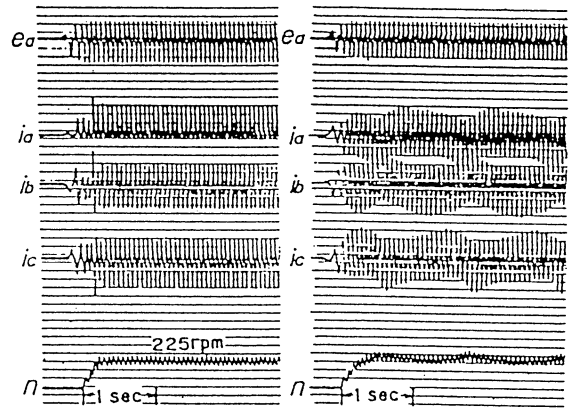


Fig.3. New inverter drive system of motor.



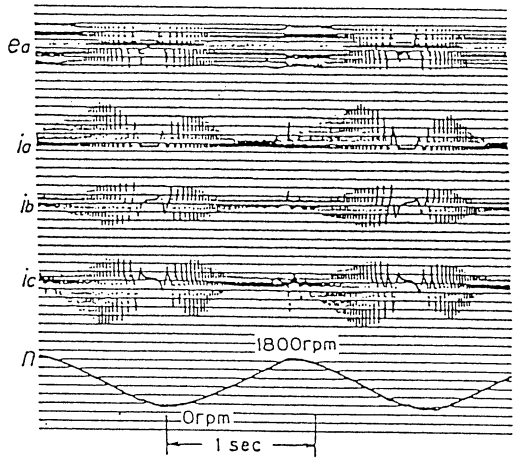
(a) Measured waveform. (b) Calculated waveform.

Fig.4. Transient response waveforms of motor.

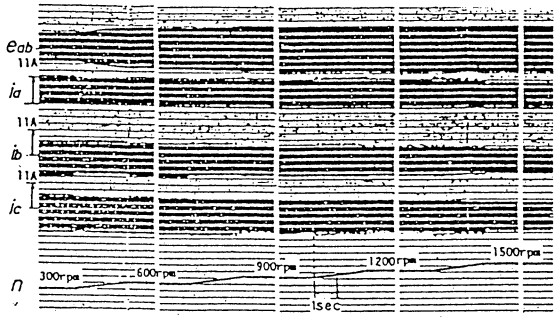


(a) Synchronous starting. (b) Induction starting.

Fig.5. Comparison with synchronous starting and induction motor starting.



(a) Oscillogram of variable-speed (light-load).



(b) Variable-speed drive of motor.

Fig.6. Variable-speed drive of proposed motor.

speed(1800rpm) of the induction motor, inverter output frequency has to be 120Hz. Therefore, magnitude of the rotating magnetic flux is reduced to a half value, consequently, it is possible to drive the motor at a voltage twice as high as the rated voltage. Taking into account the effect of core saturation due to stationary magnetic flux and withstand insulation strength of the stator winding, the maximum voltage of the motor will be 150V. A DC generator is coupled with the motor as the load.

A. Transient response of motor

The experimental and calculated waveforms of the transient response at the sudden change of the load are shown in Fig.4(a),(b), where the tested motor carrying a light load corresponding to the mechanical loss(0.14kW) or DC generator is suddenly loaded with 1.0kW. The motor has a good damping effect at transient by a series excitation characteristic owing to the half-wave rectification of the stator load current. The rapid convergence is due to damping effect of the rotor current.

B. Starting methods of motor

1) Induction motor starting

To start the proposed motor in Fig.5, we first control it by Tr4:ON in the VSI, and operate the motor as a three-phase induction motor. Transistor Tr4 is opened to insert diode when the motor speed approaches to slip 0.5. Then a current is superimposed on the stator windings to synchronizing the motor as shown in Fig.5(b)

2)Synchronous starting

The synchronous starting method is driven by transistor Tr4:OFF at low frequency in Fig.5(a).

C. Variable-speed performances

In this motor, the performances of transient response to the sudden change of load and motor speed are well damped although the test motor is not equipped with specific damper windings. In Fig.6(a),(b) and Fig.7 the oscillograms of the variable-speed drive performance driven by synchronous starting method are shown. Fig.8 illustrates the oscillograms of the variable-speed drive performances driven by transistor Tr2:OFF,Tr4:OFF. The variable-speed control is achieved by v/f control of PWM inverter. The variable-speed performances of this drive system will be applied to sensorless AC servo-motor.

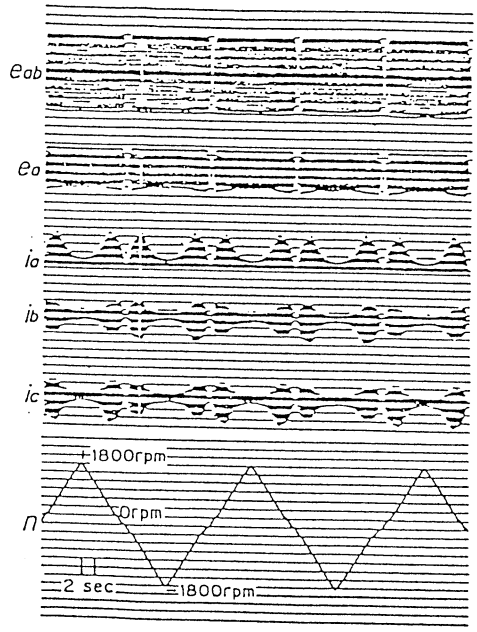


Fig.7. Oscillogram of variable-speed(no-load).

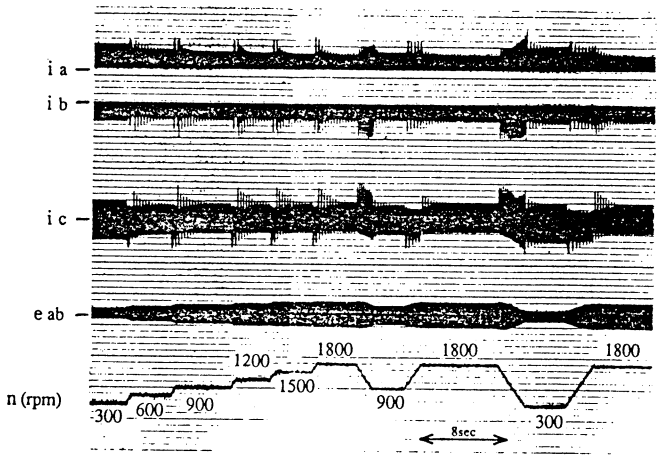


Fig.8. Oscillograms of variable-speed drive by Tr2:OFF, Tr4:OFF

5.Conclusion

The variable-speed operating of the proposed motor driven by the new inverter system has the following features;

1)simple and robust structure of motor.
2)stable operation without damper winding except q-axis damper winding.
3)stable variable-speed drive without a position detector and a feedback circuit. And
4)very simple control system

It has become clear that the proposed motor has excellent performances for constant-speed and even variable-speed drive.

References

[1]C.Schenfer, "Der synchron-induktions motor mit erregung im stator." Electrotechnik und Maschinonbau,vol.19,pp.345-354,May 1926

[2]R. L .Russel and K.H.Norsworthy. "A stator-fed half-speed synchronous motor." In Proc. Inst. Elect. Engrs., vol.104,Feb,1957,pp.77-87

[3]S. Nonaka, "A method of self-excited three-phase synchronous motor," Kansai Br. Joint Conv. Four Inst. Engrs. Japan, no. 84,1958

[4]S. Nonaka and K. Fujii, "Brushless self-excited three-phase synchronous motor driven by voltage source inverter," Trans. Inst. Elect. Engrs. Japan, Vol. 103, pp.81-89,July/Aug. 1983.

[5]S. Nonaka, K. Fujii, and T. Kawaguchi, "Performance characteristics of brushless salient-pole self-excited three-phase synchronous motor driven by transistor inverter," Trans. Inst. Elect. Engrs. Japan, Vol. 104B, pp.281-288,May 1984; Elect. Engrs. Japan, Vol. 104, pp.111-118,May./Apr. 1984

[6]S. Nonaka, K. Fujii, T. Matsumoto, and T. Kamano, "Static field superposition of brushless self-excited type half-speed synchronous motor and its performance characteristics," Technology Report of the Kyushu University, vol.54, pp.535-542,Oct.1981.

[7]S. Nonaka, K. Fujii, and T. Matsumoto, "Analysis of brushless self-excited type half-speed synchronous motor driven by voltage source inverter and its performance characteristics," Trans. Inst. Elect. Engrs. Japan, Vol. 102-B, pp.297-304, May 1982; Elect. Engrs. Japan, Vol. 102, pp.50-58,May./June 1982.

[8]S. Nonaka, and K. Fujii, " A new brushless half-speed synchronous motor with Q-axis squirrel-cage damper winding driven by voltage source inverter," IEEE Trans. Ind. App., vol. 1A-23,No.4 July/Aug. 1987.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Modeling and Complexity in Neural Networks

Kazuyuki Aihara and Natsuhiro Ichinose
Department of Mathematical Engineering and Information Physics
Graduate School of Engineering
The University of Tokyo
7 - 3 - 1 Hongo, Bunkyo-ku, Tokyo 113, JAPAN

Abstract

In this paper, we study nonlinear spatio-temporal dynamics in synchronous and asynchronous chaotic neural networks from the viewpoint of modeling and complexity in the dynamical brain. By the synchronous chaotic neural networks, possible functions of spatio-temporal neurochaos are considered for practical applications in chaos engineering. Further, rich deterministic point-process dynamics with spatio-temporal spikes is demonstrated with a biologically more plausible model of asynchronous chaotic neural networks.
Keywords: Chaos, Neural Networks, Brain, Computation, Spatio-Temporal Dynamics

1 Introduction

After great progress of researches on deterministic chaos in 1970's and 1980's, the impact of chaos has been extending not only to sciences but also to engineering and society [7, 8, 9, 18, 39]. The importance of chaos for engineering is very profound because concepts of deterministic chaos are greatly influencing such basic theories of engineering as prediction theory, control theory, computational theory, information theory and so on [7, 8, 18, 39]. This paper considers possible computation with spatio-temporal chaos in the brain, which is one of important research topics on relationship between chaos and computational theory.

2 Nonlinear Spatio-temporal Dynamics in the Brain

Biological neurons, or constituent elements of the brain are usually modeled as simple elements like the McCulloch-Pitts model with discrete 1 or 0 output and synchronous updating [28] in the field of artificial neurocomputing. However, behavior of real neurons is far more complicated and dynamical than one of such simplified neuron models. In particular, experimental studies with squid giant axons and numerical studies with the Hodgkin-Huxley equations [19] and other nerve equations have clarified that biological neurons are entirely different from linear threshold neurons like the McCulloch-Pitts model at least in the meaning that the former has excitable nonlinear "dynamics" with threshold, which produces deterministic chaos in the level of nerve membranes [3, 4, 5, 10, 11, 12, 26, 27]. Another point which should be noted is that generation and propagation of action potentials in real neurons are asynchronous.

These properties of chaotic dynamics and asynchronous updating in real neurons should generate rich spatio-temporal nonlinear phenomena in neural networks. In this paper, we consider such spatio-temporal dynamics in neural networks with models of chaotic neural networks.

3 Synchronous Chaotic Neural Network Model and Spatio-temporal Dynamics

A model of synchronous chaotic neural networks [3, 4, 6, 13] is proposed by considering physiological properties of relative refractoriness and a continuous stimulus-response curve inherent in excitable dynamics of the nerve membranes, and spatio-temporal summation of both external inputs and feedback inputs from other constituent neurons.

The dynamics of the ith chaotic neuron integrated in a synchronous chaotic neural network with M external inputs and N constituent neurons is described as follows [4, 6, 13]:

$$x_i(t+1) = f\Big[\sum_{j=1}^{M} v_{ij} \sum_{d=0}^{t} k_e^d A_j(t-d)$$

$$+ \sum_{j=1}^{N} w_{ij} \sum_{d=0}^{t} k_f^d x_j(t-d)$$

$$- \alpha \sum_{d=0}^{t} k_r^d g\{x_i(t-d)\} - \Theta_i \Big], \quad (1)$$

where $x_i(t+1)$ is the output of the neuron with a continuous value between 0 and 1; t shows the discrete time steps ($t = 0, 1, 2, \cdots$); f is the output function which is usually assumed to be the logistic function of $f(y) = 1/\{1 + \exp(-y/\varepsilon)\}$ with the steepness parameter ε; $A_j(t)$ is the strength of the jth external input at t;v_{ij} and w_{ij} are synaptic weights to the ith neuron from the jth external input and from the jth constituent neuron, respectively;k_e, k_f and k_r are the decay factors between 0 and 1 for the external inputs, the feedback inputs and the refractoriness, respectively; g is the refractory function assumed to be $g(x) = x$ for simplicity; α and Θ_i are the refractory scaling parameter and the resting threshold, respectively.

Equation(1) can be transformed into the following simplified and simultaneous form under the assumption of the exponential temporal decay in eq.(1) [4, 6, 13]:

$$\xi_i(t+1) = \sum_{j=1}^{M} v_{ij} A_j(t) + k_e \xi_i(t), \quad (2)$$

$$\eta_i(t+1) = \sum_{j=1}^{N} w_{ij} x_j(t) + k_f \eta_i(t), \quad (3)$$

$$\zeta_i(t+1) = -\alpha g\{x_i(t)\} + k_r \zeta_i(t) - \theta_i, \quad (4)$$

where $\xi_i(t+1)$ $(\equiv \sum_{j=1}^{M} v_{ij} \sum_{d=0}^{t} k_e^d A_j(t-d))$, $\eta_i(t+1)$ $(\equiv \sum_{j=1}^{N} w_{ij} \sum_{d=0}^{t} k_f^d x_j(t-d))$ and $\zeta_i(t+1)$ $(\equiv -\alpha \sum_{d=0}^{t} k_r^d g\{x_i(t-d)\} - \Theta_i)$ are internal state terms for the external inputs, the feedback inputs from the neurons in the network and the refractoriness, respectively and $\theta_i \equiv \Theta_i(1 - k_r)$. The output $x_i(t+1)$ of the ith neuron is calculated by $x_i(t+1) = f\{\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\}$. The neuron model of eqs. (2) - (4) includes some of conventional discrete–time neuron models such as the McCulloch–Pitts neurons [28] and analog neurons with the logistic output function used in backpropagation neural networks [29] as special cases; namely, the model of the synchronous chaotic neural networks is a natural extension of the conventional discrete–time neural networks for introducing chaotic dynamics into these usual neural network models in order to study possible computational roles of deterministic chaos in artificial and biological information processing by comparing the performances of the synchronous chaotic neural networks with those of the conventional neural networks [3, 4, 6, 13].

The model of synchronous chaotic neural networks generates various complex and computational neurodynamics with spatio–temporal chaos through nonlinear interactions among elemental neurons with their own chaotic dynamics. For example, the dynamics with refractoriness produces spatio–temporal chaos with fractal structure in the phase space without getting stuck at local minima. This is because any neurons can't keep firing if the decay factor k_r for the refractoriness is sufficiently close to 1 [2]. This ability has been applied to dynamical association of spatial patterns stored through a covariance weight matrix in associative memory networks [2, 4] and to dynamical searching for good approximate solutions in combinatorial optimization networks [14, 15, 33, 38].

The chaotic neurodynamics can be harnessed to converge to a fixed point by e.g. a kind of deterministic simulated annealing with transient chaos which we call chaotic simulated annealing [14, 15, 33].

The spatio–temporal summation of many synaptic inputs in chaotic neurons can be also effectively utilized for self–organization dynamics which detects and learns an unknown pattern with automatic switching between learning and retrieving modes according to distances between the input pattern and previously stored patterns [35, 36]; this self–organization dynamics is similar to experimental results observed by Freeman and his colleagues in an olfactory system [16, 31].

Although the simple model of the synchronous chaotic neural networks has little direct relation to biological neural networks, it has abundant and curious spatio–temporal dynamics with engineering applicability. In fact, computation with chaotic neural networks, or chaotic parallel distributed processing [8] is one of important research topics in chaos engineering [7, 8, 9]. Electronic circuits and analog IC have been also proposed to implement the chaotic neurons as chaotic analog devices [20, 24, 30].

4 Asynchronous Chaotic Neural Network Model and Spatio–temporal Dynamics

Although our experimental results with squid giant axons were obtained under periodic forcing [5, 10, 11, 12, 26, 27], the important fact clarified by the experiments is that the single neurons can respond not only periodically but also chaotically depending

upon parameter values of stimulation even if the input is periodic. Real response of biological neurons *in vivo*, therefore, should be dynamical indeed and more complicated. Towards more biological modeling, we can extend the model to asynchronous versions with continuous time intervals between action potentials or bursts of firing as follows [21, 22].

The effect of the external inputs is expressed by the following equation :

$$\xi_i(t) = \sum_{j=1}^{S_e} v_{ij} \sum_{n=1}^{N_{ij}} e^{-(t-t_{ij,n})/\tau_e} s_{ij,n}. \quad (5)$$

Here, ξ_i is the internal state for the external inputs, S_e is the number of external inputs, v_{ij} is the synaptic weight from the jth external input to the ith neuron, N_{ij} is the number of pulses which have stimulated the ith neuron from the jth external input, τ_e is the time constant of the temporal decay for the external input effect, $s_{ij,n}$ and $t_{ij,n}$ are, respectively, the strength and the input time of the nth pulse from the jth external input to the ith neuron.

The effect of the feedback inputs is expressed by

$$\eta_i(t) = \sum_{j=1}^{S_f} w_{ij} \sum_{n=1}^{M_j} e^{-(t-t'_{j,n}-D_{ij})/\tau_f} h_{ij}(s'_{j,n}). \quad (6)$$

Here, η_i is the internal state for the feedback inputs, S_f is the number of neurons in the network, w_{ij} is the synaptic weight from the jth neuron to the ith neuron, M_j is the total number of pulses generated by the jth neuron, D_{ij} is the pulse propagation delay including the synaptic delay from the jth neuron to the ith neuron, τ_f is the time constant of the temporal decay for the feedback input effect, h_{ij} is a function representing the waveform–shaping characteristics during the pulse propagation from the jth neuron to the ith neuron, $s'_{j,n}$ and $t'_{j,n}$ are, respectively, the strength and the generation time of the nth pulse generated by the jth neuron.

The effect of the relative refractoriness is described as follows :

$$\zeta_i(t) = -\alpha \sum_{n=1}^{M_i} e^{-(t-t_{i,n})/\tau_r} g_i(s'_{i,n}) - \theta_i, \quad (7)$$

Here, ζ_i is the internal state for the relative refractoriness, α is a scaling parameter for the strength of the relative refractoriness, τ_r is the decay time constant, θ_i is the standard threshold, and g_i is the refractory function.

The total internal state of the ith neuron is expressed by the sum of the three internal states as follows :

$$y_i(t) = \xi_i(t) + \eta_i(t) + \zeta_i(t). \quad (8)$$

Under the assumption that input spikes are asynchronous and instantaneous, we can define order on occurrences of input pulses according to their arriving time. For the external inputs, all pulses on all the input lines can be represented by an ordered sequence. Let us denote the sequence obtained this way with $\{s_n, t_n\}$ where s_n and t_n are the strength multiplied with the synaptic weight and the arriving time of the nth input spike, respectively. The change of the internal state for the external input is expressed as

$$\xi_i(t_{n+1}) = e^{-(t_{n+1}-t_n)/\tau_e} \xi_i(t_n) + s_{n+1}. \quad (9)$$

Similarly, for the feedback inputs, the set of each pulse arriving at $t'_{j,n} + D_{ij}$ produces an ordered sequence, which we denote by $\{q_n, u_n\}$. The change of the internal state for the feedback input is expressed as

$$\eta_i(u_{n+1}) = e^{-(u_{n+1}-u_n)/\tau_f} \eta_i(u_n) + q_{n+1}. \quad (10)$$

If we assume that the response of a neuron is instantaneous at arriving time of the external inputs and the feedback inputs, then the occurrence time of the output forms a mixed sequence comprising those of external and feedback inputs. If the time when an input pulse newly ordered as explained above occurs is $t'_{i,n}$, then the strength $s'_{i,n}$ of the output is defined by :

$$s'_{i,n} = f_i(y_i(t'_{i,n})). \quad (11)$$

Therefore, the change of the internal state for the refractoriness is expressed by :

$$\begin{aligned} \zeta(t'_{i,n+1}) &= e^{-(t'_{i,n+1}-t'_{i,n})\tau_r} (\zeta_i(t'_{i,n}) - \alpha g_i(s'_{i,n})) \\ &- (1 - e^{-(t'_{i,n+1}-t'_{i,n})/\tau_r})\theta_i \end{aligned} \quad (12)$$

The asynchronous chaotic neural networks can be utilized not only for studying possibilities of spatio–temporal coding and processing with timing of the events and continuous inter–event intervals [21, 22, 23, 25] but also for modeling various biological neurons from integrators to coincidence detectors [1, 32]. For example, emergence of dynamical cell assemblies in neural networks composed of coincidence detector neurons can be described by the asynchronous chaotic neural networks [17, 34, 37].

5 Conclusion

Mathematical theory of neural networks has been providing interesting nonlinear phenomena in neural networks as a typical example of complex systems. Models of synchronous and asynchronous chaotic neural networks produce abundant large–scale complex dynamics on the basis of nonlinearity of constituent elements and their interactions. Although reductionism approaches to such nonlinear complex systems as the brain are usually thought to be useless, this does not necessarily deny importance of studies on modeling and complexity of a constituent element, or a single neuron in the case of the brain. Indeed, difference of the neuron characteristics between integrators and coincidence detectors essentially influences possible mechanism of mutual interactions among neurons and possible representation of information in neural networks [17]. In this sense, modeling and complexity of a single neuron are quite important for understanding the dynamical brain.

References

[1] M. Abeles, *Israel J. Med. Sci.*, **18**, 83–92, 1982.

[2] M. Adachi and K. Aihara, *Neural Networks*, **10**/1, 83–98, 1997.

[3] K. Aihara, *Neural Computers*, Tokyo Denki University Press, Tokyo, 1988 (in Japanese).

[4] K. Aihara, in *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems* (Ed. H.Kawakami), 143, World Scientific, Singapore, 1990.

[5] K. Aihara (Ed.), *Chaos in Neural Systems*, Tokyo Denki University Press, Tokyo, 1993 (in Japanese); the Korean–language edition, Ohm Ltd., Seoul, 1994.

[6] K. Aihara (Ed.), *Neural, Fuzzy and Chaotic Computing - Introduction to a New Generation of Analog Computing*, Ohm-sha, Tokyo, 1993 (in Japanese); the Korean–language edition, Dae Kwang, Seoul, 1994.

[7] K. Aihara (Ed.), *Applied Chaos and Applicable Chaos*, Science-sha, Tokyo, 1994 (in Japanese); the Korean–language edition, Ohm Ltd., Seoul, 1996.

[8] K. Aihara and R. Katayama, *Communications of the ACM*, **38**/11, 103–107, 1995.

[9] K. Aihara, R. Katayama, *J. of Intelligent & Fuzzy Systems*, **5**/1, John Wiley & Sons, Inc., 1997.

[10] K. Aihara, G. Matsumoto and Y. Ikegaya, *J. Theor. Biol.*, **109**, 249–269, 1984.

[11] K. Aihara and G. Matsumoto, in *Chaos* (Ed. A.V. Holden), Manchester University Press, Manchester and Princeton University Press, Princeton, 257–269, 1986.

[12] K. Aihara, T. Numajiri, G. Matsumoto and M. Kotani, *Physics Letters A*, **116**, 313–317, 1986.

[13] K. Aihara, T. Takabe and M. Toyoda, *Physics Letters A*, **144**, 6/7, 333–340, 1990.

[14] L. Chen and K. Aihara, *Neural Networks*, **8**/6, 915–930, 1995.

[15] L. Chen and K. Aihara, *Physica D*, **104**, 286–325, 1997.

[16] W. J. Freeman, *Int. J. of Bifurcation and Chaos*, **2**/3, 451–481, 1992.

[17] H. Fujii, H. Ito, K. Aihara, N. Ichinose and M. Tsukada, *Neural Networks*, **9**/8, 1303–1350, 1996.

[18] C. Grebogi and J. A. Yorke (eds.), *The Impact of Chaos on Science and Society*, United Nations University Press, Tokyo, 1997.

[19] A. L. Hodgkin and A. F. Huxley, *J. Physiol.*, **117**, London, 500, 1952.

[20] Y. Horio, K. Suyama, A. Dec and K. Aihara, *Proc. WCNN94*, **4**, 690, 1994.

[21] N. Ichinose and K. Aihara, in *Towards the Harnessing of Chaos* (ed. by M.Yamaguti), Elsevier Science Publishers B.V., Amsterdam, 353, 1994.

[22] N. Ichinose and K. Aihara, *Trans. IEICE*, **J78–A**/3, 373–380, 1995 (in Japanese); the English–language translation, Electronics and Communications in Japan, Part 3, **78**/11, 52–59, Scripta Technica, Inc., John Wiley, New York, 1995.

[23] K. Judd and K. Aihara, *Neural Networks*, **6**/2, 203–215, 1993.

[24] N. Kanou, Y. Horio, K. Aihara and S. Nakamura, *Trans. IEICE*, **E76–A**/4, 642–644, 1993.

[25] W. Maass, *Advances in Neural Information Processing Systems* (Ed. G. Tesauro, D. Touretzky and T. Leen), **7**, 185–190, 1995.

[26] G. Matsumoto, K. Aihara, Y. Hanyu, N. Takahashi, S. Yoshizawa and J. Nagumo, *Physics Letters A*, **123**, 162–166, 1987.

[27] G. Matsumoto, K. Aihara, M. Ichikawa and A. Tasaki, *J. Theor. Neurobiol.*, **3**, 1–14, 1984.

[28] W. S. McCulloch and W. H. Pitts, *Bull.Math.Biophys.*, **5**, 115, 1943.

[29] D. E. Rumelhart, J. L. McClelland and the PDP Research Group, *Parallel Distributed Processing*, MIT Press, Cambridge, 1986.

[30] K. Shimizu, K. Aihara and M. Kotani, *Electronics and Communications in Japan*, Part3, **74**, 1, 51, Scripta Technica, 1991.

[31] C. A. Skarda and W. J. Freeman, *Behavioral and Brain Science*, **10**, 61, 1987.

[32] W. R. Softky and C. Koch, *J.Neuroscience*, **13**, 334–350, 1993.

[33] I. Tokuda, K. Aihara and T. Nagashima, *Neural Networks*, (in press).

[34] M. Watanabe and K. Aihara, *Neural Networks*, (in press).

[35] M. Watanabe, K. Aihara and S. Kondo, *Trans. of IEICE*, **J78–A**/6, 686–691, 1995.

[36] M. Watanabe, K. Aihara and S. Kondo, in *Control and Chaos* (Ed. by K. Judd et al.), 320–333, Birkhäuser, 1997.

[37] M. Watanabe, K. Aihara and S. Kondo, *Biol. Cybern.*, (in press).

[38] T. Yamada and K. Aihara, *J. Intelligent and Fuzzy Systems*, **5**/1, 53–68, 1997.

[39] M. Yamaguti (ed.), "Towards the Harnessing of Chaos", Elsevier, Amsterdam, 1994.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Complex Dynamics of Chaotic Neural Networks with an Asynchronous Updating

Masaharu Adachi*
Laboratory for Open Information Systems
The Institute of Physical and
Chemical Research (RIKEN)
Wako-shi, Saitama 351-01, Japan

Kazuyuki Aihara
Department of Mathematical Engineering
and Information Physics
The University of Tokyo
Bunkyo-ku, Tokyo 113, Japan

Abstract

Dynamical characteristics of chaotic neural networks with asynchronous updating rules are analyzed in the present paper. Effects of asynchronous updating of the chaotic neural network are discussed in relation to dynamical association.

Keywords: Chaos, Associative Neural Networks, Asynchronous Updating

1 Introduction

Interesting dynamical characteristics on discrete-time mathematical neuron models which include refractoriness represented by a self-recurrent inhibition (or self-feedback from engineering viewpoint) were found by Nagumo and Sato [12]. The Nagumo-Sato model assumed that the neuron output takes binary values (e.g., 0 or 1) which correspond to firing or resting states of the neuron. An extended neuron model of the Nagumo-Sato model is given by assuming that the neuron output takes graded values (e.g., [0, 1]) and the network model composed of such a neuron model is also given [4][5]. Such neuron model and neural network model are called *Chaotic neuron model* and *Chaotic neural networks*, respectively, because both single neuron and network model show deterministic chaos with certain parameter values.

It has been reported that when the chaotic neural network is applied to an associative neural network the network shows dynamical association [1][4][14]. Here, the dynamical association means that a process that is described by a non-periodic dynamical system but the output pattern sequence of the system includes all stored patterns embedded in the associative network. Such association is completely different from the associations realized in conventional associative network models since the conventional networks retrieve only one of the stored patterns [13][11][8] [15][9] or a stored cyclic sequence [6] from their nearest initial pattern and keep the steady states after the retrieval.

In the conventional associative neural networks, the updating rules for the state variables of the network are not a matter to be considered because the updating rules do not affect to the dynamics of the network; the dynamics of the constituent neurons of the network is simple; i.e., each neuron takes the weighted sum of the inputs to the neuron and just transfer it through the Heaviside function or sigmoid functions. On the other hand, in chaotic neural networks the dynamics is affected by the updating rules since the dynamics of the constituent neurons is not simple; i.e., the state of each neuron is determined not only by the present inputs to the neuron but also by the previous state of the neuron itself. In most of the previous studies on the associative chaotic neural networks, the updating of the state variables of the network was limited to the synchronous updating, where the synchronous updating means that the states of all the constituent neurons are updated simultaneously at each discrete-time step.

Recently, an asynchronous updating rule for associative chaotic neural networks is proposed and the retrieval characteristics of the networks with the updating rule are preliminary studied [2][3]. In those studies, numerical experiments show that the network with a deterministic asynchronous updating rule shows better retrieval abilities than the network with synchronous updating [2] and the network with a stochastic asynchronous updating rule [3]. In the dynamical system theoretic point of view, a numerical experimental evidence shows that the network with the deterministic updating rule is not periodic at least for 5×10^4 iterations [2].

In the present paper, a brief review on the chaotic neuron model and chaotic neural networks is given, then the associative chaotic neural networks with synchronous and asynchronous updating rules are introduced. At last, investigations on the dynamical feature

*MA is supported by the Special Postdoctoral Research Program of RIKEN.

of the network with the deterministic updating rules are shown.

2 A Chaotic Neuron Model and Chaotic Neural Networks

The chaotic neuron model [4][5] is represented by

$$x(t+1) = f\Big[A(t) - \alpha\sum_{d=0}^{t} k^d g\{x(t-d)\} - \Theta\Big], \quad (1)$$

where $x(t)$ is the neuron output at the discrete time t; f denotes the output function; $A(t)$ denotes the external stimulation at the time t; g denotes the refractory function; α, k and Θ denote the refractory scaling parameter, the refractory decay parameter and the threshold, respectively. When we assume that the neuron output takes graded value, the output function f must be a continuous function; e.g., the logistic function $f(y) = 1/\{1+\exp(-y/\varepsilon)\}$ is typically used as the output function. It must be noted that when f is a binary function like the Heaviside function the model is equivalent to the Nagumo-Sato model [12]. Defining an internal state $y(t+1)$ of the neuron by

$$y(t+1) = A(t) - \alpha\sum_{d=0}^{t} k^d g\{x(t-d)\} - \Theta, \quad (2)$$

we can obtain the following reduced difference equation on the internal state:

$$y(t+1) = ky(t) - \alpha g[f\{y(t)\}] + \theta(t), \quad (3)$$

where $\theta(t) \equiv A(t) - kA(t-1) - \Theta(1-k)$. The output $x(t+1)$ of the neuron is calculated from the internal state $y(t+1)$ through the output function f as follows:

$$x(t+1) = f\{y(t+1)\}. \quad (4)$$

The chaotic neuron model is extended to a neural network model whose constituent neurons are the chaotic neurons. Such neural network is called chaotic neural network. The dynamics of the ith constituent neuron is represented as the following equation [4][5]:

$$x_i(t+1) = f\Big[\sum_{j=1}^{M} v_{ij}\sum_{d=0}^{t} k_e^d A_j(t-d) + \sum_{j=1}^{N} w_{ij}\sum_{d=0}^{t} k_f^d x_j(t-d) - \alpha\sum_{d=0}^{t} k_r^d g\{x_i(t-d)\} - \Theta_i\Big], \quad (5)$$

where v_{ij} and w_{ij} denote synaptic weights to the ith constituent neuron from the jth external input and from the jth constituent neuron, respectively, and k_e, k_f and k_r denote the decay parameters for the external inputs, the feedback inputs and the refractoriness, respectively. As in the case of the chaotic neuron model, when we introduce three internal state variables $\xi_i(t)$, $\eta_i(t)$ and $\zeta_i(t)$ which correspond to the 1st term, the 2nd term and from the 3rd to the 4th terms in the bracket [ ] of Eq.(5), respectively, the dynamics of the ith constituent neuron is rewritten as the following three difference equations:

$$\xi_i(t+1) = k_e\xi_i(t) + \sum_{j=1}^{M} v_{ij}A_j(t), \quad (6)$$

$$\eta_i(t+1) = k_f\eta_i(t) + \sum_{j=1}^{N} w_{ij}x_j(t), \quad (7)$$

$$\zeta_i(t+1) = k_r\zeta_i(t) - \alpha g\{x_i(t)\} - \theta_i, \quad (8)$$

where $\theta_i \equiv \Theta_i(1-k_r)$. Then the neuron output is determined by

$$x_i(t+1) = f\{\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\}. \quad (9)$$

When we construct an associative chaotic neural network, we assume that the external inputs $A_j(t)$ are constant in time. With this assumption, the difference equations Eqs.(6)-(8) for the chaotic neural networks are reduced to the following two difference equations.

$$\eta_i(t+1) = k_f\eta_i(t) + \sum_{j=1}^{N} w_{ij}x_j(t), \quad (10)$$

$$\zeta_i(t+1) = k_r\zeta_i(t) - \alpha x_i(t) + a_i, \quad (11)$$

consequently the neuron output becomes

$$x_i(t+1) = f\{\eta_i(t+1) + \zeta_i(t+1)\} \quad (12)$$

where a_i denotes the sum of the threshold and the temporally constant external inputs to the ith neuron.

Figure 1 shows a bifurcation diagram of a constituent chaotic neuron in the associative chaotic neural network when it is isolated from other constituent neurons by setting $w_{ij} = 0$ for all i, j. From Fig.1 we can confirm that single chaotic neuron shows chaotic dynamics with certain parameter values.

3 Associative Chaotic Neural Networks

In order to construct an associative neural network which stores spatial patterns, the synaptic weights

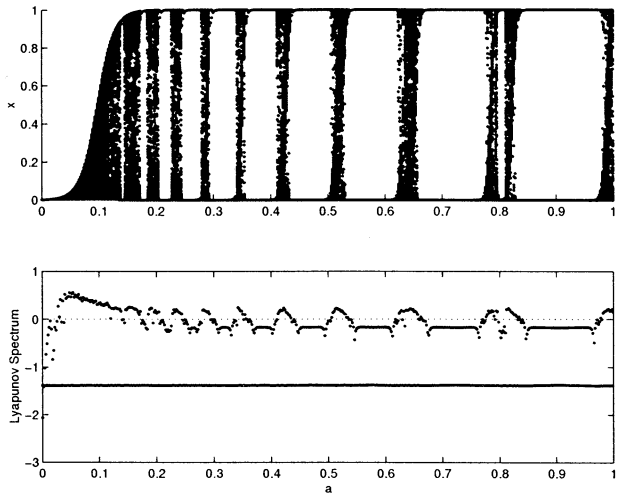


Figure 1: A bifurcation diagram of the chaotic neuron model. (Upper): Bifurcation diagram of a neuron output x_i; (Lower): The Lyapunov spectrum computed with the two internal state variables η_i and ζ_i. The parameter a_i is used as the bifurcation parameter and other parameter values are fixed to $\varepsilon = 0.015, k_f = 0.25, k_r = 0.85$ and $\alpha = 10$.

must be determined according to the spatial patterns to be stored. For example, to store the 4 spatial patterns of 100-pixels binary patterns shown in Fig.2, a neural network with 100 neurons is used and the synaptic weights are determined by

$$w_{ij} = \sum_{p=1}^{4} (x_i^p - \overline{x})(x_j^p - \overline{x}) \qquad (13)$$

where x_i^p is the ith component of the pth stored pattern[7]. Similar synaptic weights are used in conventional associative networks [13][11][8][15] [9]. The chaotic neural network of Eqs.(10)-(12) with such synaptic weights is called *Associative Chaotic Neural Network*. It must be noted that when $\alpha = k_f = k_r = 0$ the network is equivalent to the conventional associative networks [13][11] [8][15][9].

As is already mentioned in Sec.1, dynamics of the associative chaotic neural networks is affected by the updating rules for the state variables. An asynchronous deterministic updating rule which follows the descending order of the internal states $\eta_i(t)$ for the feedback inputs was proposed [2]. More precisely, at each discrete-time t the state variable $\eta_i(t)$ are compared among all the constituent neurons (for $i = 1, 2, \ldots, 100$), then the updating neuron indices i are ranked according to the descending order of $\eta_i(t)$.

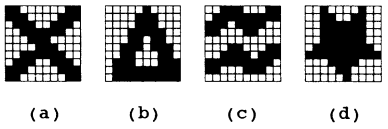


Figure 2: Stored patterns of the associative chaotic neural network. Where the patterns are displayed in the form of 10 by 10 matrices and the black and white squares represent 1 and 0, respectively.

When every neuron is updated once, the discrete-time t of the network is incremented and the next updating session for constituent neurons starts. The asynchronous deterministic updating is called *Winner updates faster*. With such an updating rule, we expect that important neurons for retrieving a stored pattern update faster than the other *less* important neurons because the larger values of $\eta_i(t)$ mean that the neurons receive strong demands to fire from the other neurons. It is already reported that the expectation is true; namely, the network with the asynchronous deterministic updating shows better retrieval ability than the networks with the synchronous updating and with a stochastic updating[2][3]. Figure 3 shows the time course of distances $d_p(t)$ between the output pattern of the network and the four stored patterns of Fig.2. The distance between the output and the pth stored pattern is defined by

$$d_p(t) = \sum_{i=1}^{100} |x_i(t) - x_i^p|$$

where the pth patterns for $p = 1, 2, 3, 4$ represent the patterns (a)-(d) of Fig.2, respectively. From Fig.3 we find that the network with the Winner updates faster retrieves all the stored patterns and the time evolution of d_p looks non-periodic.

In order to visualize the dynamics of the network with the Winner update faster in a single variable, the following quasi-energy function is used:

$$E_q(t) = -\frac{1}{2} \sum_{i \neq j} \sum_{j} w_{ij} x_i(t) x_j(t) - \sum_i a_i x_i(t). \quad (14)$$

Figure 4 shows the time evolution of the quasi-energy function in the same simulation of Fig.3. From Fig.4, we can see that the dynamics of the network looks non-periodic. It must be noted that the previous study shows that the dynamics of the network is not periodic at least for 5×10^4 iterations; the fact was shown by computing the Euclidean norm distances between internal state vectors at the end of the simulation and at each iteration step[2].

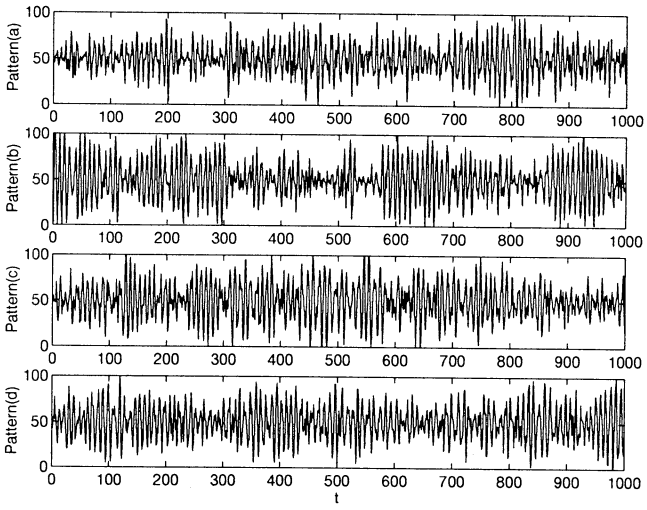


Figure 3: Distances $d_p(t)$ between the output pattern of the network and the four stored patterns with the Winner updates faster, where the parameters are set to $\varepsilon = 0.015, k_f = 0.25, k_r = 0.85, \alpha = 10$ and $a_i = 2$ for all i.

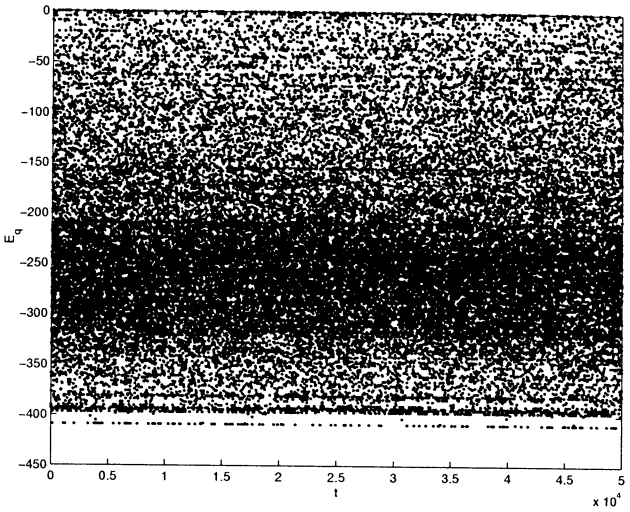


Figure 4: Time evolution of the quasi-energy function $E_q(t)$ in the same simulation of Fig.3

4 Conclusions and Discussions

The associative chaotic neural network with the deterministic asynchronous updating shows non-periodic dynamics which includes retrieval of all the stored pattern. The ability of the retrieval with the asynchronous updating is higher than that of the network with synchronous updating.

Although it is easy to implement a function like the association to the chaotic neural network with the synchronous updating[4][5], it is not easy to do so for the network with a fully asynchronous updating in continuous-time[10]. On the other hand, the former network is not biologically plausible but the latter one is biologically plausible. Therefore, study on the chaotic neural network with the deterministic asynchronous updating is useful to make a bridge between the network with the synchronous updating and the network with a fully asynchronous updating in continuous-time.

References

[1] M. Adachi & K. Aihara, "Associative Dynamics in a Chaotic Neural Network", *Neural Networks*, 10, 83–98, 1997.

[2] M. Adachi & K. Aihara, "Associative Chaotic Neural Networks with Asynchronous Updating: Winner updates faster," in *Proc. ICONIP'97*, (in printing).

[3] M. Adachi & K. Aihara, "Influence of Updating Rules on Retrievals in Associative Chaotic Neural Networks," in *Proc. NOLTA'97*, (in printing).

[4] K. Aihara, "Chaotic Neural Networks," in *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems*, ed. H. Kawakami, World Scientific, Singapore, 1990, 143–161.

[5] K. Aihara, T. Takabe & M. Toyoda, "Chaotic Neural Networks," *Phys. Lett. A*, 144, 333–340, 1990.

[6] S. Amari, "Learning patterns and pattern sequences by self-organizing nets of threshold elements," *IEEE Trans.*, C-21, 1197–1206, 1972.

[7] S. Amari, "Characteristics of sparsely encoded associative memory," *Neural Networks*, 2, 451–457, 1989.

[8] J. A. Anderson, "A simple neural network generating interactive memory," *Math . Biosciences*, 14, 197–220, 1972.

[9] J.J. Hopfield, "Neural networks and physical systems with emergent collective computation abilities," *Proc. of Natl. Acad. of Sci.*, USA, 79, 2445–2558, 1982.

[10] N. Ichinose & K. Aihara (1995): "An Analysis on Pulse Propagation Dynamics in Asynchronous Chaotic Neural Networks," *Electronics and Communications in Japan, Part 3*, 78, 52–59, 1995.

[11] T. Kohonen, "Correlation matrix memories," *IEEE Trans.*, C-21, 353–359, 1972.

[12] J. Nagumo, & S. Sato, "On a response characteristic of a mathematical neuron model," *Kybernetic*, **10**, 155-164, 1972.

[13] K. Nakano, "Associatron — a model of associative memory," *IEEE Trans.*, SMC-2, 381–388, 1972.

[14] M. Toyoda, K. Aihara, K. Shimizu, M. Adachi & M. Kotani, "Chaotic Neural Networks," *Proc. of SICE'89*, 1323–1325, 1989.

[15] H. Wigström, "A neuron model with learning capability and its relation to mechanism of association," *Kybernetic*, 12, 204–215, 1973.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III '98),
Oita, Japan, 19-21, January, 1998,

An Asynchronous Pulse Neural Network Model and Its Analog Circuit Implementation

Yoshihiko Horio, Mitsuru Hanagata and Hidekazu Yasuda
Department of Electronic Engineering
Tokyo Denki University, Tokyo, 101, Japan

Abstract

An asynchronous pulse neural network model is proposed. The model is suitable to implement a spatio-temporal coding network with an analog circuitry. Simulation results confirm that the proposed model shows complex dynamical behavior including chaotic responses. Analog circuits which implement the core part of the model are also presented.

Keywords: Neural Networks, Spatio-Temporal Coding, Chaos, Analog Nonlinear Integrated Circuits.

1 Introduction

Results from recent physiological experiments on cortical neurons suggest that a fine time structure of incoming neuronal pulses to a neuron from many other neurons plays an important role in high-level information processing [1]–[5]. At the same time, mathematical researches have revealed that a spatio-temporal coding network has high computational abilities [1, 6]–[8]. Both of these experimental and theoretical results imply that a spatio-temporal coding is essential in high-level information processing in a brain. The neuron in such a network functions as a coincident detector [1], which detects spatial correlations of the input pulses and, simultaneously, integrates the time history of the inputs. It is important to note that the state variables of the spatio-temporal coding network such as time, internal state of the neuron, nonlinear output function of the axon hillock, synaptic delay and value, and so on, are all continuous. These continuous variables enable the network to process real-numbers. This leads to a complex behavior of the network including chaos.

One disadvantage of using a *digital* computer to simulate the spatio-temporal coding network is its disability in computation of the real-numbers. Because the numbers in the digital computer are rounded-off and truncated, the digital computer cannot handle almost all real-numbers, e.g., irrational numbers. Therefore, the digital computer simulation of the dynamics of the spatio-temporal coding network may differ from the real (mathematical) one, in particular, for chaotic dynamics [9].

Another disadvantage of using digital computer simulations of the spatio-temporal coding network is computational speed. The network always consists of a vast number of nonlinear neurons whose states should be calculated in parallel. The computation of such a massively parallel nonlinear network usually takes a lot amount of time. As a consequence, a real-time processing is difficult with the digital computer because a huge computational power will be necessary for a large scale network.

Therefore, it is desirable to use an *analog* electronic circuit to implement the spatio-temporal coding network. The analog circuits can handle the real-numbers directly because their state variables such as voltage, current and time are continuous. In addition, a truly parallel computation is possible, so that the real-time processing of a large-scale network is achievable.

Even though the analog circuits can handle the real-numbers, noise in circuits is inevitable. Therefore, analog circuits never exactly simulate the original dynamics. However, the effects of noise in the analog circuits are different from those of the computational disability of the digital circuits. It is interesting to investigate the analog network with noise in comparison with the digital system with round-off and truncation.

In these respects, we have proposed several pulse neural network models suitable for analog circuit implementation [10]–[12] derived from the asynchronous chaotic neuron model [8]. Our models use a pulse with finite width, instead of an impulse used in the original model, in order to be implemented by practical hardware. The coincidence detection of the inputs, continuous pulse intervals, a relative refractoriness, an analog internal state value, a continuous output function, and synaptic delay and weight, are taken into consid-

eration in the original model [8]. In contrast, the output function of our first model was a linear threshold function. As a consequence, no chaotic response was observed [10]. We have modified this model to incorporate the continuous output function. Furthermore, an absolute refractory period and a delay in generation of the action potential have also been adopted in the models [11, 12]. As a result, the amplitude of the internal state and the time intervals of the output pulses of the neuron exhibited complex behavior. However, the output functions in these models are rather artificial.

In this paper, we further modify the former models by introducing a more natural analog output function than the former ones. The novel output function mimics a continuous characteristic of the axon hillock. Simulation results from the proposed model are shown. Moreover, analog circuits which implement the core part of the model are also presented [13].

2 The Model

Figures 1 and 2 show a schematic of the ith neuron in a network and a time-diagram of variables of the neuron, respectively. As shown in these figures, a

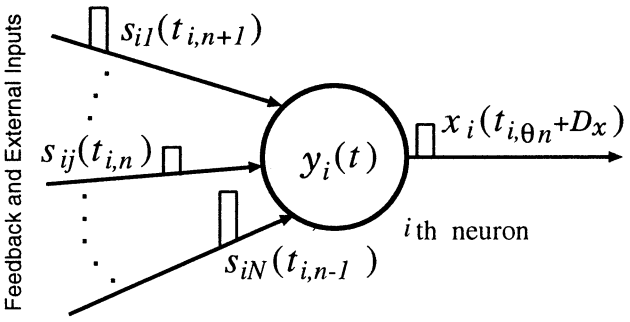


Figure 1: A schematic of the model neuron. Synaptic delays and weights are not shown.

neuron receives input pulses from other neurons and external stimuli at time, $t_{i,n}$ $(n = 0, 1, 2, \ldots)$. It is assumed that all the pulses in a network have the same pulse width of W. Moreover, the output pulse of the neuron x_i is waveform shaped by the axon with a strict threshold for propagation of action potentials initiated at the trigger zone. $h(\cdot)$ denotes such a threshold function, and is assumed to be $h(x) = 1$ for $x \geq 0.5$ and $h(x) = 0$ for $x < 0.5$. That is, pulses propagating through axons have width of W and hight of "1". By letting $UW(t_{i,n})$ denote a pulse with hight "1", width W, and rising edge at time $t_{i,n}$, the strength

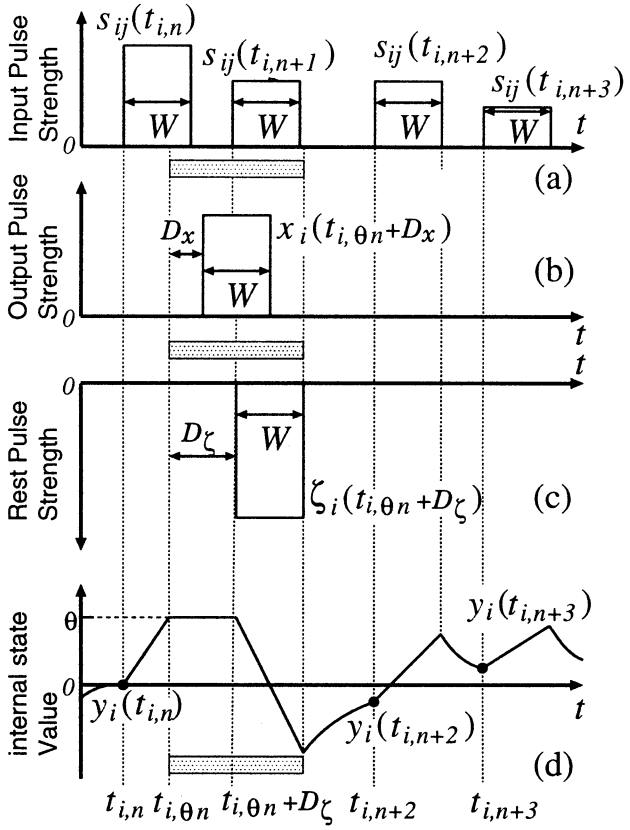


Figure 2: An example of a time diagram of the neuron. The shaded region is the absolute refractory period.

of each input pulse coming from jth neuron after a synaptic weight, w_{ij}, is expressed by $s_{ij}(t_{i,n})$, that is, $s_{ij}(t_{i,n}) = w_{ij} \times UW(t_{i,n})$. Here, w_{ij} denotes the synaptic weight from neuron j to i. Furthermore, the jth external input to the ith neuron after the synaptic weight, v_{ij}, is also expressed by $s_{ij}(t_{i,n})$, but here, $s_{ij}(t_{i,n}) = v_{ij} \times UW(t_{i,n})$. The synaptic weights, w_{ij} and v_{ij}, and synaptic delays, d_{ij} from neuron j to i and e_{ij} from jth external input, are not shown in Fig. 1 for simplicity. An internal state of the ith neuron is expressed by $y_i(t)$ at continuous time t.

When $s_{ij}(t_{i,n})$ arrives at the summing node of the neuron, the internal state changes linearly in time during W. The slope of the change at time t depends on the value of $\sum_{j=1}^{N} s_{ij}(t)$ where N is a sum of the number of the neurons in the network and that of the external input nodes to the ith neuron. If and only if the internal state reaches to the firing threshold, θ, at time $t_{i,\theta n}$ in the course of the change, then the neuron fires after a delay of $D_x (= \frac{1}{2}W$ in this paper). The amplitude of the output pulse, $x_i(t_{i,\theta n} + D_x)$, at time

$t_{i,\theta n} + D_x$ depends on $t_{i,\theta n}$ and $\sum_{j=1}^{N} s_{ij}(t_{i,\theta n})$.

This dependency corresponds to the analog output function at the axon hillock, and is given by

$$x_i(t_{i,\theta n} + D_x) = f(y'(t_{i,n} + W) - \Theta) \quad (1)$$

where $f(\cdot)$ is a monotonic nonlinear function, and Θ is an external bias. A sigmoidal function given by eq. (2) is used in this paper as $f(\cdot)$.

$$f(x) = \frac{1}{1 + e^{-x/\varepsilon}} \quad (2)$$

where ε is a constant. Furthermore, in eq. (1), $y'(t_{i,n} + W)$ is an extrapolated value of the internal state given by

$$y'(t_{i,n}+W) = y(t_{i,\theta n}) + \sum_{j=1}^{N} s_{ij}(t_{i,\theta n}) \times \frac{t_{i,n} + W - t_{i,\theta n}}{W} \quad (3)$$

Once the neuron fires, the absolute refractory period begins. Any input pulse arrives during this period is ignored. Moreover, once the internal state reaches to the firing threshold, θ, the value of the internal state is kept constant at the value of θ until a reset pulse is generated. Therefore, eq. (3) can be rewritten as

$$y'(t_{i,n}+W) = \theta + \sum_{j=1}^{N} s_{ij}(t_{i,\theta n}) \times \frac{t_{i,n} + W - t_{i,\theta n}}{W} \quad (4)$$

The neuron generates the reset pulse with a delay of $D_\zeta (= W$ in this paper).[1] The strength of the reset pulse at $t_{i,\theta n} + D_\zeta$ is denoted by $\zeta_i(t_{i,\theta n} + D_\zeta)$, which corresponds to the relative refractoriness. $\zeta_i(t_{i,\theta n} + D_\zeta)$ depends on the strength of the output pulse. In this paper, the dependency is given by

$$\zeta_i(t_{i,\theta n} + D_\zeta) = \alpha x_i(t_{i,\theta n} + D_x) \quad (5)$$

where α is a constant.

After the reset pulse is generated, the internal state decreases linearly during the pulse width W. When the reset pulse disappears, the absolute refractory period finishes. Therefore, the total length of the absolute refractory period is $D_\zeta + W$. Furthermore, when the neuron is out of the absolute refractory period and also has no input pulse, the internal state decays exponentially toward to the resting potential, i.e., 0 in this paper.

[1] The delays D_x and D_ζ should also depend on $t_{i,\theta n}$ and $s_{ij}(t_{i,n})$. In this paper, these are constant for simplicity. This assumption does not drastically change the characteristics of the model, and makes it easy to implement the model as an IC form.

2.1 Dynamical equations for the internal state

Dynamical equations for the internal state of the ith neuron at time $t_{i,n}$, i.e., $y(t_{i,n})$, is derived in order to express the proposed model as an event dynamical system [8]. Namely, the relations between $y_i(t_{i,n+1})$ and $y_i(t_{i,n})$ are explicitly given as follows.

Case (I): When $t_{i,n+1} < t_{i,\theta n} + D_\zeta$;
In this case, the neuron fires with the nth input pulse. Furthermore, the whole $(n+1)$th input pulse overlaps with the absolute refractory period generated at $t_{i,\theta n}$. The internal state, $y_i(t_{i,n+1})$, is expressed as

$$y_i(t_{i,n+1}) = \theta \quad (6)$$

Case (II): When $t_{i,\theta n} + D_\zeta < t_{i,n+1} < t_{i,\theta n} + D_\zeta + W$;
The neuron fires with the nth input pulse. Moreover, a part of the $(n+1)$th input pulse overlaps with the absolute refractory period generated at $t_{i,\theta n}$. In this case,

$$y_i(t_{i,n+1}) = \theta - \zeta(t_{i,\theta n} + D_\zeta) \cdot \frac{t_{i,n+1} - (t_{i,\theta n} + D_\zeta)}{W} \quad (7)$$

Case (III): When $t_{i,n+1} > t_{i,\theta n} + D_\zeta + W$;
The neuron fires with the nth input pulse. In this case, the $(n+1)$th input pulse does not overlap with the absolute refractory period generated at $t_{i,\theta n}$. Then,

$$y_i(t_{i,n+1}) = \{\theta - \zeta(t_{i,\theta n} + D_\zeta)\} e^{-\{t_{i,n+1} - (t_{i,\theta n} + D_\zeta + W)\}/\tau} \quad (8)$$

where τ is a time constant of the relative refractoriness.

Case (IV): When the neuron doesn't fire with nth input pulse and also $t_{i,n+1} \geq t_{i,n} + W$;
In this case, the absolute refractory period does not exist. As a result,

$$y_i(t_{i,n+1}) = \{y_i(t_{i,n}) + s_{ij}(t_{i,n})\} e^{-(t_{i,n+1} - t_{i,n} - W)/\tau} \quad (9)$$

Case (V): When the neuron doesn't fire with nth input pulse and also $t_{i,n+1} < t_{i,n} + W$;
A part of the next input pulse overlaps to the current one. Therefore,

$$y_i(t_{i,n+1}) = y_i(t_{i,n}) + s_{ij}(t_{i,n}) \cdot \frac{t_{i,n+1} - t_{i,n}}{W} \quad (10)$$

3 Simulation Results for One Neuron

Because of the paper length limitation, only some characteristics of a single neuron are shown in this paper. Detailed results for one neuron and those from a network will be shown elsewhere [14].

3.1 Bifurcation diagram of the internal state

A bifurcation diagram of the amplitude of the internal state when a periodic input pulse is applied to the neuron is shown in Fig. 3. The period of the pulse, T,

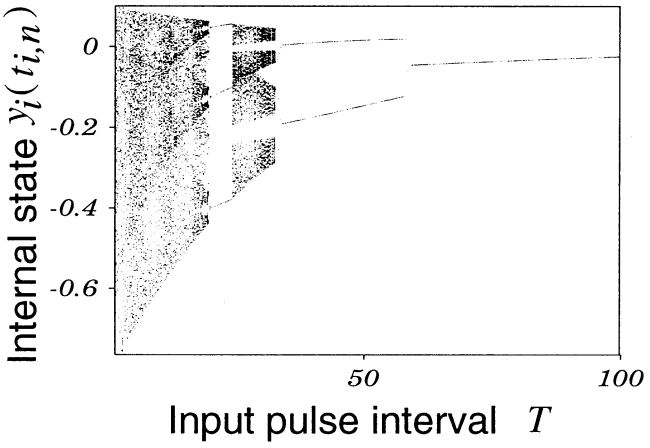


Figure 3: A bifurcation diagram of the amplitude of the internal state of the model. $s_i(t_n)$=0.2, W=1, τ=40, α=1, D_ζ=1, $\varepsilon = 0.05$, $\theta = 0.1$ and $\Theta = 0.2$.

is a bifurcation parameter. Orbit instabilities of some responses in the diagram are confirmed by numerical simulations. That is, the amplitude of the internal states in the model may exhibit chaotic behavior.

3.2 Bifurcation diagram of the output pulse interval

A bifurcation diagram of the time intervals of two successive output pulses, $x_i(t_{i,\theta n} + D_x)$ and $x_i(t_{i,\theta(n+1)}+D_x)$, is plotted in Fig. 4. The input of the neuron is a periodic pulse train with period T which is a bifurcation parameter. Some of these characteristics have orbit instabilities. Therefore, the intervals of the output pulses may be chaotic.

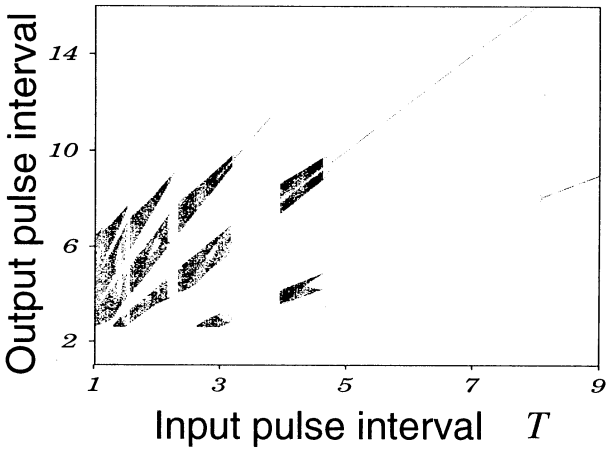


Figure 4: A bifurcation diagram of the time-intervals of two successive output pulses. $s_i(t_n)$=0.2, W=1, τ=40, α=1, D_ζ=1, $\varepsilon = 0.05$, $\theta = 0.1$ and $\Theta = 0.1$.

4 Circuit Implementation of the Model

4.1 Neuron circuit

Figure 5 shows a neuron circuit which implements a core part of the proposed model. The circuit consists of a synapse circuit (M1–M6; only one synapse is shown), a reset pulse generator (M7–M10), a Soma circuit (C1, M11–M14), a comparator (M15–M25), and an output pulse generator (C2, M26–M32). Detailed descriptions of the circuit are given in Ref. [13].

The output function of the circuit is a threshold function implemented by a comparator for simplicity. Furthermore, the circuit in the figure does not include the delay for the output pulse and absolute refractory period, that is, $D_x = D_\zeta = 0$. Circuits for the sigmoidal output function given by eqs. (1) and (2), the absolute refractoriness and the output delay will be shown at the conference presentation.

4.2 Axon Circuit

In the proposed model, the axon has a propagation delay, d_{ij} or e_{ij}. This delay includes the synaptic and dendritic delays. Figure 6 shows an axon circuit which realizes the delay. This circuit is derived from the circuit proposed in Ref. [15]. The delay time can be controlled by Vb1. Moreover, the pulse width can be determined by Vb2. Details of the circuit are described in Ref. [13].

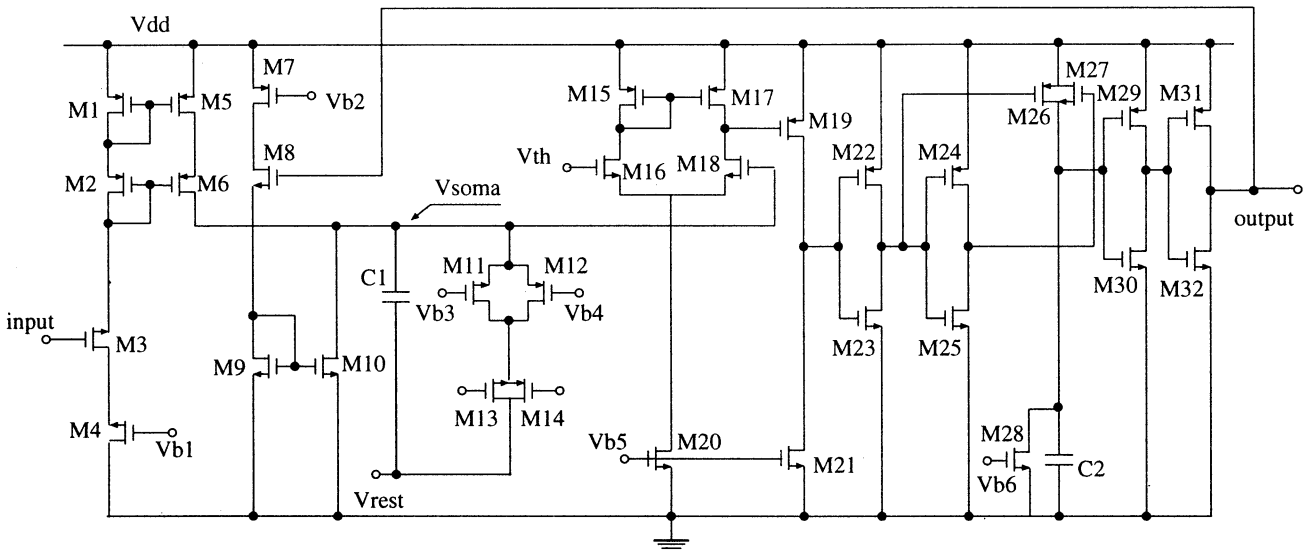


Figure 5: A circuit for the neuron part.

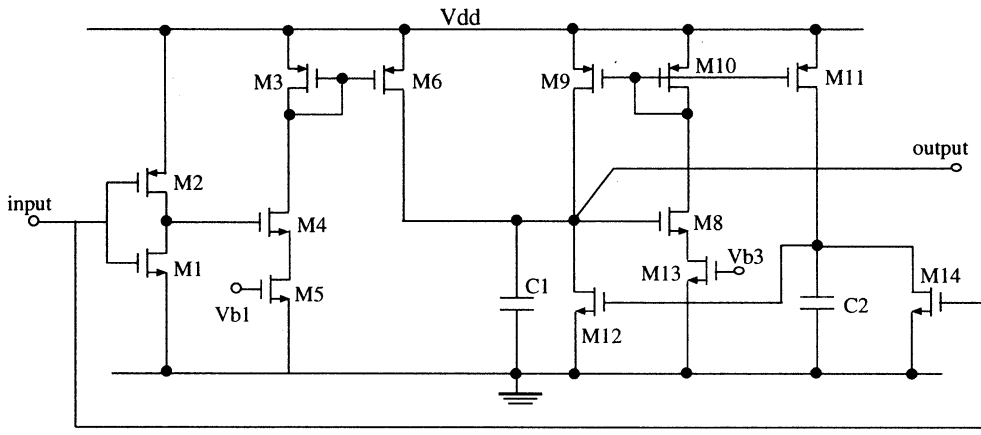


Figure 6: The axon circuit.

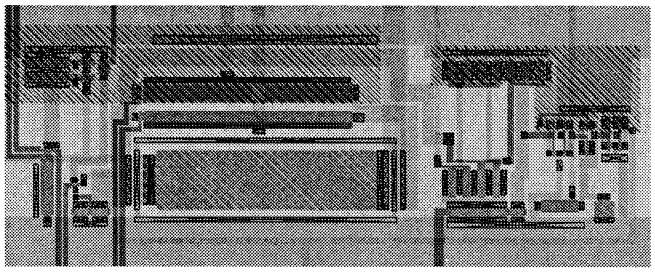


Figure 7: Layout of the neuron circuit.

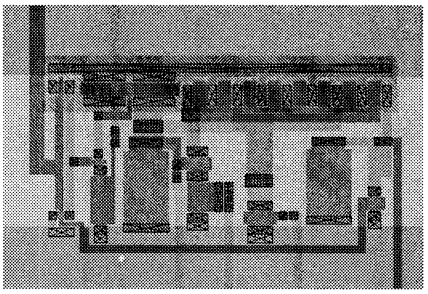


Figure 8: Layout of the axon circuit.

The above circuits can be integrated in standard CMOS process. A prototype chip is now under fabrication using HP 1.2 μm CMOS process through MOSIS [13]. The layouts of the prototype circuits are shown in Figs. 7 and 8. Results from the chip will be shown at the conference.

5 Conclusions

The asynchronous pulse neural network model which is suitable for implementation of a spatio-temporal coding network with an analog circuitry has been proposed. The coincidence detection of the inputs, continuous pulse intervals, relative and absolute refractoriness, an analog internal state value, a continuous output function, an output pulse generation delay, and synaptic delay and weight, are taken into consideration in the model. Simulation results have shown that the internal state as well as the intervals of the output pulses of the model exhibit complex behavior. Furthermore, analog circuits which implement the core part of the model have been presented. The circuits are currently under fabrication as an IC form.

As a future problem, the dynamics of a large scale network will be investigated using analog integrated circuit neurons and synapses.

Acknowledgment

The authors thank K. Aihara and N. Ichinose from Tokyo University for their valuable discussions. This work was supported in part by grants in-aid for scientific research from Ministry of Education, Science and Culture of Japan (09650081, 09044183), Research for The Future Program JSPS-RFTF (96100104), and CREST (Core Research for Evolutional Science and Technology) of Japan Science and Technology Co. (JST).

References

[1] H. Fujii, H. Ito, K. Aihara, N. Ichinose and M. Tsukada, 'Dynamical cell assembly hypothesis – Theoretical possibility of spatio-temporal coding in the cortex,' INNS Neural Networks, vol. 9, no. 8, pp. 1303-1350, 1996.

[2] W.R. Softky, 'Simple codes vs. efficient codes,' Current Opinion in Neurobiology, vol. 5, pp. 239–247, 1995.

[3] C. Koch, 'Computation and the single neuron,' Nature, vol. 385, pp. 207–210, 1997.

[4] T.J. Sejnowski, 'The year of the dendrite,' Science, vol. 275, pp. 178–179, 1997.

[5] M. Tsukada, T. Aihara, H. Saito and H. Kato, 'Hippocampal LTP depends on spatio and temporal correlation of input,' INNS Neural Networks, vol. 9, no. 8, pp. 1357-1365, 1996.

[6] C. Van der Malsburg, 'The correlation theory of brain function,' Internal Report of Max-Planck-Ins. of biophysical Chemistry, vol. 81–2, 1981.

[7] K. Judd and K. Aihara, 'Pulse propagation networks: A neural network model that uses temporal coding by action potentials,' INNS Neural Networks, vol. 6, no. 2, pp. 203-215, 1993.

[8] N. Ichinose and K. Aihara, 'An analysis on propagation dynamics in asynchronous chaotic neural networks,' Trans. of IEICE, vol. J78-A, no. 3, pp. 373-380, 1995(in Japanese); Electronics and Communications in Japan, part 3, vol. 78, no. 11, pp. 52-59, Scripta Technica Inc., New York, John Wiley (English translation).

[9] J.L. McCauley,'Chaos, Dynamics and Fractals,' Cambridge Univ. Press, 1993.

[10] M. Hanagata and Y. Horio, 'A modified asynchronous chaotic neural network model for VLSI implementation,' in Proc. IEEE ISCAS, vol. 1, pp. 657-660, 1997.

[11] M. Hanagata and Y. Horio, 'An asynchronous pulse neural network model with finite pulse width for VLSI implementation,' in Proc. ICONIP'97 (in printing), 1997.

[12] M. Hanagata and Y. Horio, 'A modified asynchronous pulse neural network model for VLSI implementation,' in Proc. NOLTA'97 (in printing), 1997.

[13] K. Yasuda, M. Hanagata, R. Kasahara, and Y. Horio, 'Analog circuit implementation of asynchronous pulse neural network model,' in Proc. NOLTA'97 (in printing), 1997.

[14] M. Hanagata and Y. Horio, 'An asynchronous pulse neural network model for analog VLSI implementation,' to be submitted.

[15] R. Sarpenshkar, L. Watts and C. Mead, 'Refractory neuron circuits,' CNS Tech. Rep., no. CNS-TR-92-08, Calif. Inst. Technol., Pasadena, 1992.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Local and Global Modeling for Complex Time Series

T. Ikeguchi
†Department of Applied Electronics
Science University of Tokyo
Chiba 278, Japan

K. Aihara
†† Department of Mathematical Engineering and
Information Physics
The University of Tokyo
Tokyo 113, Japan

Abstract

We analyze complex time series by Local versus Global (LVG) Plots on Lyapunov exponents. The analyzed time series are not only deterministic chaos, which are produced from the conventional low dimensional nonlinear deterministic systems, but also so called "complex chaos," which are generated from the generalized shift maps. Correct values of Lyapunov exponents are estimated only with local modeling, and spurious values appear with global modeling for the conventional chaos. For generalized shift maps, it is indicated that the assumption of modeling by local linear approximation does not hold, spurious values are estimated.

1 Introduction

Identifying deterministic chaos and its quantitative characterization are very important from the viewpoint of time series analysis on nonlinear dynamical systems theory. For quantitative characterization of deterministic chaos, there are several statistics, for example, the fractal dimensions, the Lyapunov exponents, the metric entropies and so on.

Among them, Lyapunov characteristic exponent is one of the important quantities, which evaluates orbital instability. In the case of non-periodic behavior produced from deterministic nonlinear dynamical systems, this quantity measures not only the orbital instability but also can give a good estimate of an upper limit of Kolmogorov-Sinai entropy.

In order to estimate Lyapunov exponents from observed time series, several algorithms have been proposed. For example, the sophisticated approach, which estimates local dynamics of original dynamical systems, has already been proposed: this approach is usually called the Jacobian estimation method [10, 3].

In any case, it is very important to estimate the Lyapunov spectra, or at least the largest Lyapunov exponents, since it quantifies how chaotic the dynamical system is, which facts means that Lyapunov exponents give important information to modelers of complex time series. Moreover, the sum of the positive Lyapunov exponents gives a good estimates on Kolmogorov-Sinai entropy, that gives the critical time for prediction. Namely, estimating algorithms described above play very important roles on analyzing and modeling complex observational time series data for investigating possibilities of chaotic dynamics.

It has already been reported that even if the observable is only a single-variable time series, the Lyapunov exponents and its spectrum of original dynamical systems can be estimated with high accuracy in the case of observing enough number of data points. However, it was also claimed that if Lyapunov spectral analysis is blindly applied to random data, under an assumption on existence of a deterministic system, it has been suggested that spurious positive Lyapunov exponents would appear [6]. In order to avoid this situation, we have already proposed the Local versus Global (LVG) plots on Lyapunov spectra [6, 5], which evaluates the variation of Lyapunov exponents with smooth changes of modeling strategy from local to global one, and can estimate correct values of Lyapunov exponents, possibly not only the largest one but also all exponents. In this paper, we apply LVG plots both on conventional chaotic dynamics and complex chaos, and we will show that LVG plots can be a good tool from the view point of modeling complex time series, if the assumption on modeling fits for the time series.

2 Local versus Global Plots

If the analyzed time series are observed from nonlinear dynamical systems, it has been shown that the above algorithms [10, 3] work well. However, if the

number of data points are not so enough, or the time series are heavily contaminated by noise, it is also reported to be harder to obtain reliable estimates on Lyapunov exponents than fractal dimensions. Moreover, it is indicated by the present authors that a blind application of the algorithm on random data would lead to spurious estimates of positive largest Lyapunov exponents [6, 5]. In the case of analyzing real time series data, it is unavoidable that the observed time series is corrupted by noise, then it is very important to consider more careful adoption.

One of the essential source for chaotic behavior is nonlinearity, and nonlinearity can often be approximated by the local linear modeling. This is why the local modelings have been adopted not only for estimating Lyapunov exponents [10, 3] but also for prediction [4]. That is, if the observed time series is realization from nonlinear dynamics, the local modeling could be a good one for the observed time series.

Then, if we change the modeling strategy from local modeling to global modeling smoothly, which could reveal essential aspects of complicated dynamics of the analyzed time series; it is expected that in the case of local modeling the estimated Lyapunov exponents would be correct, while in the case of global modeling the estimated values would be estimated as far different values from those of local modeling. Namely, the results of estimating Lyapunov exponents as a function of smooth change on modeling strategy would tell much about correct values of Lyapunov exponents, or at least the existence of positive Lyapunov exponents of underlying attractors.

Here, we should refer that Casdagli has already discussed the similar way to evaluate how the dynamics of the time series should be described by deterministic versus stochastic modeling [2] from the view points of nonlinear prediction errors.

3 Numerical Examples

3.1 Low-dimensional Chaos

In this section, we analyze low dimensional nonlinear dynamical system by LVG plots. The example is the Hénon map, which is described as follows:

$$\begin{cases} x(t+1) &= 1 - ax(t)^2 + y(t) \\ y(t+1) &= bx(t) \end{cases} \qquad (1)$$

where $a = 1.4$ and $b = 0.3$.

In order to avoid the nuisance problems on reconstructing attractor from the observed single variable time series, we just apply our algorithm to the trajectory with full state spaces in this section.

In Figure 1, we show the results for the Hénon maps with noise. It is clearly seen that the estimated values of the largest Lyapunov exponents show their correct values, ~ 0.418. The correct value is confirmed from computer simulation with the above parameter values.

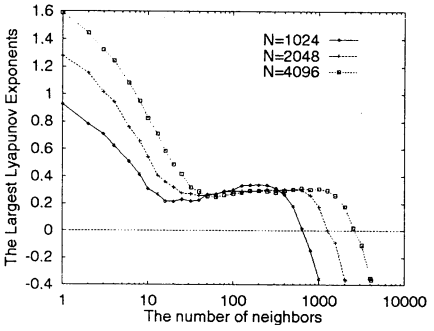


Figure 1: Local versus Global Plots on the Hénon maps with 20[dB] observation noise.

3.2 Complex Chaos

Let us take another examples from nonlinear dynamical systems which posses more complicated dynamics. In Refs.[9, 8, 12], it was shown that even two-dimensional dynamical systems could have very complicated dynamics. The example is the generalize shift map [9, 8, 12]. It is shown that a kind of shift-like dynamical systems can have more complex dynamics than usual low-dimensional nonlinear dynamical systems. In Refs.[9, 8], several examples are examined and in Ref.[12], 121 examples are analyzed. As examples, we take the Moore's original map [9] and No.16 map from Yoshikawa's thesis [12]. While the Moore's map has very complicated behavior, the Yoshikawa's No.16 map behaves the same as conventional chaos. In Table 1, we show the function Φ of the original Moore's map, and that of Yoshikawa's No.16 map.

The upper bits and the lower bits are transformed to the real numbers on $[0, 1]$, then the analyzed time series are two dimensional. LVG plots are estimated on these two-dimensional time series. The different definition of Lyapunov exponents

$$\lambda = \frac{\| \sum_{t=0}^{N-1} F(A(t)) \| \log_e 2}{N}, \qquad (2)$$

Table 1: The function Φ of the Moore's map and the Yoshikawa's No.16 map.

Moore			Yoshikawa(No.16)		
$a_{-1}.a_0a_1$	F	G	$a_{-1}.a_0a_1$	F	G
0.00	−1	0.11	0.00	+1	1.11
0.01	+1	1.01	0.01	−1	0.00
0.10	+1	1.11	0.10	+1	1.01
0.11	−1	0.00	0.11	−1	0.11
1.00	+1	0.01	1.00	+1	0.01
1.01	−1	0.10	1.01	−1	0.01
1.10	+1	0.11	1.10	−1	0.10
1.11	−1	0.01	1.11	+1	0.11

where N is the number of iterations, $A(t)$ is (infinite) bit sequence at the time t on which the rule Φ is applied, tells that the Lyapunov exponent of the complex chaos is converged to 0, and that of the conventional chaos (No.16 map) converges to positive value (~ 0.07).

From Figures 2 and 3, the results of estimated Lyapunov exponents are shown. From these figures, the different values are estimated. For the complex chaos, the first exponents is estimated positive, ~ 0.25 and for the conventional chaos, ~ 0.5.

In the case of applying LVG plots on unknown time series, we assume that the observed time series are produced from low dimensional nonlinear dynamical systems and can be modeled with the local linear approximation method. For the conventional map, such as the Hènon maps, the above assumption is good, however for generalized shift maps, it is not. Namely, it is very important to consider the issue that *can we really assume that the applied model is good for the target time series.* When the assumption is broken, even if the method is moderately and carefully applied, the results become harm and give no information for modeling complex time series.

4 Experimental Time Series

4.1 Squid Axon Response

It has been experimentally shown that when squid giant axons are forced by periodic pulses, timings of firing in the nerve membranes are irregular, or chaotic, under some conditions. We have already analyzed the real data of squid axon response stimulated by periodic pulses with deterministic nonlinear prediction,

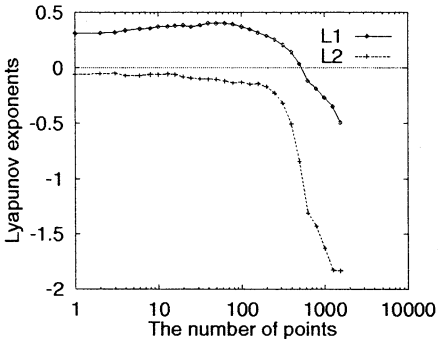


Figure 2: Local versus Global Plots on the Generalized shift. The function Φ is shown in Table 1[9]. L1 indicates the first Lyapunov exponents and L2 the seconds.

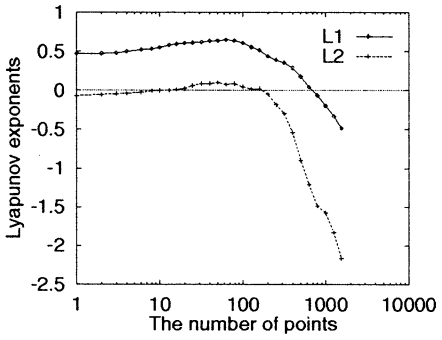


Figure 3: Local versus Global Plots on the Generalized shift. The function Φ is taken from the table No.16 (Table 1) [12]. L1 indicates the first Lyapunov exponents and L2 the seconds.

and shown existence of chaotic dynamics behind a complex response of squid axon data quantitatively [7, 1].

In this section, we revisit squid axon response that is stimulated by sinusoidal external force with LVG plots. In Figure 4, only the largest Lyapunov exponents are plotted. In lower dimensional reconstructed state space, for example $m = 2, 3$, the variation of exponents are similar to that of random variables [6]. However in higher than four dimensions, we can find the plateau regions in the case of small number of near neighbors, and the estimated values are converged to about 0.2. The convergence at $m = 4$ is in good agreement with our previous results by nonlinear prediction [1] in combination of the method of surrogate data.

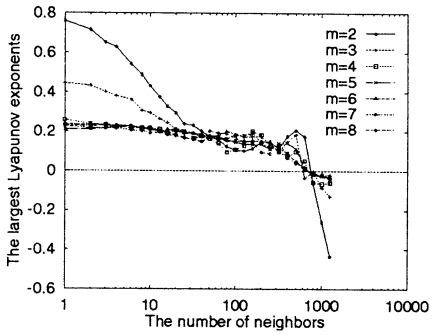


Figure 4: LVG plots on the squid axon data. m denotes the reconstructed dimension.

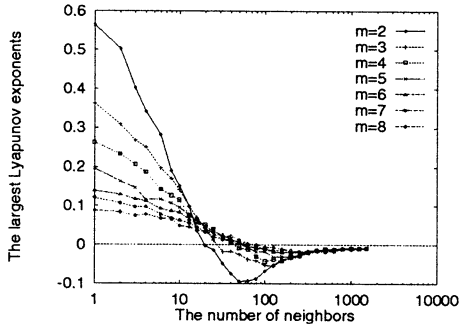


Figure 5: LVG plots on the financial time series. m denotes the reconstructed dimension.

4.2 Exchange rate between the Japanese Yen to the U.S. Dollar

It has also been claimed that the unpredictable behavior of financial time series is due to deterministic nonlinear dynamical systems. Although many authors report on deterministic nonlinear aspects on these financial time series, there is still no clear explanation on the origin of these data. We have also analyzed several financial time series from the view points of nonlinear dynamical systems theory[11]. However, we could not find any clear evidence that the time series are produced from deterministic dynamical systems. In this section, we analyze the exchange rate between the Japanese Yen to the U.S. dollars.

In Figure 5, we show the results on the exchange rate between the Japanese Yen to the U.S. dollar with LVG plots. We cannot find any clear plateau regions from the results, and this tendency is very similar to the result that the time series is stochastic case.

5 Conclusion

We analyze complex time series by the local versus global plots. LVG plots examines variation of Lyapunov exponents with changes of the modeling strategy and is useful for analyzing real time series data.

We apply the proposed algorithm to numerical examples from mathematical models of nonlinear dynamical systems. In the case of nonlinear dynamical systems, even if we can only observe the time series corrupted by noise, it is possible to find the convergence of the Lyapunov exponents, although the values of estimated Lyapunov exponents are underestimated. In the case of complex chaos, since the assumption of the model, the existence of local linear dynamics, does not match the time series, different values are estimated by LVG plots. This result implies very important future direction of the present issue of modeling complex time series, since it is almost impossible to know the information of the time series.

We also apply the LVG plots to real time series. For the squid axon data, convergence of the plateau region in LVG plots means the existence of the positive Lyapunov exponents. However, for the financial time series, we cannot find any evidence of its existence.

References

[1] K. Aihara, T. Ikeguchi, and G. Matsumoto. Deterministic nonlinear dynamics of a forced oscillation experimentally observed with a squid gian axon, submitted to *International Journal of Bifurcation and Chaos*, 1997.

[2] M. Casdagli. *J. Royal Stat. Soc. B*, 54(2):303–328, 1992.

[3] J. P. Eckmann, S. O. Kamphorst, D. Ruelle, and S. Ciliberto. *Phys. Rev. A*, 34(6):4971–4979, 1986.

[4] J. D. Farmer and J. J. Sidorowich. *Phys. Rev. Lett.*, 59(8):845–848, 1987.

[5] T. Ikeguchi and K. Aihara. Tech. Rep. of IEICE, 96(477):27–104, 1996.

[6] T. Ikeguchi and K. Aihara. *Int. J. Bifurcation and Chaos*, 7(6), 1997.

[7] A. Mees, K. Aihara, M. Adachi, K. Judd, T. Ikeguchi, and G. Matsumoto. *Phys. Lett. A*, 169(1&2):41–45, 1992.

[8] C. Moore. *Nonlinearity*, 4:199–230, 1991.

[9] C. Moore. *Phys. Rev. Lett.*, 64(20):2354–2357, 1997.

[10] M. Sano and Y. Sawada. *Phys. Rev. Lett.*, 55(10):1082–1085, 1985.

[11] T. Terasaki, T. Ikeguchi, K. Aihara, and S. Tanaka. *Trans. of IEICE*, J78-A(12):1601–1617, 1995.

[12] T. Yoshikawa. An analysis of nonlinear dynamics of generalized shifts, Master Thesis, Dept. of Electrical Engineering, Science Univ. of Tokyo, 1993.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Mixing Invariant Densities

Gary Froyland*

Department of Mathematical Engineering
The University of Tokyo, 7-3-1 Hongo
Bunkyo-ku, Tokyo 113, JAPAN
Email: gary@sat.t.u-tokyo.ac.jp
URL: www.sat.t.u-tokyo.ac.jp/~gary

Abstract— For many chaotic systems there appears to be a single asymptotic distribution describing the frequency with which almost all trajectories visit different regions of state space. Such a distribution is commonly called the "physical" invariant measure or "natural" invariant measure of the system. Consider the evolution of a cluster of initial conditions under the action of the system. This cluster will quickly spread out and begin to distribute itself according to the asymptotic distribution of the system. When the cluster reaches the asymptotic distribution, we say that the cluster of initial conditions has attained the *equilibrium state* of the system. In this paper we are concerned with the rate at which initial distributions approach equilibrium. Often this rate is exponential, and we describe a new numerical method for rigorously bounding (and sometimes estimating) the rate. Our method provides a considerable improvement over current theoretical bounds.

I. Introduction

We begin by illustrating the salient objects with a short example. Suppose that we are operating a machine which produces automobile paint. Such paints often consist of a mixture of pigment particles and a binding solution to hold the pigment particles together. For optimal appearance of the paint coating, it is essential that the pigment particles be relatively evenly distributed throughout the binding solution. To manufacture the paint, the pigment particles will initially be injected into the binding solution and there will be machines to mix together the two constituents of the paint. We are interested in the efficiency of such machines and would like them to mix the pigment particles around as quickly as possible to save time, energy, and machine wear and tear.

In dynamical systems terms, let the drum of paint (pigment and binder) be our state space, and the action of one revolution of our mixing machine represent one iteration of our discrete model $T : M \circlearrowleft$. When the contents of the pigment are dumped into the drum, we are given an initial configuration of the system, described by the density of pigment particles in the paint. As we iterate our system (the mixing machine operates), the pigment particles are mixed around, hopefully approaching an even density (equilibrium) within the paint. In this paper, we present a rigorous numerical method for determining a lower bound for the rate at which initial densities approach this equilibrium state of the system.

Theoretical bounds for smooth initial distributions have been produced by Rychlik [1] and Liverani [2]. Their estimates are, however, extremely conservative; for example, the map $x \mapsto 2x \pmod 1$, $x \in S^1$, has $1/2$ as its rate of approach to equilibrium, while the bounds of [1] and [2] are 0.9986 and 0.5 respectively. In this case the bound of [2] is optimal, but these bounds quickly worsen as nonlinearities are added to the system. We compare our bounds with these theoretical bounds in a later section.

II. Formalities and Main Results

Let M be a compact Riemannian manifold and $\varphi : M \to \mathbb{R}^+$ satisfy $\int_M \varphi(x)\, dm(x) = 1$, where m is normalized volume measure on M. The function φ is known as a *density function* (or simply a *density*), and represents a distribution of mass on M. We denote the space of all density functions by $\mathcal{D}$. Distributions of mass on M which may not be represented as density functions (for example singular distributions with respect to Lebesgue measure) shall be simply called *distributions.*

As we are after rigorous results, we must restrict ourselves to situations where it is known that (i) a unique invariant distribution exists, and (ii) the rate of approach to equilibrium is at some exponential rate strictly between 0 and 1. Only in such situations does it make sense for us try to bound or estimate the exponential rate of approach to equilibrium. We therefore assume that T is piecewise $C^{1+\gamma}$, and fur-

*This work was completed at the Centre for Applied Dynamics and Optimization, The University of Western Australia, with partial support by a grant from the Australian Research Council.

ther that T is either (i) expanding, (ii) Anosov, or (iii) Axiom A (attractor or non-attractor). In case (ii), the invariant distribution is sometimes not a density and in case (iii) the invariant distribution is always not. For ease of presentation, we will deal with case (i) and those situations in case (ii) where one does have an invariant density. We mention the other cases only to indicate the generality of our results.

As the map T is iterated, this distribution of mass is pushed around under the action of T. The operator which pushes densities around is the *Perron-Frobenius* operator $\mathcal{P} : \mathcal{D} \circlearrowleft$, defined by

$$(\mathcal{P}\varphi)(x) = \sum_{y \in T^{-1}x} \frac{\varphi(y)}{|\det D_y T|}.$$

In the sequel, we will consider $\mathcal{P}$ to act on the space $\mathcal{D}^\perp = \{\varphi \in C^\gamma(M, \mathbb{C}) : \int_M \varphi \cdot \varphi^* \, dm = 0\}$. We will be calculating the spectrum of $\mathcal{P}$, and so must include complex-valued eigenfunctions to obtain the fullest spectrum. However, to obtain sensible answers, we must restrict ourselves to functions which are at least piecewise Hölder. If we do not restrict ourselves to functions with a certain degree of smoothness, we will not obtain a value for r between 0 and 1. For example, if $M = \mathbb{T}^2$ the 2-torus, and $T : \mathbb{T}^2 \circlearrowleft$ is defined by $T(x_1, x_2) = (2x_1 + x_2, x_1 + x_2)$ (mod 1), then Crawford and Cary [3] have shown that the rate of approach of a C^0 function to the invariant density $\varphi^* \equiv 1$ is at an algebraic (and not geometric) rate. Lastly, we know that $\mathcal{P}$ has 1 as an eigenvalue of unit multiplicity, with corresponding eigenfunction φ^*. We do not wish to consider this eigenfunction, and so we simply choose $\mathcal{P}$ to act on those functions orthogonal to the one-dimensional subspace spanned by φ^*.

Because all of the functions in $\mathcal{D}^\perp$ have no φ^* component, they should all decay to the zero function at an exponential rate under the action of $\mathcal{P}$. Since

$$\int_M |\mathcal{P}^k \varphi| \, dm \leq \|\varphi\|_\gamma \|\mathcal{P}^k\|_\gamma,$$

we therefore search for the minimal value of r_0 such that there exists a constant $C < \infty$ satisfying $\|\mathcal{P}^k\|_\gamma \leq C r^k$ for all $r > r_0$. The spectral radius theorem says that r_0 is given by the spectral radius of $\mathcal{P}$ acting on $\mathcal{D}^\perp$, so it is the spectrum of $\mathcal{P}$ (denoted by $\sigma(\mathcal{P})$) that we shall try to approximate.

The spectrum of $\mathcal{P}$ consists of two major parts, the first of which is the *essential spectrum*; see Figure 1. The radius of the essential spectrum, denoted r_{ess}, is defined as the smallest number between 0 and 1 for which elements of $\sigma(\mathcal{P})$ outside the disk $\{z \in \mathbb{C} : |z| \leq r_{\text{ess}}\}$ are isolated eigenvalues of finite multiplicity. Reasonable bounds for r_{ess}

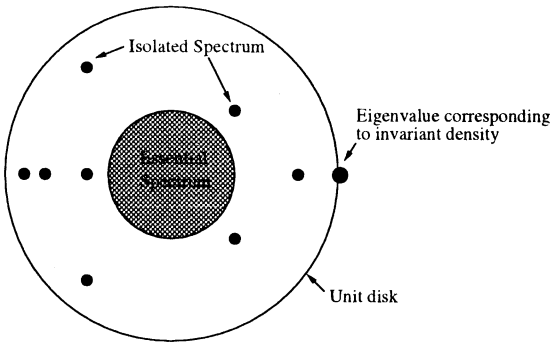


Figure 1: Schematic representation of the spectrum (as a subset of the complex plane) of the Perron-Frobenius operator $\mathcal{P} : \mathcal{D}^\perp \oplus \{\varphi^*\} \circlearrowleft$.

have been obtained by Ruelle [4]: $r_{\text{ess}} \leq \lambda^\gamma$, where $\lambda = 1/(\inf_{x \in M} \inf_{v \in T_x M} \|D_x T(v)\|/\|v\|)$. It is the spectral points *outside* the essential spectral radius that cause theoretical estimates of the rate of decay to be so poor, and it is these that we concentrate on computing bounds for.

To be able to estimate the spectrum of the infinite dimensional operator $\mathcal{P}$ on a computer, we require some finite dimensional approximation. We divide the state space into a finite number of connected sets $A_1, \ldots, A_n$ in an effort to discretise the dynamics. We then form the $n \times n$ transition matrix

$$P_{ij} = \frac{m(A_i \cap T^{-1} A_j)}{m(A_i)}, \tag{1}$$

where m is normalised volume measure on M, to describe the evolution from one discrete region to another. One may think of the entry P_{ij} as representing the probability of a point in the region A_i moving into the region A_j in one step. Ulam [5] originally proposed this matrix as a finite-dimensional approximation to the Perron-Frobenius operator for expanding interval maps. In [6] the author proved that provided that the regions $A_1, \ldots, A_n$ are carefully chosen, the matrix P is a very good approximation of the Perron-Frobenius operator for both multidimensional expanding maps and uniformly hyperbolic maps, in the sense that the unique vector fixed under left multiplication of the stochastic matrix P defines a good approximation of the equilibrium distribution. Here we report some of the results obtained in [7], which show that the matrix P also provides us with a bound on the rate of approach to equilibrium.

Theorem 1 (Main Result): *Let $T : M \circlearrowleft$ be a $C^{1+\gamma}$, $(0 < \gamma \leq 1)$, expanding (resp. Anosov) map of a compact d-dimensional (resp. 2-dimensional) Riemannian manifold M. Denote by $\{\mathfrak{P}_n\}_{n=n_0}^\infty$ a sequence of Markov partitions for T on M, with the property that $\max_{A \in \mathfrak{P}_n} \operatorname{diam} A \to 0$ as $n \to \infty$.*

Construct the matrix

$$P_{n,ij} = \frac{m(A_{n,i} \cap T^{-1}A_{n,j})}{m(A_{n,j})},$$

$A_{n,i}, A_{n,j} \in \mathfrak{P}_n$, $1 \le i,j \le \mathrm{card}\,\mathfrak{P}_n$. *Denote by* $\sigma'(P_n)$ *the spectral values of* P_n *that lie in the region* $|z| > \lambda^{\gamma'}$, *and by* $\sigma(\mathcal{P})$ *the spectrum of* $\mathcal{P}$: $C^{\gamma'}(M,\mathbb{C}) \circlearrowleft$, $(0 < \gamma' < \gamma)$. *Then* $\sigma'(P_n)$ *converges to a set containing* $\sigma(\mathcal{P}) \setminus \{|z| \le \lambda^{\gamma'}\}$.

In particular, an upper bound for the rate of approach to the equilibrium density φ^* *for* T *is:*

Expanding T*:*

$$r_0 \le \max\left\{\lambda^{\gamma'}, \lim_{n\to\infty} \max_{z\in\sigma(P_n)\setminus\{1\}} |z|\right\}.$$

Anosov T*:*

$$r_0 \le \left(\max\left\{\lambda^{\gamma'}, \lim_{n\to\infty} \max_{z\in\sigma(P_n)\setminus\{1\}} |z|\right\}\right)^{1/3}.$$

III. An Example

For ease of presentation, we illustrate our technique for a one-dimensional system. We choose the following piecewise quartic (piecewise C^2, so $\gamma = 1$) interval map defined by[1]

$$Tx = \begin{cases} x^4 - 0.81x^3 + 0.14x^2 + 1.78x + 0.11, \\ \quad \text{for } 0.12 \le x \le 1/2; \\ 1.16x^4 - 3.59x^3 + 4.08x^2 - 3.78x + 2.25, \\ \quad \text{for } 1/2 \le x \le 1. \end{cases} \tag{2}$$

This map is a slight nonlinear perturbation of a piecewise linear Markov map for which it is known that the rate of approach to equilibrium is governed by isolated spectral points. The graph of

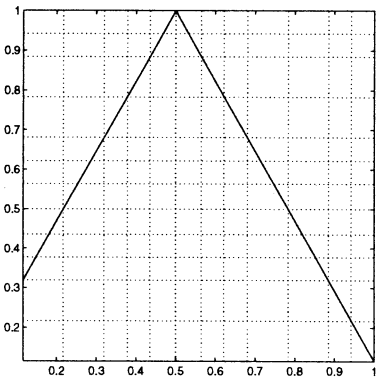


Figure 2: Graph of T as defined in (2)

the invariant density φ^* of T (or the equilibrium distribution) is shown in Figure 3. A bound for the essential spectral radius is given by $\lambda = 1/\left(\min_{x\in[0.1154,1]} |T'(x)|\right) \approx 0.5706$. If there are no isolated spectral points outside the disk $\{|z| \le 0.5706\}$, then we will take 0.5706 as a bound for the rate of approach to equilibrium. If however, we find isolated spectral points (approximated as eigenvalues of the matrices P_n) outside this disk, we take the largest modulus of such points (not counting the eigenvalue 1) as an *estimate* of the rate.

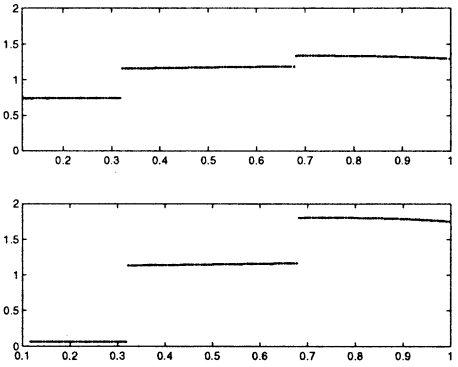


Figure 3: Graph of the equilibrium distribution φ^* of T (top); Graph of the density which approaches the equilibrium distribution φ^* at a rate determined by the isolated eigenvalues $-0.5014 \pm 0.3261i$ (bottom).

We now begin our construction of the matrices P_n. An initial Markov partition of $\mathfrak{P} = \{[0.1154, 0.3194], [0.3194, 0.5], [0.5, 0.6806], [0.6806, 1]\}$ is formed. Refined Markov partitions are constructed by simply taking the join of $\mathfrak{P}$ with its inverse images, that is, define $\mathfrak{P}^{(N)} = \bigvee_{i=0}^{N-1} T^{-i}\mathfrak{P}$, where $\mathfrak{A} \vee \mathfrak{B} := \{A \cap B : A \in \mathfrak{A}, B \in \mathfrak{B}\}$. The 12 set partition $\mathfrak{P}_{12}^{(2)}$ is illustrated in Figure 2. Figure 4 shows the position of the eigenvalues of P_{212} (corresponding to $N = 7$) relative to the region bounding the essential spectrum. From these results, we can confidently state that $\mathcal{P}$ *does* have a non-trivial isolated spectrum, and that therefore, it is this isolated spectrum that will control the rate of approach to equilibrium. Theorem 1 says that a *bound* for the rate of decay is $\max\left\{\lambda^{\gamma'}, \lim_{n\to\infty} \max_{z\in\sigma(P_n)\setminus\{1\}} |z|\right\}$, for any $\gamma' < 1$. We can do better than this, though, as the eigenvectors of the eigenvalues $-0.5014 \pm 0.3261i$ lie in $\mathcal{D}^{\perp}$. Thus there do exist densities which approach equilibrium at this slower rate (see Figure 3), and $0.5982 = |-0.5014 \pm 0.3261i|$ is an *estimate* of this rate. If it so happened that all of the non-unit eigenvalues of P_{212} were contained in $\{|z| \le 0.5706^{\gamma}\}$, we would simply take $0.5706^{\gamma'}$ as our upper bound

[1] All numerical values in this section are approximate only.

Table 1: Our computed-assisted bound for the rate of approach to equilibrium compared to known theoretical bounds

Map	Ruelle bound (essential spectrum)	Our bound	Liverani bound	Rychlik bound
$\tilde{T}x = -1.7525\|x - 1/2\| + 1$	0.5652	0.6009	0.6009	0.9986
T as in (2)	0.5706	0.599	0.9162	0.99999915

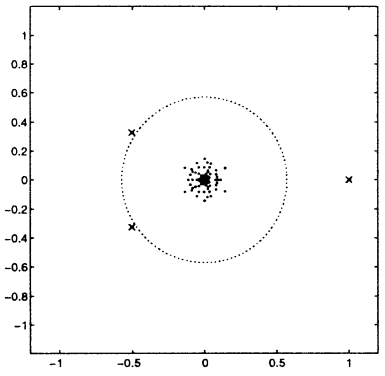


Figure 4: A plot of the spectrum of the 212×212 matrix P_{212}. The dotted circle is the boundary of the disk $\{|z| \leq 0.5706\}$ which contains the essential spectrum. The crosses are the eigenvalues of P_{212} which estimate the isolated spectrum of $\mathcal{P}$.

instead.

A comparison of our computed estimates with known theoretical estimates is made in Table 1. The map $\tilde{T}$ has a graph almost identical to that displayed in Figure 2. The small nonlinearities added to the piecewise linear $\tilde{T}$ have a large effect on the theoretical bounds. While the theoretical bounds always worsen with increasing nonlinearity, our estimate of the rate of approach to equilibrium actually decreases from 0.6009 to around 0.598, so that in this case, the extra nonlinearity appears to *speed up* the mixing rate.

We conclude by acknowledging that the construction of Markov partitions may be difficult for some systems, especially in higher dimensions. An alternative is to use some arbitrary partition of connected sets to construct the matrices (1); for example, a uniform grid, or a triangulation. In this case, rigorous results are not available, but in the author's experience, such constructions tend to give pessimistic estimates for the mixing rate, with the magnitude of the largest non-unit eigenvalue of P_n being larger than the numerically observed rate of approach to equilibrium. For example, we constructed transition matrices for Anosov's Cat Map: $(x_1, x_2) \mapsto (2x_1 + x_2, x_1 + x_2) \pmod 1$, $(x_1, x_2) \in \mathbb{T}^2$, using a triangulation of the 2-torus as a partition. The rate of approach to equilibrium is known to be $2/(\sqrt{5}+3) \approx 0.3820$, while the magnitude of the largest non-unit eigenvalue of an 800×800 transition matrix is 0.5769; see Figure 5. A more detailed

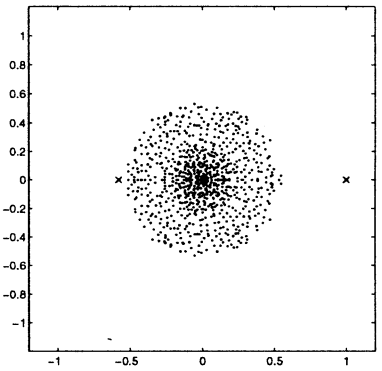


Figure 5: Spectrum of an 800×800 transition matrix for Anosov's Cat Map, generated from a partition of 800 triangles. The two crosses are the two largest eigenvalues in magnitude.

discussion of these results is contained in [8].

References

[1] Marek Rychlik. Regularity of the metric entropy for expanding maps. *Trans. Amer. Math. Soc.*, 315(2):833–847, 1989.

[2] Carlangelo Liverani. Decay of correlations. *Ann. of Math.*, 142(2):239–301, 1995.

[3] John Crawford and John Cary. Decay of correlations in a chaotic measure-preserving transformation. *Phys. D*, 6(2):223–232, 1983.

[4] David Ruelle. The thermodynamic formalism for expanding maps. *Comm. Math. Phys.*, 125:239–262, 1989.

[5] S. M. Ulam. *Problems in Modern Mathematics.* Interscience, 1964.

[6] Gary Froyland. Finite approximation of Sinai-Bowen-Ruelle measures of Anosov systems in two dimensions. *Random Comput. Dynam.*, 3(4):251–264, 1995.

[7] Gary Froyland. Computer-assisted bounds for the rate of decay of correlations. *Comm. Math. Phys.*, 1997. In press.

[8] Gary Froyland. *Estimating Physical Invariant Measures and Space Averages of Dynamical Systems Indicators.* PhD thesis, The University of Western Australia, Perth, 1996. Available at `http://www.sat.t.u-tokyo.ac.jp/~gary/`.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Nonlinear Dynamical Analysis of Japanese Vowels

Takaya Miyano
Sumitomo Metal Industries, Ltd.
1-8 Fusocho, Amagasaki, Hyogo 660, Japan

Akira Nagami
Sumitomo Metal Industries, Ltd.
1-8 Fusocho, Amagasaki, Hyogo 660, Japan

Isao Tokuda
Muroran Institute of Technology
Mizumoto, Muroran, Hokkaido 050, Japan

Kazuyuki Aihara
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

Abstract

Dynamical properties of voice signals of Japanese vowels are characterized with time series analysis associated with the surrogate method in terms of short-term predictability and smoothness of time evolution. Linear autoregression and regularization networks with $(1+\cosh x)^{-1}$ as the basis function are employed as the predictors. Chaotic dynamics is likely to participate in generating voiced sounds.

1 Introduction

In the standard approach to automatic speech recognition and speech synthesis voiced sounds are characterized basically in terms of formant frequencies associated with linear autoregressive (AR) coefficients through autocorrelation function. Nonlinear deterministic ingredients of the sounds, if any that may lurk in the broadband structure of the power spectrum are out of consideration or implicitly presumed to provide no significant feature of speech signals. Recently surprising suggestions have been made that vocal signals of Japanese vowels may include chaotic fluctuations, from time series analysis of the voice signals [1, 2]. Although it is unclear that nonlinear deterministic feature of voiced sounds, if any could play an important role in speech recognition and speech systhesis, it is an intriguing issue in itself whether or not a chaotic attractor exists in voiced sounds.

In this paper we extend the previous works, aiming at settling the issue of possible existence of a chaotic attractor in Japanese vowels. We take inverse approach with time series analysis of voice signals of Japanese vowels and their surrogate data synthesized from the voice data to preserve statistical properties such as the power spectrum and the amplitude distribution [3, 4]. In a preliminary work to the present work, nonlinear forecasting with a Sugihara-May predictor [5] was applied to the vocal time series. The decay of predictability with the prediction time was found, although the scaling property of the prediction error as a function of the prediction time did not provide clear indication of low-dimensional chaos. The target issue of the present work is thus to clarify whether irregular ingredients included in the vocal time series represent deterministic chaos or correlated ramdom noise. To test for determinism, we apply diagnostic algorithm based on short-term predictability [5] and on smoothness in time evolution [6, 7]. Difference in dynamical properties inferred between the original and the surrogate data is supposed to indicate that there should be nonlinear determinism that can not be captured by linear statistical analysis such as power spectral estimation. Nonlinear aspects are also examined in terms of the difference in the predictive performance between a linear AR and a nonlinear predictor. In the present work generalized regularization networks [8] with a particular class of bell-shaped basis function $(1+\cosh x)^{-1}$ are employed as the nonlinear predictor. The basis function has a virtue that the exact shape of the input-output curve can be generated by a simple circuit system consisting of bipolar transistors [9], which would be effective when hardware-implimenting regularization networks is required.

2 Theoretical Preliminaries

A first diagnostic algorithm is based on nolinear forecasting [5]. Given a time series data $\{x(t)\}_{t=1}^{N}$, we first generate phase space with lagged sequences of the data points as $\boldsymbol{x}(t) = (x(t), x(t-\Delta t), \cdots, x(t-(D-1)\Delta t))$, where Δt is an appropriate time lag and D is the embedding dimension. Generalized regularization net-

works used in this work as a predictor are of the form:

$$x(t+\tau\Delta t) = \sum_{h=1}^{N_h} w_h[1+\cosh(-\beta_h \parallel \boldsymbol{x}(t)-\boldsymbol{x}_h \parallel)]^{-1} + \boldsymbol{a}\cdot\boldsymbol{x}(t)+b+\epsilon(\tau)\ . \quad (1)$$

where w_h, β_h, and $\boldsymbol{x}_h$ are network parameters, $\boldsymbol{a}$ and b are linear AR coefficients optimized by least-mean-squared-error fitting, N_h is the number of the basis functions, and $\epsilon(\tau)$ denotes a random variable representing the prediction error. The network parameters are determined with learning rule based on stochastic gradient descent [10] so as to minimize the residual of the linear AR predictor. When making linear forecasts, w_h are set to zero. Degrees of visible determinism are measured in terms of the prediction error given in the present work as the root-mean-squared error between predicted and the corresponding actual values normalized by the standard deviation of the original time series, denoted as $E(\tau)$. Suppose that the time series include a significant amount of nonlinear ingredients. Then the regularization network would provide better predictive performance than the AR predictor.

A second diagnostic algorithm is based on the method developed by Wayland et al. [7] as a simpler variant of the algorithm introduced by Kaplan and Glass [6]. The central idea of this algorithm is that neighboring trajectories generated in the embedding space should point in similar directions, if determinism is visible in a time series. We first find K nearest neighbors of M randomly chosen vectors $\boldsymbol{x}_0(t_i)$ in the sense of Euclidean distance, denoting as $\boldsymbol{x}_k(t_i)$ $(i=1,\cdots,M; k=0,1,\cdots,K)$. We next generate a time evolution of the vectors as $\boldsymbol{x}_k(t_i+T\Delta t)$ where T denotes an appropriate time interval. Let the temporal change of the vectors be expressed by $\boldsymbol{v}_k(t_i) = \boldsymbol{x}_k(t_i+T\Delta t) - \boldsymbol{x}_k(t_i)$. Then the diversity of the time evolution can be gauged in terms of the translation error E_{trans} defined by

$$E_{trans} = \frac{1}{K+1}\sum_{k=0}^{K}\frac{|\boldsymbol{v}_k(t_i) - <\boldsymbol{v}(t_i)>|^2}{|<\boldsymbol{v}(t_i)>|^2}\ , \quad (2)$$

$$<\boldsymbol{v}(t_i)> = \frac{1}{K+1}\sum_{k=0}^{K}\boldsymbol{v}_k(t_i)\ . \quad (3)$$

To reduce the stochastic error, we estimate the median of E_{trans} for Q sets of M randomly chosen $\boldsymbol{x}_0(t_i)$, then take the average over the Q medians, according to Wayland et al. As determinism becomes visible, E_{trans} tends to zero. In a previous numerical work, E_{trans} was shown to exceed 0.5 for correlated and uncorrelated random noise [11].

3 Time Series Data

The data to be analysed are time series of Japanese vowels /a/, /i/, /u/, /e/, and /o/ observed for two male and two female persons in the standard ATR (Advanced Telecommunications Research Institute International) data base. In this paper we show the analysis only about vowel /a/. The data identification numbers are mausy003 and mmssy003 for male /a/, and fsusy003 and fynsy003 for female /a/.The signals were low-pass filtered with a cut-off frequency of 8 kHz, digitized with a sampling rate of 20 kHz in a 16-bit resolution. For each time series the initial transient part has been removed to make the data stationary. Two categories of surrogate data have been generated to preserve specific statistical properties of the original data: (1) phase randomized surrogate time series generated by inverse Fourier transform with phase randomization to preserve the power spectrum, and (2) Fourier shuffled surrogate time series generated by inverse Fourier transform with phase randomization and invariant rank order to preserve both the power spectrum and the amplitutde distribution. Each original and surrogate time series consist of 2048 data points. For details on the algorithm generating the surrogate data see [3, 4].

4 Time Series Analysis and Discussion

We first apply a diagnostic algorithm of Wayland et al. to the original and the surrogate data under a condition of $K=4$, $T=5$, $\Delta t=10$, $M=300$, $Q=20$. Figures 1–4 show the average median of E_{trans} as a function of D. A considerable difference in E_{trans} between the original and the surrogate data can be recognized in each case. To confirm that the difference does not stem from the algorithmic error in generating the surrogate data, the similar surrogate method is applied to numerical sine wave added to correlated random noise actually observed as the leakage-current across a silicon metal-oxide-semiconductor capacitor [12]. The noise level has been controlled so as to realize similar values of E_{trans} to each original voice data. The synthetic time series simultates voiced sounds contaminated with additional correlated random noise. Estimates are shown in Fig.5, where no significant difference in E_{trans} can be detected.

From these estimates, the null hypothesis can be rejected that the irregular ingredients included in the original time series should represent correlated random noise. To test for nonlinearity in the irregular ingredients, we next apply time series prediction with regularization networks and linear AR predictors to compare

the predictive performance. Figure 6 illustrates the comparison of the prediction error for each original time series. The predictors have been optimized using library patterns consisting of input vectors and the corresponding output values generated from the first half of the original time series to forecast the last half ($D = 12$ and $N_h = 6$ for mausy003, $D = 4$ and $N_h = 4$ for mmssy003, $D = 8$ and $N_h = 6$ for fsusy003, $D = 7$ and $N_h = 8$ for fynsy003). The optimal embedding dimension was determined by the preliminary work based on the Sugihara-May method. The number of the basis functions have been optimized by a trial-and-error method. The nonlinear predictor provides slightly better predictive performance than the linear AR predictor, which may agree with the existence of nonlinear determinism in the original data. The corresponding results on the surrogate data are shown in Figs.7 and 8, where the predictors have been trained for the first half of each original time series, that is, the predictors have learned the dynamics underlying the original voice data. The nonlinear and the linear predictors generate large prediction errors (generalization errors) in each case. No significant difference in the predictive performance between the predictors can be observed for the surrogate data in which nonlinear determinism, if any has been destroyed. These observations seem to suggest the existence of nonlinear determinism in the original data.

5 Summary

The present analysis indicates that a low-dimensional chaotic attractor is likely to exist in voiced sounds of Japanese vowel /a/. Similar results are obtained for time series data of other Japanese vowels. A remaining issue of interest might be the influence of chaotic feature of voiced sounds on speech recognition and speech synthesis, which will be our future work.

Acknowledgment

We would like to thank Associate Professor Tohru Ikeguchi of the Science University of Tokyo for helpful advice.

References

[1] I. Tokuda, R. Tokunaga, and K. Aihara, "A simple geometrical structure underlying speech signals of Japanese vowel /a/", *Int. J. Bifurcation and Chaos*, Vol.6, pp.149–160, 1996.

[2] T. Miyano, "Are Japanese vowels chaotic?", *Proc. 4th Int. Conf. on Soft Computing*, Vol.2, pp.634–637, 1996.

[3] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: the method of surrogate data", *Physica* D, Vol.58, pp.77–94, 1992.

[4] T. Chang, S. J. Schiff, T. Sauer, J. -P. Gossard, and R. E. Burke, "Stochastic versus deterministic variability in simple neuronal circuits: I. Monosynaptic spinal cord reflexes", *Biophysical Journal*, Vol.67, pp.671–683, 1994.

[5] G. Sugihara and R. M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series", *Nature*, Vol.344, pp.734–741, 1990.

[6] D. T. Kaplan and L. Glass, "Coarse-grained embeddings of time series: Random walks, Gaussian random processes, and deterministic chaos", *Physica* D, Vol.64, pp.431–454, 1993.

[7] R. Wayland, D. Bromley, D. Pickett, and A. Passamante, "Recognizing determinism in a time series", *Phys. Rev. Lett.*, Vol.70, pp.580–582, 1993.

[8] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures", *Neural Computation*, Vol.7, pp.219–269, 1995.

[9] A. Nagami, H. Inada, and T. Miyano, unpublished.

[10] T. Miyano and F. Girosi, "Forecasting global temperature variations by neural networks", *A. I. Memo*, No.1447, 1994 (Artificial Intelligence Laboratory, Massachusetts Institute of Technology).

[11] T. Miyano, "Time series analysis of complex dynamical behavior contaminated with observational noise", *Int. J. Bifurcation and Chaos*, Vol.6, pp.2031–2045, 1996.

[12] T. Miyano, M. Fujito, K. Fujimoto, and A. Sanjoh, "Fractal analysis of leakage-current fluctuations of Si metal-oxide-semiconductor capacitors for the characterization of dry-etching damage", *Appl. Phys. Lett.*, Vol.61, pp.2521–2523, 1992.

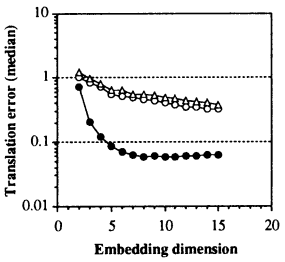


Figure 1: Wayland tests on male /a/ (mausy003): original data (●), phase randomized surrogate (○), and Fourier shuffled surrogate (△).

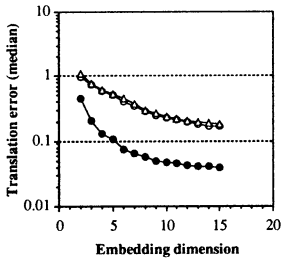


Figure 2: Wayland tests on male /a/ (mmssy003): original data (●), phase randomized surrogate (○), and Fourier shuffled surrogate (△).

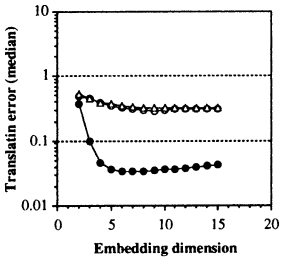


Figure 3: Wayland tests on female /a/ (fsusy003): original data (●), phase randomized surrogate (○), and Fourier shuffled surrogate (△).

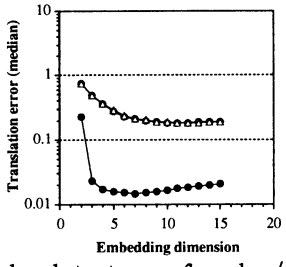


Figure 4: Wayland tests on female /a/ (fynsy003): original data (●), phase randomized surrogate (○), and Fourier shuffled surrogate (△).

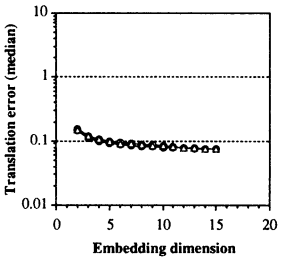


Figure 5: Wayland tests on numerical sine wave added to actually observed correlated random noise: original data (●), phase randomized surrogate (○), and Fourier shuffled surrogate (△).

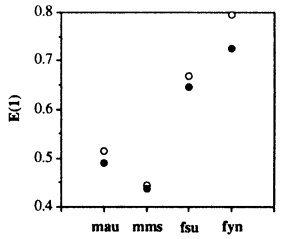


Figure 6: Prediction error about the last half of each original time series in the nonlinear (●) and the linear (○) forecasting. $\tau = 1, \Delta t = 10$.

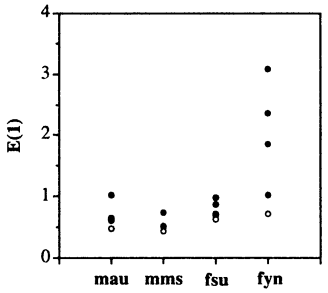


Figure 7: Prediction error about each original (○) and the corresponding surrogate time series (●) in the nonlinear forecasting. $\tau = 1, \Delta t = 10$.

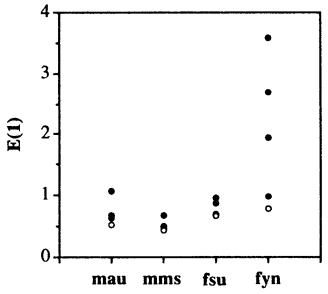


Figure 8: Prediction error about each original (○) and the corresponding surrogate time series (●) in the linear forecasting. $\tau = 1, \Delta t = 10$.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A Pruning Algorithm for Higher Order Neural Networks

Lixin Ma, Bing Xu and Hiromi Miyajima
Fac. of Engineering, Kagoshima University
Korimoto 1-21-40 Kagoshima-shi, 890, JAPAN

Abstract

In order to determine the structure of higher order neural networks, this paper describes a pruning back propagation method for three-layered second order neural networks and shows the validity of them by numerical examples.

key words: pruning algorithm, higher order neural networks, back propagation, three-layered second order networks, two spiral problem

1. Introduction

Many studies from theory to application have been done with neural networks. So far, properties and possibilities of them have been clarified. Therefore, some studies about new evolution of this model start and higher order neural networks which use the weighted sum of products of input variables, have been proposed as a new concept[1,2,4]. In some applications by using them, it is shown that they are superior in ability to the traditional neural networks. But, little is known about the effectiveness of pruning algorithms for higher order neural networks(HONNs).[4]

In this paper, a pruning algorithm for HONNs are proposed and the effectiveness of them is shown by numerical simulations. First HONNs and back propagation methods are given. And a pruning algorithm is proposed. Further numerical simulations for pruning BP method are performed.

2. Higher order neural networks and back propagation method

First, we consider HONNs each layer of which consists of N_m(m=0,---,M-1) units, where the number of layers is M, the 0-th and the (M-1)-th layers are called input and output layers, respectively. The i-th (higher order neural) element of the m-th layer has N_{m-1} inputs and one input and each value of input takes any real number from -1 to 1 and output takes any real numbers from 0 to 1. The state of a unit, the output, is represented as a function of potential u. The potential u increases in proportion to the weighted sum $\sum_{[L_p]} w_{i[L_p]} y_{l_1} \cdots y_{l_p}$ of all combinations of products of p pieces of input variables y_i ,$1 \leqq i \leqq N_{m-1}$ and a threshold s_i, where $[L_p]=l_1 \cdots lp$ and $\sum_{[L_p]}$ is defined as

$$\sum_{l_1=1}^{N_{m-1}-p+1} \sum_{l_2=l_1+1}^{N_{m-1}-p+2} \cdots \sum_{l_p=l_{p-1}+1}^{N_{m-1}}$$

to exclude the overlapping of variables and $w_{i[L_p]}$ is called a weight of products of p pieces of input variables. Then, each output of the i-th unit of the m-th layer, $y_i^{(m)}$(i=1,---,n, m=1,---,M-1) , is determined as follows.

$$y_i^{(m)}=F\left(\sum_{p=1}^{k} \sum_{[L_p]} w_{i[L_p]}^{(m)} y_{l_1}(m-1) \cdots y_{l_p}(m-1)+s_i^{(m)}\right),$$

$$F(x)=\frac{1}{1+e^{-x}} \qquad (1)$$

, where k is the maximum number of order for HONNs. Fig.1 shows an example with k=2 and M=3 of HONNs. For input data $\mathbf{y}^{(0)}$ to input layer, let the desirable value of output layer be represented by $\mathbf{d}$, where

$$\boldsymbol{y}^{(0)} = (y_1^{(0)}, \cdots, y_{N_0}^{(0)}), \boldsymbol{d} = (d_1, \cdots, d_{N_{M-1}}).$$

Now the output $y_i^{(m)}$ for $\mathbf{y}^{(0)}$ is computed from the Eq.1.

The square error E between output $y_i^{(M-1)}$ of output layer for $\mathbf{y}^{(0)}$ and the desirable output d_i is defined as follows.

$$E=\frac{1}{2} \sum_{i=1}^{N_{M-1}} (y_i^{(M-1)}-d_i)^2 \qquad (2)$$

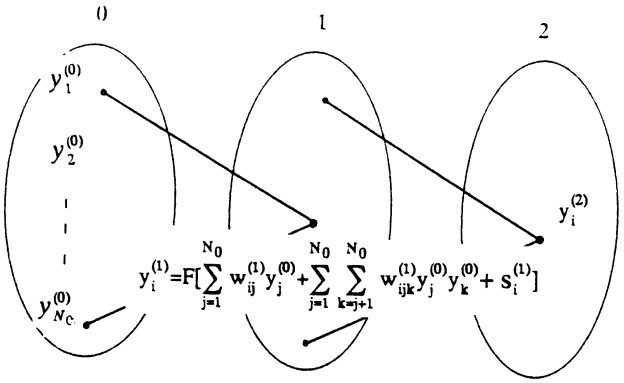


Fig.1 A HONN with k=2 and M=3.

In order to decrease E, weights of each layer are regulated according to the Eq.3.

$$w_{i[L\rho]}^{(m)}(t+1)=w_{i[L\rho]}^{(m)}(t)-K\frac{\partial E}{\partial w_{i[L\rho]}^{(m)}} \qquad (3)$$

The method is called back propagation(BP) method.[2]

In the following, we will deal with the case of k=2, M=3 and N_2=1 without loss of generality.

3. Pruning BP method for higher order neural networks

From the Eqs. 1, 2 and 3, back propagation method for HONNs with k=2, M=3 N_2=1 is shown as follows. Let P be the number of input-output data and r=1,---,P.

[Step a1]

Let r=1.

[Step a2]

An initial setting of weights is conducted first. The initial value of them is set to random real numbers.

The input-output data $(y_1^{(0)r}, - - - , y_{N_0}^{(0)r}, d^r)$ is given.

[Step a3]

For input $y_i^{(0)r}$ (i=1,---,N_0), output of each layer is computed as follows;

$$y_i^{(m)r}=F[\sum_{j=1}^{N_{m-1}} w_{ij}^{(m)} y_j^{(m-1)r} + \sum_{j=1}^{N_{m-1}}\sum_{k=j+1}^{N_{m-1}} w_{ijk}^{(m)} y_j^{(m-1)r} y_k^{(m-1)r} + s_i^{(m)}] \qquad (4)$$

, where m=1,2.

[Step a4]

The objective function E over weights $w_{ij}^{(m)}$ and $w_{ijk}^{(m)}$ are computed as follows.

(a) With output layer,

$$\frac{\partial E}{\partial y_i^{(2)}}=y_i^{(2)r}-d_i^r, \quad \frac{\partial E}{\partial x_i^{(2)}}=\frac{\partial E}{\partial y_i^{(2)}} y_i^{(2)r}(1-y_i^{(2)r}),$$

$$\frac{\partial E}{\partial w_{ij}^{(2)}}=\frac{\partial E}{\partial x_i^{(2)}} y_j^{(1)r}, \quad \frac{\partial E}{\partial w_{ijk}^{(2)}}=\frac{\partial E}{\partial x_i^{(2)}} y_j^{(1)r} y_k^{(1)r}. \qquad (5)$$

(b) With hidden layer,

$$\frac{\partial E}{\partial y_l^{(1)}}=\sum_{i=1}^{N_2}\frac{\partial E}{\partial x_i^{(2)}}\frac{\partial x_i^{(2)}}{\partial y_l^{(1)}},$$

$$\frac{\partial x_i^{(2)}}{\partial y_l^{(1)}}=w_{il}^{(2)}+G(l)\sum_{j=1}^{l-1} w_{ijl}^{(2)} y_j^{(1)r}+H(l)\sum_{k=l+1}^{N_1} w_{ilk}^{(2)} y_k^{(1)r},$$

$$\frac{\partial E}{\partial x_i^{(1)}}=\frac{\partial E}{\partial y_l^{(1)}} y_l^{(1)r}(1-y_l^{(1)r}), \quad \frac{\partial E}{\partial w_{lh}^{(1)}}=\frac{\partial E}{\partial x_i^{(1)}} y_h^{(0)r},$$

$$\frac{\partial E}{\partial w_{lhg}^{(1)}}=\frac{\partial E}{\partial x_i^{(1)}} y_h^{(0)r} y_g^{(0)r} \qquad (6)$$

, where G(l)=0 for l=1 and 1 for l+1, and H(l)=0 for l=N_2 and 1 for l=N_2.

[Step a5]

The weights $w_{ij}^{(m)}$ and $w_{ijk}^{(m)}$ are improved according to the following

$$w_{ij}^{(m)}(t+1)=w_{ij}^{(m)}(t)-K_1\frac{\partial E}{\partial w_{ij}^{(m)}}+K_2(w_{ij}^{(m)}(t)-w_{ij}^{(m)}(t-1))$$

$$w_{ijk}^{(m)}(t+1)=w_{ijk}^{(m)}(t)-K_1\frac{\partial E}{\partial w_{ijk}^{(m)}}+K_2(w_{ijk}^{(m)}(t)-w_{ijk}^{(m)}(t-1)) \qquad (7)$$

[Step a6]

If r=P then the algorithm terminates, if r≠P then go to Step a2 with r←r+1.

A pruning algorithm for HONNs is proposed using the above method.

[Step b1]

An initial setting of many prepared weights is conducted first. The threshold T is set. Let t=1. Let the set of ordered numbers of weights {1,---,r} be denoted by **R**. Let T_0 be the maximum number of learning.

[Step b2]

The steps Step a1 to Step a6 are performed.

[Step b3]

The error $E_0(t)$ for the desirable output is calculated as follows.

$$E_0(t) = \frac{1}{P}\sum_{r=1}^{P} |y^{(2)r} - d^r|$$

[Step b4]

The mean squared error $E_0(t)$ and the threshold T are compared. Then

if $E_0(t) \leqq T$ then go to Step b5,

if $E_0(t) > T$ and $t < T_0$ then go to Step b2 with $t \leftarrow t+1$

otherwise the algorithm terminates.

[Step b5]

Let $E_k(t)$ be the error in the case where the k-th weight is deleted. That is, when the set of weight numbers is **R**-{k}, the following error is computed:

$$E_k(t) = \frac{1}{P}\sum_{r=1}^{P} |y^{(2)r} - d^r|$$

Then we select a weight k_d such that

$$E_{k_d}(t) = \min E_k(t)$$

It means that the mean squared error is minimum value in the case where the k_d-th weight is deleted. Therefore, the k_d-th weight is deleted.

[Step b6]

If $E_{k_d}(t) > T$ then the algorithm terminates otherwise go to Step b2 as $\mathbf{R} \leftarrow \mathbf{R}$-{$k_d$}, and $t \leftarrow t+1$.

4. Numerical simulations

In this section, function approximation and pattern classification as numerical simulations are performed.

4.1 Function approximations

In order to demonstrate the validity of the proposed algorithms, some input-output data are prepared for each system. The systems are identified as neural networks. Here are three systems each of which is specified by each function as follows. Each function has two inputs x, y and one output z.

(1) $z = \frac{(xy)^2+1}{2}$

(2) $z = \frac{4\sin(3.14x)+2\cos(3.14y)}{12} + 0.5$

(3) $z = 1.9\{1.35 + e^x \sin[13(x-0.6)^2]e^{-y}\sin(7y)\}$

Each input variable is normalized within [-1,1]. The output data z is normalized within [0,1]. Five hundred data are employed for identification in each system. Each method stops the learning when the error $E_0(t)$ is less than 0.002. Table 1 shows mean squared errors (mse) for 500 testing data.

4.2 Pattern classification

A benchmark, two-spiral problem, is used to test the proposed method. This problem discriminates between two sets of training points that lie on two distinct spirals in the X-Y plane. Each spiral has 96 input-output pairs in each spiral. It appears to be a very difficult task for BP for first order NNs[3]. A second order neural network can reach 100 percent of classification in a limited time, while first order one seems to get stuck in a minimum after a long time, fails to reach 100 percent of classification. Fig.2 shows results when pruning methods reach 95 percent of classification for training set. For the first order neural network, the number of weights reduces from121 to 101. For the second order neural network, the number of weights reduces from126 to 91.

As a result, the proposed method is superior to the other methods even when the same number of weights are used.

5. Conclusion

This paper describes a pruning back propagation algorithm for HONNs. First a pruning algorithm is proposed and it is applied to function approximation and classification problem, two spiral problem. It is shown that the proposed method is superior to the traditional BP one for HONNs. Specifically, the proposed method for second order NNs is superior to one for first order NNs. In future work, we will compare the proposed method with the other pruning ones.

References

[1] D. E. Rumelhart and J. L. McClelland , "Parallel Distributed Processing," MIT Press, Cambridge, 1986.

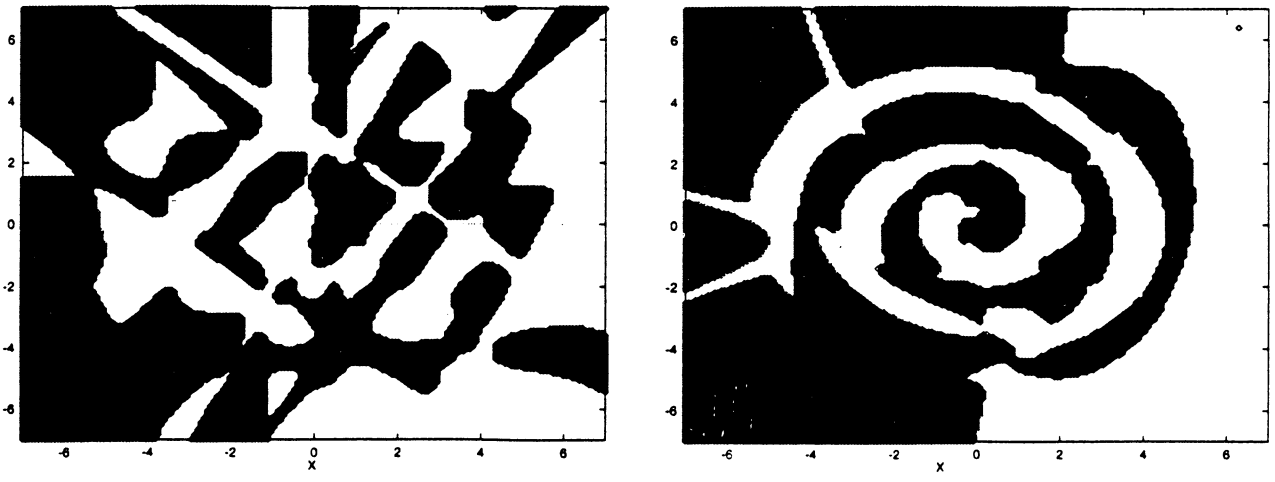
[2] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors, " Nature, 323, pp.533-536, 1986.

[3] Y. Shang and B. W. Wah,"Global optimization for neural network training," Computer, vol.29, no.3, pp.45-54, 1996.

[4] K. Mehrotra, C. K. Mohan, and S. Ranka, "Elements of artificial neural networks," MIT Press, 1996.

Table 1 Result of function approximation.
For a/b, a and b mean the number of weights and mse, respectively.
In pruning method a→b means that the number of weights reduces from a to b.

Method / System	The number of weights is fixed.		Pruning method	
	First order	Second order	First Order	Second Order
(1)	21 / 0.0226	9 / 0.0049	33→24 / 0.001059	28→11 / 0.000764
(2)	21 / 0.0692	18 / 0.0685	33→19 / 0.002455	39→14 / 0.002630
(3)	49 / 0.0115	51 / 0.113	65→43 / 0.004410	64→36 / 0.004473



(a) Pruned first order neural network.

(b) Pruned second order neural network.

Fig.2 Two dimensional graph for two spiral problem.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A Learning Method for Large-scale Recurrent Neural Networks

N. Honma
Col. of Medical Sciences
Tohoku University, Sendai 980-77, JAPAN
(homma@abe.ecei.tohoku.ac.jp)

K. Abe
Faculty of Eng.

H. Takeda
Faculty of Eng.
Tohoku Gakuin University
Tagajyo 985, JAPAN

Abstract

The paper demonstrates that the recurrent neural network dynamics in the vicinity of the edge of chaos are effectual for estimation problems of unknown complicated nonlinear dynamics. The emphasis of this paper is on new theory to prove why this distinguished dynamics are effectual for recurrent neural network's learning.

The theory proves that the edge of chaos is necessary to learn the attractors of any continuous dissipative dynamical systems except for a fixed point and that the correspondence of the Lyapunov spectra is also necessary for learning. Based on the new theory, we propose a new learning algorithm to keep the network dynamics in the edge of chaos. In this algorithm the standard parameters of neurons are changed also according to the global error measure calculated by the well known back propagation algorithm.

Key words: *Recurrent neural networks, Chaos, Complex systems, Learning, The edge of chaos.*

1 Introduction

Recurrent neural networks with dynamics can be useful for identification of nonlinear complicated dynamical systems[4]. Therefore various learning algorithms for the networks have been proposed and discussed. Most of these algorithms[6, 7, 5], however, are based on the back-propagation algorithm for feedforward neural networks. In addition, the global error measure is the only criterion of the learning algorithm for the feedforward networks. Since the properties of their recurrent network dynamics are not considered in these algorithms, the learning in these networks has many difficulties. For example, these algorithms have lots of local minima, require numerous memories, and are computationally expensive for large-scale networks which are composed of a number of neurons. Especially, in the case of complex time series such as chaos, the learning tends to be trapped to local minima such as the average value of the training time series. This fact means that the complexity of the network dynamics is an important factor for the recurrent network learning.

In our previous paper, properties of the network dynamics are investigated and relations between the complexity of dynamics and the learning ability are revealed by using new parameters[1]. We showed excellent learning ability at the edge of chaos from several simulation results.

In this paper, we prove that the edge of chaos is necessary to learn the attractors of any continuous dissipative dynamical systems except for a fixed point and that the correspondence of the Lyapunov spectra is also necessary for learning. Then our learning algorithm based on the advantages of this distinguished dynamics is improved by new theory.

2 Recurrent neural networks and learning objects

2.1 Recurrent neural networks

To consider the various interactions, we use recurrent neural networks without restraint on configurations of the connections: *fully connected recurrent neural networks*(Fig. 1). The networks consist of neurons whose internal states x_i, $i = 1, 2, \cdots, N$, are governed by

$$x_i(t+1) = \frac{1}{1+\exp(-2s_i(t+1)/s_0)}, \quad (1)$$

$$s_i(t+1) = \sum_j^N w_{ij}x_j(t) + \theta_i, \quad (2)$$

$$\hat{y}_i(t) = x_i(t), \qquad (t = 1, 2, \cdots). \quad (3)$$

$t = 1, 2, \cdots$, where s_i and $\hat{y}_i$ denote inputs and outputs, respectively. N is the number of neurons. w_{ij},

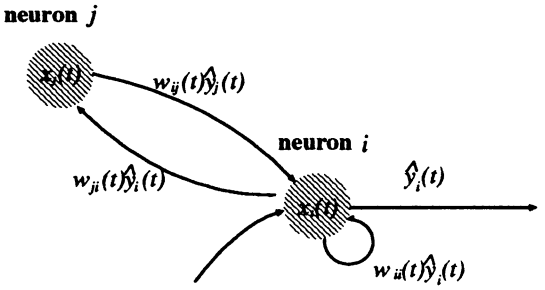


Figure 1: Fully connected recurrent neural networks

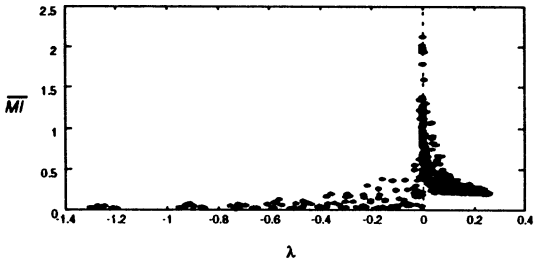


Figure 2: Average mutual information $\overline{MI}$ versus Lyapunov exponent λ.

s_0 and θ_i represent synaptic connection strengths, the gain of sigmoids and thresholds, respectively.

2.2 Learning objects

We consider a n-dimensional continuous dissipative dynamical system defined by

$$\begin{aligned} \dot{\boldsymbol{y}}(t) &= \boldsymbol{f}(\boldsymbol{y}(t)), \\ \boldsymbol{y}(t) &= [y_1(t), y_2(t), \cdots, y_n(t)]^T, \\ \boldsymbol{f}(\boldsymbol{y}) &= [f_1(\boldsymbol{y}), f_2(\boldsymbol{y}), \cdots, f_n(\boldsymbol{y})]^T, \end{aligned} \tag{4}$$

as a target system which is an object of our recurrent networks' learning. The sampled time series of its attractor $\boldsymbol{y}(t), \boldsymbol{y}(t+\Delta t), \boldsymbol{y}(t+2\Delta t), \cdots$ are fed to our recurrent neural network as the training data. Here Δt is the sampling period and is chosen for much shorter one than the attractor's, in the case of a strange attractor, shorter one than the mean orbital period. For simplicity, Δt is equal to 1, i.e., the interval of the time step in the recurrent neural networks

3 The edge of chaos and the learning ability

The dynamical complexity of the attractors of the object systems given by (4) can be classified by Lyapunov spectra $(\lambda_1, \lambda_2, \cdots, \lambda_n)$, since the signs of the Lyapunov exponents provide a qualitative picture of a system's dynamics[8]. For example, in a three-dimensional continuous dissipative dynamical system the only possible spectra, and the attractors they describe, are as follows: $(-,-,-)$, a fixed point; $(0,-,-)$, a limit cycle; $(0,0,-)$, a two-dimensional torus; and $(+,0,-)$, a strange attractor.

Note that one of the Lyapunov exponents λ_t, which is the Lyapunov exponent of the *temporal* direction, is zero for all of the attractors except for a fixed point. Therefore, making λ_t of the recurrent neural networks be 0 is necessary, that is, the edge of chaos is necessary for learning most of the attractors.

4 Learning Algorithm

4.1 Concept and its criteria

Our previous learning method is based on the advantages of the edge of chaos where the networks can represent various objects since mutual information are high such as in figure 2. This is only a reason why this distinguished dynamics are effectual and it is shown by simulation results[2, 3].

On the other hand, our new theory proves why the edge of chaos is effectual. To learn a object requires that the Lyapunov spectra of the network is equal to the Lyapunov spectra of the learning object. It is however computationally expensive to make the complete correspondence. In this paper, two Lyapunov exponents, the largest Lyapunov exponent λ_1 and the Lyapunov exponent of temporal direction, λ_t, are identified to the exponents of the object, respectively. Letting λ_1^{obj} be the largest Lyapunov exponent of the learning object, our new algorithm changes the network parameters w_{ij}, θ_i to make the equivalences $\lambda_1 = \lambda_1^{obj}$, $\lambda_t = 0$(the edge of chaos).

At first, we define the two error measures.

$$e_1 = \frac{1}{2}(\lambda_1 - \lambda_1^{obj})^2, \tag{5}$$

$$e_t = \frac{1}{2}\lambda_t^2. \tag{6}$$

Also we use the conventional error measure.

$$e_c = \frac{1}{2}\sum_{t=1}^{T}(\hat{y}(t) - y(t))^2, \tag{7}$$

where T denote the length of the learning time series. Then we define total error measure $e = e(e_c, e_1, e_t)$. For simplicity, we use

$$e = e_c + e_1 + e_t. \tag{8}$$

We get the changes of parameters, Δw_{ij}, to decrease the error e by the gradient method as follows.

$$\begin{aligned}\Delta w_{ij} &= -\eta \frac{\partial e}{\partial w_{ij}}, \\ &= -\eta \left(\frac{\partial e_c}{\partial w_{ij}} \right. \\ &\quad \left. +(\lambda_1 - \lambda_1^{obj}) \frac{\partial \lambda_1}{\partial w_{ij}} + \lambda_t \frac{\partial \lambda_t}{\partial w_{ij}} \right). \end{aligned} \tag{9}$$

where η is a positive constant.

Here $\partial e_c / \partial w_{ij}$ can be calculated by the conventional learning algorithms[6, 7, 5]. Therefore the changes Δw_{ij} are calculated by computing λ_1, λ_t, $\partial \lambda_1 / \partial w_{ij}$ and $\partial \lambda_t / \partial w_{ij}$.

4.2 Implementation

4.2.1 Computation of λ_1 and λ_t

Given a continuous dynamical system in an n-dimensional phase space, we monitor the long-term evolution of an infinitesimal n-sphere of initial conditions; the sphere will become an n-ellipsoid due to the locally deforming nature of the flow. The i-th one dimensional Lyapunov exponent λ_i is then defined in terms of the length of the ellipsoidal principal axis vector ${}^i\boldsymbol{L}(T)$

$$\lambda_i = \lim_{T \to \infty} \frac{1}{T} \log \left[\frac{\| \, {}^i\boldsymbol{L}(T) \, \|}{\| \, {}^i\boldsymbol{L}(0) \, \|} \right], \tag{10}$$

where the λ_i are ordered from largest to smallest[8].

In the case of n-dimensional difference equation $\boldsymbol{x}(t+1) = \boldsymbol{F}(\boldsymbol{x}(t))$, by letting $\boldsymbol{DF}(\boldsymbol{x}(t))$ be Jacobi matrix at the point $\boldsymbol{x}(t)$, and linearizing ${}^i\boldsymbol{L}(t+1) = \boldsymbol{DF}(\boldsymbol{x}(t)) \, {}^i\boldsymbol{L}(t)$, we get

$$\lambda_i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \log \left[\frac{\| \, \boldsymbol{DF}(\boldsymbol{x}(t)) \, {}^i\boldsymbol{L}(t) \, \|}{\| \, {}^i\boldsymbol{L}(t) \, \|} \right]. \tag{11}$$

To avoid the computational divergence of ${}^i\boldsymbol{L}(t+1)$, we normalize ${}^i\boldsymbol{L}(t+1)$ as

$$\delta \, {}^i\boldsymbol{L}(t+1) = \frac{\boldsymbol{DF}(\boldsymbol{x}(t)) \delta \, {}^i\boldsymbol{L}(t)}{\| \, \boldsymbol{DF}(\boldsymbol{x}(t)) \delta \, {}^i\boldsymbol{L}(t) \, \|}. \tag{12}$$

We denote ${}^1L(t) = \| \, {}^1\boldsymbol{L}(t) \, \|$, then

$$\lambda_1 = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \log \, {}^1L(t). \tag{13}$$

In recurrent networks of (1)~(3), the (i,j)-element of the Jacobi matrix, $DF_{ij}(\boldsymbol{x}(t))$, are calculated by

$$DF_{ij}(\boldsymbol{x}(t)) = \frac{2}{s_0} w_{ij} x_i(t+1)(1 - x_i(t+1)). \tag{14}$$

Therefore the element of ${}^1\boldsymbol{L}(t)$, ${}^1L_i(t), i = 1, 2, \cdots, N$ are given by

$${}^1L_i(t) = \frac{2}{s_0} X_i(t+1) \sum_{j=1}^{N} w_{ij} \delta \, {}^1L_j(t), \tag{15}$$

where, $\delta \, {}^1\boldsymbol{L}(t) = [\delta \, {}^1L_1(t), \delta \, {}^1L_2(t), \cdots, \delta \, {}^1L_N(t)]^T$, and $X_i(t) = x_i(t)(1 - x_i(t))$.

Finally, we can get

$${}^1L(t) = \sqrt{\sum_{i=1}^{N} (\, {}^1L_i(t))^2}. \tag{16}$$

That is, preparing arbitrary n-dimensional vector such that $\| \, \delta \, {}^1\boldsymbol{L}(0) \, \| = 1$ then we can calculate the largest Lyapunov exponent λ_1 using time series in learning period $\boldsymbol{x}(0), \boldsymbol{x}(1), \cdots, \boldsymbol{x}(T)$ and the connection strengths w_{ij}.

In the case of computation of λ_t, we can use the following approximation.

$${}^t\boldsymbol{L}(t) \simeq \boldsymbol{x}(t+1) - \boldsymbol{x}(t). \tag{17}$$

Therefore

$${}^tL(t) \simeq \sqrt{\sum_{i=1}^{N} (x_i(t+1) - x_i(t))^2}. \tag{18}$$

Then we can calculate

$$\lambda_t = \lim_{T \to \infty} \frac{1}{T} \log \frac{{}^tL(T)}{{}^tL(0)}. \tag{19}$$

4.2.2 Computation of $\partial \lambda_1 / \partial w_{ij}$ and $\partial \lambda_t / \partial w_{ij}$

For simplicity, let ${}^1\boldsymbol{L}(t)$ denote $\boldsymbol{L}(t)$. From (13)

$$\begin{aligned}\frac{\partial \lambda_1}{\partial w_{ij}} &= \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\partial \log L(t)}{\partial w_{ij}}, \\ &= \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{1}{L(t)} \frac{\partial L(t)}{\partial w_{ij}}. \end{aligned} \tag{20}$$

From (16)

$$\frac{\partial L(t)}{\partial w_{ij}} = \frac{1}{L(t)} \sum_{k=1}^{N} L_k(t) \frac{\partial L_k(t)}{\partial w_{ij}}. \quad (21)$$

From (15)

$$\frac{\partial L_k(t)}{\partial w_{ij}} = \frac{2}{s_0} \left(\frac{\partial X_k(t+1)}{\partial w_{ij}} \sum_{m=1}^{N} w_{km} \delta L_m(t) + X_k(t+1) \left[\Delta_{ik} \delta L_j(t) + \sum_{m=1}^{N} w_{km} \frac{\partial \delta L_m(t)}{\partial w_{ij}} \right] \right). \quad (22)$$

Here $\Delta_{ik} = 1$ in the case of $i = k$, otherwise $\Delta_{ik} = 0$ and

$$\frac{\partial X_k(t)}{\partial w_{ij}} = (1 - 2x_k(t)) \frac{\partial x_k(t)}{\partial w_{ij}}. \quad (23)$$

$\partial x_k(t)/\partial w_{ij}$ can be calculated by the conventional learning algorithms[6, 7, 5].

Also we have

$$\frac{\partial \delta L_m(t)}{\partial w_{ij}} = \frac{1}{L^2(t-1)} \left(\frac{\partial L_m(t-1)}{\partial w_{ij}} L(t-1) - \frac{\partial L(t-1)}{\partial w_{ij}} L_m(t-1) \right). \quad (24)$$

Here letting $\partial \delta L_m(0)/\partial w_{ij}$ be 0, from (22) we can calculate $\partial \delta L_k(1)/\partial w_{ij}$ and $\partial L(1)/\partial w_{ij}$. Then from (24), we can calculate $\partial \delta L_m(1)/\partial w_{ij}$, and then we can calculate $\partial L(t)/\partial w_{ij}$, and then $\partial \lambda_1/\partial w_{ij}$.

And last we describe how to calculate $\partial \lambda_t/\partial w_{ij}$. From (19)

$$\frac{\partial \lambda_t}{\partial w_{ij}} = \lim_{T \to \infty} \frac{1}{T} \left(\frac{1}{{}^tL(T)} \frac{\partial\, {}^tL(T)}{\partial w_{ij}} - \frac{1}{{}^tL(0)} \frac{\partial\, {}^tL(0)}{\partial w_{ij}} \right). \quad (25)$$

From (18)

$$\frac{\partial\, {}^tL(t)}{\partial w_{ij}} = \frac{1}{{}^tL(t)} \sum_{k=1}^{N} {}^tL_k(t) \left(\frac{\partial x_k(t+1)}{\partial w_{ij}} - \frac{\partial x_k(t)}{\partial w_{ij}} \right). \quad (26)$$

Therefore we can calculate $\partial \lambda_t/\partial w_{ij}$ by calculating $\partial x_k(t)/\partial w_{ij}$ using the conventional methods.

5 Conclusion

It is shown that the edge of chaos is necessary to learn the attractors of any continuous dissipative dynamical systems except for a fixed point and that the correspondence of the Lyapunov spectra is also necessary for learning. Simulation results illustrating the effectiveness of the proposed algorithm will be shown at the conference.

Further research is to evaluate computational expensiveness and amount of required memories of proposed algorithm.

References

[1] N. Honma, K. Abe, M. Sato and H. Takeda, "On Emergent Evolution of Holon Networks Using an Autonomous Decentralized Method", Proc. of 13th World Congress of IFAC, vol. I, pp.237-242, 1996.

[2] N. Honma, K. Kitagawa and K. Abe: Effect of Complexity on Learning Ability of Recurrent Neural Networks, Proc. of the 2nd AROB, pp.58-61, 1997.

[3] N. Honma, K. Kitagawa, K. Abe and H. Takeda: An Autonomous Criterion of Learning Methods for Recurrent Neural Networks, Proc. of the 2nd ASCC, **II**, pp.219-222, 1997.

[4] K. S. Narendra, and K. Parthasarathy: Identification and control of dynamical systems using neural networks, Neural Networks, vol. 1, No. 1, pp.4-27, 1990

[5] B. A. Pearlmutter, "Learning State Space Trajectories in Recurrent Neural Networks", Neural Computation, vol. 1, No. 2, pp.263-269, 1989.

[6] F.J.Pineda, "Generalization of Back-Propagation to Recurrent Neural Networks", Physical Review Letters, vol. 59, No. 19, pp.2229-2232, 1987.

[7] R. J. Williams and D. Zipser, "A learning Algorithm for Continually Running Fully Recurrent Neural Networks", Neural Computation, vol. 1, No. 2, pp.270-280, 1989.

[8] A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano: Determining Lyapunov Exponents from a time series, Physica, **16D**, pp.285-315, 1989.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Behavior Planning of Autonomous Mobile Robots based on Cooperative Neural Networks

Mitsunobu Murata, Morikazu Nakamura, Takashi Ono, and Kenji Onaga
Dept. of Information Engineering, University of the Ryukyus
Nishihara Okinawa 903-01 Japan

Abstract

In this paper we propose an approach to obtain behavior planning of autonomous mobile robots based on cooperative neural networks. The merits of our approach are (1) reusability of neural networks already obtained, (2) simplicity of each neural network even if the global purpose is complex, and (3) lesser learning cost. Moreover, our method can be extended to the multiple robots case, that is, applicable to cooperative works of multiple robots. As future works, we have to evaluate our proposed method in detail and apply to more complex behavior planning.

1 Introduction

In this paper, we propose a method for behavior planning of autonomous mobile robots based on cooperative neural networks. We treat behavior planning of single robot and that of multiple robots.

Let us consider behavior planning for some purpose. Usually, we can divide the purpose into multiple simple ones. For example, in the case of navigation for approaching to a light source, we can have two simple purposes, to approach the light source, and to avoid obstacles. When we construct behavior plan based on neural networks, traditional approaches try acquiring one neural network realizing the purpose. However, in our approach we first construct neural networks for the simple purposes and then combine the networks for the global purpose. That is, we develop a mechanism of cooperation of the neural networks. The neural networks cooperate each other for the global purpose. The merits of our approach are (1) reusability of neural networks already acquired, (2) simplicity of each neural network even if the global purpose is complex, and (3) lesser learning cost (computation cost).

We can extend this concept to the multiple robots case if we consider that neural networks on different robots are also cooperate each other. In this case cooperation between neural networks on different robots should be realized in a radio communication environment.

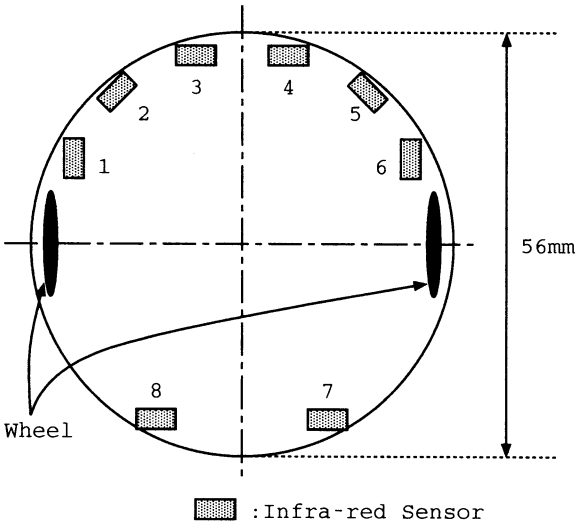


Figure 1: Mobile robot Khepera

In our experiment we use a miniature mobile robot Khepera[7], shown in Figure 1, trademarked by K-Team SA. Khepera is very useful for researches on mobile robots[4][5][6]. Many papers related with robotics were simulation-based but not real-robot-based.

References[1][2][3] collected advanced research results with respect to robotics. Floreano and Mondada showed efficient learning of navigation of a mobile robot based on a genetic algorithm and neural network using very simple facility, a Khepera and a personal computer.

2 Evolutionary Behavior Planning

In this section, we explain Floreano and Mondada's way [4][5], say a *conventional method*, of behavior planing based on evolutionary computation with a neural

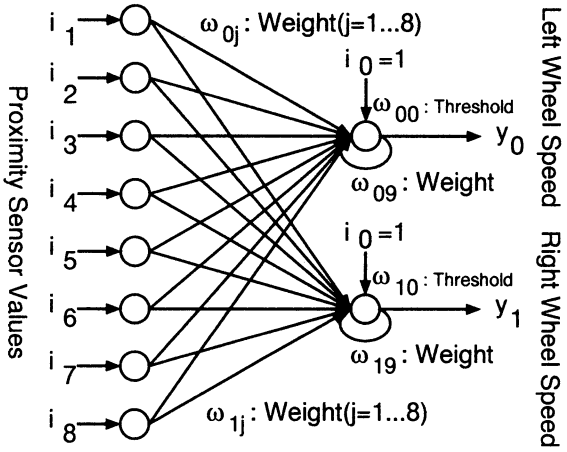


Figure 2: Neural network for robot controller

network.

Let us consider a behavior of a Khepera for avoiding obstacles.

2.1 Robot Controller based on Neural Networks

Figure 2 shows a neural network to control the robot. The inputs of the first layer, that is, the inputs of this network, corresponds to proximity sensors of the robot, while the output of the second layer, namely the outputs of the network, associate with left or right wheel speed. All the input and output values range between 0 and 1. The outputs of the first layer connect to the inputs to the second layer.

If an output value of the network equals 0.5, the wheel dose not rotate. Larger values than 0.5 let the wheel turn forward and smaller ones than 0.5 turn backward.

The outputs of the network are calculated by the following equation.

$$y_k = F(X_k) = \frac{1}{1+e^{-X_k}} \tag{1}$$

$$X_k = \sum_{j=0}^{9} \omega_{kj} i_j (i_0 = 1) \tag{2}$$

where the function F is called the *sigmoid* function, i_j, $j = 1, 2, ..., 8$, is an input value of the network, that is, corresponding proximity sensor value, y_k, $k = 0, 1$, is an output value, that is, the left/right wheel speed, ω_{kj}, $j = 1, ..., 8$, and $k = 0, 1$ means the input weight of the neuron k of the output layer corresponding to the input neuron j of the input layer, and ω_{k0} represents the threshold of the neuron k of the output layer.

2.2 Evolutionary Learning

In this section we describe evolutionary learning of robot's behavior. A chromosome considered here is represented by a sequence of the input weights and thresholds of the network, that is, a chromosome should be a sequence of real numbers.

Chromosomes in the initial population are generated randomly such that the range of each gene is between -0.5 and $+0.5$.

The whole procedure of the evolutionary learning can be described as follows.

1. Evaluate each individual. That is, for each individual,
   (a) let the robot run randomly for a while and stop.
   (b) set the neural network corresponding to the individual to the robot
   (c) get the input data from the sensors and calculate the output of the network.
   (d) let the robot move according to the output of the network.
   (e) calculate the fitness value by the following equation:

$$\Phi = V(1 - \sqrt{\Delta v})(1 - i) \tag{3}$$

   where V is the speed of the robot during the robot runs, Δv is the absolute value of the subtraction between left and right wheels speeds, i is the biggest among values of eight proximity sensors.

2. Select chromosomes for the next generation by roulette selection.

3. Apply crossover and mutation operations according to their probability.

4. Go to the step 1 unless the terminate condition holds.

After a enough number of generation, we get neural networks learned avoiding obstacles.

3 Robot Controller based on Cooperative Neural Networks

In the previous section, we described a conventional approach for evolutionary learning proposed in [4][5]. This approach is very useful for simple behavior navigation. However, when it comes to the case of complex behavior navigation, the approach does not seem to be useful. In this section we propose another approach based on cooperative neural networks. It can be easily extended to the case of multiple robots cooperation works.

Let us consider learning of navigation for robots which need to avoid obstacles and approach a light source. We use this example here to explain our approach for easier understanding though it is not enough complicate to emphasize the merit of our approach comparing with the previous conventional one.

3.1 Cooperative Neural Networks

We can generally divide the given (complex) purpose into multiple simple ones. For the previous example, we can have two simple purposes, to avoid obstacles and to reach the light source.

When we construct behavior plan based on neural networks, traditional approaches try acquiring one neural network for the purpose. However, in our approach we first construct neural networks for the simple purposes and then combine the networks for the global purpose. That is, we develop a mechanism of cooperation of the neural networks. Therefore, at the first step, for each of the two simple purposes, to avoid obstacles and to reach the light source, we construct a robot controller based on a neural network, for example, by the method described in Section 2. Note that these two neural networks should have the same output layer since both of them compute the next behavior(go forward, backward, turn to the left, or to the right) of the robot. The inputs for the neural network to avoid obstacles are connected to the proximity sensors, on the other hand those for the other one are combined with the light sensors. The mobile robot Khepera has the both sensors on its front and back. These two networks to avoid obstacles and approach the light source should be learned with their fitness functions as follows:

$$\Phi_1 = 1 - i, \tag{4}$$

$$\Phi_2 = j, \tag{5}$$

respectively, where i is the biggest among values of eight proximity sensors and j is that of eight light sensors.

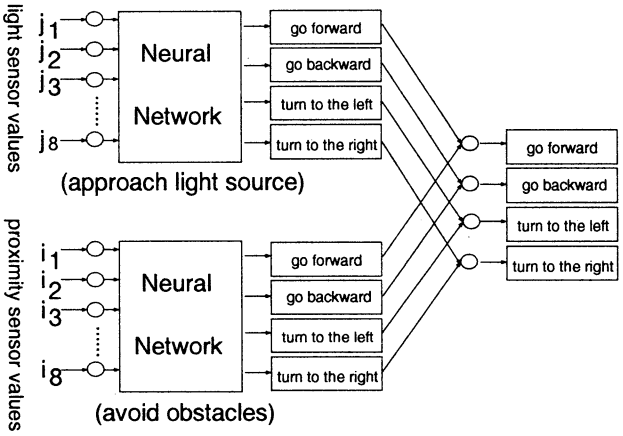


Figure 3: Cooperative Neural Networks

Secondly, we combine the neural networks for the global purpose, to avoid obstacles and approach the light source. As depicted in Figure 3, in the neural network for the global purpose, two neural networks for simple purposes are connected to the new output layer. At this step, only the weights and thresholds of the new output layer are learned. For the global learning, the following fitness function is employed:

$$\Phi_{global} = F(\Phi_1, \Phi_2) \tag{6}$$

where F is called *cooperation* function. In our experiment, we use

$$F(x_1, x_2) = x_1 \cdot \sqrt{x_2} \cdot V \tag{7}$$

where V is the speed of the robot.

After learning, we can get a neural network for the global purpose based on cooperation of neural networks.

The merits of our approach are listed up as follows:

1. simplicity of each neural network even if the global purpose is complex.
2. lesser learning cost.
3. reusability of neural networks already acquired.

However, division a global purpose into some simple ones is not easy when the purpose is so complicated.

3.2 Cooperative Works of Multiple Robots

In this section we extend the cooperative neural networks to a multiple robots case. Let us consider a

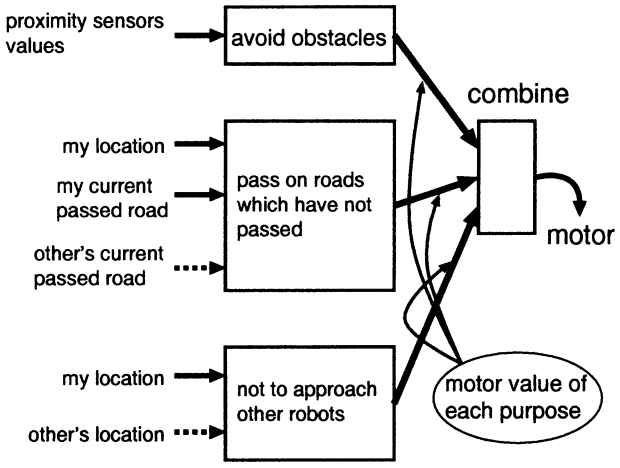


Figure 4: A neural networks structure for road traveling by multiple robots

problem that let the robots run on the roads as many as possible within a fixed time, that is, each robot try to pass on roads which no robot including itself has passed. This problem can be solved by cooperation of multiple robots. We assume that robots can communicate each other by radio. (A mobile robot Khepera has no function of radio communication but we construct an experiment environment so that robots can communicate each other.)

The global purpose of this problem can be described as follows:
"Move on the road where no obstacle exists, no robot including itself has been passed, and go as straight as possible."
We divide the global purpose into three simple purposes, to avoid obstacles, to pass on the road which have not been passed, and not to approach other robots. In this environment, robots can communicate information on passed road and their location each other.

For each simple purpose, a robot controller based on a neural network is constructed and then they are combined for the global purpose. Figure 4 depicts the structure of the robot controller for the global purpose. In this figure, the dotted line shows data transfer by radio, that is, communication between different robots while the others are communication within a robot.

4 Concluding Remarks

In this paper we proposed an approach to obtain behavior planning of autonomous mobile robots based on cooperative neural networks. The merits of our approach are (1) reusability of neural networks already obtained, (2) simplicity of each neural network even if the global purpose is complex, and (3) lesser learning cost. Moreover, our method can be extended to the multiple robots case, that is, applicable to cooperative works of multiple robots. As future works, we have to evaluate our proposed method in detail and apply to more complex behavior planning.

References

[1] "From Intelligent Robots to Artificial Life" *Third International Symposium on Evolutionary Robotics* (1995)

[2] "From Intelligent Robots to Artificial Life" *Fourth International Symposium on Evolutionary Robotics* (1995)

[3] "From Intelligent Robots to Artificial Life" *Collection of Papers ER'97* (1995)

[4] Francesco Mondada and Dario Floreano: "Evolution of Neural Control Structures: Some experiments on Mobile Robots" *Robotics and Autonomous Systems*, No. 16, pp. 183-195, (1995)

[5] Dario Floreano, Francesco Mondada: "Evolution of Homing Navigation in a Real Mobile Robot" *IEEE Transactions on Systems, Man and Cybernetics* (1996)

[6] Keisuke Kondo, Ikuko Nishikawa, Hidekatsu Tokumaru: "A study on an autonomous mobile robot control with behavior network" *Second International Symposium on Artificial Life and Robotics* (1997)

[7] "Khepera USER MANUAL Version 4.06" K-Team SA (1995)

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III '98),
Oita, Japan, 19-21, January, 1998,

Food search behavior and Neural Networks

Teturo Kitazoe, Yasunari Yoshitomi and Junichi Tomiyama
Department of Computer Science and Systems Engineering,
Faculty of Engineering, Miyazaki University
1-1 Gakuen Kibanadai Nishi Miyazaki 889-21 Japan
kitazoe@cs.miyazaki-u.ac.jp

keywords: neural network, competition,
decision-making, information processing of creature, quit process

Abstract

An animal or an insect has to quit to take food if there is obstacles to reach to it. We study how to implement the quit processes in the neural net equations with competition. The results are shown for flower search and for boss changing processes.

1 Introduction

It seems that selforganizing processes with competition in neural networks have a role in many kinds of decision-making processes. A well known example is that a frog processes the input information in the neural networks in the tectum, that is the visual region of the midbrain, and decides which bug to catch [1]. A fox runs after a specific rabbit even if there are many rabbits in its sight. These decision-making processes are called as winner-take-all process (WTA) in neural networks. In WTA, a neuron is excited and becomes a winner, while the other neurons with weaker input become losers. It is known that the WTA process is very stable. The winner neuron remains excited if its input stimulus is reduced very much. The fox will continue running after the aimed rabbit even if it finds better rabbit nearby. However, the fox has to quit the rabbit if the fox finds it impossible to catch it.

We see many examples of these quit processes in the daily life of insects or animals, especially in food search behavior. Think about an ant on a leaf which finds a nice food on the another leaf. However, it must quit the food because it cannot fly to it. The quit ability is one of the most important functions for survival. In the present study we consider how to implement these quit processes in the neural nets equations. We take a neural network equation by Amari and Arbib[1]. We also discuss about WTA equations proposed by Yuille and Grzywacz[2].

2 Neural network model with competition

To approach to our problem, we take the primitive competition model for neural nets by Amari and Arbib and consider how to modify the equation to implement our process.

$$\dot{u}_i = -u + w_1 f(u_i) - w_2 g(v) - h_1 + s_i \quad i = 1, 2, \cdots, n \tag{1}$$

$$\tau \dot{v} = -v + \sum_{i=1}^{n} f(u_i) - h_2 + s_0 \tag{2}$$

Here u_i is the potential of i th element,s_i the input and v the potential of the inhibitory element. The function f,g are given as

$$f(u) = 1[u] = \begin{cases} 1, & u > 0 \\ 0, & u \leq 0 \end{cases} \tag{3}$$

$$g(u) = u^+ = \begin{cases} u, & u > 0 \\ 0, & u \leq 0 \end{cases} \tag{4}$$

To maintain the characteristic features of competition ,four conditions are imposed for thresholds h_1,h_2 and weight w_1,w_2 as

$$h_1 = s_{min} > 0 \tag{5}$$

$$1 > h_2 > 0 \tag{6}$$

$$w_2 > \frac{s_{max} - h_1}{1 - h_2} \tag{7}$$

$$(2 - h_2)w_2 + h_1 - s_{max} > w_1 > (1 - h_2)w_2 + h_1 \tag{8}$$

Amari and Arbib concluded that these conditions produced only one stable equilibrium element with maximum stimulus as a winner and the others as losers.The eminent thing is that an element will continue always excited once it becomes a winner, even if its stimulus becomes lower than the other stimuli[1].

3 Problems in food seaching processes

Consider a bee which leaves its nest,flies to flowers to take honey and comes back to its nest after obtaining enough honey.When it begins to look for flower ,it will decide to go to the most beautiful flower which will have a good honey. The neural nets in a bee processes the input stimuli of many flowers and the neural network with competition chooses a flower with maximum stimulus input. Then,the bee decides to fly to it. Now consider the selected flower is covered with a net and the bee cannot reach to it.The primitive neural nets equation tells that the flower continues to be a winner even if its stimulus becomes zero.The bee will never stop trying to get to the flower. It will eventually be tired and die. Thus ,we need a modification for the equation if we want to help the bee. We introduce a threshold function ct_i and write the equation as

$$\dot{u}_i = -u_i + w_1 f(u_i) - w_2 g(v) - (h_1 + ct_i) + s_i \tag{9}$$

,where t_i is the number of collisions against the net and c is a constant . By this modification, we show the simulation in Fig.1 that the first selected flower(F4) loses its potential every time it strikes against the net and that finally the next flower (F3) with the second strongest stimulus is selected.

Next let us consider the process that the bee comes back to its nest, after eating enough honey.In this case,we introduce another threshold function as

$$h_1 \rightarrow h_1 + ct_i + M \tag{10}$$

,where M denotes the total weight of honey it has eaten.However,it is noted that we do not introduce M for the equation corresponding to the nest neural element. The result is shown in

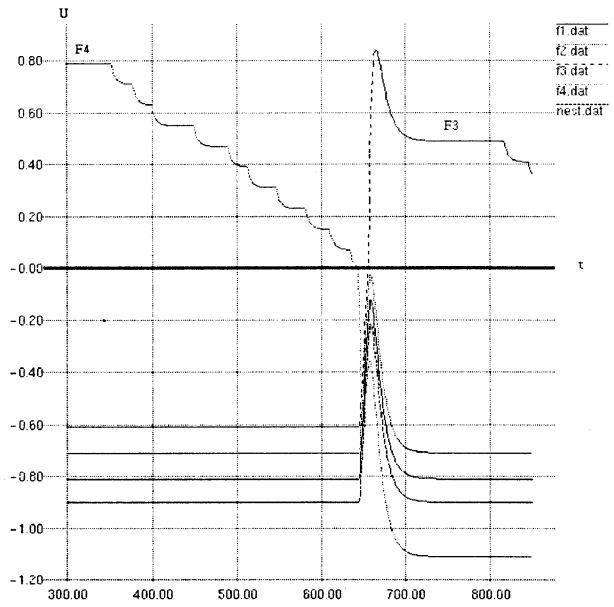


Figure 1: The aimed flower is changed from F4 to F3 after many collisions to a net surrounding F4.

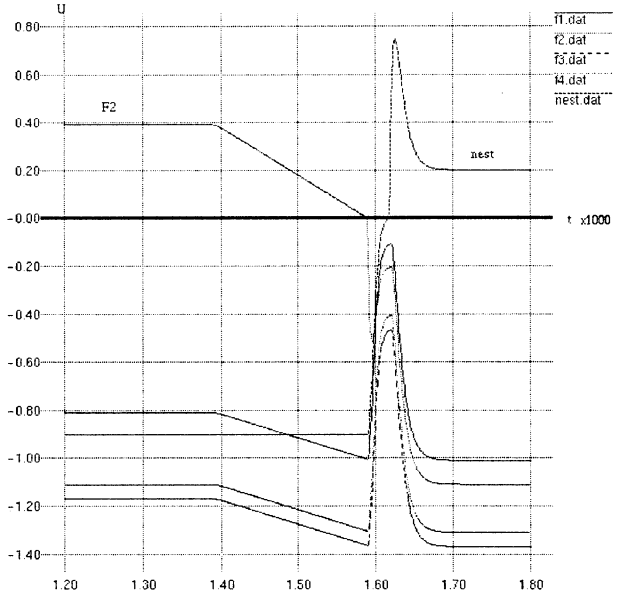


Figure 2: Nest is selected after bee's stomach is fulfilled with food.

Fig2 that the nest is selected after its stomach is fulfilled.

4 Boss and boss change processes

We consider a process to select a boss as an example of group formation among creatures(called as monkey hereafter).A group of monkeys is formed around a boss monkey after the boss is decided.We suppose that each monkey has its own neural nets and receives input from other monkeys.In this case the input are considered as influence power of monkey if we assume there exist such quantities.Each monkey processes these input in its brain,decides most powerful monkey as a boss and makes a group around the boss.Notice that a monkey has to receive his own power as a input(otherwise the most powerful monkey chooses the second one as a boss).We made a simulations by using primitive competition model.We put 100 monkeys on a lattice(500x500) at random(Fig.3). We define a communication range as a maximum distance a monkey can send or receive signal. If the range is larger,the smaller number of groups is formed.(Fig.5)Fig4 shows a group formation for range 100,assuming that monkeys approach to their bosses.

However,the group formation is suddenly destroyed when the range is increased more than 200(Fig.5). This shows the critical ability to process the competitive information and may correspond to a catastrophe.The basic competition model tells that the boss continues as a boss even if its power is weakened very much.Such a boss with very weak power must be changed to the second powerful monkey.We implement the boss changing process in the equation,considering that the boss has made many mistakes.Thus,we introduce the thresh-

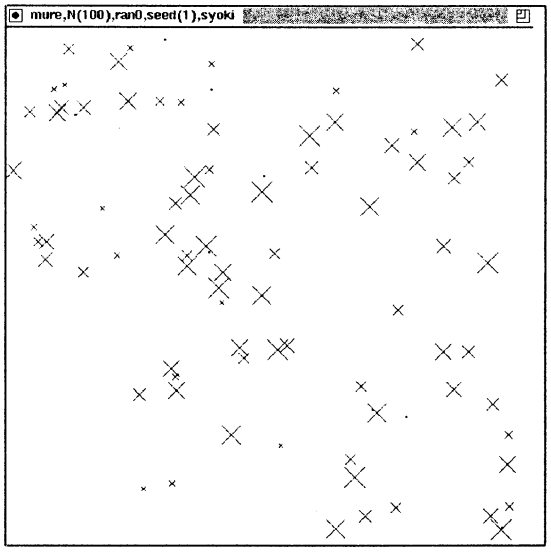


Figure 3: Initial distribution of monkeys.larger "x" shows a monkey with bigger power.

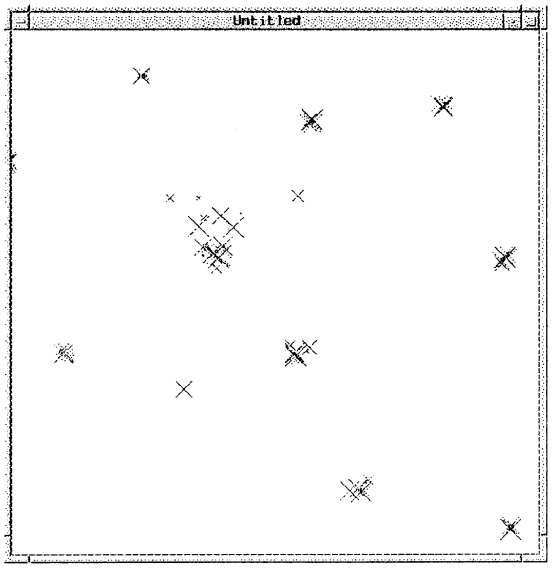


Figure 4: Group formation for communication range 100,letting monkeys approach to their bosses.

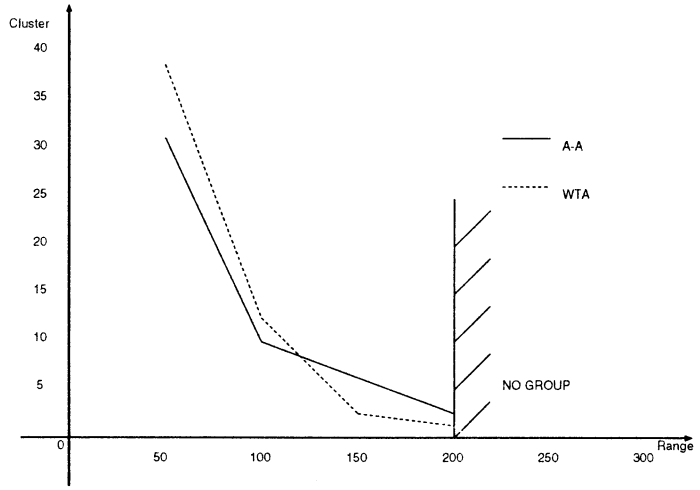


Figure 5: The range is larger,the smaller number of groups is formed.

old functions of ith stimulus

$$h_1 \rightarrow h_1 + ct_i \qquad (11)$$

,where t_i is the number of mistakes a monkey has made.Fig.6 shows a result of group formation after the primitive competition equations are applied.We assume monkeys do not move to their bosses in this case.It is noticed that there appears a kind of hierarchical structure of groups with primary and secondary ranking; The boss of secondary group belongs to the boss of the primary group. Fig.7 and 8 show features before and after the boss change.Since the former boss 47 had many mistakes, the second powerful monkey 58 in the same group becomes a boss in the group of monkeys 58,85,47,35,64,66. However,the new boss cannot become a primary group boss,belonging to the another primary group whose boss is 62.

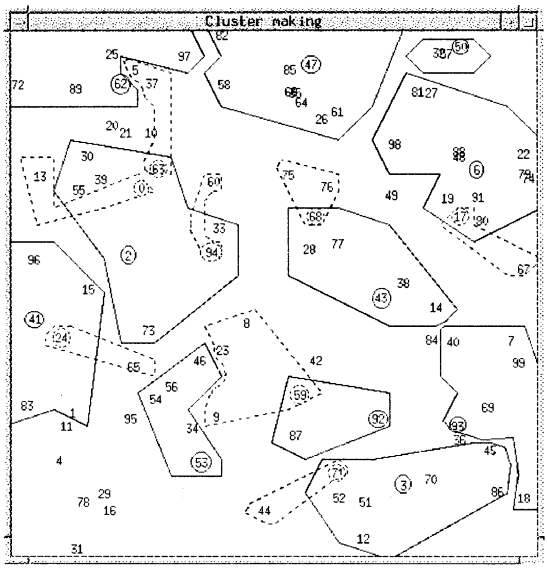


Figure 6: Group formation for range 100.Groups enclosed by a solid line show primary groups whose bosses are circled by a solid line.Dashed lines and Dashed circles show secondary groups and their bosses,respectively.

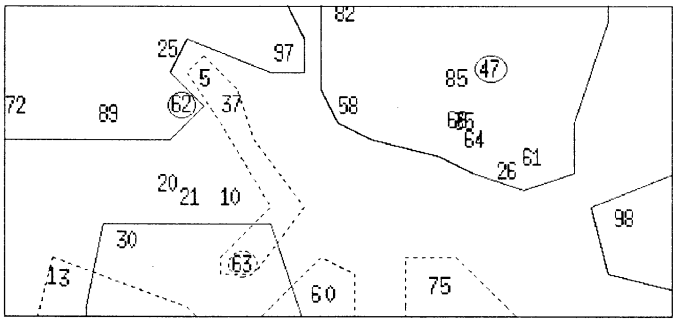


Figure 7: Group map before the boss 47 changes to 58

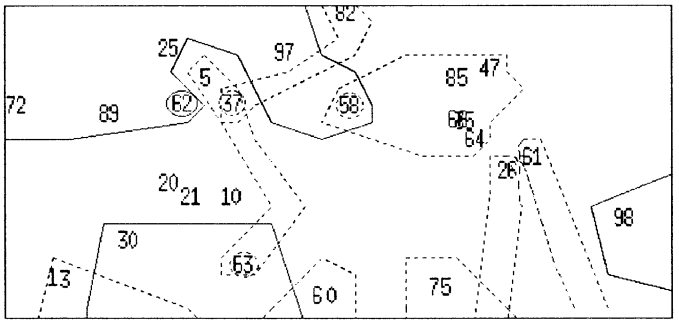


Figure 8: Group map after the boss 47 changed to 58.

5 Quit processes in WTA equation

In 1989,another WTA equation was proposed by Yuille and Grzywacz.Though it does not have a correspondence to the actual neural nets in a brain,it has the same competitive function which has smaller number of parameters and seems to work more effective than the primitive competitive model.Their equation is written as

$$\tau \dot{x}_i = -x_i + I_i K(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N) \tag{12}$$

,where x_i corresponds to the state of i-th unit and I_i is its input. $K$is written as

$$K(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N) = e^{-\lambda \sum_{j \neq i} x_j} \tag{13}$$

In this system,x_i with the largest input I_i becomes maximum and other units become zero after a competition.To implement qutting mechanism in this equation,we replace the input I_i with threshold functions as

$$I_i \rightarrow I_i - ct_i - M \tag{14}$$

,where t_i,M and c are the same introduced in the section3,4.The results of simulations for

bee-flower-nest system and for monkey group making and boss changing processes,shows essentially the same features as the ones obtained from the primitive competition models.

6 Conclusion

To do quit processes is one of the most important flexible abilities of creatures, avoiding the obstacled circumstances and preserving their own lives.We tried to implement these function in the neural net work equations. The simulations show successful results for bee-flower-nest system. The same implementation is applied for boss changing processes of monkey. We notice hierarchical structure in group formation and reorganization of group member when a boss is changed to the secondary monkey.

References

[1] S.Amari,M.A.Arbib, Competition and Cooperation in Neural Nets, *System Neuroscience*,Academic Press,119-165,1977.

[2] A.L.Yuille,N.M.Grzywacz, A Winner-Take-All Mechanism Based on Presynaptic Inhibition Feedback, *Neural Computation*,1,334-347,1989.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Reinforcement Learning of Airship Control with Adaptive State Space Segmentation

Yuka AKISATO † Keiji SUZUKI † Azuma OHUCHI †

† Research Group of Complex systems Engineerings
Graduated School of Engineering, Hokkaido University
North-13 West-8, Sapporo 060, Japan

Abstract

The purpose of this research is to acquire the adaptive control policy of an airship in a dynamic and continuous environment based on the reinforcement learning combined with evolutionary construction of the structure. In general, the state space for the reinforcement learning will become huge in this problem, because the airship has a great inertia and must sense huge information from the continuous environment to behave appropriately. To reduce the huge state space and construct the suitable segmentation of the state space, we propose the combination of the CMAC-based Q-learning and the evolutionary construction of the state space layers of it. The simulation result of the proposed method reveals the acquisition of the state space segmentation with which the airships can learn more effectively.

Keywords : reinforcement learning, state space segmentation, CMAC, GA, airship control

1 Introduction

In recently, the reinforcement learning has been receiving increased attention as a method for robot learning with little or no a priori knowledge. The leaning methods have realized the higher capability of reactive and adaptive behaviors in several applications. However, the learning capability depends on how to design the learner, e.g., selection of the sensors' information, segmentation of these sensors' information and the way of reward providing. As concerns the state space, if the segmentation of the state space is coarse, the structure will cause perceptual aliasing problem by which the learner can not discriminate the important states to accomplish the task. On the other hand, the fine segmentation will produce too many states to generalize the experiences. Although the state space segmentation is important to learn effectively, the appropriate structure of it depends on both of the embodiment of the learner and the given tasks. Thus, it is difficult to design the structure only with the experiences or insights of the designers to guarantee the optimal one for learning the control policy.

In this paper, we apply a reinforcement learning to acquiring the control policy for an airship that tries to escape from another airship as purser. The airship has great inertia and is in a dynamic and continuous environment, so that the suitable state space segmentation will become important to learn the task effectively. Based on this problem domain, we propose the combination methods of the CMAC-based Q-learning and the evolutionary construction of the state space layers of it for realizing the effective learning of the airship control.

2 Problem Domain

We consider a simulated airship control on a continuous environment. Two airships exist in the two-dimensional square room surrounded by walls as shown in Figure 1. The task for an airship as *Evader* is to evade another airship as *Pursuer* and to avoid the crash into the walls.

The environment and the structure of the airship are assumed as follows: The area of the room is assumed as 5[m] × 5[m] surrounded by the walls. In this room, it is assumed that there is no current of air in space. The airship has two propellers P_1, P_2. The distance between the airship's center gravity M_0 and each propeller is $l = 10.0$[cm]. The mass of this airship is $M = 7.0$[g] and the mass of each propeller is $M_p = 1.0$[g]. In this simulation, we assume the airplane has no volume, that is, it is a point with mass. Therefore, Evader is considered to be caught when the distance between Evader and Pursuer is within a constant $d = 30.0$[cm]. A propeller $P_i (i = 1, 2)$

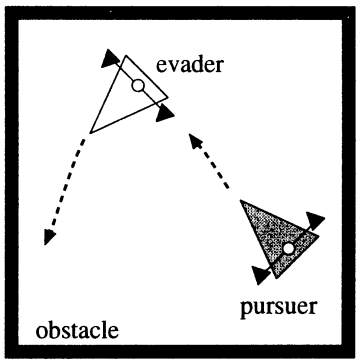


Figure 1: Airship as Evader Controlling Problem

can produce a force $f_i (0 \leq |f_i| \leq F_{max} = 17.5[\mathrm{N}])$. Driving these propellers, the airship can move in the two-dimensional space freely.

The resistance coefficients for driving force and for rotating force are $D_v = 17.0[\mathrm{kg/s}]$ and $D_w = 7000.0[\mathrm{kg/s}]$, respectively. Also there are secondary delay because of effect of inertia.

The equations of the airship's motion are

$$M\ddot{x}_G + D_v\dot{x}_G + F_x = 0, \quad (1)$$

$$M\ddot{y}_G + D_v\dot{y}_G + F_y = 0, \quad (2)$$

$$I\ddot{\theta} + D_w\dot{\theta} + \tau = 0, \quad (3)$$

where (x_G, y_G) is the airship's coordinate on the plane, (F_x, F_y) and τ is the resultant force vector and the torque, respectively, and I is the rotate inertia moment $l_1^2 M_1 + l_2^2 M_2$. Simulations are carried out with the Runge-Kutta method of 0.002 [sec] step. The control sampling step is 1.0 [sec] step.

Each airship has six kinds of sensors, we assume, to detect the state of the airships and their environment, that is (1) the relative distance to the other, (2) the relative direction to the other, (3) the relative direction to the other orient, (4) the absolute velocity of the other, (5) the absolute velocity of itself and (6) the relative distance to the wall in the decided direction. The ranges of these sensors are [0,707.1](cm) for concerning the distances, [0, 2π](rad) for concerning the directions and [-20.0, 20.0](cm/s) for concerning the velocities.

3 CMAC based Q-Learning Algorithm

We apply a reinforcement learning to acquiring the control policy for the Evader. The airship control problem has a continuous state space with an infinite number of states. When the problem space is too large to explore completely, the learner must deal with the generalization problem. We applied *CMAC-based Q-learning* algorithm (Q-CAMC)[1], which is *Q-learning* algorithm using *CMAC* as an approach to the generalization problem.

The structure of the state space in Q-CMAC consists of the overlapped tiles. Each tiles is allowed to have the different state space segmentation. Thus the Q-values are distributed into these tiles as the Q-tables. Therefore, the Q-CMAC can generate the huge state-action values from the combination of the overlapped tiles even if each tile consists of relatively small Q-table.

Let $q_i(x, a)$ be a Q-value for taking action a in state s held by the i-th tile in the Q-CMAC, then the state-action value $Q(x, a)$ can be written as

$$Q(x, a) = \frac{1}{m}\sum_{i=1}^{m} q_i(x, a) \quad (4)$$

where m is a number of tiles. We suppose that the force produced by each propeller has three states, F_{max}, 0 or $-F_{max}$. Accordingly, the number of actions in this Q-CMAC is $3 \times 3 = 9$.

When the airship takes the action a from state x, it receives the reward r, and then it moves to the next state y. In this sequence, $q_i(x, a)$ $(i = 1, ..., m)$ is updated as follows:

$$q_i(x, a) \leftarrow q_i(x, a) - \alpha e, \quad (5)$$

$$e = Q(x, a) - (r + \gamma \max_{b \in actions} Q(y, b)) \quad (6)$$

where $\alpha (0 < \alpha < 1)$ is the learning rate and $\gamma (0 \leq \gamma < 1)$ is the discount parameter.

In this problem, the Evader will receive the negative reward p_e when captured by the Pursuer or collides against walls. The positive reward r_{step} is given to the Evader for every step while it succeeds to evade Pursuer and to avoid the walls.

Given a current state x and an available action a_i, the airship selects each action a with the probability determined by the Boltzmann distribution:

$$p(a_i|x) = \frac{e^{Q(x,a_i)/T}}{\sum_{k \in actions} e^{Q(x,a_k)/T}} \quad (7)$$

where T is the temperature parameter.

Each run consits of t_t trials. Each trial begins with both airships located randomly in the room. The trials are terminated when the Evader is captured, either of airship collide against wall, or time step is over t_o.

4 Evolutional Segmentation

The state space segmentation is one of what the learning capability depends on. It is necessary to segment the sensor space suitable for learning to perform the given task efficiently. We attempt to use the *genetic algorithm* (GA) to construct the appropriate sensor space segmentations in the Q-CMAC. That is, the GA should be found the number of divisions and dividing points for each sensor information in each tiles.

In search progress, we make the GA search the segmentation of one tile per several generations called a *period.* When a period end, the tile with the best fitness is added to the Q-CMAC. Therefore, according to the repeating of the periods, the number of tiles segmented by the GA is increased.

To realize the evolutional tile segmentations for the Q-CMAC, specific type of chromosomes is introduced. The length of the chromosome is variable and the order of the genes included in the chromosomes is arbitrary. The genes are composed of two parts. One part is a *type* which is a sort of the sensor information and another part is a real *value* which represents the dividing point within the range of its sensor information. Thus, if a chromosome includes n genes concerned with t type, the tile decoded from the chromosome is divided the range of the t type sensor into $(n+1)$ regions in accordance with the values in these genes. If the chromosome doesn't include the genes concerned with a certain type of the sensor, there is no division $(n=0)$ in the range of the sensor information. The meaning of the no division is that the sensor information isn't perceptible in the tile. In addition, if the better tiles found by the GA has no divisions concerning the type t sensor, it seems that the sensor information, type t, isn't required for achieving the given task. Hence, it can be expected that the GA can not only find the better state space segmentations for the reinforcement learning but also exclude the useless sensor information for the learning task.

For example, if a tile has two divisions a_1 and a_2 for the sensor type α, three divisions b_1, b_2 and b_3 for the sensor type β, and no divisions for the sensor type γ, the chromosome representing the tile is as follows:

$$\text{Chromosome } A : (\beta, b_1)(\alpha, a_2)(\beta, b_3)(\beta, b_2)(\alpha, a_1)$$

where (t, v) denotes the gene consisted of t type and the value v.

An initial population is constructed at random but that the string length is less than a constant lm_{init}.

The fitness of the chromosome is evaluated with the performance of the Q-CMAC of which the tile is segmented by the chromosome and the tiles are decided until last period. The fitness value is calculated from the average number of step for last t_e trials in a run as follows:

$$fitness = \frac{1}{t_e} \sum_{i=t_t-t_e}^{t_t} s_i \tag{8}$$

where s_i denote the number of steps in i-th trial. In other words, equation (8) means the average number of steps Evader runs without the captured by the Purser and the collisions with the walls.

The offspring is produced by the following genetic operations.

1. **Selection**
   One is selected with the elitist strategy and the others are selected with the roulette strategy.

2. **Cross-over**
   The alternative of the two cross-over operations is applied to these offspring with cross-over rate p_c.

   - Recombination:
     Each of the parents' chromosomes are combined to form the offspring's in arbitrary one-point crossover.
   - Value Cross-over:
     Two genes including same type are selected arbitrarily from the parents' chromosomes. Let the values of these genes be assumed as v_1,v_2 $(v_1 < v_2)$. From these genes, two new genes including the same type and having the values u_1, u_2 are generated. If the range of this type sensor is $[L_l, L_u]$, the two values u_1, u_2 are calculated as follows:

     $$u_1 = \frac{v_1 + v_2}{2}, \tag{9}$$

     $$u_2 = \begin{cases} \frac{L_l+v_1}{2} & (u_1 < \frac{L_l+L_u}{2}) \\ \frac{v_2+L_u}{2} & (u_1 > \frac{L_l+L_u}{2}) \end{cases} . \tag{10}$$

     The new two genes are put back at the positions where the old genes were. The operation is done once per every type of genes.

3. **Mutation**
   The type and the value of the gene are arbitrarily changed with mutation rate p_c.

4. **Reduction**
   If the Q-table derived from the chromosome can not allocated a memory, the gene selected from the chromosome arbitrary is deleted.

5 Experiment

To confirm the efficiency of the proposed method, we carry out the experiment of the Evader learning. In the Evader learning, 5 tiles of the Q-CMAC constructed with the GA. The periods for constructing the tiles were given in order as 5, 10, 10, 15 and 15. The rest of GA parameters were determined such as the size of population is 30, $p_c = 0.8$, $p_m = 0.03$, $lm_{init} = 20$. The learning parameters in (5),(6) and (7) were determined such as $\alpha = 0.3, \gamma = 0.9, T = 0.05$. The rewards were assigned as $p_e = -5.0, r_{step} = 0.1$. The parameters concerning the number of trials were $t_t = 150, t_n = 50, t_o = 500$. The Pursuer has trained to pursue the Evader by means of the co-learning with 30000 trials. In this experiment, The trained Pursuer didn't learn while the Evader was learning in this experiment. Each run began with the Q-tables of the Evader initialized to 0.

Figure 5 shows the evolutionary progress of the best and the average fitness of chromosomes. It indicates that the combination the GA with the Q-CMAC can perform the efficient learning ability. Figure 3 shows

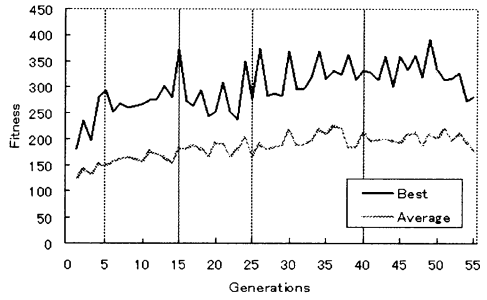


Figure 2: the fitness

the total segmentations in 5 tiles which constructed by the best chromosomes in all periods. As for the sensor of the distance between the Pursuer, the segmentations are increased and become fine according to the Purser closing. Therefore, the Evader will becomes sensitive to approach of the Pursuer. The similar tendency can be observed with the sensing of the distance between the walls.

We examined the strategy of the Evader with these best segmentations. Figure 4 shows the Evader's trajectories in term of the distance to the Pursuer and the distance to the nearest wall. The results of the trajectories were of last 10 trials among 150 trials. In the successful trials, the Evader tends to stay at the same position while the Pursuer is far from the Evader. If the Pursuer approaches to the Evader, the Evader gradually will move toward the wall to keep

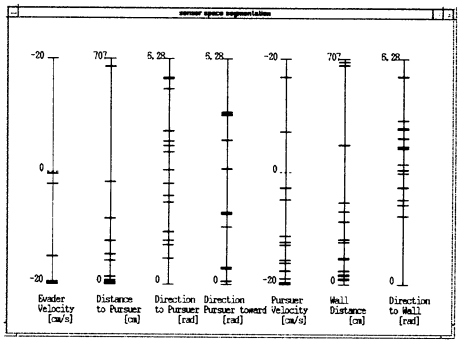


Figure 3: the state space segmentation

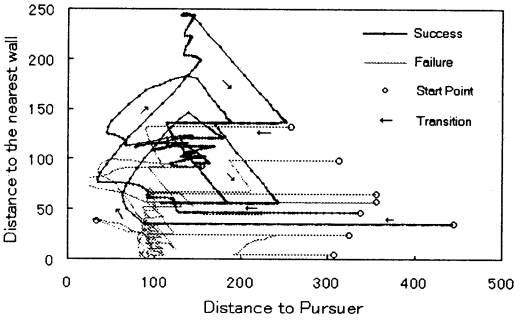


Figure 4: The trajectories in term of the distance to Pursuer and the distance to the nearest wall

the distance between the Pursuer. Apparently, the sensitivities concerned with those distances make the feasibility to perform the Evader's strategy. In addition, the sensitivities are acquired by the evolutionary segmentations.

6 Conclusion

In this paper, in order to acquire the adaptive control policy of an airship in continuous environments, we proposed the combination of the GA and the CMAC-based Q-learning. In the proposed method, the GA can construct the structure of the Q-CMAC, that is, the segmentations of the sensor spaces. Through the experiment, we realized the autonomous control of the airship effectively for evading the Purser in the problem domain.

References

[1] Ono N,Fukuta Y(1996)Learning to Coordinate in a Continuous Environment. ICMAS'96 Workshop Notes on Learning, Interactions and Organizations in Multiagent Environment, Kyoto Japan, Dec.10.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Application of Genetic Algorithm on Designing Nonlinear Model Following Control Systems

Weizheng Liu Shigenori Okubo Tianbing Li
Faculty of Engineering, Yamagata University
4-3-16 Jonan, Yonezawa, 992 Japan

Abstract

Using the nonlinear state feedback control, many plants which do not satisfy the inner product condition superficially can be modified to satisfy the inner-product condition, thus can be dealt with by Okubo's method. This paper presents the related proofs and simulations using genetic algorithm.

Keywords: nonlinear control , genetic algorithm , positive real , inner product , global boundedness

1 Introduction

A design method for nonlinear model following control systems, which guaranteed bounded internal states, was proposed by Okubo [1] for a family of plants with separable linear and nonlinear parts, but the requirement for positive-real(PR) and inner-product conditions for the nonlinear part limits the method's application even though Okubo improved the method by using the nonlinear characteristics actively for the plants with zero-unstable linear parts [2]. In this paper, we present a new approach for Okubo's method by extending the state feedback control for linear systems to nonlinear systems, and relaxes greatly the requirement for PR and inner product conditions for the nonlinear parts . By using the nonlinear state feedback, many plants which do not satisfy the inner product condition superficially can be modified to satisfy the inner product condition, thus can be dealt with by Okubo's method. Furthermore, a nonsingular matrix R is brought in to eliminate the requirement for PR condition successfully.

A genetic algorithm is used to create the nonlinear state feedback vector, and results the approach suits the nonlinear plants which can be expressed by the Kronecker vector with any highest degree definitely.

2 The Design of Nonlinear Model Following Control Systems

The plants in this paper are MIMO nonlinear systems, described by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + B\mathbf{u}(t) + \mathbf{d}(t) \quad (1)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + \mathbf{d}_0(t), \quad (2)$$

where $\mathbf{x}(t) \in R^n$ is the internal state vector whose elements are available, $\mathbf{u}(t) \in R^l$ is the control vector, $\mathbf{y}(t) \in R^l$ is the output vector, $\mathbf{d}(t) \in R^n$ and $\mathbf{d}_0(t) \in R^l$ are bounded linear disturbances, $\mathbf{f}(\mathbf{x}(t)) \in R^n$ is the known vector of the nonlinear part. $\mathbf{f}(\mathbf{x}(t))$ is assumed to satisfy

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{f}_1(\mathbf{x}(t)) + A_{[1,2N-1]}\mathbf{x}^{[2N-1]}(t) \quad (3)$$
$$\|\mathbf{f}_1(\mathbf{x}(t))\| \leqq \alpha_1 + \beta_1\|\mathbf{x}(t)\|^{\gamma_1}, \quad (4)$$

with constants $\alpha_1, \beta_1, \gamma_1$, where $\alpha_1 \geqq 0, \beta_1 \geqq 0$,and $0 \leqq \gamma_1 < 2N-1$ (N is a natural number and $N > 1$); $\mathbf{x}^{[2N-1]}(t)$ is the state vector to the Kronecker power of $(2N-1)$, $A_{[1,2N-1]}$ is a tensor matrix consisting of coefficients of the highest power in $\mathbf{x}(t)$ of the nonlinear part $\mathbf{f}(\mathbf{x}(t))$, and $\mathbf{f}_1(\mathbf{x}(t))$ is the remain part. $\|\cdot\|$ denotes the Euclidean norm. $\mathbf{d}(t), \mathbf{d}_0(t)$ can satisfy (5) if a known scalar characteristic polynomial $D_d(p)$ is selected to satisfy

$$D_d(p)\mathbf{d}(t) = \mathbf{0} \quad D_d(p)\mathbf{d}_0(t) = \mathbf{0}. \quad (5)$$

The following model is given by

$$D_m(p)\mathbf{y}_m(t) = N_m(p)\mathbf{r}_m(t), \quad (6)$$

with $\mathbf{y}_m(t) \in R^l, \mathbf{r}_m(t) \in R^{l_m}$, where $D_m(p)$ is a monic diagonal matrix given by

$$D_m(p) = diag[D_{m_k}(p)]; (l \times l), \partial D_{m_k}(p) = n_{m_k}, \quad (7)$$

here, numerator $N_m(p)$ is a $l \times l_m$ polynomial matrix with row degree $\partial_{r_k}(N_m(p)) = \sigma_{m_k}$. $\partial(\cdot)$ denotes the

degree of polynomial $(\cdot)$, and $\partial_{r_k}(\cdot)$ row k's degree of polynomial matrix $(\cdot)$.

The output error equation is

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{y}_m(t). \tag{8}$$

We first define $\mathbf{g}(\mathbf{x}(t))$ in (9) with a positive constant $\lambda > 0$, and divide $\mathbf{u}(t)$ into $\mathbf{u}_1(t)$ and $\mathbf{u}_2(t)$ in (10)

$$\mathbf{g}(\mathbf{x}(t)) = \lambda \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t)) \tag{9}$$

$$\mathbf{u}(t) = \mathbf{u}_1(t) + \mathbf{u}_2(t). \tag{10}$$

So, (1)(2) can be rewritten as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= -\lambda \mathbf{x}(t) + \mathbf{g}(\mathbf{x}(t)) \\ &\quad + B\mathbf{u}_1(t) + B\mathbf{u}_2(t) + \mathbf{d}(t) \qquad (11) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + \mathbf{d}_0(t). \qquad (12) \end{aligned}$$

Then, we use $\mathbf{u}_1(t)$ for modifying the nonlinearity of the plants and $\mathbf{u}_2(t)$ for the model following control respectively.

$$\mathbf{u}_1(t) = K_{[1,2N-1]}\mathbf{x}^{[2N-1]}(t), \tag{13}$$

where $\mathbf{x}^{[2N-1]}(t)$ is the state vector $\mathbf{x}(t)$ to the Kronecker power of $(2N-1)$, and $K_{[1,2N-1]}$ is the tensor coefficient matrix which relates to

$$\begin{aligned} &\mathbf{x}^T(t)R[A_{[1,2N-1]} + BK_{[1,2N-1]}]\mathbf{x}^{[2N-1]}(t) \\ &\quad = -\mathbf{x}^{T[N]}(t)Q_{[N,N]}\mathbf{x}^{[N]}(t), \qquad (14) \end{aligned}$$

here, R is a nonsingular $n \times n$ parameter matrix which is selected such that $Q_{[N,N]}$ becomes a positive definite symmetric matrix easily under the requirement of PR in Theorem 1. But it is difficult to get definitely the analytic solution of $K_{[1,2N-1]}$ all time from (14). So, a genetic algorithm(GA) is used to find a solution of $K_{[1,2N-1]}$. By removing the duplicate elements from $\mathbf{x}^{[2N-1]}(t)$, the compacted vector $\mathbf{x}^{<2N-1>}(t)$ is available. If, for example, $n = 2$, and $N = 2$, $\mathbf{x}^{[2N-1]}(t)$ is compacted to $\mathbf{x}^{<2N-1>}(t)$ as shown in follows,

$$\begin{aligned} \mathbf{x}^{[2N-1]}(t) &= [x_1^3, x_1^2x_2, x_1^2x_2, x_1x_2^2, \\ &\quad x_1^2x_2, x_1x_2^2, x_1x_2^2, x_2^3]^T, \qquad (15) \\ \mathbf{x}^{<2N-1>}(t) &= [x_1^3, x_1^2x_2, x_1x_2^2, x_2^3]^T, \qquad (16) \end{aligned}$$

and there exits a matrix $A_{<1,2N-1>}$ satisfying

$$A_{[1,2N-1]}\mathbf{x}^{[2N-1]}(t) = A_{<1,2N-1>}\mathbf{x}^{<2N-1>}(t). \tag{17}$$

Simplify and rewite (14) as

$$\begin{aligned} &\mathbf{x}^T(t)R[A_{<1,2N-1>} + BK_{<1,2N-1>}]\mathbf{x}(t)^{<2N-1>} \\ &\quad = -\mathbf{x}^{T<N>}(t) < \{Q_{<N,N>}\} > \mathbf{x}^{<N>}(t). \qquad (18) \end{aligned}$$

Coding $K_{<1,2N-1>}$ and R randomly to generate initial population of trials first, then evaluating and storing the performance of each trial by the objective function of $f_Q = \lambda_{min}(< \{Q_{<N,N>}\} >)$, where $\lambda_{min}(\cdot)$ is the minimum eigenvalue of the matrix $Q_{<N,N>}$, and $< \{Q_{<N,N>}\} >$ is the compacted form of $Q_{<N,N>}$, the genetic algorithm is realized to search a solution of $K_{<1,2N-1>}$ satisfying (18). If $f_Q \leq 0$, that is to say $Q_{<N,N>}$ is not a positive definite matrix, the $K_{<1,2N-1>}$ is not a solution of (18), so some trails are selected from $K_{<1,2N-1>}, R$ code set using a probability distribution criterion. One or more genetic operators such like crossover, mutation is applied to the selected trials to produce new offspring. The next generation is formed by refreshing the $K_{<1,2N-1>}, R$ code set with the new offspring. The GA process is terminated after a $K_{<1,2N-1>}$ is found to satisfy $f_Q > 0$.

$$\begin{aligned} &\mathbf{h}(\mathbf{x}(t)) = \mathbf{g}(\mathbf{x}(t)) + B\mathbf{u}_1(t) \\ &= \mathbf{g}(\mathbf{x}(t)) + BK_{<1,2N-1>}\mathbf{x}^{<2N-1>}(t). \qquad (19) \end{aligned}$$

Then, (11) can be rewritten as

$$\dot{\mathbf{x}}(t) = -\lambda \mathbf{x}(t) + \mathbf{h}(\mathbf{x}(t)) + B\mathbf{u}_2(t) + \mathbf{d}(t), \tag{20}$$

so, (21) is available by combining (2) with (20)

$$\begin{aligned} (p+\lambda)\mathbf{y}(t) &= C\mathbf{h}(\mathbf{x}(t)) + CB\mathbf{u}_2(t) \\ &\quad + C\mathbf{d}(t) + (p+\lambda)\mathbf{d}_0(t). \qquad (21) \end{aligned}$$

Define $D(p), N_r, \mathbf{w}(t)$ as

$$D(p) = (p+\lambda)I \tag{22}$$

$$N_r = CB \tag{23}$$

$$\mathbf{w}(t) = C\mathbf{d}(t) + (p+\lambda)\mathbf{d}_0(t), \tag{24}$$

with the unit matrix I , and the nonsingular matrix N_r , then (21) becomes

$$D(p)\mathbf{y}(t) = C\mathbf{h}(\mathbf{x}(t)) + N_r\mathbf{u}_2(t) + \mathbf{w}(t). \tag{25}$$

Let $T(p)$ be a diagonal stable polynomial matrix in p

$$T(p) = diag(T_k(p)), \tag{26}$$

with $T(p) \in R^{(l \times l)}$, $\partial T_k(p) = \rho_k$, $\partial D_d(p) = n_d$, and

$$\rho_k = n_d - n_{m_k} + 1 \geqq 0, \tag{27}$$

where n_{m_k} is selectable. Divide $T(p)D_m(p)$ by $D_d(p)$, let $D(p)$ be the quotient and $S(p)$ the remainder, i.e.

$$T(p)D_m(p) = D_d(p)D(p) + S(p), \tag{28}$$

then $D(p)$ is also a diagonal polynomial matrix.

In order to construct $\mathbf{u}_2(t)$, we let $Q(p)$ be a monic and proper stable transfer function from $\mathbf{u}_2(t)$ to $\mathbf{y}(t)$, namely,

$$Q(p) = diag(Q_k(p)), \tag{29}$$

with $Q(p) \in R^{l\times l}$, $\partial Q_k(p) = n_d$. So, from (6),(8),(25),and (28), we can get easily

$$\begin{aligned} T(p)D_m(p)\mathbf{e}(t) = D_d(p)D(p)\mathbf{y}(t) + S(p)\mathbf{y}(t) \\ -T(p)N_m(p)\mathbf{r}_m(t) \end{aligned} \tag{30}$$

$$\begin{aligned} \mathbf{e}(t) &= \frac{1}{T(p)D_m(p)}\{(D_d(p) - Q(p))N_r\mathbf{u}_2(t) \\ &+Q(p)N_r\mathbf{u}_2(t) + D_d(p)C\mathbf{h}(\mathbf{x}(t)) \\ &+S(p)\mathbf{y}(t) - T(p)N_m(p)\mathbf{r}_m(t)\}, \end{aligned} \tag{31}$$

since $|N_r| \neq 0$, $\mathbf{u}_2(t)$ can be obtained by letting the right-hand side of (31) be equal to zero,

$$\begin{aligned} \mathbf{u}_2(t) &= -N_r^{-1}Q(p)^{-1}(D_d(p) - Q(p))N_r\mathbf{u}_2(t) \\ &-N_r^{-1}Q(p)^{-1}D_d(p)C\mathbf{h}(\mathbf{x}(t)) \\ &-N_r^{-1}Q(p)^{-1}S(p)\mathbf{y}(t) \\ &+N_r^{-1}Q(p)^{-1}T(p)N_m(p)\mathbf{r}_m(t). \end{aligned} \tag{32}$$

Where σ_{m_k} is selected by the condition of

$$n_{m_k} - \sigma_{m_k} \geqq 1, \tag{33}$$

in order to make the transfer function of $\mathbf{u}_2(t)$ be proper.

$\mathbf{u}_2(t)$ can be realized to make the output error equal zero, but we must prove that the internal states are bounded.

Theorem 1 *For a system with a input matrix B of full rank, a nonsingular matrix R exists and satisfies the following PR condition*

$$\begin{aligned} &\{I - B(CB)^{-1}C\}R^{-1} \\ &= {R^{-1}}^T\{I - C^T(B^TC^T)^{-1}B^T\} \geqq 0. \end{aligned} \tag{34}$$

(proof omitted)

R can be obtained by the GA, thus the PR condition in Okubo[1),(2)] is eliminated successfully.

3 The Boundedness of the Internal States

In order to prove the boundedness of the internal states, we define several filters,

$$\dot{\xi}_1(t) = F_1\xi_1(t) + G_1\mathbf{u}_2(t) \tag{35}$$

$$\dot{\xi}_2(t) = F_2\xi_2(t) + G_2\mathbf{y}(t) \tag{36}$$

$$\dot{\xi}_3(t) = F_3\xi_3(t) + G_3\mathbf{h}(\mathbf{x}(t)), \tag{37}$$

and rewrite $\mathbf{u}_2(t)$ as

$$\begin{aligned} \mathbf{u}_2(t) &= -H_1\xi_1(t) - E_2\mathbf{y}(t) - H_2\xi_2 \\ &- E_3\mathbf{h}(\mathbf{x}(t)) - H_3\xi_3 + \mathbf{u}_m. \end{aligned} \tag{38}$$

Note that there is connections between the polynomial matrices and the system matrices,

$$\begin{aligned} H_1(pI - F_1)^{-1}G_1 &= N_r^{-1}Q(p)^{-1}(D_d(p) \\ &- Q(p))N_r \end{aligned} \tag{39}$$

$$E_2 + H_2(pI - F_2)^{-1}G_2 = N_r^{-1}Q(p)^{-1}S(p) \tag{40}$$

$$E_3 + H_3(pI - F_3)^{-1}G_3 = N_r^{-1}Q(p)^{-1}D_d(p)C, \tag{41}$$

where $|pI - F_i| = |Q(p)| \quad (i = 1, 2, 3)$.

Since

$$\mathbf{u}_m(t) = N_r^{-1}Q(p)^{-1}T(p)N_m(p)\mathbf{r}_m(t), \tag{42}$$

thus the system can be realized in state-space. Consequently, denote a vector $\mathbf{z}(t)$ as

$$\mathbf{z}(t) = \left[\mathbf{x}^T(t), \xi_1^T(t), \xi_2^T(t), \xi_3^T(t)\right]^T. \tag{43}$$

So, the closed system can be rewritten as

$$\dot{\mathbf{z}}(t) = A_s\mathbf{z}(t) + B_s\mathbf{h}(\mathbf{x}(t)) + \mathbf{d}_s(t) \tag{44}$$

$$\mathbf{x}(t) = C_s\mathbf{z}(t), \tag{45}$$

where A_s, B_s, d_s are constant matrices,and $\mathbf{d}_s(t)$ is bounded. Thus, the internal states are bounded if we can prove $Z(t)$ is bounded. The characteristic polynomial of A_s is [2)]

$$|pI - A_s| = |T(p)||D_m(p)||Q(p)|^2(p + \lambda)^{n-l}, \tag{46}$$

with stable polynomials of $(p+\lambda)$, $|T(p)|$, $|D_m(p)|$, and $|Q(p)|$.Therefore, A_s is also a stable system matrix.

$$\mathbf{x}(t) = H(p)R\mathbf{h}(\mathbf{x}(t)) + \bar{\bar{d}}_x(t), \tag{47}$$

here, [2)]

$$H(p) = \frac{1}{(p + \lambda)}[I - B(CB)^{-1}C]R^{-1}. \tag{48}$$

$\bar{\bar{d}}_x(t)$ is bounded. From theorem 1, we can select a R to make $H(p)$ positive real[2)]. $\bar{A}$ is a stable system matrix if $(\bar{C}, \bar{A}, \bar{B})$ is a minimum realization of H(p).The state-space realization of (47) becomes

$$\dot{\bar{\mathbf{z}}}(t) = \bar{A}\bar{\mathbf{z}}(t) + \bar{B}R\mathbf{h}(\mathbf{x}(t)) \tag{49}$$

$$\mathbf{x}(t) = \bar{C}\bar{\mathbf{z}}(t) + \bar{\bar{d}}_x(t). \tag{50}$$

Since $H(p)$ is positive real, there exists a positive definite matrix P_s and a semi-positive definite matrix Q_s satisfying

$$P_s\bar{A}^T + \bar{A}P_s = -Q_s \leqq 0, \qquad P_s\bar{B} = \bar{C}^T. \quad (51)$$

Consider the following Lyapunov function

$$V(t) = \frac{1}{2}\bar{\mathbf{z}}^T(t)P_s\bar{\mathbf{z}}(t) > 0, \quad (52)$$

using (49)

$$\dot{V}(t) = -\frac{1}{2}\bar{\mathbf{z}}^T Q_s\bar{\mathbf{z}}(t) + \bar{\mathbf{z}}^T(t)P_s\bar{B}R\mathbf{h}(\mathbf{x}(t)) \quad (53)$$

We can rewrite (53) as

$$\dot{V}(t) \leqq \alpha_8 - \beta_8\|\mathbf{x}(t)\|^{2N}, \quad (54)$$

but with $\alpha_i \geqq 0 (i = 1 \sim 8)$, $\beta_i \geqq 0, (i = 1 \sim 6)$, and $\beta_i > 0, (i = 7 \sim 8)$, so we can conclude that $\mathbf{x}(t)$ and $\mathbf{z}(t)$ is bounded by the reduction to absurdity [1].

Theorem 2 *For the nonlinear systems of (1)(2), using the method in this paper to design model following control systems, all internal states are bounded if*

① $\mathbf{f}_1(\mathbf{x}(t)) \leqq \alpha_1 + \beta_1\|\mathbf{x}(t)\|^{\gamma_1}$

② $\mathbf{d}(t)$,$\mathbf{d}_0(t)$ *is bounded*

③ *there exists a* $K_{<1,2N-1>}$ *such that* $< \{Q_{<N,N>}\} >> 0$

We have eliminated the inner product condition in Okubo[2] successfully.

4 A Example of Design

Consider the plant of

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} x_1^3 - 2x_1^2x_2 - x_2^3 + x_1x_2 \\ -3x_1^2x_2 - x_1x_2^2 + x_2^3 - x_1^2 \end{bmatrix} + \begin{bmatrix} -1 \\ 2 \end{bmatrix} u(t) + \begin{bmatrix} d(t) \\ d(t) \end{bmatrix} \quad (55)$$

$$y(t) = [1 \quad -3]\,\mathbf{x}(t) + d_0(t), \quad (56)$$

with step disturbances of $d(t)$ and $d_0(t)$,and the model of

$$y_m(t) = \frac{3}{p+5}\sin(t) + 4.5, \quad (57)$$

Let the initial population of trials be 50, research domain $-15 \sim 15$, and the number of generations 500, using GA, the nonlinear state feedback vector can be obtained. The objective function f_Q becomes positive at the 9th generation.

$$u_1(t) = [1.27 \quad 1.26 \quad 0.62 \quad -0.96]\,x^{<3>} \quad (58)$$

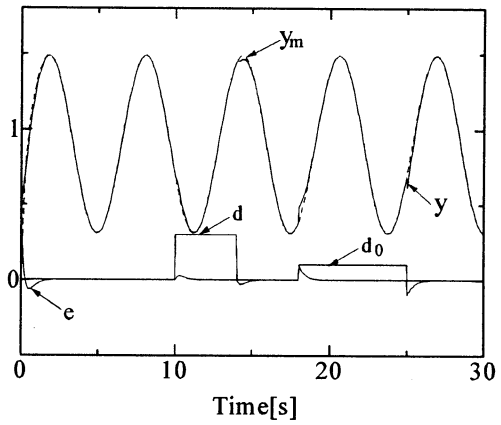


Fig.1 Output responses of nonlinear model following control system

The output response in Fig. 1 shows that the control is valid for the disturbances $\mathbf{d}(t)$ and $\mathbf{d}_0(t)$, and the output error converges to zero.

5 Conclusion

Using the nonlinear state feedback, a new approach has been presented above and extended greatly the application domain of Okubo's method. The method can be used conveniently for plants which is difficult to verify the conditions of PR and inner product. Especially, GA is brought in to simplify the design of nonlinear control systems with high nonlinear degrees.

References

[1] Shigenori Okubo," A Design of Nonlinear Model Following Control System with Disturbance", Trans. of the Society of Instrument and Control Engineers , Vol.21,No.8,pp.792-799,1985

[2] Shigenori Okubo," Nonlinear Model Following Control System Using Stable Zero Assignment", Trans. of the Society of Instrument and Control Engineers, Vol.25,No.8,pp.939-946,1992

[3] Shigenori Okubo," Globally Stable MRACS fro Multi-Input and Multi-Output Nonlinear System", Trans. of the Society of Instrument and Control Engineers, Vol.26,No.1,pp.46-53,1990

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

Application of a Model Independent Control Strategy

Ranajit Chatterjee Fumitoshi Matsuno

Department of Computational Intelligence and System Science

Tokyo Institute of Technology

4259 Nagatsuta, Midori, Yokohama, 226 Japan

Tel/Fax: (045)-924-5546 e-mail: matsuno@dis.titech.ac.jp

Abstract

The present work discusses a possible control strategy in the absence of a proper system model. The control action is based on the information of the present state only and thus independent of the structure or complexity of the system. The proposed strategy is applied on a real system and the controller performance is presented.

1. Introduction

The model dependence of the conventional controller design algorithms insists on having the complete system knowledge. But the uncertainty and nonlinearity of the real systems gives rise to complex control algorithms with associate-d implementaional difficulties [1]. Some of the approaches which try to avoid the necessity of a detail quantitative model to decide the control actions are the qualitative reasoning approach of the fuzzy knowledge based controllers [2,5] and utilization of the model free learning strategies of neural controllers [3,4]. Those approaches also have their strong points as well as limitations. Designing the fuzzy controllers need atleast some qualitative knowledge about the system behavior over the necessary operating region. The neuro-controllers also require a proper learning phase using apriori information about the system or about the performance of other controllers.

In the present discussion a quantitative approach is explored for systems where information at the present system state are available or measurable. The algorithm being independent of the system structure or complexity, it is equally applicable to the linear as well as nonlinear systems within the specified limitations. In order to demonstrate the proposed control strategy, an experiment to control the vibration of a flexible link, is considered here.

2. Problem Formulation

The dynamics of the class of system considered in the present discussion is given by,

$$dx/dt = f(x) + b(x)u \qquad (1)$$

where $x(t)$ and $u(t)$ are the state vector and input vector, and $f(x)$ and $b(x)$ are nonlinear functions with respect to $x(t)$.

For discrete computation eqn.1 may be written as

$$\Delta x/\Delta t = f(x) + b(x)u \qquad (2)$$

where, Δt is the computation step. Thus the vector corresponding to the differential change in the position vector of the state point in state space may be written as,

$$\Delta x = \Delta t f(x) + \Delta t b(x)u \qquad (3)$$

which may be re-written as,

$$\Delta x(t) = \phi(x) + \psi(x)u(t)$$

or, $$\Delta x(t) = \phi(x) + \beta(t) \qquad (4)$$

where, $\phi(x) = \Delta t f(x)$; $\psi(x) = \Delta t b(x)$

and $\beta = \Delta t b(x)u(t) = \psi(x)u(t)$

Thus, the net change in the states in time Δt could be represented as the resultant of the system component ϕ and input component β. Now if we could compute the required β from a desired Δx and ϕ at any state, the required effect might be obtained. The u could subsequently be computed from the knowledge of ψ. But the possibility is decided by the relative dimensions of the ϕ vector and u vector, and it may not always be possible to create the required vector $\beta = (\Delta x - \phi)$. In the following discussion the case of two dimensional systems with single input is considered and the possible

control actions nearest to the desired one are studied.

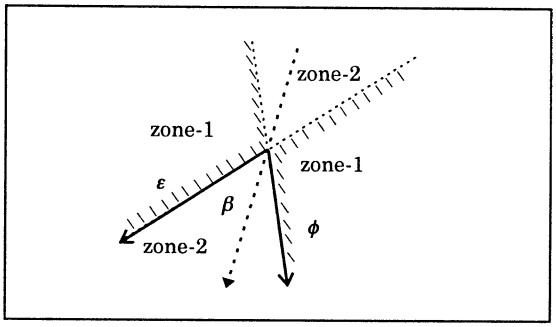


Fig.1. The two possible cases of relative position of β

The desired objective is to create a Δx vector along the error vector if possible, i.e., to approach the target state along the minimum distance. The next objective of the control is to reduce the error vector $\varepsilon=(x_d-x)$ as much possible at each step, i.e., to bring the state nearest to the desired state x_d.

3. The Control Strategy

Let us define, the state vector x = [x1 x2]^T; the error, ε = [ε1 ε2]^T; the input component, ψ = [ψ1 ψ2]^T; and the system component, ϕ = [φ1 φ2]^T.

The angles they make with the x1 axis are given by,

$\theta_\psi = \tan^{-1}(\psi 2 / \psi 1)$; $\theta_\varepsilon = \tan^{-1}(\varepsilon 2 / \varepsilon 1)$ and $\theta_\phi = \tan^{-1}(\phi 2 / \phi 1)$.

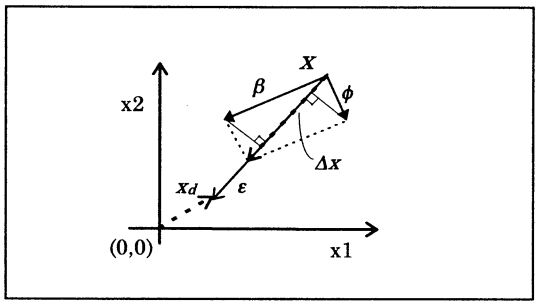


Fig.2. The zone-1 computation

Now, from fig.1 it can be seen that the relative orientation of the β vector with respect to the ε and ϕ vector leads to two different situations. While the β lies within zone-1 the Δx vector may be created along the ε direction i.e., the state may approach the target directly, as shown in fig.2. But in zone-2, since no amount of u can make the resultant along ε. The tip of the resultant vector will always lie of the B1-B2 line, as shown in fig.3. Thus the desired action could be to move the system to the point nearest to the desired state (the point P in the fig.3).

Thus computation zone may be selected as,

if sgn(sin(θ_ε-θ_ψ))=sgn(sin(θ_ϕ-θ_ψ))
then zone=1; else zone=2;

In zone 1, as shown in fig 2, we may compute the desired β as,

$\beta = u|\psi| = -(|\phi|\sin(\theta_\phi-\theta_\varepsilon) / \sin(\theta_\psi-\theta_\varepsilon))$

and, in the case of zone 2 (fig.3),

$\beta = u|\psi| = -(|\phi|\cos(\theta_\psi-\theta_\phi) - |\varepsilon|\cos(\theta_\psi-\theta_\varepsilon))$

The necessary u can subsequently be found from the knowledge of ψ at that state.

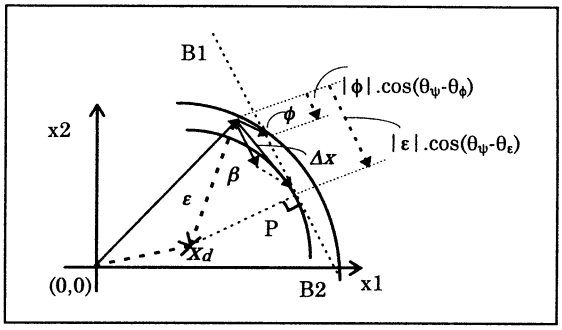


Fig.3. The zone-2 computation

3.1. Identification of the necessary parameters

As observed in the above computations (eqn.4), the calculation of u require knowledge of ϕ and ψ at any state point. The lack of availability of the system model require some experimental methods to find them. But since the algorithm only require the values at the present state thus complete system identification may be avoided.

3.1.1. Identification of $\phi(x)$ vector

The above computation considers that Δt is small enough such that f is constant over that duration. Now if a duration T=Δt/2 is selected, the f may be considered constant over both half of Δt.

Now the earlier Δt duration for control may be divided in two parts,

1. $\phi(x)$ measurement for the first T duration and
2. control for the next T duration.

As seen from eqn.4, during the first T period if u=0, $\Delta x=\phi$. In the subsequent T duration this $\phi(x)$ may be used to compute the corresponding control action. Thus the approach results in an intermittent (pulsed) control action. As the controller is used on a certain system the information about the f vector over the state-space will be available and the information could be used to avoid the measurement cycle eventually resulting in a smoother control .

3.1.2. Identification of the $\psi(x)$ vector

In the case of unknown systems the experimentation for identification of the ψ vector could be done as follows.

As can be observed from eqn.4, if ϕ can be ensured to be zero, for certain x, then, $\psi=\Delta x$ /u for a known amount of input.

But in general there may not be any basis of the above assumption. So as a general case, the system may be perturbed by a certain known input pulse. Thus the system eventually evolves under the action of the system vector ϕ only. Thus the information about the ϕ values for corresponding x could be retained along the trajectory. If the system could be reset to the same initial condition and the same input is applied, the system is expected to follow a similar trajectory. Now if a non-zero input, as shown in fig.4, is applied at a state point x_a, for which the $\phi(x_a)$ is known, the $\psi(x_a)$ could be computed from eqn.4.

In the cases where different unaccountable uncertainties prevent the repeatability of the same trajectory by the system, a number of initial perturbation experiments may be conducted and the $\phi(x_a)$ may be computed by interpolation of the available data.

3.2. Assumptions and limitations of the strategy

The applicability of the present version of the algorithm considers the following system properties and information.

1. The $f(x)$ and $b(x)$ are independent.
2. All the states of the system should be measurable.
3. The system should be time invariant (in the case of unknown system model).

4. Study of the Controller on a real System

The control strategy discussed above, is applied to control the vibration of a flexible link. The system description and the controller performance are discussed in the following sections.

4.1. Description of the system under control

A schematic diagram of the system is shown in fig.5. The flexible link is a thin rectangular stainless steel plate with length 0.6 m, width 50 mm and thickness 2mm. Rigidity modulas E= 2.06e11 Nm^2 and I=4.17e-12 m^4. The link is connected with a direct drive d.c. motor with output torque of 5.625 Nm/volt. At the base of the link two pairs of strain-gauges are used to measure the oscillation of the link.

4.2. Performance of the controller

The fig.6 shows the oscillation of the link, due a input perturbation with a pulse of 4v for a duration of 3ms, in the form of strain variation with time. Fig.7 shows the corresponding evolution of the system in state-space in the absence of any control.

The proposed controller is tried on the system to control the vibration, without using any model of the system. To control the system, the strain (s) measured at the base of the link and the time derivative of it (ds/dt) are used as the system states. The ψ vector is identified initially by a separate perturbation experiment on the system, as described in the last section. The computation interval is set to be 1ms, and the inter-control-action delay is set to a value of 8ms. The temporal and state-space response of the controlled system, under the same perturbation, are shown in fig.8 and fig.9. Fig.10 shows the controller output. As can be observed, the amplitude of the vibration of the link, under the action of the proposed controller, is reduced faster than the uncontrolled case.

5. Conclusions

The present work explores the applicability of a model independent control algorithm on a real system. The algorithm, for computing the control action, is described for two dimensional systems with single input, and the limitations of the method are mentioned. The performance of such control is presented which shows the applicability of the proposed control strategy in the absence of proper system model. The control action, thus computed may be used as the basis to tune other structured controllers, to make them adapt to unknown systems lacking a precise mathematical model. The algorithm, being system model independent, is equally applicable to linear or nonlinear systems. The strategy is easily extendible to control the multiple-input-multiple-output multi-dimensional systems as well.

6. References

[1] A. Isidori, Nonlinear Control Systems, Springer, 1995.

[2] D.Driankov, H.Hellendoorn and M. Reinfrank, An Introduction to Fuzzy Control, Springer, 1993.

[3] J.M.Zurada, Introduction to Artificial Neural Systems, Jaico Publishing House, 1995.

[4] S.Haykin, Neural Networks : a comprehensive foundation, Macmillan College Publishing Co. 1994.

[5] D.E.Thomas and B.Armstrong. Heloury, "Fuzzy Logic Control - A Taxonomy of Demonstrated Benefits", *Proc. of the IEEE*, vol.83, no.3, pp. 407-421, march 1995.

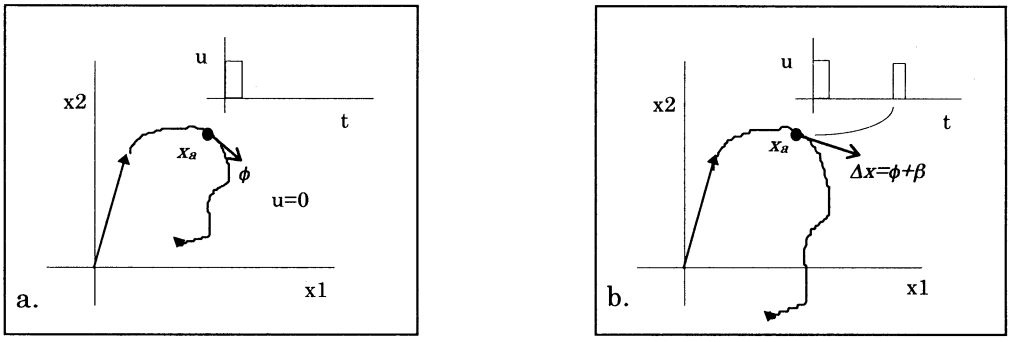


Fig.4 Experimental identification of β, (a). measuring $\phi(x_a)$, (b). measuring the resultant with known $\phi(x_a)$ and u

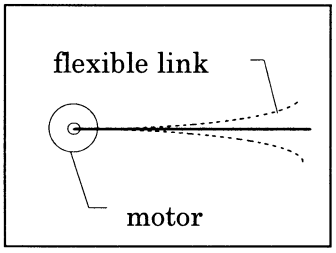


Fig.5. The system schematic

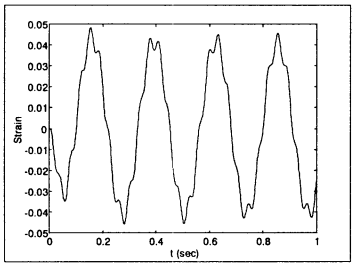


Fig.6. Strain oscillation over time

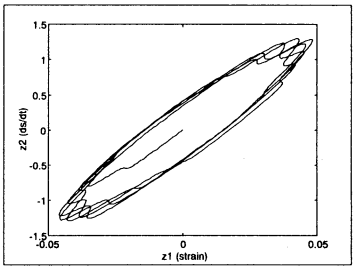


Fig.7. State-space plot of the system

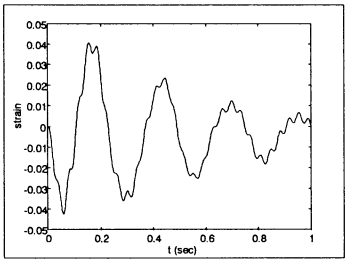


Fig.8. Strain oscillation with controller

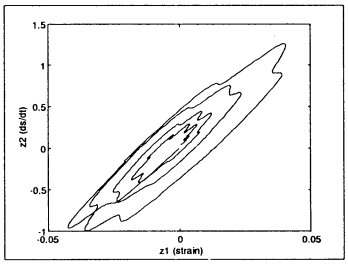


Fig.9. The state-space plot with control

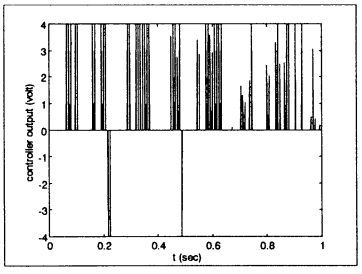


Fig.10. The controller output

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

A probabilistic Hilbert Space Approach towards understanding nonlinear dynamical systems in an Observer's endo-and exo-variables

T E Schilhabel
Electronics and Computer Science
Image, Speech and Intelligent Systems
Research Group
University of Southampton, SO17 1BJ UK

C J Harris
Electronics and Computer Science
Image, Speech and Intelligent Systems
Research Group
University of Southampton, SO17 1BJ UK

Keywords: endo- and exo-physics, continuity equation, Hilbert space, Schrödinger equation.

Abstract

This paper deals with the probabilistic description of a general *a priori* unknown linear/nonlinear dynamical physical system in an endo-observer related coordinate system. The complex relationships between endo-and exo-physics lead to a formulation of evolving probablity densities in the Hilbert space. The probability densities can be represented in form of the Schrödinger equation.

1 Physical variables – Relations among observers

This paragraph intends to make clear that there exist systems as sub-elements of the real world which are measured by experiments which include an observer who carries out this experiments and who lives inside these systems.

We are part of the real world and its objects we would like to analyze with our experiments. The definitions in the following are borrowed from Primas [2]. The *universe of discourse* splits this world up into the *observing system* and an *objects* which has to be observed. The system without information about an external observer outside the interrelations of object and observer given by experiments is an *endo-system.* That means that all our models based purely on measurements are of *endo*-character. The *exo-system* is defined by a set of necessary rules of how to relate measured entities of the endo-system with variables of models created by someone living in the exo-system. A *cut* is defined as the methatheoretical distinction which seperates an endo-system into an exo-system and an object.

We believe to live in a three-dimensional Euclidean space where objects can be related to each other by describing their spatial coordinates. Spatial coordinates are therefore *a priori* endo-variables where each hypothetical endo-observer at any point of the Euclidean space may relate his coordinate system to that of the others. The corresponding physical process is called spatial coordinate measurement. It is known that the relative coordinates between different observers remain unchanged as long as the transformation from one observer to the next one is a rigid one, *i.e.* translation and rotation. That means it is possible to define an abstract exo-physical observer with an arbitrary coordinate system where the condition of rigid displacements is fulfilled.

As it has just been demonstrated, a single observer cannot measure the complete set of exo-physical variables required in order to represent the dynamical system in state space form. The only possible solution to this dilemma is the employment of several observers who compare their measurements in order to extrapolate the desired exo-variables. The *cut* is therefore defined as the observability of exo-variables in terms of comparative measurements between endo-observers.

This section introduced the physical problems of describing general physical objects in their state space models, when some components of the state vectors are not directly observable. The concepts of exo- and endo-physics have been introduced and we have been demonstrated the difficulties of any endo-physical observer trying to relate its endo-variables to *a priori* inaccessible exo-variables. In order to overcome this problem a *cut* has been defined which allows the deduction of exo-variables by employing relative coordinates.

2 The probabilisitc representation of physical variables

Physical experiments incorporate three sources of error when a system has to be mathematically modelled: the corruption of measurements by noise due to imprecisions, the influence of noise which enters the system due to the lack of closure of the experiment against the environment, and the over-simplification of models of the underlying real physical system. The most appropriate approach may be, therefore, a probabilistic model which assigns to each coordinate in the *endo-observer's* frame of reference a probability (density) which quantifies the occurrence of a given value of that particular coordinate. It might be more desireable to specify the probability (density) of the coordinates which are related to the physical system, but this knowledge is, however, not available to us because of all the generic erroneous influences mentioned above.

In the last section it was argued that the lack of direct access to exo-variables requires the inclusion of several endo-observers in the neighbourhood of a particular endo-observer of interest to deduce indirectly the desired information by comparison of endo-variables among neighbouring observers. How this relates to the probabilistic description of exo-variables is described in section below.

The mean of any physical variable v at time t is given by

$$\langle v \rangle (t) = \int_{-\infty}^{\infty} dv \int_{-\infty}^{\infty} dr \varrho(r,t) \varrho(v|r,t) v \qquad (1)$$

which exists as long as $\varrho(r,t)\varrho(v|r,t)v$ and $\varrho(r,t)\varrho(v|r,t)$ are integrable with respect to r. Due to Fubini's and Tonelli's theorems, these conditions indeed hold. We define that $\langle v \rangle (r,t) = \int_{-\infty}^{\infty} dv \varrho(v|r,t)v$, which is the variable an exo-observer would observe at position r, when the number of experimental repetitions tends to infinity. The endo-observer at position r has to deduce the in principle measureable variable $\langle v \rangle (r,t)$ by comparison of an endo-variable with its neighbours. The independence of the spatial probability density $\varrho(r,t)$ from the velocity $\langle v \rangle (r,t)$ is evident. The crucial endo-and-exo-physical variable here is this probability densitiy which, due to the rigid displacement inavariance, is independent of the endo-observers. What is required is a representation of the probability densities which leaves its magnitudes invariant but incorporates at the same time the information about the velocity $\langle v \rangle (r,t)$ at the position $r \in \mathbb{R}$. The only nontranslational arbitrary transformation between two neighbouring endo-coordinate systems which leaves the distance constant is the rotation. Consider an endo-observer with respect to an exo-observer to be at spatial coordinate r. When the probability density $\varrho(r,t)$ is represented as a vector $\vec{\varrho}(r,t) = \vec{\varrho'}(0',t)$ with lenght $\varrho(r,t)$, the rotation of this vector between $0'$ and $0' + dr'$ may be viewed as shown in figure 1. When this endo-observer $0'$ compares his endo-variable–here an arbitrary initial argument–with the argument of the neighbour-endo-observer $0' + dr'$, the difference between both arguments $d\mathbf{U}$ represents up to a constant the desired exo-variable $\vec{\varrho}(0',t)$. The just mentioned (unitary) transformation may be derived from a general unitary operator $\mathbf{U}(r',t)$, which depends on a single continuous parameter, by employing the Taylor series. The first order term is then

$$\left.\frac{d\mathbf{U}(r',t)}{dr'}\right|_{r'=0} = i2h \langle v \rangle (r,t) \qquad (2)$$

(In this case it can be shown that $\mathbf{U}(0') = \mathbf{I}$ and $\mathbf{U}(r'_1 + r'_2) = \mathbf{U}(r'_1)\mathbf{U}(r'_2)$. The inverse operator is given by $\mathbf{U}^{-1}(r') = \mathbf{U}(-r') = \mathbf{U}^{\dagger}(r')$, with $\dagger$ denoting the adjoint operator.) Because of the rigid transformation between endo-observers, Equ. (2) may be generalized to exo-variables. Due to the properties of strongly continuous one-parameter groups mentioned above and the rigidity condition, Equ. (2) becomes immediately

$$\left.\frac{\partial \mathbf{U}(r_1 + r_2, t)}{\partial r_1}\right|_{r_1=0} = \left.\frac{d\mathbf{U}(r,t)}{dr}\right|_{r=r_2} \qquad (3)$$

$$= i2h \langle v \rangle (0,t)\mathbf{U}(r_2,t) \qquad (4)$$

with $r = r_1 + r_2$, which is a first-order differential equation with the unique solution

$$\mathbf{U}(r,t) = e^{i2h \int_0^r ds \langle v \rangle (s,t)} \qquad (5)$$

for the initial condition $\mathbf{U}(0,t) = \mathbf{I}$. Any probability density represented by a vector $\vec{\varrho}(\bullet)$ at the origin of $\mathbb{C} \times \mathbb{R}$ which is then translated to the point r, transforms into the vector

$$\vec{\varrho}(r,t) = \mathbf{U}(r,t)\vec{\varrho}(0,t). \qquad (6)$$

When a step of the size dr' is performed with respect to an arbitrary endo-observer to the neighbouring one, the transformation has to be of magnitude proportional to $\langle v \rangle (0',t)dr'$. A probability density

$\varrho(r',t)$ which is then represented as a vector in the endo-observers coordinate system $\mathbb{C} \times \mathbb{R}$ must fulfill the same relation between $\vec{\varrho}(0',t)$ and $\vec{\varrho}(r',t)$ as it does for the exo-observer's $\vec{\varrho}(r_0)$ and $\vec{\varrho}(r'+r_0,t)$ in case of an arbitrary coordinate transformation $r = r' + r_0$. This interrelation between exo- and endo-variables of spatial coordinates allows now the formulation of *a priori* endo-physical measurements in terms of exo-physical representations $\mathbf{U}(r,t)$. Instead of utilizing the probability density, the square root of $\varrho(r,t)$ may be employed, which then defines the new function*

$$\psi(r,t) = \sqrt{\varrho(r,t)} e^{ihG(r,t)} \tag{8}$$

with $G(r,t) = \int_0^r ds \langle v \rangle (s,t)$.

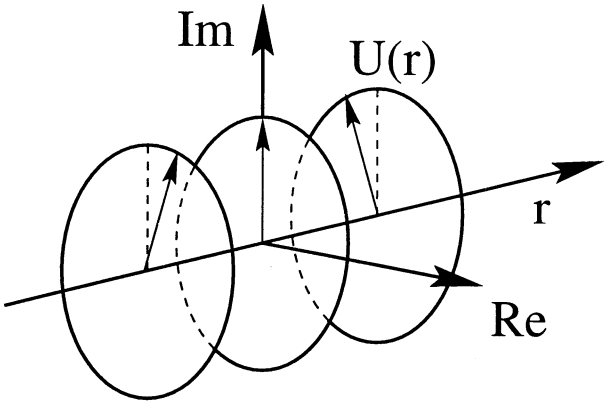


Figure 1: *The transformation of $\vec{\varrho}(0)$ to the spatial coordinate r is a pure rotation in the complex plane, as indicated by the circles.*

3 The Fourier integral

Equ. (8) suggests the introduction of a new quantity, the magnitude of a special case operator $\mathbf{U}(-r) = e^{-iGr}$ with G =const. to the representation (8) for $r \in \mathbb{R}$. The endo-physical transformation described by that $\mathbf{U}(-r)$ is the assumption as if there were just the exo-variable $\langle v \rangle$ on the entire real line, or equivalently the operator $U(r)$ then demands that the transformation between neighboured endo-observers has to be given by $U(r_2')U(-r_1') = e^{iG\Delta r'}$ with $\Delta r' = r_2' - r_1'$ the difference vector between two arbitrary endo-observers. It is the relative argument between neighbouring observers, not the absolute argument between $\vec{\varrho}(r')$ and $\vec{\varrho}(r'+r_0')$ the which determine the contribution of each dr' to the total magnitude. It can be shown that the Fourier-integral is the suitable tool for computing the contribution of each operator e^{iGr} to the wave train (8). In order to show that it does indeed, it has to be prooved that $e^{iG_1 r}$ and $e^{iG_2 r}$ are orthogonal to each other as long as $G_1 \neq G_2$, and that the magnitude of each transformed $\psi(r,t)$ is independent of any absolute vector's argument anywhere for $r \in \mathbb{R}$. The Fourier integral

$$\mathcal{F}(\psi(r,t)) = e^{i\alpha} \int_{-\infty}^{\infty} dr \psi(r,t) e^{-iGr} \tag{9}$$

won't change its magnitude for any arbitrary α, *i.e.* the change of integration variable to $dx = dr$, $\alpha = Gr_0$ and $x = r - r_0$ leaves the value of $\mathcal{F}^\star(\psi(r,t))\mathcal{F}(\psi(r,t))$ unchanged. Further, due to the "orthogonality" $\delta(G_1 - G_2) = \int dr e^{i(G_1 - G_2)r}$ is unique and allows the interpretation of $\mathcal{F}^\star(\psi(r,t))\mathcal{F}(\psi(r,t))$ as probability densities.

4 The Liouville equation

The probabilistic description for physical variables seems to be the most suitable representation of gained information of an observer about a physical system. When there is a exo-physical model for an arbitary nonlinear dynamical system, then there exists due to the uniqueness theorem of phase space models the *continuity equation* as evolution equation of probability densities $\varrho(\mathbf{x})$ due to the evolution laws $\{F_i\}$ in the deterministic equation $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$. The *continuity equation* describes the change of the probability density at particular point $\mathbf{x}$ due to the gain/loss of density through the circumscribing surface about this very point.

$$\frac{\partial \varrho(\mathbf{x})}{\partial t} + \sum_i \frac{\partial}{\partial x_i}(\varrho F_i(\mathbf{x})) = 0 \tag{10}$$

The sum in Equ. (10) is often abbreviated as $\mathrm{div}(\mathbf{j}(\mathbf{x},t))$ with the flux $\mathbf{j}(\mathbf{x},t)$.

However, problems occur when there is information lacking, in our case the endo-observer cannot

*Equ. (8) describes a *slowly varying wavetrain* which is approximately sinusoidal everywhere with changing amplitude $\sqrt{\varrho(r,t)}$, varying phase $G(r,t)$, and the local wavenumber $k(r,t)$, and local frequency $\omega(r,t)$ are defined by

$$k(r,t) = \frac{\partial G}{\partial r}(r,t), \qquad \omega(r,t) = \frac{\partial G}{\partial t}(r,t). \tag{7}$$

Although the analogies with oscillating physical systems are evident, Equ. (8) has, apart from the mathematical structure, nothing in common with those systems.

provide the probability densities of all required dimensions in the state space. A *probability flux* may still be defined as in the example stated above by utilizing

$$\mathbf{j}(\mathbf{r},t) := \varrho(\mathbf{r},t)\,\langle \mathbf{v}\rangle\,(\mathbf{r},t) \tag{11}$$

Instead of employing the the continuity equation for the state space model (exo-observer) the dimensional reduced version may be utilized

$$\frac{\partial \varrho(\mathbf{x}_r)}{\partial t} + \mathrm{div}\left(\varrho(\mathbf{x}_r)\int d^q x\, \varrho(\mathbf{x}_q|\mathbf{x}_r,t)\mathbf{x}_q\right) = 0 \tag{12}$$

which describes the probabilities in the r-dimensional subspace of the n-dimensional state space, where the information of q dimensions is not directly accessible. Due to the arguments from the section above, the integral $\langle \mathbf{x}_q\rangle\,(\mathbf{x}_r,t) = \int d^q x\, \varrho(\mathbf{x}_q|\mathbf{x}_r,t)\mathbf{x}_q$ may indirectly be measurable. In case that the integrability condition for $\langle \mathbf{x}_q\rangle\,(\mathbf{x}_r,t)$ if fulfilled with respect to all components of $\mathbf{x}_r$ and the function $<\bullet>\,(\mathbf{x}_r)$ is integrable with respect to $d\mathbf{x}_r$, then it can be shown ([4]) that there exists a potential function $G(\mathbf{x}_r,t)$, such that $\nabla_{\mathbf{x}_r} G(\mathbf{x}_r,t) = \langle \mathbf{x}_q\rangle\,(\mathbf{x}_r,t)$, and according to Equ. (8) functions in the Hilbert space be defined by

$$\psi(\mathbf{x}_r,t) = \sqrt{\varrho(\mathbf{x}_r,t)}e^{ihG(\mathbf{x}_r,t)}. \tag{13}$$

The desired quantity is the probability flux $\mathbf{j}(\mathbf{x}_r,t) = \varrho(\mathbf{x}_r,t)\nabla_{\mathbf{x}_r}G(\mathbf{x}_r,t)$ which may be computed by

$$\mathbf{j}(\mathbf{x}_r,t) = \frac{i}{2h}\left(\psi^\star(\mathbf{x}_r,t)\nabla_{\mathbf{x}_r}\psi(\mathbf{x}_r,t) - \psi(\mathbf{x}_r,t)\nabla_{\mathbf{x}_r}\psi^\star(\mathbf{x}_r,t)\right) \tag{14}$$

The divergence of the probability flux leads then to

$$\frac{\partial}{\partial t}\psi^\star(\mathbf{x}_r,t)\psi(\mathbf{x}_r,t) = \frac{i}{2h}\left(\psi^\star(\mathbf{x}_r,t)\nabla^2_{\mathbf{x}_r}\psi(\mathbf{x}_r,t) - \psi(\mathbf{x}_r,t)\nabla^2_{\mathbf{x}_r}\psi^\star(\mathbf{x}_r,t)\right) \tag{15}$$

which may, up to an arbitrary scalar function $V(\mathbf{x}_r)$, uniquely define the relation

$$\frac{\partial}{\partial t}\psi(\mathbf{x}_r,t) = \left(\frac{i}{2h}\nabla^2_{\mathbf{x}_r} + V(\mathbf{x}_r)\right)\psi(\mathbf{x}_r,t). \tag{16}$$

5 Conclusion

It has been demonstrated that the state space description of a dynamical system requires insight into an object which is *a priori* not available (exo-variables). Physical limitations of real life observers (endo-bservers) may need a different, indirect method to measure physical variables which are directly not accessible. Together with the probabilistic formulation of accessible exo-variables, our approach leads to a representation of probability amplitudes in the form of a "wavetrain" with phases which represent exo-variables in terms of required endo-variables. The Liouville or continuity equation for probabilities in the reduced state space may be transformed into the Schrödinger equation. The Fourier transform enables the extraction of spatial invariant unitary transforms, which represent invariant exo-variables in $\mathbb{R}$.

Acknowledgements

The authors would like to thank DRA Malvern and Lucas Aerospace (and all colleagues which contributed to this paper) for this work.

References

[1] L.E. Ballentine. *Quantum Mechanics.* Prentice Hall, Inc., 1990.

[2] H. Primas. Endo- and exo-theories of matter. In H. Atmanspacher and G.J. Dalenort, editors, *Inside versus Outside, Endo- and Exo-Concepts of Observation and Knowledge in Physics, Philosophy and Cognitive Science*, volume 63 of *Springer Series in Synergetics.* Springer-Verlag, Berlin, 1994.

[3] D.H. Sattinger and O.L. Weaver. *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics.* Number 61 in Applied Mathematica Sciences. Springer-Verlag, New York Inc., 1986.

[4] T.E. Schilhabel and C.J. Harris. A probabilistic framework for understanding nonlinear dynamical systems in the hilbert space. In *Proceedings of the 12th International Conference on Systems Engineering ICSE'97.* Coventry University, September 1997.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III' 98),
Oita, Japan, 19-21, January, 1998,

An Experiment Dynamic Modelling Scheme for Flexible Systems

Qiao Guo, Yue Wu and Ji-lian Lu
Robotics Research Center, Beijing Institute of Technology, Beijing, 100081, P. R. of China

Abstract

This paper proposes an experiment dynamic modelling scheme for flexible systems, especially for flexible robot manipulators. It uses the improved algorithm and structure of the ANN (Artificial Neural Networks). All of the data used in the process of modelling came from our experiments based on the very flexible link which was fixed on a FANUC Robot S-Model 300. This is why we call it the experiment modelling method. The idea and the scheme of the modelling is suitable for a general flexible system if its input and output signals can be measured. Our work includes the two main parts. One is to collect the experimental data, another one is to improve the algorithm and structure of the ANN, which was proposed by Vogl et al. The theoretical analysis and experiment results showed that this modelling scheme is more siutable for the flexible system with the characteristics of fast change dynamics, and also it can be more accurate than others and more convenient for real-time use.

1 Introduction

In the natural world, the live system like human body or animals are all flexible. They have more advantages than the mechanical system which are usually built up by the metals or some other hard materials.

In order to realize control of a flexible system and to get the desired working performance, people have to make a good model of the plant to be controlled first. Traditionally, flexible system modelling is a very complex problem. Such a system can be fully described by a group of partial differential equations (usually with high-order), or a group of partial differential and integrated equations [1]. Although this kind of models can express the distribution characteristics of a flexible system in some sense, it is only useful for the analysis of low-frequency oscillation problem of a simple shape and mass uniform-distributed link. It is too difficult to use this method to describe more complex flexible system, so that usually people have to use the reduced lower-order models in the practice. On the other hand, there are many other modelling schemes for flexible system [2], [3]. Most of them are analytic, and have many difficulties to obtain the satisfied solutions.

In this paper we propose an experiment dynamic modelling scheme for flexible systems, especially for flexible robot manipulators It uses the improved structure and algorithm of the ANN. All of the data used in the process of modelling came from the experiments based on the very flexible link which was fixed on a FANUC Robot S-Model 300. The idea and the method of the modelling is effective for a general flexible system only if its input and output signals can be measured. Our work includes the two main parts. One is to collect the experimental data, another one is to improve the algorithm and structure of the ANN which was proposed by Vogl et al [4]. The theoretical analysis and experiment results showed that this modelling scheme is more siutable for the flexible system with the characteristics of fast change dynamics, and also it can be more accurate and more convenient for real-time use.

The data collection experiment is shown in section II. The modelling ideas and scheme are described in Section III. The analysis of the modelling scheme and the comparison of the results using the different modelling methods are given in Section IV. Section V sumarizes the main contributions of the paper.

2 The Collection of the Experiment Data

In order to get the experiment data, we need to build up a data collection system, which consists of a very flexible thin copper link and the Robot Performance Test System. The link is long 700 mm, its diameter is 5 mm. It was fixed on the flange union of the FANUC Robot S-Model 300 in our Lab. We took the fixed side of the link as the input, another side as the output. There was one flat plane on the each side of the link. The two planes were parallel each other. There were the three infrared beams on the each plane. There were totally the six infrared beams on the two sides, which were used to produce the signals of the

motion trajectories of the input and output sides of the link.

On the other hand, we also needed the Robot Performance Test System (RPTS) to collect the experiment data of the flexible link. The system consists of a software and the 3D motion analysis system named OPTOTRAK which is made by the Northern Digital Corporation in Canada. The software named RPT was developed by our Research Center a few years ago. It consists of five parts, i.e. system settings, data collection, performance caculation, graph analysis, file management. According to ISO9283, this system can do the six kinds of performance tests of robot manipulators: posture, displacement, drift, stability, trajectory and turning. Each test has their own independent parameter settings, test steps and data collection ways.

Now we are ready to do the experiment of data collection. Using the software named RPTCOLL we did the nine experiments totally. Each time we set a different test number. All of the works of the data collection were down automatically.

In order to make the motion trajectories of the input of the flexible link approach the step functions as well as possible, we set the motion speed of the robot to be 2000 mm/s which is the highest speed of the robot.

3 Modelling Scheme

In consideration of the strong nonlinear characteristic of the flexible system, we selected the feedforward structure and the basic BP and the Vogl's fast algorithms [4] of the ANN. However the ANNs in existence can not fit what the task needs so we have to improve the structure and the algorithm of the ANN which was proposed by Volg et al. The modelling results show that the improved ANN is better for flexible system modelling than other kinds of ANNs.

3.1 ANN Model and Learning Algorithm

there are two main types of ANNs - the feedforward network and feedback network. The most popular ANN is the multilayered perceptron (MLP) model which is a feedforward network. Kolmogorov [4] gave the proof of the theorem, in which it was pointed out that for any given function and allowed error $\epsilon > 0$, there always exists a three-layers feedforward network, the relationship of its input and output can approach the given function with the allowed error. In the other words, a three-layers feedforward ANN can fit any function with any given errors.

The process of a feedforward ANN fitting a function is the realization of a high-dimension nonlinear mapping. It can also be seen as the process of recurrence computation and regulation. The most typical algorithm is the error back propagation algorithm, it's also named BP algorithm. Any complex feedforward mapping and the boundary of the decision can be realized by the BP algorithm and MLP. So this kind of methods have been widely used in many fields such as pattern recognition, intelligence control, signal process, coding theory and function curve fitting etc.

For the reason of the slow convergence speed of the basic BP network, Vogl et al [4] proposed a fast algorithm of the ANN with the idea of 'batch processing'.

3.2 The Improvement of the ANN's Algorithm and Structure

In the modelling process, we found out that the ANN with the basic BP algorithm can not work well because of the large quantity of computations, the slow speed of convengence, the long time to escape from the insensitive areas of computations. And also the modelling results showed that the algorithm is only good at the part of the changes of the input signals. However it almost doesn't work at the part of the flat input signals. We thought the reason is that the network structure used by us is not as complex as the task needs, and the numbers of the neurons are not enough for the kind of input data. However if the numbers of the neurons are increased largely, the regulation of the network will become more complex, and the learning process will have to take very long time.

On the other hand, we also found out that the Vogl's fast algorithm is more dependent on the initial settings of the parameters (e.g. learning rate etc.). The regulation of the network parameters is difficult at the beginning. And the more important thing is that the results of the network output are similar with the previous ones even though its computation qauntity is reduced, and its convengence speed is faster.

In order to solve the above problems we modified the Vogl's ANN algorithm from the following two sides of the view. One is to add a factor γ_i into the Signoid function in order to make the outputs of the network more sensitive to the input signals when the ANN works at the part of the flat input signal. Another one is to make up a large scale ANN which consists of many subANNs with the similar structures.

As we knew that the item $y_i(1 - y_i)$ will approach to zero, and $(\tilde{y}_i - y_i)/neq0$ (here y_i is the output of the network, $\tilde{y}_i$ is the desired output of the network,) when the ANN works at the part of the flat input

signals. In order to make y_i escape from the insensitive area quickly, we think that the error function should be modified at the time, i.e. add a factor $\gamma_i > 1$ into the Signoid function,

$$f(x_i) = \frac{1}{1 + e^{-(x_i - \theta_i)/\gamma_i}} \quad (1)$$

The γ_i can change the shape of the Signoid function. The larger γ_i is, the flater the shape of the Signoid function is. We will set $\gamma_i = 1$ when y_i is out of the local optimization point or the flat area.

On the other hand, in our case if the common ANN is used we have to need many neurons in the hidden layers of the ANN so that the training has to take the very long time even in order to get the preliminary learning results. At the same time, the regulation of such a network is very difficult because of the large quantity of the samples (2250 groups actully) which show the strong nonlinear characteristics with the high-frequency.

In order to overcome the above difficulties, after a lot of tries we finally made up a kind of large scale ANN which consists of many subANNs with the simple and similar structure. Each subANN is in charge of treating a part of the samples. The training of such a subANN is very easy and more efficient because of the small quantity of computations, fast speed, and higher precision.

When the training of one of the subANNs is finished, the learning of the other subANNs can be done on the base of the training results of its previous subANN because of the similarity between each other. The large scale ANN can work well and get the needed outputs because it can switch to the proper subANN automatically according to the conditions of the input samples

In such a way we solved the mismatch problem of the common ANN structure and the requirements of a complex task , and got the satisfied results finally.

4 The Analysis of the Modelling Results

Without loss of generality, we used the expriment data group, named 3000 in our modelling process, which are corresponding to the straight displacement 300 mm only along the z-direction. During the learning/training process we took the data of file 3000TR00.POS and 3000TR01.POS as the samples of the input and output.

First, we ran the three-layers feedforward ANN and basic BP algorithm, the average error was 0.0029 after the network ran and regulated its weights 21000 times. The modelling result is shown in Fig. 1.

Second, we used the Vogl's fast algorithm and the same ANN structure as before, the average error was reduced to 0.0027 after the network ran and regulated its weights 11000 times. The result is given in Fig. 2.

Finally, the improved structure and algorithm of the ANN were used to produce about 130 subANNs according to the modes of the input samples. Here we still took the data of file 3000TR00.POS and 3000TR01.POS as the samples of the input and output. Fig. 4 shows the error curve of the actual and desired outputs along the z-direction. Now we can make the following conclusions:

- The speed of basic BP algorithm is slow. And it needs the large quantity of computations for training. It takes very long time to escape from the insensitive areas. However it can get the good precision of its outputs after a very long time training.
- The speed of the Vogl's algorithm is fast. Its computation quantity is reduced a lot. It does not need very long time to escape from the insensitive areas. However its parameters (such as learning rate etc.) are more dependant on the initial settings so that the regulation of the network is difficult at the beginning.
- The improved ANN has the simple structure and the easy training process, the small quantity of the computations, and the good learning results (i.e. the high precision of modelling) because it consists of the many subANNs, which structures are similar each other. If the training for one of the subANNs is done, the training of the other subANNs is based upon the results of the previous subANN, and becomes easier. On the other hand, this kind of the large scale ANN can solve the mismatch problem caused by the input information and the common ANN structure because each subANN is only responsible for treating a part of the data.

5 Conclusions

In order to build up the precise dynamic model of a flexible system in a more practical way, we made totally use of the three modelling schemes, i.e. the three kinds of learning algorithms and two types of the structures of the ANN in consideration of the strong nonlinear characteristics and the capability of each ANN.

It has to take very long time for us to get the needed results when we use the basic BP algorithm and the structure of the three layers feedforward ANN. And using the Vogl's fast algorithm and the same ANN structure, we got the similar modelling results but in the shorter time. However we have to pay much more attention on the initial settings of the ANN parameters.

The most severe problem in the above two modelling schemes is mismatching between the ANN's structure used and the modelling task requirments, which caused that the training results corresponding to the part of the flat input signals were much worse than the reults corresponding to the part of the change of the input signal. We improved the Vogl's algorithm and the original structure of the ANN in order to solve the problem. The improved algorithm and structure of the ANN show the much better performance for the modelling of the flexible system.

The further goal of our work will be to bulid up a closed-loop feedback compensation control system using the improved ANN dynamic modelling scheme in order to make the flexible robot system works well.

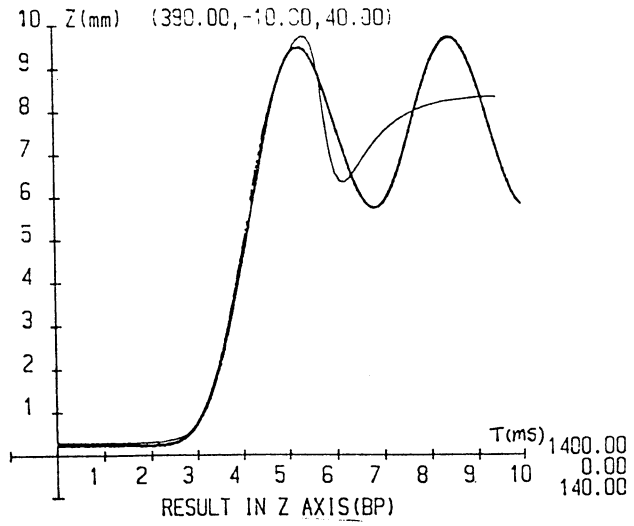


Fig. 1 The Modelling Results using the Common BP algorithm

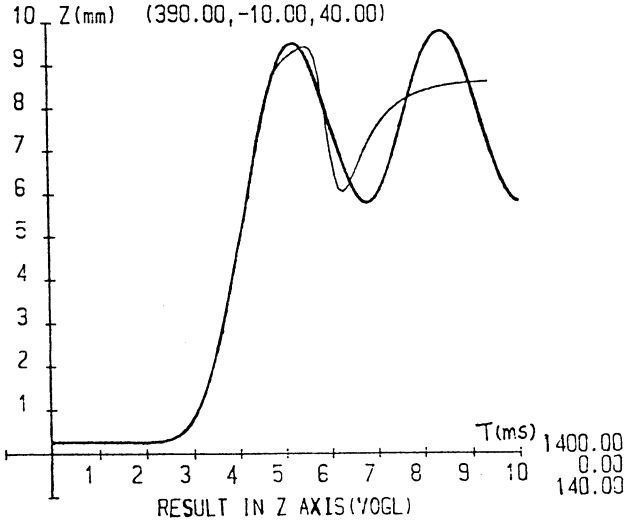


Fig. 2 The Modelling Results using the Vogl's Fast algorithm

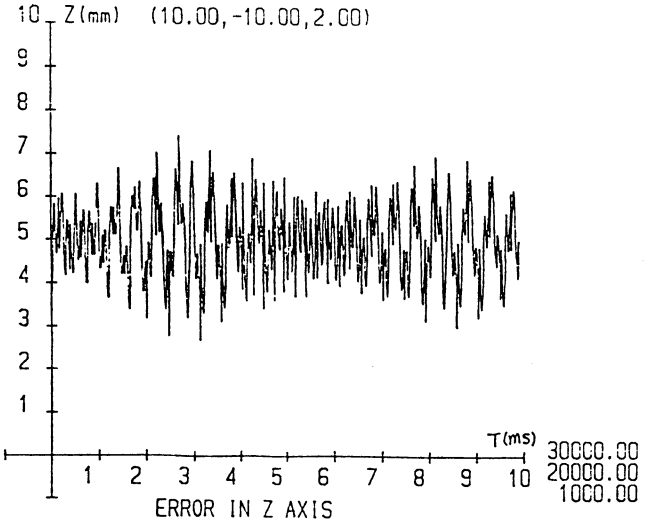


Fig. 3 The Modelling Results using the Improved ANN

References

[1] X. Ding, T. J. Tarn and A. K. Bejcry, "A General Dynamic Model of Flexible Robot Arms for Control", *Proc. IEEE International Conference on Robotics and Automation*, U.S.A., 1989.

[2] Patrizio Tomei, Antonio Tornambe, "Approximate Modelling of Robots Having Elastic Links", *IEEE Trans. on the Systems, Man and Cybernetics*, Vol. 18, No. 5, pp 831-840, 1988.

[3] Hideak Kanoh, "Distributed Parameter Models of Flexible Robot Arms", *Advanced Robotics*, Vol. 5, No. 1, 1991.

[4] L. Jiao, "The computations of the Artificial Neural Network", the Press of XEST University, 1993.

Proc. of Third Int. Symp. on Artificial Life, and Robotics (AROB III'98),
Oita, Japan, 19-21, January, 1998,

Reinforcement Learning for Multiple Robots with Organizational Learning *

○ Keiki Takadama
ATR Human Information Processing Research Labs. / The University of Tokyo

Koichiro Hajiri
ATR Media Integration & Communications Research Labs. / Ritsumeikan University

Tatsuya Nomura
ATR Human Information Processing Research Labs.

Shinichi Nakasuka
The University of Tokyo

Katsunori Shimohara
ATR Human Information Processing Research Labs.

2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan, Tel:+81-774-95-1093 Fax:+81-774-95-1008
Email: takadama@hip.atr.co.jp/keiki@ai.rcast.u-tokyo.ac.jp

Abstract

This paper proposes a novel reinforcement learning mechanism for multiple robots with a concept of organizational learning, and shows that robots with our reinforcement learning mechanism get out of deadlock situations and complete given tasks in both few steps and few iterations. Through intensive simulations of truss construction by multiple robots, the following experimental results have suggested: (1) A certain amount of positive rewards is needed to get out of deadlock situations; (2) Behavior rules performed in both the first and the last steps must be reinforced with the large rewards to complete given tasks; (3) Reinforcement with the combination of geometrical functions reduces both the steps and the iterations.

Keywords: reinforcement learning, organizational learning, organizational structure, multiple robots, emergence, adaptive behavior

1 Introduction

Reinforcement learning generally aims to maximize rewards per an action and to reduce the number of iterations as much as possible. The former approach focuses on *exploration* which extends the search space to maximize rewards, and the latter approach focuses on *exploitation* which utilizes the previous behaviors to reduce the number of iterations. In a single agent learning environment, various researches based on either exploitation [4, 5, 6] or exploration [12] are proposed. Other researches also address the trade-off between exploitation and exploration [8]. However, we cannot apply these methods to a multiple robots environment, because discrete Markov processes and the identification of the environments usually cannot be assumed any more, and because the landscape of the solution space usually becomes multi-modal. To overcome the above issues, we have proposed a reinforcement learning for multiple robots which distributes a negative reward to only the last selected behavior when robots get into deadlock situations [9]. However, since this method only focuses on exploitation to minimize the steps for completing given tasks, this paper proposes a new method which focuses on both exploitation and exploration for multiple robots control.

This paper is organized as follows. Next section starts to explain the concept of the organizational learning, and section 3 gives the details of the organizational learning model with learning classifier systems. In section 4, new reinforcement learning for multiple robots is proposed. An example of multiple robots for truss construction task is given in section 5, and section 6 discusses the experimental results. Finally, the conclusion is presented in section 7.

2 Organizational Learning

2.1 Multiple Robots and Organizational Learning

In the multiple robots learning, it is important for robots to have some mechanisms for self-evaluating their adaptive behaviors to improve the collective performance. The easiest way of evaluating behaviors is to employ a global evaluation function as an explicit control. However, robots must communicate with the central control system or must broadcast information to get the results of a global evaluation function. This approach makes communication cost high. Especially in space, since an amount of communication is limited by the weight or electrical power of robots, it is quite importance to reduce the communication cost. Furthermore, since communication traffic increases as the number of robots increases, it is difficult for robots to perform the tasks with cooperating in real time. Moreover, it is generally difficult to define an appropriate global evaluation function which assigns an optimal and rational behavior to all robots. Therefore, we employ only local evaluation functions instead of a global one, and aim to make robots perform inter-

* Paper submitted to the Third International Symposium on ARTIFICIAL LIFE AND ROBOTICS (AROB'98)

actions among their neighborhoods and acquire their own functions for adaptive collective behaviors only with local evaluation functions.

This kind of bottom-up approach is studied in the context of management organizational sciences as organizational learning, and a lot of researches focus on economical market systems or human organizations [1, 2, 3, 7]. In management organizational sciences, organizational learning is generally defined as follows: "*Organizational Learning* is the creating, acquiring and transferring of *distinctions* and *practices* in the organizations. It is effective if it increases the organization's fit to its environment. Organizational learning implies behavior modification, including changes in relationships, in order to create the conditions for creating, acquiring and transferring distinctions and practices [3]". Since this concept enables robots to autonomously form an organizational structure which completes given tasks effectively, we have proposed Organizational-learning oriented Classifier System which introduces the properties of organizational learning [10, 11].

2.2 Organizational Structure and Functions

Before we explain our organizational learning model, two technical terms are defined as follows.

- **Organizational Structure:** A structure in which robots acquire their own functions.
- **Function:** A sequence of behaviors determined by rules.

Since the organizational structure dynamically changes, robots must adapt to a new organization to maintain or improve the organizational performance. For effective adaptation to the new organization, robots learn adaptive strategies as the order of using the rules, and acquire the effective long sequence of behaviors. Since the long sequence of behaviors produces a lot of variety of behaviors, this variety provides a chance to improve the total organizational performance.

3 Organizational-Learning Oriented Classifier System

In this section, we explain Organizational-learning oriented Classifier System (OCS) [10, 11] which is an extended multiagent version of Learning Classifier System (LCS) with the concept of organizational learning. OCS enables robots to acquire their own appropriate functions through the local interaction among them and form organizational structure which improves organizational performance. In OCS, robots behave only according to their own local evaluation functions, instead of a global evaluation function as an explicit control. Although Michigan approach and Pittsburgh approach for the LCS are proposed, we modify both approaches with introducing the concept of the organizational learning as follows.

- As shown in Fig. 1, each robot does not employ the explicit global evaluation function, but employs the local evaluation function which is composed of the following four parts: (1) Rule base composed of a lot of rules (CFs: Classifiers) in LCS, (2) Working Memory (WM) for storing information on the environment state, (3) Memory for storing the behavior rule sequence which is cleared after rewards are distributed to robots, and (4) Evaluation Mechanism to evaluate the sequence of behavior rules.

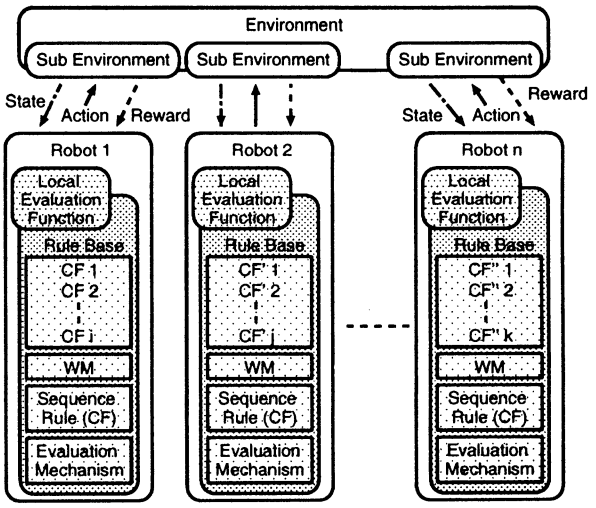


Figure 1: **Architecture of OCS**

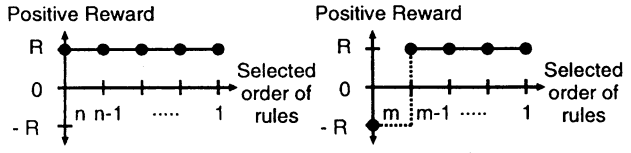
- Robots can obtain their own environmental states, however the total environment states around robots cannot be obtained. Robots determine the behavior according to the state.
- The crossover operation used in Michigan approach is not performed between the rules, and a global evaluation function for selecting elite LCS used in Pittsburgh approach is not also employed.
- All robots have a set of the same simple and rational rules except for the strength of their own rules, and modify or acquire their own functions by self-organizing the strength when rewards are distributed to robots.

In OCS, the following mechanisms and assumptions are also employed.

- A robot performs one behavior by selecting one rule.
- A rule is selected according to a roulette selection used in usual LCS.
- At the beginning, all strength are set to the same value.

4 Reinforcement Learning for Multiple Robots

In OCS, we employ Profit Sharing [4] as an evaluation mechanism shown in Fig. 1. Since a cooperation is quite important for multiple robots to complete given tasks, we have proposed the reinforcement learning which distributes positive rewards to all performed behavior rules when robots complete given tasks and which distributes a negative reward to only the last selected behavior when robots get into deadlock situations [9]. This is because the behaviors performed before getting into deadlock situations contribute to completing tasks with cooperating each other. Fig. 2 (a) indicates the reinforcement learning method for task completion, and the strength in behavior rules is calculated according to Eq. (1). Fig. 2 (b), on the other hand, indicates the method for deadlock situations, and the strength is calculated according to Eq. (2). In this figure, the vertical axis indicates the size of rewards, the horizontal axis indicates the selected order of rules. In the following equations, ST represents the strength of CF, i represents the selected order of CF, n and m represent the max number of the selected CFs, and R represents the size of rewards. In the case that i is small, CF(i) is selected within the first few selections.



R: Size of Reward n,m : Selected order when robots acquire reward

Figure 2: **Previous Reinforcement learning**

- **Task Completion:**

$$ST(i) = ST(i) + R, where\ i = n, n-1, \cdots, 1 \quad (1)$$

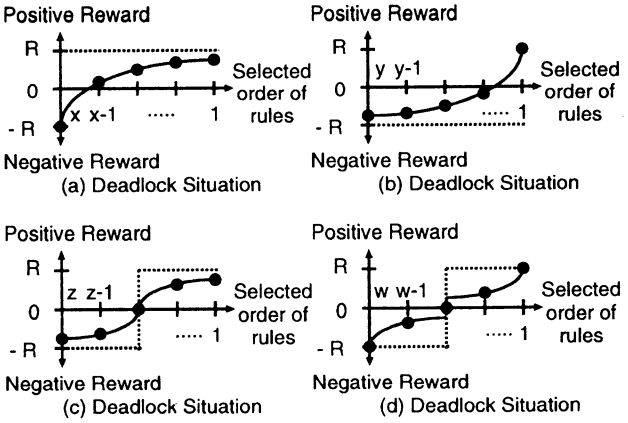
- **Deadlock Situation:**

$$ST(i) = ST(i) + R, where\ i = m-1, \cdots, 1$$
$$ST(i) = ST(i) - R, where\ i = m \quad (2)$$

Since the previous method in Fig. 2 has the following defects in the case of deadlock situations, this paper improves the reinforcement learning for deadlock situations. The method for task completion, on the other hand, is the same method in Fig. 2.

1. Robots cannot complete given tasks when the second last selected rule leads deadlock situations. The scheme in Fig. 2 is valid only when robots can get out of deadlock situations by performing the only one behavior.
2. Although this scheme focuses on exploitation to reduce the iterations, exploration is not focused on to search the fewer steps for completing given tasks.

In order to overcome the above problems, we propose four methods for deadlock situations as shown in Fig. 3. In the following equations, G represents the geometric ratio with the range of $0 < G < 1$.



R: Size of Reward
x,y,z,w : Selected order when robots acquire reward

Figure 3: **New Reinforcement learning**

- **Deadlock Situation (Fig.3 (a)) :**

$$ST(i) = ST(i) + R \cdot (1 - 2 \cdot G^{x-i}),\ where\ i = x, x-1, \cdots, 1 \quad (3)$$

- **Deadlock Situation (Fig.3 (b)) :**

$$ST(i) = ST(i) + R \cdot (2 \cdot G^{i-1} - 1),\ where\ i = y, y-1, \cdots, 1 \quad (4)$$

- **Deadlock Situation (Fig.3 (c)) :**

$$ST(i) = ST(i) + R \cdot (G^{i-z/2-1} - 1),$$
$$where\ i = z, z-1, \cdots, z/2+1$$
$$ST(i) = ST(i) + R \cdot (1 - G^{z/2-i}),$$
$$where\ i = z/2, z/2-1, \cdots, 1 \quad (5)$$

- **Deadlock Situation (Fig.3 (d)) :**

$$ST(i) = ST(i) - R \cdot G^{w-i},$$
$$where\ i = w, w-1, \cdots, w/2+1$$
$$ST(i) = ST(i) + R \cdot G^{i-1},$$
$$where\ i = w/2, w/2-1, \cdots, 1 \quad (6)$$

The third method is composed of scheme in Fig. 3 (b) and (a), and the fourth method is composed of scheme in Fig. 3 (a) and (b) which is an opposite scheme of the third method.

5 Multiple Robots in Space

5.1 Example

To verify the effect of our new reinforcement learning with OCS, we perform experiments of truss construction combined by beams in space as quickly as possible. In order to complete this task, robots must perform the following matters.

- **Cooperation: Behavior order**
  Robots must learn the behavior order to cooperate. For example, two robots hold beams respectively with setting a desired location in a desired angle between the two beams, and the third robot welds the two beams.

- **Acquisition of Different Functions**
  Robots must acquire different functions even if the situation is the same. For example, robots acquire the following functions to cooperate with each other: "hold a beam", "weld beams", or "the combination of them".

5.2 Robots

In order to reduce the weigh of robots, each robot has one arm and holds either a beam or a welding tool. Robots hold their own beams at the station and wait for welding beams at the beam constructing location. The other robots go to the welding location to weld beams or return to the station to get another beams. As other assumptions or limitations, the following matters are taken into consideration.

- Robots can move or rotate for setting a beam at the desired location and angle.
- Robots which wait at the beam constructing location cannot release beams until beams are welded by welding robots.
- Robots cannot pass a beam to other robots.
- Robots cannot weld beams at the welding location until the beams are set at the desired location with the desired angle.
- Robots do not know their appropriate functions beforehand.

In this task, robots get into deadlock situations when all robots hold their own beams or hold the welding tools, or when some robots cannot go to the beam constructing or welding location in the case that other robots wait in the course of the beam constructing or welding location.

6 Simulation

6.1 Experiments

We perform the following five experiments: Previous method in Fig. 2 and New four methods in Fig. 3 (a) $\sim$ (d), and compare the results of the above five methods in term of both the steps and iteration counts. In the experiments, we count one step when all robots perform one behavior, and count one iteration when robots complete a task or get into the deadlock situation. In each case, 5 to 10 robots perform to construct the truss with combining 13 beams, and continue to complete the same task to acquire more appropriate functions. Furthermore, R is defined as 1 and G is defined as 0.5.

6.2 Results

Table 1 and 2 show two results: Final steps for completing tasks and Iteration counts. The former is the results of the steps after robots acquire their own appropriate functions, and the latter is the results of the iteration counts which indicates the counts from the beginning to the convergence of acquiring functions. As shown in this table, robots with the method in Fig.3 (b) and (c) cannot complete given tasks because they cannot get out of deadlock situations. Next, the final steps with the method in Fig.3 (d) are always fewer than those of both the previous method and the method in Fig.3 (a). This relationship is also found in the case of iteration counts. However, there is no relationship in both the final steps and iteration counts between the previous method and the method in Fig.3 (a).

Table 1: **Final Steps for Completing Task**

Number of Robots	Previous Method Fig.2	New Method Fig.3 (a)	(b)	(c)	(d)
5	232	247	—	—	223
6	222	222	—	—	215
7	205	163	—	—	150
8	132	154	—	—	128
9	151	129	—	—	126
10	122	128	—	—	117

Table 2: **Iteration Counts**

Number of Robots	Previous Method Fig.2	New Method Fig.3 (a)	(b)	(c)	(d)
5	78	73	—	—	64
6	66	70	—	—	56
7	61	68	—	—	61
8	112	71	—	—	56
9	42	52	—	—	42
10	74	62	—	—	62

6.3 Findings and Discussions

From the experimental results in the above simulations of truss construction with multiple robots, the following findings are obtained.

- **Importance of positive reward:**
  Since an amount of positive rewards is smaller than that of negative rewards in the case of the method as shown in Fig. 3 (b), the functions of robots are always changed by negative rewards. Therefore, it is difficult for robots to acquire their own appropriate functions, and this unstable function acquisition prevents from getting out of deadlock situations. From the above consideration, a certain amount of positive rewards is needed to get out of deadlock situations.
- **Importance of large rewards in both first and last step:**
  In the case of the method as shown in Fig. 3 (c), rewards in both the first and the last steps are not so large especially when the number of selected behavior rules is small. Therefore, it is

difficult for robots to decide their own appropriate functions quickly, and this unstable function acquisition prevents from getting out of deadlock situations. However, in the case of the method as shown in Fig. 3 (a), robots can complete a given task because the large negative rewards are distributed to the behavior rules performed in the last step. From these factors, behavior rules performed in both the first and the last steps must be reinforced with the large rewards to complete given tasks.

- **Effectiveness of combination of geometrical functions:**
  Since behavior rules performed in both the first and the last steps are reinforced with the large rewards and a certain amount of positive rewards is guaranteed to be distributed to the first several selected rules, robots with the method in Fig. 3 (d) complete a given task with fewer steps and fewer iterations than those with the previous method. This result indicates that this method contributes to reducing the steps for completing tasks by exploring the search space and reducing the iterations by exploiting the previous functions acquisition. Furthermore, since the method in Fig. 3 (d) reinforces the behavior rules according to the geometrical functions unlike the previous method, this method also performs effectively for the problems that robots cannot get out of deadlock situations by performing the only one behavior.

- **Advantage of geometrical functions:**
  As an advantage of geometrical functions, some steps or iterations with the method in Fig. 3 (a) are fewer than those of the previous method. This result shows that the method in Fig. 3 (a) performs effectively when the last few selected rules prevent from searching other solutions. Since we cannot know whether the problem can be solved with changing to select the only last rule or not in advance, the geometrical functions are effective for general problems. In this experiment, the problem structure is different according to the number of robots.

7 Conclusion

This paper has proposed a reinforcement learning method for multiple robots with OCS, and has shown its effectiveness with the task construction problems. By addressing both exploitation and exploration in the multiple robots environment, our method which combines geometrical functions enables robots to complete given tasks with fewer steps and fewer iterations than those with the previous method. Furthermore, we have found that a certain amount of positive rewards is needed to get out of deadlock situations, and also found that behavior rules performed in both the first and the last steps must be reinforced with the large rewards to complete given tasks. Future research includes:

- Theoretical analysis in the proposed reinforcement learning method for multiple robots.
- Verification of the effectiveness of our method when the number of robots increases more than this experiment.
- Verification of the effectiveness of our method in reconstruction not only when new robots are added but also when a certain number of robots get inactive.
- Investigation of the effectiveness of our method with real robots.

References

[1] C. Argyris, D. A. Schon: "*Organizational Learning*", Addison-Wesley, 1978.

[2] R. Duncan, A. Weiss: "*Organizational Learning: Implications for organizational design*", in B. M. Staw (Ed.), Research in organizational behavior, Vol. 1, JAI Press, pp. 75-123, 1979.

[3] R. Espejo, W. Schuhmann, M. Schwaninger, U. Bilello: "*Organizational Transformation and Learning*", John Wiley & Sons, 1996.

[4] J. J. Grefenstette: "*Credit Assignment in Rule Discovery Systems Base on Genetic Algorithms*", Machine Learning , Vol 3. pp. 225-245, 1988.

[5] D. E. Goldberg: "*Genetic Algorithms in Search, Optimization, and Machine Learning*", Addison-Wesley, 1989.

[6] J. H. Holland: "*Properties of the Bucket Brigade Algorithm* ", The 1st International Conference on Genetic Algorithms (ICGA '85), pp. 1-7, 1985.

[7] J. G. March:, "*Exploration and Exploitation in Organizational Learning*", Organizational Science, Vol. 2, No. 1, pp. 71-87, 1991.

[8] K. Miyazaki, M. Yamamura, S. Kobayashi: "*MarcoPolo: A Reinforcement Learning System Considering Tradeoff Exploitation and Exploration under Markovian Environments*", (in Japanese), Journal of Japanese Society for Artificial Intelligence, vol. 12, no.1, pp. 78-89, 1997.

[9] K. Takadama, K. Hajiri, T. Nomura, M. Okada, K. Shimohara: "*Learning Model for Adaptive Behaviors as an Organized Group of Swarm Robots*", The 2nd International Symposium on Artificial Life and Robotics (AROB'97), pp. 103-106, 1997.

[10] K. Takadama, T. Terano: "*Good Solutions will Emerge without a Global Objective Function: Applying Organizational Classifier System to Printed Circuit Board Design*", The IEEE International Conference on Systems, Man, and Cybernetics (SMC'97), pp. 3355-3360, 1997.

[11] K. Takadama, S. Nakasuka, T. Terano: "*Printed Circuit Board Design via Organizational-Learning Agents*", Applied Intelligence: Special Issue on Intelligent Adaptive Agents, 1998, to appear.

[12] C. J. C. H. Watkins, P. Dayan: Technical Note: "*Q-Learning*", Machine Learning, Vol 8. pp. 55-68, 1992.

Author Index