PROCEEDINGS OF THE SIXTH INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB 6th '01)

Vol.1

Jan. 15-Jan. 17, 2001
U-Port, Gotanda, Tokyo, JAPAN

Editors : Masanori Sugisaka and Hiroshi Tanaka

ISBN4-9900462-1-8

Proceedings of The Sixth International Symposium on

ARTIFICIAL LIFE AND ROBOTICS

(AROB 6th '01)

for Cognitive and Behavioral Intelligent Artificial Liferobot

January 15-17, 2001
U-Port, Gotanda, Tokyo, JAPAN

Editors: Masanori Sugisaka and Hiroshi Tanaka

THE SIXTH INTERNATIONAL SYMPOSIUM
ON

ARTIFICIAL LIFE AND ROBOTICS

(AROB 6th '01)

ORGANIZED BY

Organizing Committee of International Symposium on Artificial Life and Robotics(AROB)

CO-SPONSORED BY

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

CO-OPERATED BY

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 City Hall
Kanto Bureau of Economy, Trade and Industry (to be expected)
Kyushu Bureau of Economy, Trade and Industry
NHK Oita Station
Nihon Keizai Shimbun, INC.
THE NIKKAN KOGYO SHINBUN, LTD
OBS Broadcast Company
Oita Asahi Broadcasting
Oita Municipal Government
Oita Prefectural Government
Oitagodo Shinbunsya
PRESS THE NISHINPPON
Science and Technology Agency
TOS Broadcast Company
The Mainichi Newspapers
THE YOMIURI SHIMBUN

HONORARY PRESIDENT

M. Hiramatsu (Governor, Oita Prefecture)

ADVISORY CCOMMITTEE CHAIRMAN

F. Harashima (President, Tokyo Metropolitan Institute of Technology, Japan)

GENERAL CHAIRMAN

Masanori Sugisaka (Oita University)

CO-GENERAL CHAIRMAN (PROGRAM)

Hiroshi Tanaka (Tokyo Medical & Dental University)

CO-CHAIRMAN

John Casti (Santa Fe Institute)

ADVISORY COMMITTEE

S.Fujimura (The University of Tokyo, Japan)
F.Harashima (President, Tokyo Metropolitan Institute of Technology, Japan)(Chairman)
H.Kimura (The University of Tokyo, Japan)
S.Ueno (Kyoto University, Japan)

INTERNATIONAL ORGANIZING COMMITTEE

W.B.Arthur (Santa Fe Institute, USA)
Z.Bubnicki (Wroclaw University of Technology, Poland)
W.Banzhaf (University of Dortmund, Germany)
C.Barrett (Los Alamos National Laboratory, USA)
J.L.Casti (Santa Fe Institute, USA)
J.P.Crutchfield (Santa Fe Institute, USA)
J.M.Epstein (Santa Fe Institute, USA)
T.Fukuda (Nagoya University, JAPAN)
D.J.G.James (Coventry University, UK)
S.Kauffman (Santa Fe Institute, USA)
C.G.Langton (Santa Fe Institute, USA)
J.J.Lee (KAIST, Korea)
Y.Li (Tsinghua University, China)
R.G.Palmer (Santa Fe Institute, USA)
S.Rasmussen (Santa Fe Institute, USA)
T.S.Ray (University of Oklahoma, USA)
P.Schuster (Santa Fe Institute, USA)
M.Sugisaka (Oita University, Japan) (Chairman)
H.Tanaka (Tokyo Medical & Dental University, Japan)
C.Taylor (University of California-Los Angeles, USA)
W.R.Wells (University of Nevada-Las Vegas, USA)
Y.G.Zhang (Academia Sinica, China)

INTERNATIONAL STEERING COMMITTEE

M.Bedau (Reed College, USA)
Z.Bubnicki (Wroclaw University of Technology, Poland)
J.L.Casti (Santa Fe Institute, USA) (Co-chairman)
S.Fujimura (The University of Tokyo, Japan)
T.Fukuda (Nagoya University, Japan)
D.J.G.James (Coventrry University, UK)
J.J.Lee (KAIST, Korea)
G.Matsumoto (RIKEN, Japan)
M.Nakamura (Saga University, Japan)
S.Rasumussen (Santa Fe Institute, USA)
T.S.Ray (University of Oklahoma, USA)
M.Sugisaka (Oita University, Japan) (Chairman)
H.Tanaka (Tokyo Medical & Dental University, Japan)
C.Taylor (University of California-Los Angeles, USA)
K.Tsuchiya (Kyoto University, Japan)
W.R.Wells (University of Nevada-Las Vegas, USA)
Y.G.Zhang (Academia Sinica, China)

INTERNATIONAL PROGRAM COMMITTEE

K.Abe (Tohoku University, Japan)
K.Aihara (The University of Tokyo, Japan) (Co-chairman)
M.Bedau (Reed College, USA)
R.Belew (University of California- San Diego, USA)
Z.Bubnicki (Wroclaw University of Technology, Poland)
T.Christaller (CMD-German National Research Center for Information Technology, Germany)
T.Fujii (RIKEN, Japan)
M.Gen (Ashikaga Institute of Technology, Japan)
T.Gomi (AAI, Canada)
I.Harvey (University of Sussex, UK)
H.Hashimoto (The University of Tokyo, Japan) (Co-chairman)
H.Hirayama(Asahikawa Medical College, Japan)
P.Husbands (University of Sussex, UK)
K.Ito (Tokyo Institute of Technology, Japan)
J.Johnson (The Open University, UK)
Y.Kakazu (Hokkaidou University, Japan)
R.E.Kalaba (University of Southern California, USA)
H.Kashiwagi (Kumamoto University, Japan)
O.Katai (Kyoto University, Japan)
S.Kawaji(Kumamoto University, Japan)
S.Kawata (Tokyo Metropolitan University, Japan)
J.H.Kim (KAIST, Korea)
S.Kitamura (Kobe University, Japan)
H.Kitano (Sony Computer Science Laboratory Inc., Japan)
T.Kitazoe (Miyazaki University, Japan)
S.Kumagai (Osaka University, Japan)
Carl G. Looney (University of Nevada-Reno)
H.H.Lund (University of Aarhus, Denmark)

M.Nakamura (Saga University, Japan)
R.Nakatsu (Santa Fe Institute, Japan)
H.H.Natsuyama (Advanced Industrial Materials, USA)
S.Omatsu (University of Osaka Prefecture, Japan)
T.Omori (Tokyo University of Agriculture &Technology, Japan)
R.Pfeifer (University of Zurich-Irchel, SwitZerland)
T.S.Ray (University of Oklahoma, USA) (Co- chairman)
Y.Sankai (University of Tsukuba, Japan)
T.Sawaragi (Kyoto University, Japan)
T.Shibata (MITI, MEL, Japan)
K.Shimohara (ATR, Japan)
L.Steels (VUBAI Laboratory, Belgium)
M.Sugisaka (Oita University, Japan)
S.Tamura (Osaka University, Japan)
H.Tanaka (Tokyo Medical & Dental University, Japan) (Chairman)
N.Tosa (ATR, Japan)
K.Ueda (Kobe University, Japan)
A.P.Wang (Arizona State University, USA)
K.Watanabe (Saga University, Japan)
X.Yao (The University of New South Wales, Australia)
W.R.Zimmer(GMD-Japan Research Laboratory, Japan)

LOCAL ARRANGEMENT COMMITTEE

K.Nakano (University of Electro-Communications, Japan)
K.Okazaki (Fukui University, Japan)
S.Sato (Director, Research and Development Center, Oita University, Japan)
K.Shibata (Tokyo Institute of Technology, Japan)
K.Shigemitsu (Oita Industrial Research Institute, Japan)
M.Sugisaka (Oita University, Japan)
Y.Suzuki (Tokyo Medical&Dental University)
H.Tsukune (Director, Oita Industrial Research Institute, Japan)
X.Wang (Oita Institute of Technology, Japan)
I.Yoshihara (Miyazaki University, Japan)

TOPICS

Hardware Oriented Topics are welcome
in the fields given by

Artificial Brain Research
Artificial Intelligence
Artificial Life
Artificial Liferobotics
Artificial Living
Artificial Mind Research
Bioinformaitcs
Brain Science
Chaos
Cognitive Science

Complexity
Computer Graphics
Evolutionary Computations
Fuzzy Control
Genetic Algorithms
Human-Machine Cooperative Systems
Human-Welfare Robotics
Innovative Computations
Intelligent Control and Modeling
Micromachines
Micro-Robot World Cup Soccer Tournament
Mobile Vehicles
Molecular Biology
Neural Networks
Neurocomputers
Neurocomputing Technologies and Their Applications for Hardware
Robotics
Robust Virtual Engineering
Virtual Reality

COPYRIGHTS

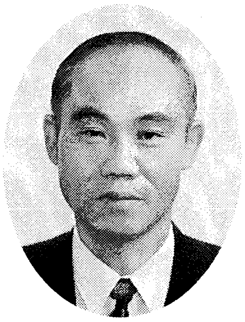
Accepted papers will be published in the proceeding of AROB and some of high quality papers in the proceeding will be requested to re-submit for the consideration of publication in an international journal ARTIFICIAL LIFE AND ROBOTICS (Springer) and APPLIED MATHEMATICS AND COMPUTATION (North- Holland).

All correspondence relating to the symposium should be addressed to:

Prof. Masanori Sugisaka
General Chairman of International Symposium
On Artificial Life and Robotics
(AROB)
AROB Secretariat
Dept. of Electrical and Electronic Engineering
Oita University
700 Dannoharu, Oita 870-1192
JAPAN
TEL 001-81-97-554-7831
FAX 001-81-97-554-7841
E-MAIL msugi@cc.oita-u.ac.jp
arobsecr@oita-cc.cc.oita-u.ac.jp

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

MESSAGE



Masanori Sugisaka
General Chairman of AROB
(Professor, Oita University)

It is my great honor to invite you all to the upcoming International Symposium on Artificial Life and Robotics. The first symposium was held in February (18-20) 1996, B-Con Plaza, Beppu, Oita, Japan. That symposium was organized by Oita University under the sponsorship of the Japanese Ministry of Education, Science, Sports, and Culture (Monbusho), and co-sponsored by Santa Fe Institute (USA), SICE, RSJ, and IEEJ, (Japan). This symposium invites you to discuss the development of new technologies in the 21st century, concerning Artificial Life and Robotics, based on simulation and hardware.

I would like to express my sincere thanks to the Science and International Affairs Bureau, Monbusho, Japanese Government, for their repeated support.

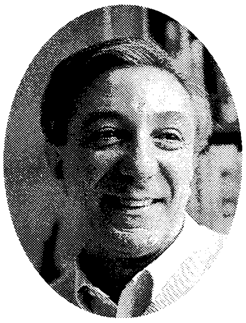
We hope that AROB will facilitate the establishment of an international joint research institute on Artificial Life and Robotics. I hope that you will obtain fruitful results from the exchange of ideas during the symposium.

Masanori Sugisaka
M. Sugisaka

January 5, 2001

MESSAGE

John L. Casti
Vice Chairman of AROB
(Professor, Santa Fe Institute, USA)



For the past 300 years or more, science has focused on understanding the material structure of systems. This has been evidenced by the primacy of physics as the science par excellence, with its concern for what things are made of. The most basic fact about science in the 21st century will be the replacement of matter by information. What this means is that the central focus will shift from the material composition of systems—what they are—to their functional characteristics—what they do. The ascendancy of fields like artificial intelligence, cognitive science, and now artificial life are just tips of this iceberg.

But to create scientific theories of the functional/informational structure of a system requires employment of a totally different type of laboratory than one filled with retorts, test tubes or bunsen burners. Rather than these labs and their equipment designed to probe the material structure of objects, we now require laboratories that allow us to study the way components of systems are connected, what happens when we add/subtract connections, and in general, experiment with how individual agents interact to create emergent, global behavioral patterns.

Not only are these "information labs" different from their "matter labs" counterparts. There is a further distinction to be made even within the class of information labs. Just as even the most well-equipped chemistry lab will help not one bit in examining the material structure of, say, a frog or a proton, a would-be world designed to explore traders in a financial market will shed little, if any, light on molecular evolution.

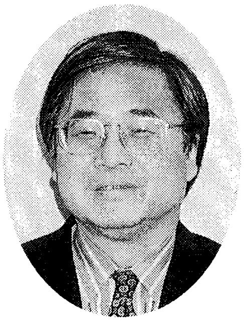
Since the very first Artificial Life meeting in 1987 in Los Alamos, New Mexico, the Santa Fe Institute (SFI) has been at the forefront of this shift in emphasis from matter to information. By the same token, SFI has actively supported such research initiatives in every corner of the world. This support has extended to the Artificial Life and Robotics meetings here in Japan, since the time of the very first meeting in 1996. Each year, researchers from the SFI faculty have come to Japan to meet with others at these AROB meetings, in order to present edge-of-the-frontier ideas and to exchange views on how the fields of ALife and robotics are progressing. So it is a great pleasure for me to again represent SFI on the Organizing Committee of AROB6, and to welcome everyone to this event.

John CASTI
J.L. Casti

January 5, 2001

MESSAGE

Hiroshi Tanaka
Program chairman of AROB
(Professor, Tokyo Medical and Dental University)



On behalf of the program committee, it is truly my great honor to invite you all to the Sixth International Symposium on Artificial Life and Robotics (AROB 6th '01). This symposium is made possible owing to the cooperation of Oita University and Santa Fe Institute. We are also debt to Japanese academic associations such as SICE, RSJ, and several private companies. I would like to express my sincere thanks to all of those who make this symposium possible.

As is needless to say, the complex systems or Alife approach now attracts wide interests as a new paradigm of science and engineering. Take an example in the field of bioscience. As is well known by the name of HGP (Human Genome Project), vast amount of genome information brings about not only from human genome but also various species like several bacterias, yeast, warm, fly. However, as a plenty of genome data becomes available, it becomes sincerely recognized that the framework by which these genome data can be understood to make a whole picture of life is critically needed. The complex systems or Alife approach is now actually expected to be an efficient methodology to integrate this vast amount of data.

This example shows the complex system approach is very promising and becomes widely accepted as a paradigm of next generation of science and engineering. We hope this symposium becomes a forum for exchange of the ideas of the attendants from various fields who are interested in the future possibility of complex systems approach.

I am looking forward to meeting you in Tokyo.

Hiroshi Tanaka

H. Tanaka

January 5, 2001

TECHNICAL PAPER INDEX

Plenary Lecture (Plenary Talks)

MA1: Intelligent Control and Modeling I (General Session)(Room Sakura)

MA2: Brain Science and Complexity I (General Session)(Room Sakura)

MA3: Artificial Life I (General Session)(Room Sakura)

MB1:The Significance and Meaning for Emergence (Organized Session)(Room Kaede)

MB2: Communications with Life-like Creatures and Robots (Organized Session) (Room Kaede)

MB3: Neural Network, Recognition and Behavior Control (Organized Session) (Room Kaede)

MC1: Genetic Algorithms and Evolutionary Computation I (General Session) (Room Natsume)

※This paper would be changed from oral session to poster session.

MC2: Neural Networks I (General Session)(Room Natsume)

MC3: Robotics I (General Session)(Room Natsume)

TB-1: Softcomputing in Robotic Intelligence (Organized Session)(Room Kaede)

TB-2: Artificial Brain (Organized Session)(Room Kaede)

TB-3: Methodology of Emergent Synthesis (Organized Session)(Room Kaede)

TC1: Brain Science and Complexity II (General Session)(Room Natsume)

TC2: Intelligent Control and Modeling II (General Session)(Room Natsume)

TC3: Artificial Life II (General Session)(Room Natsume)

TD1: Mobiles Vehicles I (General Session)(Room Aoi)

TD2: Related fields I (General Session)(Room Aoi)

TD3: Robotics II (.General Session)(Room Aoi)

WA1:Artificial LifeⅢ (General Session)(Room Sakura)

WA2: Mobiles Vehicles II (General Session)(Room Sakura)

WA3: Related fields II (General Session)(Room Sakura)

WA4:Genetic Algorithms II (General Session)(Room Sakura)

WB-1: Biodiversity I (Organized Session)(Room Kaede)

WB-2: Biodiversity Ⅱ(Organized Session)(Room Kaede)

WB-3: Non-conventional Computing and Moleware Architecture (Organized Session) (Room Kaede)

WB-4: Biologically Inspired Systems (Organized Session)(Room Kaede)

WC1:Molecular Biology I (General Session)(Room Natsume)

WC2: Intelligent Control and ModelingⅢ(General Session)(Room Natsume)

WC3: RoboticsⅢ(General Session)(Room Natsume)

WC4: Neural Networks II (General Session)(Room Natsume)

MF: Poster Session (Room Karinn)

TF: Poster Session (Room Karinn)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

ADAPTIVE FUZZY CONTROLLER FOR A CLASS OF UNCERTAIN OPERATION SYSTEMS

Leszek Siwek

Institute of Control and Systems Engineering,
Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, POLAND
Phone: +48 71 320 33 28, email: lsiwek@ists.pwr.wroc.pl

Keywords: control system, fuzzy logic, computer network, adaptation via identification

Abstract

The paper is concerned with a class of service operation systems and the main function of control which consists in decision making about the admission of customers. One of those operation systems is the ATM computer network where the admission control is a crucial function for correct work of the network.

The fuzzy controller for the admission control in ATM network is presented and the adaptation of the controller using the current values of performance index is considered. In the paper the self-adjusting process consists in the current updating of two parameters in selected membership functions. The results of two-stage identification of the optimization plant are given. The adaptation algorithms in closed-loop system obtained with the use of identification results are described. The problem of how to determine the parameters of adaptive algorithms to ensure the convergence condition is shortly discussed.

1. Introduction

It is not easy for the admission control system in ATM network to make decision about accepting or rejecting a new customer. The reason of this is the unknown precise mathematical model of the control plant. This system may be considered as a specific two level queuing system. On the lower level we have the calls of customers appearing randomly. Each customer is associated with the stream of cells. On the higher level of the system under consideration we have the stream of cells entering the queue. Cells from the queue are transmitted by the output link. The aim of the control is proper admission of calls on the lower level, maximizing the intensity of cells leaving the queue and keeping the cell loss rate below the given value. Because the precise mathematical model of the plant to be controlled is unknown, it is very difficult to develop a precise admission control algorithm. Therefore, it is reasonable to apply algorithms that are based on neural networks [1], logical knowledge representation [2] and fuzzy logic, where a three-part fuzzy controller has been described (see [3]). In paper [4], a two-part fuzzy controller has been introduced and the adaptation consisting in the self-adjusting of **one parameter** of selected membership functions has been presented. In this paper the structure of fuzzy controller is similar to the one presented in [4]. But the algorithm of the adaptation via so called two-stage identification consists in self-adjusting of **two parameters**. The control system under consideration is described in sec. 2. The two-stage identification concept is given in sec. 3. The algorithms of adaptation in closed-loop system designed with the use of identification results are described in sec. 4.

2. Model of the control plant and the fuzzy controller

Let us denote by t_n the arrival moment of the n-th call, by τ_m the moment of the parameters measurement, $\tau_{m+1} - \tau_m \overset{\text{Ä}}{=} \text{Ä}\tau = \text{const.}$, by $\text{Ä}\tau(n)$ the last time interval of length $\Delta\tau$, preceding t_n. For characterisation of the control plant let us introduce the following variables. **Variables describing the n-th call**: $z_n^{(1)}$ is a maximum value of cells intensity (i.e. the intensity of transferring the cells by a customer to the system), $z_n^{(2)}$ is a mean value of the cells intensity, $z_n^{(3)}$ is the period of time in which the cells intensity equals $z_n^{(1)}$. The variables $z_n^{(1)}$, $z_n^{(2)}$ and $z_n^{(3)}$ determine v_n, i.e. the substitute cells intensity for the n-th call. The variable $z_n^{(4)}$ is the sequence of signals about leaving the system by customers accepted earlier ($z_{i,n}^{(4)} = 0$ if the i-th customer has already left the system, $z_{i,n}^{(4)} = 1$ if the i-th customer is still in the system). The **control decision** u_n is equal to 0 (rejection) or 1 (acceptance). The **output variables**: (i.e. variables characterising the plant, dependent on z_n and u_n) $y_n^{(1)}$ – the total substitute intensity of the cells entering the queue, i.e. the sum of v_n and v_i of all earlier accepted calls which are still in the system, i.e.

$$y_n^{(1)} = \sum_{i=1}^{n-1} z_{i,n}^{(4)} \cdot u_i \cdot v_i + v_n ,$$

$y_n^{(2)}$ – the cell loss rate, i.e.

$$y_n^{(2)} = \frac{P_n}{C \cdot \Delta\tau}$$

where P_n is the number of cells lost in Äô(n), C is a maximum intensity of cells leaving the queue; $y_n^{(3)}$ – the intensity of cells leaving the queue at the moment t_n, i.e. ratio between the number of cells leaving the queue in Ä$\tau(n)$ and the length Äτ, $y_n^{(4)}$ – the current length of the queue, i.e. the number of cells in the queue at the moment t_n; $y_n^{(5)}$ – the change of the length of the queue in Ä$\tau(n)$, i.e. if $d(t)$ is the length of queue at the moment t then $y_n^{(5)} = d(\tau_m) - d(\tau_{m-1})$.

In the system under consideration $y_n^{(3)}$ is used to define the performance index

$$Q_N = \frac{1}{2N+1} \sum_{m=-N}^{+N} w_m$$

where N is a horizon of observation, w_m is the intensity of cells leaving the queue at the moment $\hat{o}_m$, i.e. ratio between the number of cells which left the queue in $[\hat{o}_{m-1}, \hat{o}_m]$ and Äô.

The aim of the control in the system under consideration is to find such a control algorithm which will generate a sequence u_n maximizing the performance index Q_N with the constraint $y_n^{(2)} \leq á$ where α is a given number.

The executor of the control algorithm in this paper is the fuzzy controller consisting of two parts (R_z, R_d in Fig. 1). In each part the determination of the outputs is based on a fuzzy approach. Block R_d using $z_n^{(1)}, z_n^{(2)}, z_n^{(3)}$ calculates v_n and then $y_n^{(1)}$. For determining v_n the fuzzy controller with 18 fuzzy rules has been suggested (see [5]). In block R_z decision u_n is determined using $y_n^{(1)}, y_n^{(2)}, y_n^{(4)}, y_n^{(5)}$. For this block 24 fuzzy rules have been described in [4]. Each rule has the form of implication. For example, one of the rules in block R_z is the following: "If $y_n^{(1)}$ is medium and $y_n^{(2)}$ is small and $y_n^{(4)}$ is large and $y_n^{(5)}$ is small then $u_n = 1$". The fuzzy rules used in [4] and in this paper are based on the rules presented in [3], and on the known properties and strategies of ATM networks, such as the equivalent capacity strategy, the bandwidth allocation strategy [6], and on the way of cells buffering in ATM networks [7]. For each property in the fuzzy rules, triangular and trapezoidal membership functions are suggested. The determination of $y_n^{(1)}$ in the block R_d for the given z_n, and the determination of u_n in the block R_z for the given y_n are based on the known procedures using the fuzzy rules (see e.g. [8]). In the case of u, the membership function is reduced to two values: $\mu_u(0)$ and $\mu_u(1)$ which denote the certainty factors that $u = 0$ and $u = 1$, respectively. The defuzzification consists in the determination of the mean values.

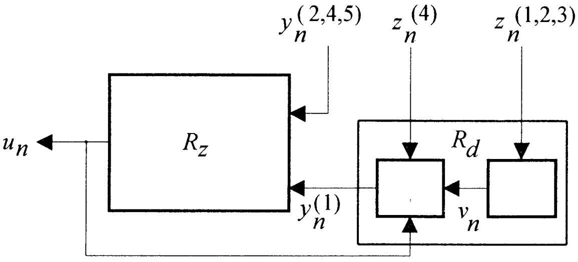


Fig. 1.

3. Two-stage identification of the optimization plant

Let us assume that the process of adaptation consists in self-adjusting of the parameters a, b in the membership functions of the property concerning v_n and $y_n^{(1)}$, respectively (see Figs. 2 and 3). In paper [4] we considered the parameter b which was changed in the process of adaptation and the parameter in the probability density of $z_n^{(1)}$ denoted by d. Consequently, the control system with fuzzy controller has been considered as a static optimization plant with the output Q_l and inputs: b_l, d_l where l denotes the l-th period of adaptation

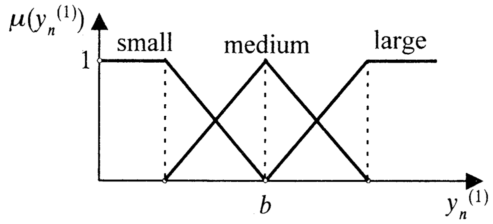


Fig. 2.

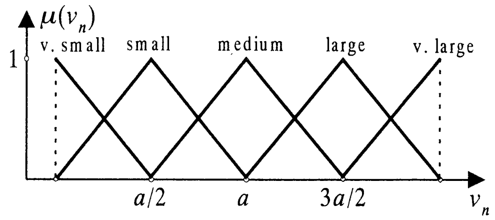


Fig. 3.

containing the moments $m = (l-1)N+1,\dots,lN$,

$$Q_l(b) = \frac{1}{N} \sum_{m=(l-1)N+1}^{lN} w_m \,,$$

b_l and d_l are the values of b and d in the l-th period of adaptation, respectively. Now the optimization plant has an output Q_l and two inputs: a_l, b_l, i.e. the values of a and b in the l-th period of adaptation, respectively. For the plant under consideration a method of two-stage identification [9] has been used. On the first stage of identification, for the fixed constant value a we change b and determine a vector of parameters $c_1 = (c_1^{(1)}, c_1^{(2)}, c_1^{(3)})$ in the relationship between Q and b. The identification on the first stage should be repeated for different values a and as a result we determine a vector of parameters $c_2 = (c_{21}^{(1)}, c_{21}^{(2)}, c_{21}^{(3)}, c_{22}^{(1)}, c_{22}^{(2)}, c_{22}^{(3)}, c_{23}^{(1)}, c_{23}^{(2)}, c_{23}^{(3)})$ in the relationship between c_1 and a (identification on the second stage). On the first stage the model

$$Q_j \overset{\text{Ä}}{=} F_1(b, c_{1j}) = c_{1j}^{(1)} b^2 + c_{1j}^{(2)} b + c_{1j}^{(3)} \,, \qquad (1)$$

where $j = 1, 2, \dots, J$, has been assumed. On the second stage the model

$$c_1 \overset{\text{Ä}}{=} F_2(a, c_2)$$

has been assumed in the following form

$$c_1^{(1)} = c_{21}^{(1)} a^2 + c_{21}^{(2)} a + c_{21}^{(3)}, \quad c_1^{(2)} = c_{22}^{(1)} a^2 + c_{22}^{(2)} a + c_{22}^{(3)},$$

$$c_1^{(3)} = c_{23}^{(1)} a^2 + c_{23}^{(2)} a + c_{23}^{(3)} \qquad (2)$$

where c_{21}, c_{22}, c_{23} are the parameters which should be determined on the second stage. Finally, using (1) and (2) we obtain the model of our plant:

$$Q \overset{\text{Ä}}{=} F_1[b, F_2(a, c_2)] = (c_{21}^{(1)} a^2 + c_{21}^{(2)} a + c_{21}^{(3)}) b^2 +$$

$$+ (c_{22}^{(1)} a^2 + c_{22}^{(2)} a + c_{22}^{(3)}) b + c_{23}^{(1)} a^2 + c_{23}^{(2)} a + c_{23}^{(3)} \,. \qquad (3)$$

For the numerical data obtained during the simulation, the values of parameters have been calculated by using the least square method (see e.g. [10]). Examples of the results obtained on the first and second stage of identification are shown in Figs. 4 and 5, respectively.
The final results of identification are as follows:

$$c_{21}^{(1)} = -1.52, \; c_{21}^{(2)} = -0.08, \; c_{21}^{(3)} = 0.63, \; c_{22}^{(1)} = 2.25,$$

$$c_{22}^{(2)} = 0.02, \; c_{22}^{(3)} = -0.53, \; c_{23}^{(1)} = 0.24, \; c_{23}^{(2)} = 0.02,$$

$$c_{23}^{(3)} = 1.32.$$

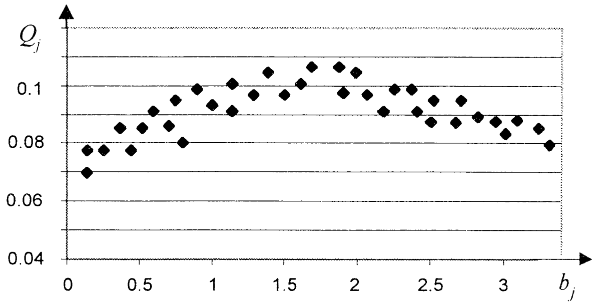


Fig. 4.

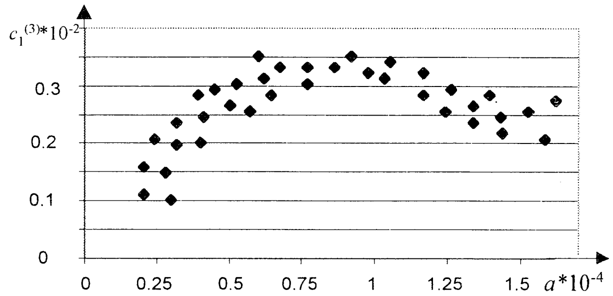


Fig. 5.

4. Algorithms of adaptation via identification

In paper [4] two versions of adaptation algorithms (in open-loop and closed-loop system) have been presented. Now, we propose only the adaptation in a closed-loop system because the model of the plant is the result of identification which has been made for constant value d ($d = 0.02$). We consider two versions of the adaptation algorithm in closed-loop system: with the application of a gradient of the function $Q(a,b)$ and with the application of trial steps. Let g denote a vector of parameters in the control algorithm with p components, i.e. parameters in the membership functions. Then in the first version

$$g_{l+1} = g_l + K \cdot \underset{gl}{grad\; Q_l} \qquad (4)$$

where Q_l (the model of the plant obtained as a result of identification) is the known function of g_l , K is a $p \times p$ matrix of coefficients which may be chosen by a designer. In the case considered in sec. 3, g is a two-dimensional parameter ($p = 2$). According to formula (3) $Q_l = F(a_l, b_l, c_2)$, and (4) is reduced to the following adaptation algorithm

$$\begin{bmatrix} a_{l+1} \\ b_{l+1} \end{bmatrix} = \begin{bmatrix} a_l \\ b_l \end{bmatrix} + K \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

where

$$q_1 = (2c_{21}^{(1)} a_l + c_{21}^{(2)}) b_l^2 + (2c_{22}^{(1)} a_l + c_{22}^{(2)}) b_l + 2c_{23}^{(1)} a_l + c_{23}^{(2)},$$

$$q_2 = 2(c_{21}^{(1)} a_l^2 + c_{21}^{(2)} a_l + c_{21}^{(3)}) b_l + c_{22}^{(1)} a_l^2 + c_{22}^{(2)} a_l + c_{22}^{(3)} .$$

In the second version

$$g_{l+1} = g_l + K w_l .$$

For the known model of the plant $Q_l = F(g_l)$, the i-th component of the vector w_l is:

$$w_l^{(i)} = \frac{F(g_l + \ddot{a}_i) - F(g_l - \ddot{a}_i)}{2\acute{o}_i}, \quad i = 1, 2, ..., p \qquad (5)$$

where δ_i is a vector with zero components, except for the i-th component which equals σ_i (the value of the trial step). If the model is unknown then

$$w_l^{(i)} = \frac{Q_l(g_l + \ddot{a}_i) - Q_l(g_l - \ddot{a}_i)}{2\acute{o}_i}, \quad i = 1, 2, ..., p \qquad (6)$$

where $Q_l(g_l)$ is the value of the output when at the input $g = g_l$ is put.

As in [4], the model of the plant determined as a result of identification may be used for determination of convergence conditions for the process of adaptation in case (4) and (5), and for estimation of the time of convergence in case (6). The way of obtaining the convergence conditions for case (4) and (5) is similar to the one presented in [4]. It is worth noting that here K is not one-dimensional but is a matrix of coefficients.

5 Conclusions

Using adaptation presented in [4], i.e. self-adjusting of one parameter in the selected membership functions, significantly improves the performance index. The results of simulation showed that the performance index can be further improved by changing two parameters in different membership functions. That is why in this paper the adaptation algorithms consisting in the changing of two parameters have been developed. One should note that the process of two-stage identification took place on the simulator of the queuing system, not on the real-life plant. During this simulation the parameter in the probability density of $z_n^{(1)}$ denoted by d was constant (in [4] was changing). It is the reason why in this paper the adaptive algorithm in open-loop system has not been developed. In further works, using three-stage identification, a model of a plant depending on two parameters in membership functions and on d (disturbance) will be developed. This result can be used to develop the adaptation algorithm in open-loop system. The model of the plant resulting from three-stage identification may be expected to be much more similar to the real one.

Acknowledgements

This work was supported by the Polish State Committee for Scientific Research under the grant no. 8 T11C 012 16.

References

[1] Hiramatsu, A., (1990), ATM communications network control by neural networks, *IEEE Trans. Neural Networks*, 1, no. 1.

[2] Rapior, P. (1998), The Bubnicki method in knowledge based admission control for ATM networks, Proc. of the 13th Inter. Conf. on Systems Science, Wroclaw, Poland, pp. 238-243.

[3] Chang, C. J., Cheng, R. G. (1996), Design of a fuzzy traffic controller for ATM Networks, *IEEE Trans. on Networking*, 4, no. 3.

[4] Siwek, L., (2000), On the adaptive controller for the admission control in two-level service system, Proc. 14th Int. Conf. on Systems Engineering, Coventry, UK, pp.498-503.

[5] Siwek, L. (2000), Control System of Calls Admission in Computer Network Using Fuzzy Approach, Proc. of the 16th IMACS World Congress, Lausanna, Switzerland.

[6] Guerin R. ,Ahmadi H., Naghshineh M., (1991), Equivalent capacity and its application to bandwidth allocation in high-speed networks, *IEEE J. Select. Areas Commun.*, 9, no. 7.

[7] Saito, H. (1992), Call admission control in an ATM network using upper bound of cell loss probability, *IEEE Trans. Commun.*, 40, no. 9, pp. 1512-1521.

[8] Yager, R.R., Filev P.D. (1994), Essentials of fuzzy modelling and control, *John Wiley & Sons, Inc.*.

[9] Œwi¹tek, J. (1987), Two-stage identification and its technical and biomedical applications, *Wydawnictwo Politechniki Wroc³awskiej*, Wroclaw, (in Polish).

[10] Bubnicki, Z. (1980), Identification of Control Plant, *Elsevier*, New York, Oxford, Amsterdam.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Design and Optimization of Fuzzy Controllers based on the Operator's Knowledge

Hyeon Bae, Sungshin Kim, Man Hyung Lee*

Dept. of Electrical Engineering, Pusan National University
* School of Mechanical Engineering, Pusna National University
Changjun-dong, Kumjung-gu, Pusan, 609-735, Korea

E-mail : baehyeon@pusan.ac.kr, sskim0@pusan.ac.kr, mahlee@pusan.ac.kr

Abstract — Industrial processes are normally operated by a skilled human who has the cumulative and logical information about the systems. Fuzzy control has been investigated for many applications which could not represented by an accurate mathematical model or could not controlled by conventional methods. Intelligent control approaches based on fuzzy logic have a chance to include human thinking that is represented by a natural language. In this paper, we design a controller based upon operator's knowledge without mathematical model of the system and optimize the controller. The tested system is constructed for sending a ball to the goal position using wind from two DC motors in the path. A vision camera to mimic human eyes detects the boll position. The system used in this experiment could be hardly modeled by mathematical methods and could not be easily controlled by conventional manners. The controller is designed based on the input-output data and experimental knowledge obtained by trials, and optimized under predefined performance criterion.

Keywords – Fuzzy logic, intelligent control, optimization.

I. Introduction

During the last several years, fuzzy controller is investigated to improve manufacturing processes [1], [2]. Conventional controllers normally need mathematical models and cannot easily handle models because of the incompleteness or uncertainty. On the contrary, intelligent control approaches based on fuzzy logic can include human's inference mechanism without information of the system's dynamic equation. Therefore, we can make control rules without quantization [6]. The controller for the experimented system in this paper is designed based on the empirical knowledge because of the feature of the used system. The fuzzy singleton method is used for the consequent part.

II. Ball Positioning System

As shown in Fig. 1, the experimental tested system consists of independent two fans operated by DC motors. The purpose of this experiment is that a ball is sent to the final goal positions using two fans. This system contains non-linearity and uncertainty because the aerodynamics inside the path. In this experiment, the wind flowed from DC motors will crash to surfaces of the inner pass and it makes an eddy. This eddy changes strength of wind and makes the difference of the wind strength at each position. Therefore, the ball is not moved as the power is applied to the each path because of the uncertainty. Analysis of effects caused by the wind is difficult and modeling of the system is also not easy because of the much changeable phenomenon from time to time. Therefore, the goal of this experiment is that the fuzzy controller is initially designed by operator's knowledge and optimized itself after playing with the system.

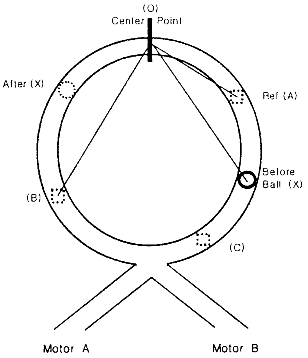


Fig. 1. Test system.

III. Image Processing

Real images are transmitted to a computer by a CCD camera and a frame-grabber. The two-dimensional ball position is recognized from the captured image. In this case, colors of ball, plant, and predefined path are different each another: background is black, path is green, and ball is yellow. Black and green colors have low red value compared with yellow color. The difference of each image is measured for finding moving ball position [8]. The difference coding reduces the effect of environment mixed with similar colors. In the image processing, the center of ball is obtained by difference values of two images that stored in arrays. The threshold process is employed to eliminate noise after getting the difference values of images. If the value of a pixel is higher than required threshold value, one is assigned otherwise zero is assigned to that pixel. Finally, the center of area is used for measuring the ball center point [7].

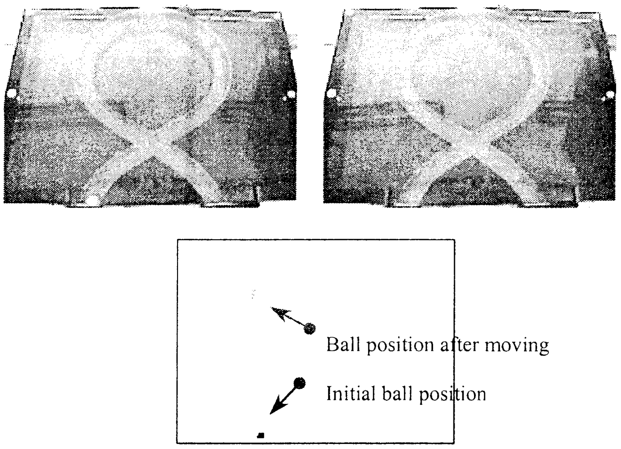


Fig. 2. Image processing for finding the ball position.

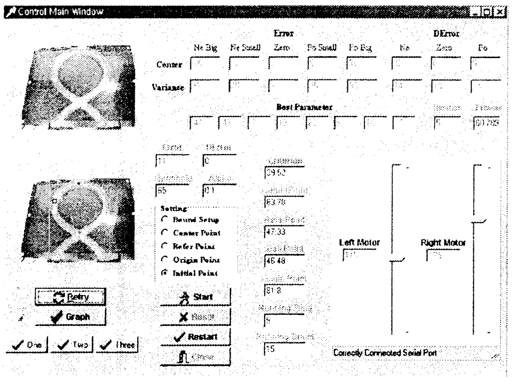


Fig. 3. Control main window.

IV. Fuzzy System

A. Measuring Error Value

Equation of error values is needed for evaluating. Error values are defined by the difference of distances between the final goal and the present ball position. In this experiment, error and derivative of error are used for inputs. Input voltages of motors are calculated by the membership functions in the consequent part. The direction of error value has + or – sign by the distance measure. If the ball is locating at the opposite direction of the goal position, an error value should be big. The following equation is to measure error values.

$$e(\bullet) = \sqrt{(B_x - O_x)^2 + (B_y - O_y)^2} - \sqrt{(R_x - O_x)^2 + (R_y - O_y)^2} \quad (1)$$

where B, O, and R indicate a ball position, an original position, and a goal position respectively.

B. Fuzzy Rules and Membership Function

Gaussian functions are used for membership functions of a premise part and five singleton values are used for the consequent part. As shown in Table I, the skilled human could define how to handle or control the system. This is called the operator's knowledge. Fuzzy rules are constructed based upon operator's knowledge as shown in Table II that is used to control the system.

Table I Operator's knowledge

Initial ball position : Right	
Higher than goal	Motor R: decreasing speed slowly Motor L: increasing speed slowly
Opposite side	Motor R: decreasing speed rapidly Motor L: increasing speed rapidly
Lower than goal	Motor R: increasing speed slowly Motor L: maintaining speed

Table II Fuzzy rules

Right Motor

$\dot{e}$ \ e	PB	PS	ZE	NS	NB
N	NS	ZE	ZE	PB	PB
ZE	NB	NS	ZE	ZE	PS
P	NB	NS	NS	PS	PS

Left Motor

$\dot{e}$ \ e	PB	PS	ZE	NS	NB
N	PS	PS	PS	NS	NS
ZE	PB	PS	ZE	NB	NB
P	PB	PB	PS	NB	NB

C. Inference and Defuzzification

The Mamdani fuzzy inference system was proposed for controlling system with a set of linguistic rules obtained from experienced operators and min-max composition is used for inference [3], [5]. In Mamdani fuzzy model, center of areas or mean of maximum is used for defuzzification. In this paper, fuzzy singleton is used which is a special case of the Mamdani fuzzy inference system. Each rule's consequent is specified by a fuzzy singleton [5]. The fuzzy singleton is free of computation works. Therefore, it is suitable for real time applications.

V. Optimization

The system condition is changed each experiment due to different environment conditions. Therefore, the adjustment of parameters in the membership functions is necessary. In this paper, two types of optimization methods are applied for finding good parameters. The eight parameters for error and derivative error membership function are adjusted by an optimization. Each membership functions are symmetrically constructed, so adjusted parameters are reduced. In this experiment, hybrid genetic algorithm and simulated annealing are implemented for optimization.

A. Genetic Algorithm

A genetic algorithm is a global optimization method to find the required solutions. The GA obtains new populations using reproduction, crossover, and mutation [5]. In this paper, 8-bits are composed for searching each parameter. Therefore, total 64-bits genes are used [4]. Reasonable ranges of each parameter are determined by operator's knowledge. If a proper range is used in a GA decoding process, high quality solution can be obtained without much population and generation. This is suitable for this plant. And it could not break the fuzzy linguistic reasoning due to keeping the regular range.

B. Hybrid Genetic Algorithm

In GA, a search space in a GA is discretized by its resolution R_i for the parameter x_i with L_i bits. It could be formulated as a following equation.

$$R_i = \frac{UB_i - LB_i}{2^{L_i} - 1} \qquad (2)$$

If it is increased by k bits, R_i is increased by 2^k times, and search space is increased by $(2^k)^n$ times. Hybrid genetic algorithm is the method that a GA and Nelder-Mead's simplex method are combined. The hybrid approach yields the fast convergence rate without sacrificing the accuracy of the solution. Therefore it can reduce the computational cost.

C. Simulated Annealing

Simulated annealing (SA) is derivative free Simulated annealing (SA) is a derivative free optimization method that recently drawn much attention for being as suitable for continuous as for discrete optimization problems [5]. The most important part of SA is the annealing schedule, which specifies how rapidly the temperature is lowered from high to low values. This is usually application specific and requires some experimentation by trial and error. In this paper, the algorithm starts at a high starting temperature and it is declined gradually while iterations are repeated.

VI. Experiment Result

A. Performance Criterion

Performance criteria are necessary to evaluate the performance. Several types of criteria are used for evaluation. In this paper, the combined performance criterion is defined as follow which considers a convergence time and error values simultaneously.

$$f(e) = \alpha \sum_{t=t_1}^{t_2} t|e(t)| + (1-\alpha)\sum_{t=t_1}^{t_2} |e(t)| \qquad (3)$$

where α is contributed value and t_1, t_2 is the period time of measurement error.

B. Operator's Test

In a operator's experiment, convergence rates and error values are measured for comparing the performance. It is achieved by the performance criterion so called the objective function and fuzzy rules are fixed based on the empirical knowledge.

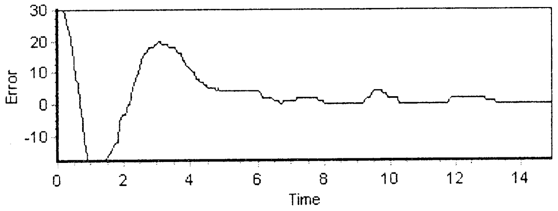


Fig. 4. Error graph with respect to time.

Fig. 4 shows an error value that converges to 0 with time going. The ball is oscillated much before it approaches near a goal position and then the oscillation is rapidly decreased. In operator's test, random parameters are firstly selected, and parameters are adjusted by evaluated performance. This is much important information for the optimization. Results of the test are shown in Table III.

Table III Parameters of membership function

	Goal : Left upper					
	1	2	3	4	5	Average
A	41.81	13.95	14.24	32.20	12.36	22.91
B	16.69	8.59	30.74	13.72	13.02	16.55
C	17.21	14.08	34.44	18.40	17.42	20.31
D	19.83	13.19	12.72	12.34	9.37	13.49
E	10.44	8.60	9.20	27.11	9.63	12.99
F	18.09	19.33	19.64	19.88	18.24	19.04

C. Hybrid Genetic Algorithm

30% of crossover and 1.5% of mutation rate are implemented for genetic algorithm in this test. And elitism is used for storage and transfer of the best population. It could improve the optimization performance. The first graph of Fig. 5 represents the performance with respect to each membership functions. The performance values jump to worse value in the middle of a graph, but they could be declined gradually with iteration. Shapely changed values are eliminated at the consequent part. After processing of GAs, the performance represents a worse value than previous values in the simplex method as shown in the second graph of Fig. 5 because the difference of the performance exists in real experiments even though the same membership functions are used. This phenomenon could occur due to features of systems or uncertainty of wind. Another reason is that the best performance values are kept continuously in GAs and then it will transfer to the next generation. On the other hand, all parameters transferred from GAs are used for operating the system and then performances are evaluated again in the simplex method, so little bit gap could exist. But the performance value is lowed finally as shown in the second graph of Fig. 5. This algorithm can be regarded as a proper optimization method for the tested system.

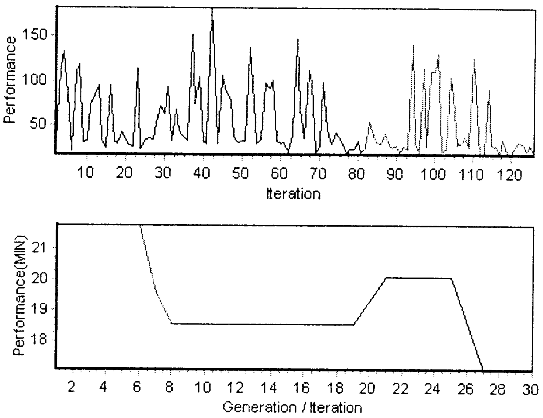


Fig. 5. Test performance result by hybrid GA.

C. Simulated Annealing

As shown in Fig. 6, two graphs represent result performance of SA algorithm. In the initial part of the second graph of Fig. 6, the performance value is dropped much and then it is lowed to better values while iteration is repeated. SA can search good solutions fast even though repeated iteration times are fewer than iteration times in GA. When SA is used for the optimization, selecting initial values is very important. In this experiment, SA starts with searched parameters based on operator's knowledge. A high computation cost is avoided using known initial parameters. Initially a high temperature coefficient is selected for the global search and then it is lowed gradually during the processing for the local search. The optimized solution is searched through this processing.

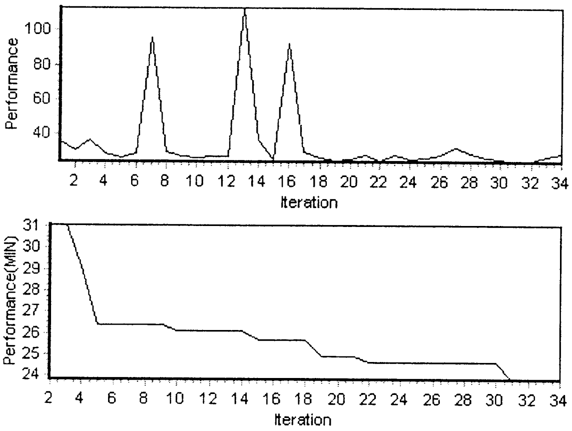


Fig. 6. Test performance result by SA.

VII. Conclusion

The primary goal of this paper is to design and optimize a fuzzy controller from operator's knowledge and running process. It uses characteristics of the fuzzy controller. The fuzzy controller can control systems using fixed rules based on operator's knowledge without mathematical models. When the system is operated, the environment condition can be changed from time to time. In this case, fixed parameters of fuzzy membership function are not suitable for controlling the system properly. Therefore center and variance values are adjusted automatically. It causes output values of controller to change and then it can be adaptable for variable environment condition. In this paper, Gaussian functions are employed for fuzzy membership functions. Two parameters of each function are optimized. And 8 membership functions are used for error and derivative error functions. Thus, total 16 parameters are handled. The hybrid approach that combines a GA with a stochastic variant of the simplex method is implemented and applied for alleviating this difficulty. Also, SA is tested as an optimization. Operator's knowledge is used for finding reasonable ranges in a hybrid GA and initial parameters in SA. In this paper, both of them are compared considering convergence rate and accuracy. SA is better than hybrid GAs considering convergence rate. On the other hand, a hybrid GA is better than SA considering accuracy. It is difficult to decide which one is the better optimization approach. But we thought SA is a more proper optimization algorithm for the system because its behavior is more similar to human's thinking.

References

[1] C. C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller, Part I ", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 20, pp. 404-418, 1990.

[2] C. C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller, Part I ", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 20, pp. 419-435, 1990.

[3] C. T. Chao, Y. J. Chen, and C. C. Teng, "Simplification of Fuzzy-Neural Systems Using Similarity Analysis", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, pp. 344-354, 1996.

[4] D. E. Glodberg. "Genetic Algorithms in Search, Optimization, and Machine Learning", *Addison-Welsey*, pp. 59-83, 1989.

[5] J. S. R. Jang, C. T. Sun and E. Mizutani, "Neuro-Fuzzy and Soft Computing", *Prentice-Hall*, pp. 73-85, 1997.

[6] L. X. Wang., "A Course in Fuzzy Systems and Control", *Prentice-Hall*, pp 208-213, 1997.

[7] R. Jain, R, Kasturi and B, G, Schunck, "Machine Vision", *McGraw-Hill*, pp. 30-33, 1995.

[8] R. M. Haralick, L. G. Shapiro, "Computer and Robot Vision", *Addison Wesley*, 1992.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Fuzzy Identification of Chaotic and Complex Behavior of Human Operator Stabilizing an Inverted Pendulum on a Cart

Yoshihiko Kawazoe
Dept. of Mechanical Engineering, Saitama Institute of Technology,
1690 Okabe-machi, Ohsatogun, Saitama 369-0293 Japan. E-mail: ykawa@sit.ac.jp

In order to stabilize the inherent unstable system like the inverted pendulum on a cart, severe judgment of situation is required. Accordingly, it can be expected that the human operators exhibit a complex behavior occasionally. This paper tries to identify the individual difference and skill up process of human behavior by using fuzzy inference in order to investigate the chaotic behavior of human operator and the possibility of the formation of complex system in the learning process of the human operator with difficult control objects. The operators in the experiment are skilled to some extent in stabilizing the inverted pendulum by training, and the data of ten trials per person were successively taken for an analysis, where the waveforms of pendulum angle and cart displacement were recorded. The maximum Lyapunov exponents were estimated from experimental time series data against embedding dimensions. It was found that the operator's behavior indicates chaotic feature with the positive maximum Lyapunov exponent. It was also found that the human behaviors have a large amount of disorder according to the result of the estimated entropy from the time series data. Furthermore the rules identified for a fuzzy controller from time series data of each trial of each operator show well the human-generated decision-making characteristics with the chaos and the large amount of disorder.

Key Words: Chaos, Complexity, Fuzzy Identification, Fuzzy Control, Inverted Pendulum, Human Operator

1. INTRODUCTION

Machinery and human being are absolutely of different nature at the present situation. Most of the past research work have dealt with the linear characteristics of human behavior in the man-machine system [1]. Many studies on control systems for stabilizing the inherent unstable system like the inverted pendulum on a cart, have been also presented in the past. However, there seem to be few studies and a number of unknown points regarding the nonlinear characteristics of human behavior in the man-machine system as well as in the learning process of the human operator with difficult control objects [2]-[4].

In order to stabilize the unstable system like the inverted pendulum, the severe judgment of situation is required. Accordingly, it can be expected that the human operators exhibit a complex behavior occasionally. In the author's previous papers [5]-[9], it was found that there are various nonlinear features in the stabilizing behavior of human operator. Being nonlinearly stable or stabilizing in this study means that the inverted pendulum does not fall down for 60 seconds.

This study investigates the chaotic behavior of human operator exhibited in stabilizing an inverted pendulum on a cart, which can move along a sliding rail of limited length. The operators in the experiment are skilled to some extent in stabilizing the inverted pendulum by training, and the time series data of ten trials per person were successively taken for an analysis. The maximum Lyapunov exponents might be estimated from time series data against embedding dimensions. It also investigates the possibility of formation of a complex system in stabilizing control of an inverted pendulum by a human operator. The entropy is estimated from the measured time series data. The entropy may be interpreted as a measure of the amount of disorder in the system and the maximum entropy means a random process with a uniform probability. Furthermore the rules are identified for a fuzzy controller from time series data of each trial of each operator to show the chaotic behavior of human-generated decision-making with the amount of disorder.

2. CHAOTIC BEHAVIORS AND FORMARTION OF COMPLEXITY DURING STABILIZING CONTROL BY A HUMAN OPERATOR

Figure 1 shows the experimental situation. The inverted

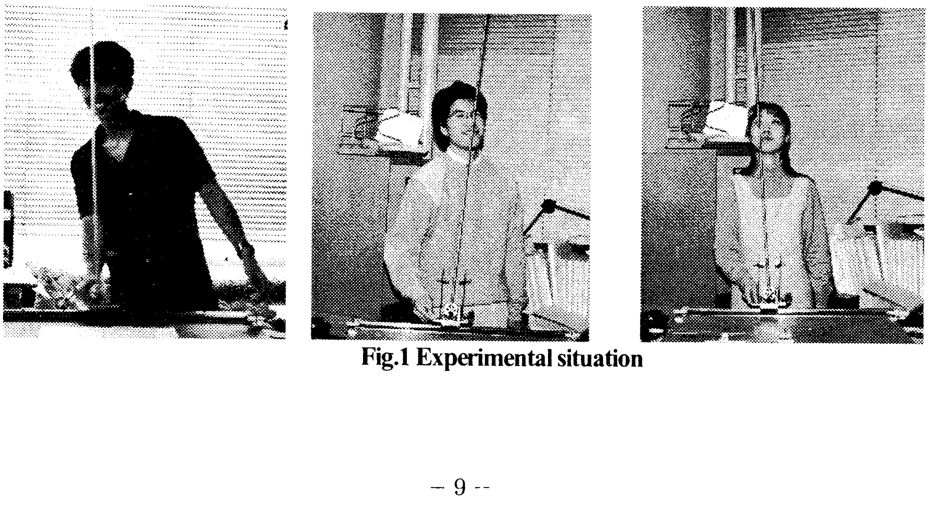
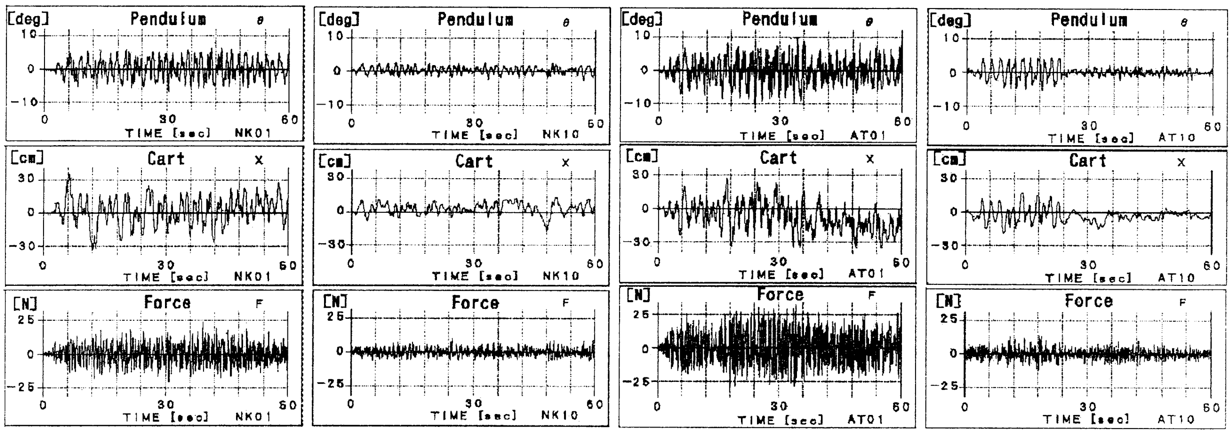


Fig.1 Experimental situation



(a) NK's 1st Trial: NK01 (b) NK's 10th Trial: NK10 (c) AT's 1st Trial: AT01 (d) AT's 10th Trial: AT10

Fig.2 Measured waveforms of pendulum angle, cart displacement, and force applied to the cart during the stabilizing control of an inverted pendulum by human operators.

two sliding rails of limited length, being hinged to the cart so as to rotate in the plane. A human operator manipulates a cart directly by hand. Although it takes some time and is needed intensive training for a human operator to succeed in stabilizing the pendulum for 60 seconds, it is not so difficult after the first success of stabilizing. Figure 2 shows the measured wave forms of pendulum angle, cart displacement, and force stabilized by the human operators.

Methods for dynamical analysis of experimental time series are still developing, but a common method is a two-step process: (1) reconstruction of the strange attracter of the unknown dynamical system from the time series, and (2) determination of certain invariant quantities of the system from the reconstructed attracter. It is possible to glean the dynamics from a single time series without reference to other physical variables [10]. This concept was given a rigorous mathematical basis by Takens [11] and Mane [12]. Since the attracter dimension is unknown for experimental data, and therefore the required embedding dimension M is unknown, the dimension of the embedding space is increased by 1 until 15 in this study. The largest Lyapunov exponent can be obtained from a time series using an algorithm given by Wolf *et al.* [13]. If Lyapunov exponent is positive, near by trajectories diverge; the evolution is sensitive to initial conditions and therefore chaotic.

Consider a hypothetical statistical system for which the outcome of a certain measurement must be located on the unit interval. If the line is subdivided into N subintervals, we can associate a probability p_i with the ith subinterval containing a particular range of possible outcomes. The entropy of the system is then defined as

$$S = -\sum_{i=1}^{N_C} p_i \log_e p_i \qquad (1)$$

This quantity may be interpreted as a measure of the amount of disorder in the system or as the information necessary to specify the state of the system. If the subintervals are equally probable so that $p_i = 1/N$ for all i, then the entropy reduces to $S = \log_e N$, which can be shown to be its maximum value. Conversely, if the outcome is known to be in a particular subinterval, then $S = 0$, the minimum value [10][14]. We applied this formulation to the experimental time series by establishing N bins or subintervals of the unit interval into which the value of time series data may fall. We define the net entropy of S calculated with Eq.(1) and the entropy ratio of $S/\log_e N$.

3. FUZZY IDENTIFICATION OF CHAOTIC BEHAVIORS AND FORMARTION OF COMPLEXITY DURING STABILIZING CONTROL OF AN INVERTED PENDULUM BY HUMAN OPERATORS

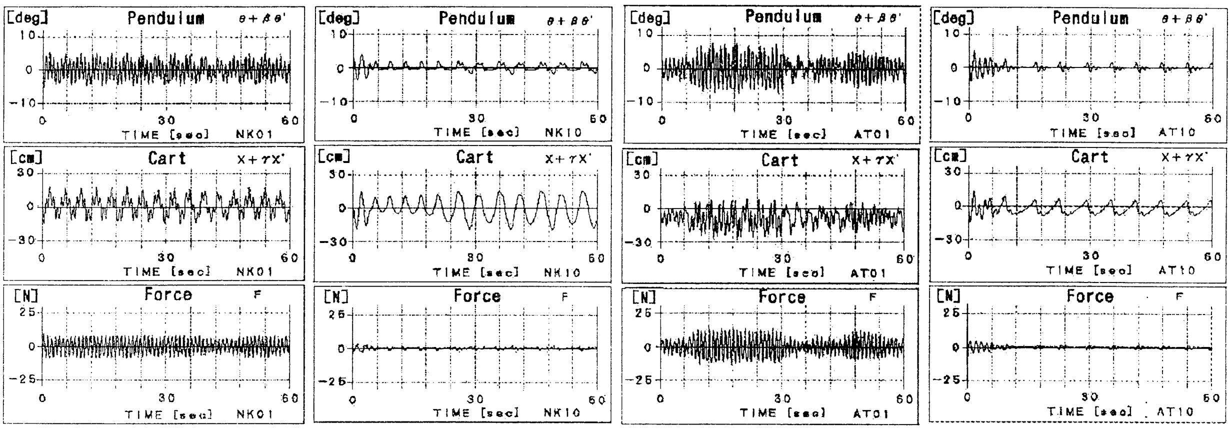
We choose the pendulum angle θ_t, angular velocity θ'_t, the cart displacement X_t and its velocity X'_t as input variables, and the force F_t that moves the cart as output of the fuzzy controller, trying to identify the nonlinear characteristics of the human operator from the experimental time series data. Furthermore, we choose the combined variables $\theta_t + \beta\theta'_t$ and $X_t + \gamma X'_t$ as inputs so as to eliminate the complexity of the control rule table. The β and γ are the combination variables. How to make the membership functions and the control rules were given by Kawazoe [8].

Figure 3 shows the simulated results using the fuzzy controller with rules and membership functions constructed from the experimental time series data by using fuzzy inference. Figure 4 shows the block diagram of stabilizing control simulation of the pendulum using the constructed fuzzy controller from human operators' time series data.

The simulated results exhibited the feature of those of each trial of the experiment.

Figure 5 shows the identified fuzzy rules constructed from the experimental time series data. during stabilizing control of an inverted pendulum by human operators.

Figure 6 shows the example of maximum Lyapunov exponents estimated from the simulated time series against embedding dimensions, being compared with those of the experimental.



(a) NK01 (β=0.0325, γ=0.100) (b) NK10 (β=0.06, γ=0.242) (c) AT01 (β=0.06, γ=0.22) (d) AT10 (β=0.045, γ=0.19)

Fig. 3 Simulated results with identified control rules from manual control data using fuzzy reasoning.

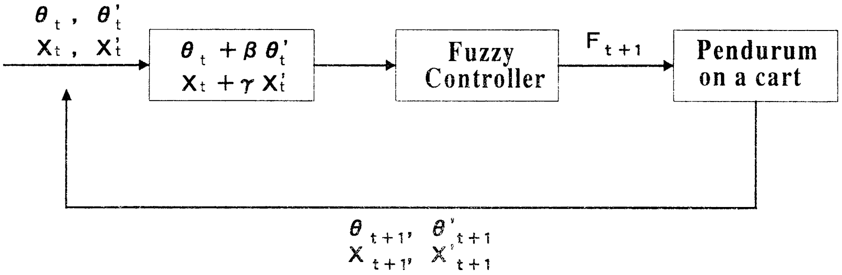


Fig.4 Block diagram of stabilizing control simulation of the pendulum using the constructed fuzzy controller from human operators' time series data.

		$\theta+\beta\theta'$				
		NB	NS	ZR	PS	PB
X+γX'	NB	PS	PMS	NMS	NMB	ZR
	NS	PMB	PMS	NMS	NMB	ZR
	ZR	PB	PS	ZR	NS	NB
	PS	ZR	PMB	PMS	NMS	NMB
	PB	ZR	ZR	PMS	NMS	NS

(a) NK01

		$\theta+\beta\theta'$				
		NB	NS	ZR	PS	PB
X+γX'	NB	PS	ZR	NMS	NS	ZR
	NS	PMB	PMS	NMS	NS	NS
	ZR	PMB	PS	ZR	NS	NMB
	PS	ZR	PS	NMS	NMS	NS
	PB	ZR	ZR	NMS	NMS	NS

(b) NK10

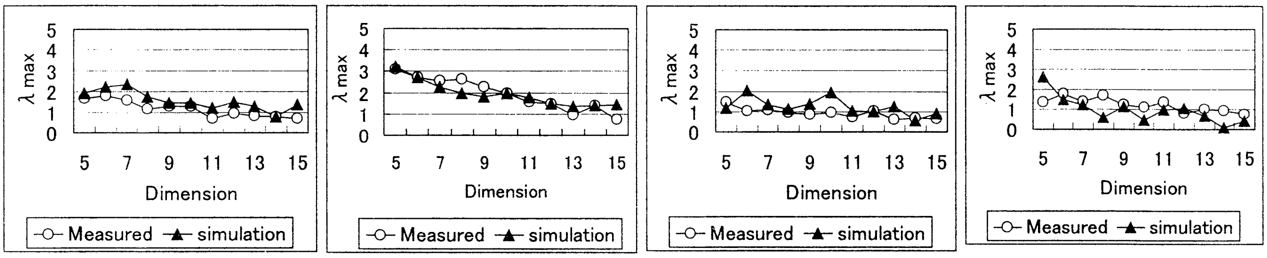
		$\theta+\beta\theta'$				
		NB	NS	ZR	PS	PB
X+γX'	NB	PMB	PMS	ZR	ZR	ZR
	NS	PB	PS	NMS	ZR	ZR
	ZR	PB	PMB	ZR	NMB	NB
	PS	ZR	PB	PMS	NS	NMB
	PB	ZR	ZR	PMS	NMS	NMB

(c) AT01

		$\theta+\beta\theta'$				
		NB	NS	ZR	PS	PB
X+γX'	NB	PMB	ZR	NS	ZR	ZR
	NS	PMB	PS	NS	NB	ZR
	ZR	PMB	PS	ZR	NS	NB
	PS	ZR	PMB	PMS	NMS	NMB
	PB	ZR	PB	PMS	NMS	NMB

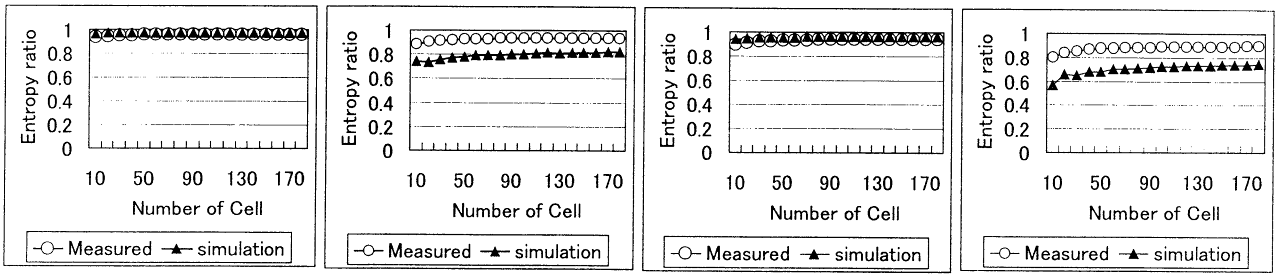
(d) AT10

Fig.5 Identified fuzzy rules from manual control data for stabilizing control of an inverted pendulum on a cart.



(a) NK01 (b) NK10 (c) AT01 (d) AT10

Fig.6 Maximum Lyapunov Exponent.



(a) NK01 **(b) NK10** **(c) AT01** **(d) AT10**

Fig.7 Ratio of entropy against maximum entropy.

Figure 7 shows the entropy ratios of the simulation being compared with those of the experiment.

The result indicates that the rules identified for a fuzzy controller from time series data of each trial of each fairly skilled operator show well thc human-generated decision-making characteristics with the chaos and the large amount of disorder. It is seen that the individual difference and skill up process of chaotic and complex human operation could be identified with fuzzy inference.

4. CONCLUSION

This paper investigated the chaotic behavior of human operator stabilizing an inverted pendulum on a cart. The maximum Lyapunov exponents were estimated from experimental time series data against embedding dimensions. It was found that the operator's behavior indicates chaotic feature with the positive maximum Lyapunov exponent. It was also found that the human behaviors have a large amount of disorder according to the result of the estimated entropy from the time series data. Furthermore the rules identified for a fuzzy controller from time series data of each trial of each operator show well the human-generated decision-making characteristics with the chaos and the large amount of disorder.

It was seen that the individual difference and skill up process of chaotic and complex human operation could be identified with fuzzy inference.

The author is grateful to Mr. K. Tanaka for his suggestions and encouragement. Sincere thanks are extended to Messrs K. Hashimoto, T. Ohta, N. Suyama and M. Sekine for their help in carrying out this study when senior students at Saitama Institute of Technology.

This work was supported by the High- Tech Research Center of Saitama Institute of Technology.

REFERENCES

[1] Iguchi M (1970), Man-machine system (in Japanese), Kyoritsu Shupan.

[2] Ito K. & Ito M (1976), On the human operator's learning process and nonlinear behavior in stabilizing an unstable system (in Japanese), Trans. IEE of Japan, Vol.96, No.5, pp.109-115.

[3] Fujii K & Taguchi J (1981), The role of human controller in man machine system (in Japanese), System and control, Vol.25, No.6, pp.328- 335.

[4] Inooka H (1991), Swinging control of a pendulum: manual control approrch, Theoretical and Applied Mechanics, Vol.40, pp.3- 9.

[5] Kawazoe Y (1992), Manual control and computer control of an inverted pendulum on a cart, Proc. of the First International Conf. on Motion and Vibration Control, pp.930 -935.

[6] Kawazoe Y & Ju DY (1994), Nonlinear characteristics of human operator with the stabilizing control of an inverted pendulum on a cart, Proc. 2nd International Conf. on Motion and Vibration Control, pp.645- 650.

[7] Kawazoe Y, Ohta T, Tanaka K & Nagai K (1997), Nonlinear behavior in stabilizing control of an inverted pendulum on a cart by a human operator: remarks on chaotic behavior and a complex system (in Japanese), Proc. 5th symposium on Motion and Vibration Control, No.97-31, pp.395 - 398.

[8] Kawazoe Y (1999), Nonlinear characteristics of a human operator during stabilizing control of an inverted pendulum on a cart: Fuzzy identification from experimental time series data and Fuzzy control simulation, Motion and vibration control in Mechatronics, Edited by Seto K, Mizuno T & Watanebe T, pp.133-138.

[9] Kawazoe Y (2000), Measurement and analysis of chaotic behavior of human operator stabilizing an inverted pendulum on a cart, Proc. of ICMA2000-Human friendly mechatronics, pp.457-462.

[10] Baker GL and Gollub JP (1996), Chaotic dynamics: an introduction, Cambridge University Press.

[11] Takens F (1981), Detecting strange attracters in turbulence, In Rand DA and Young LS (ed), Lecture Notes in Mathematics, Vol.898, Springer-Verlag, Berlin, 366-81.

[12] Mane R (1981), On the dimension of the compact invariant sets of certain nonlinear maps. In Rand DA and Young LS (ed), *ibid*, Vol.898, Springer-Verlag, Berlin, 230-42.

[13] Wolf A, Swift JB, Swinney HL, and Vastano JA (1985), Determining Lyapunov exponents from a time series, Physica, 16D, 285.

[14] Baierlein R (1971), Atoms and information theory, W.H.Freeman & Co., San Francisco, Chapter 3.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Adaptive Visual Control of a SM5 Robot with a Hand-Eye Coordinated Camera

H. D. Kim[1] , D. Y. Jeong[2] , S. H. Han[2], M. H. Lee[3] and H. Hashimoto[4]

1. Dept. of Mold Design of Yenam Collage, Chinju, Korea, Tel : +82-591-751-2075
2. Division of Mechanical and Automation Eng., Kyungnam Univ., Masan, 631-701, Korea.
   E-mail: shhan@kyungnam.ac.kr Tel: +82-551-49-2624 Fax: +82-551-43-8133
3. School of Mechanical Engineering, Pusan National Univ., Pusan, 609-735, Korea.
   E-mail: mahlee@hyowon.cc.pusan.ac.kr Tel: +82-51-510-1456 Fax: +82-51-510-2331
4. Institude of Industrial Science, Univ. of Tokyo, Japan

ABSTRACT

In this paper, we present a visual tracking structure combining image information and the control of a robot arm in order to track moving target. We use the sum of square difference optical flow method to figures in each sampling time. This data is converted into a velocity command that is used to control the robot arm by a self-tuning regulator. Finally, the proposed algorithm was verified on a SCARA robot with 4-axis arm.

1. INTRODUCTION

High automation and production are the goals of today's industry because time is money. Traditional processing, especially in the assembly of products, wastes much time. However, a robot with visual ability can quickly adapt to the requirements of unknown work and make proper responses to sudden changes of environment. This makes the robot more flexible.

Many scholars have used dynamic visual feedback to implement the visual tracking control of a robot arm, which combines the image signal of a target and the control commands of a robot arm. The control architecture can be divided into two classes via certain differences of feedback image information. One trance forms the visual signal into three dimensional spatial information which is used directly as a control signal of the system. The other uses two dimensional image information directly as a control signal of the system. From the classification of Sanderson and Weiss [6], the former is called the Position-based method and the latter is called the Feature-based method.

In general, the choice of whether to use the Position-based method or Feature-based method in a visual tracking system depends on the position of the camera. If the camera is fixed on the ground (normally requiring two camera) we usually use the Position-based method. If a single camera is mounted on the end-effector of a robot arm, we usually use the Feature-based method. For one camera, 2-D image information can not be transformed into 3-D space information. In this paper, the camera is mounted on the end-effector, so we use the Feature-based method.

For the system controller, we make reference to the method of Papanikolopoulos [5], which uses the SCARA robot model. Improving on this method, however, we consider the problem of system coupling. Using an adaptive algorithm, we compute the velocity value of a moving robot arm from the measured image signal and then generate a velocity command to drive the robot arm.

This paper is organized into eight section. In section 2, the adaptive algorithm is introduced. In section 3, we offer an easy way to calibrate the parameters of the camera and then present a brief introduction to image processing. In section 4, we derive the mathematical model of the overall visual tracking system and the visual tracking theory. the controller of robot arm is introduced in section 5. In section 6, the visual tracking theory and the controller of robot arm are implemented in terms of our hardware and experimental results are presented. Finally, in section 7, the conclusion is summarized.

2. CONTROL ALGORITHM

The proposed adaptive control system is different from other fixed control systems because it can adapt to unpredictable changes from the inner system or the environment. It also has high adaptability for system uncertainty and errors in design and measurement. In visual tracking system, there is unavoidable noise in the image signal, and it is difficult to calibration the camera parameters. Therefore, we chose an adaptive controller for our visual tracking system.

This section describes the proposed algorithm for system parameter estimation and the proposed algorithm for minimum variance control of MIMO systems:

We will use the four-axies robot model to express the system dynamics. The robot model is expressed as follows:

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) + C(q^{-1})n(k) \qquad (1)$$

where

$u(k)$: system input
$y(k)$: system output
$n(k)$: noise input, which is independent of $n(k-1)$, $u(k)$ and $y(k-1)$; $n(k)$ is white noise. i.e. zero mean $E[n(k)] = 0$
m : system order

We express the equation in discrete from;

$$\begin{aligned} y(k) = & -a_1 y(k-1) - a_2 y(k-2) \cdots - a_m y(k-m) \\ & + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) \cdots + b_m u(k-m) \\ & + c_0 n(k) + c_1 n(k-1) + \cdots + c_m n(k-m) \end{aligned} \qquad (2)$$

If $c_0 = c_1 = c_2 = c_3 = \cdots\cdots = c_m = 0$,

$$\begin{aligned} y(k) = & -a_1 y(k-1) - a_2 y(k-2) \cdots - a_m y(k-m) \\ & + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) \cdots + b_m u(k-m) \end{aligned} \qquad (3)$$

Equation (3) is the robot model equation.

This paper uses the recursive least squares algorithm to estimate the parameters a_i and b_i If $c_1, c_1, c_2, \cdots\cdots c_m$ are not all equal to zero, the parameters include $c_1, c_1, c_2, \cdots\cdots c_m$, which express the noise dynamics. This algorithm is called the extended least squares parameter estimation. It is expressed as:

$$\underline{\hat\theta}(k+1) = \underline{\hat\theta}(k) + \underline{R}(k) e_{k+1}$$

$$\underline{R}(k) = \underline{\underline{P}}(k)\ \underline{\phi}^T(k+1)[1 + \underline{\phi}(k+1)\ \underline{\underline{P}}(k)\ \underline{\phi}^T(k+1)]^{-1} \qquad (4)$$

$$\underline{\underline{P}}(k+1) = [\ \underline{\underline{I}} - \underline{R}(k)\ \underline{\phi}(k+1)]\ \underline{\underline{P}}(k)$$

$$e_{k+1} = \hat y_{k+1} - \underline{\phi}(k+1)\ \underline{\hat\theta}(k) \qquad (5)$$

where

$$\underline{\hat\theta}(k) = [a_1 \quad a_2 \quad \cdots \quad a_m \ b_0 \ b_1 \quad \cdots \quad b_m \ c_0 \ c_1 \quad \cdots \quad c_m]^T$$

$$\begin{aligned} \underline{\phi}(k) = [&-y(k-1) \quad -y(k-2) \quad \cdots \quad -y(k-m) \\ & u(k) \quad u(k-1) \quad \cdots \quad u(k-m) \\ & n(k) \quad n(k-1) \quad \cdots \quad n(k-m)] \end{aligned}$$

We use the SCARA robot model to express the system dynamics and consider the system tome delay, d, as follows:

$$\underline{\underline{A}}(q^{-1})\ \underline{y}(k) = q^{-d}\ \underline{\underline{B}}(q^{-1})\ \underline{u}(k) + \underline{\underline{C}}(q^{-1})\ \underline{n}(k) \qquad (6)$$

where

$\underline{y}(k)$, $\underline{u}(k)$, $\underline{n}(k) \in m \times 1$, $\underline{\underline{A}}(q^{-1})$, $\underline{\underline{B}}(q^{-1})$, $\underline{\underline{C}}(q^{-1})$ are $m \times m$ polynomial matrices and all the zeros of $\underline{\underline{C}}(q^{-1})$ are inside the unit circle.

We can get the best predictive value of system output as follower:

$$\underline{\underline{C}}(q^{-1}) = \underline{\underline{A}}(q^{-1})\ \underline{\underline{F}}(q^{-1}) + q^{-d}\ \underline{\underline{G}}(q^{-1}) \qquad (7)$$

$$\underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{C}}(q^{-1}) = \underline{\underline{\bar C}}(q^{-1})\ \underline{\underline{F}}(q^{-1}) \qquad (8)$$

$$\underline{\underline{\bar G}}(q^{-1}) = q^d[\ \underline{\underline{\bar C}}(q^{-1}) - \underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{A}}(q^{-1})] \qquad (9)$$

where

$$\underline{\underline{F}}(q^{-1}) = \underline{\underline{I}} + \underline{\underline{f}}_1 q^{-1} + \cdots\cdots + \underline{\underline{f}}_{d-1} q^{-d+1}$$

$$\underline{\underline{G}}(q^{-1}) = \underline{\underline{g}}_0 + \underline{\underline{g}}_1 q^{-1} + \cdots\cdots + \underline{\underline{g}}_{n-1} q^{-n+1}$$

$$\underline{\underline{f}}_0 = \underline{\underline{I}},\ det\ \underline{\underline{C}}(z) = det\ \underline{\underline{\bar C}}(z)$$

Multiplying both sides of equation(7) by $\bar F(q^{-1})$, we get

$$\begin{aligned} & \underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{A}}(q^{-1})\ \underline{y}(k) \\ & = q^{-d}\ \underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{B}}(q^{-1}) + \underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{C}}(q^{-1})\ \underline{n}(k) \end{aligned} \qquad (10)$$

Substituting equation (8) and (9) into equation (10) yields

$$\begin{aligned} & [\ \underline{\underline{\bar C}}(q^{-1}) - q^{-d}\ \underline{\underline{\bar G}}(q^{-1})]\ \underline{y}(k) \\ & = q^{-d}\ \underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{B}}(q^{-1})\ \underline{u}(k) + \underline{\underline{\bar C}}(q^{-1})\ \underline{\underline{F}}(q^{-1})\ \underline{n}(k) \end{aligned}$$

Therefore,

$$\begin{aligned} & \underline{\underline{\bar C}}(q^{-1})\ [\ \underline{y}(k) - \underline{\underline{F}}(q^{-1})\ \underline{n}(k)] \\ & = q^{-d}\ \underline{\underline{\bar G}}(q^{-1})\ \underline{y}(k) + q^{-d}\ \underline{\underline{\bar F}}(q^{-1})\ \underline{\underline{B}}(q^{-1})\ \underline{u}(k) \end{aligned} \qquad (11)$$

Define:

$$\underline{y}^0(k/k-d) = \underline{y}(k) - \underline{\underline{F}}(q^{-1})\ \underline{n}(k) \qquad (12)$$

Via equation(11), we get

$$\begin{aligned} & \underline{\bar C}\ (q^{-1})\ \underline{y}^0(k+d/k) \\ & = \underline{\bar G}(q^{-1})\ \underline{y}(k) + \underline{\bar F}(q^{-1})\ \underline{B}(q^{-1})\ \underline{u}(k) \end{aligned} \qquad (13)$$

From equation (13), we know that $\underline{y}^0(k+d/k)$ and the measurement value, $\underline{Y}^k = \{\ \underline{y}(k),\ \underline{y}(k-1)\cdots\cdots\}$, are dependent.

Therefore,

$$\begin{aligned} \underline{y}^0(k+d/k) & = E\left\{\ \underline{y}^0(k+d/k)\ \ \underline{Y}^k\right\} \\ & = E\left\{\ \underline{y}(k+d) - \underline{\underline{F}}(q^{-1})\ \underline{n}(k+d)\ \ \underline{Y}^k\right\} \\ & = E\left\{\ \underline{y}(k+d) - \sum_{j=0}^{d-1} \underline{\underline{f}}_j\ \underline{n}(k+d-j)\ \ \underline{Y}^k\right\} \\ & = E\left\{\ \underline{y}(k+d)\ \ \underline{Y}^k\right\} \end{aligned} \qquad (14)$$

From equation(14), we know that $\underline{y}^0(k+d/k)$ is the best d-step predictive value of $y(k)$.

The objective of a minimum variance controller for a MIMO system is to minimize the mean square error of the reference input and system output. We define the performance index as:

$$\begin{aligned} J(\ \underline{u}(k)) = E\{\ [\ \underline{y}(k+d) - \underline{r}(k+d)]^T \\ [\ \underline{y}(k+d) - \underline{r}(k+d)]\} \end{aligned} \qquad (15)$$

where, $\underline{r}(k) \in m \times 1$, is the reference input.

The condition which satisfies the minimum performance index is:

$$\frac{\partial J(\ \underline{u}(k))}{\partial\ \underline{u}(k)} = 0 \qquad (16)$$

In order to satisfy equation(16):

$$\underline{y}^0(k+d/k) = \underline{r}(k+d) \qquad (17)$$

Substituting equation (13) into equation (17), we derive the minimum variance control rule:

$$\underline{\underline{\overline{C}}}(q^{-1})\ \underline{r}(k+d) = \underline{\overline{G}}(q^{-1})\ \underline{y}(k) + \underline{\underline{\overline{F}}}(q^{-1})\ \underline{\underline{B}}(q^{-1})\ \underline{u}(k) \tag{18}$$

3. CAMERA MODEL AND IMAGE PROCESSING

It is difficult to determine exactly the image coordinates via the image information because of the moving target and the image noise. Especially, short image processing time is required for stability. Thus we use the sum of squared difference method to find the image coordinates of the target quickly.

Define a point $p(k-1)=(u(k-1), v(k-1))$ in image $(k-1)$ and a point $p(k)=(u(k), v(k))$ We assume that the intensity value in the neighborhood N of $p(k-1)$ remains a constant value, $p(k)$, over time. i.e

$$\underset{(u,\,v)\in N}{I}(u(k-1), v(k-1)) = \underset{(u,\,v)\in N}{I}(u(k), v(k))$$

where $I(u(k), v(k))$ is the gray level value of point (u, v) in image (k).

For any point $p(k-1)=(u(k-1), v(k=-1))$, we can find a displacement, $d=(s, t)$, so that the SSD index function is minimized, the SSD index function being

$$e(p(k-1), d) = \sum_{(m,\,n)\in N}[I(u(k-1)+m, v(k-1)+n) - I(u(k-1)+m+s, v(k-1)+n+t)]^2 \tag{19}$$

where $(s, t)\in\Omega$, Ω is the maximum possible range of displacement and $(m, n)\in N$, N being the neighborhood of point p. The magnitude of Ω is dependent on the velocity of the target. If the target moves more quickly, the size of Ω is larger and of course the SSD computation time is larger. Therefore, computation time sets a limit on the velocity the system can successfully track, and there is a trade off between trackable velocity and computation time. Given limited computer capabilities, design approaches must emphasize speedy algorithms. A complementary approach, however, utilizes computational hardware improvements such as faster computers and parallel processing networks. Our proposed method uses both algorithm and hardware improvements, as will be seen.

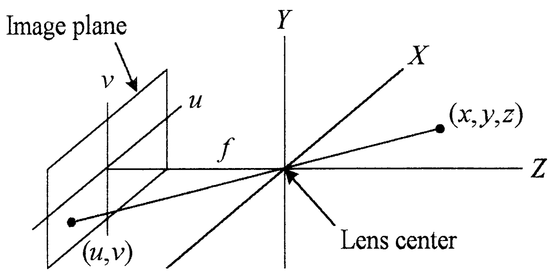


Fig. 1. The projection relationship.

For calibration of camera parameters, the origin of the camera coordinates is at the center of the lens, and the Z axis is parallel to the optic axis, as seen in figure1. From figure1, we have

$$\frac{x}{uS_u} = \frac{y}{vS_v} = \frac{z}{f} \tag{20}$$

which can be expressed as

$$\left\{x = uS_u\left(\frac{z}{f}\right),\ y = vS_v\left(\frac{z}{f}\right)\right\} \tag{21}$$

where

f : the focal length of camera

S_u, S_v : the pixel's unit length along the u, v direction in the image plane(mm/pixel)

It is possible that (u, v) is negative from figure 1. We give an offset such that the value (u, v) in the computation is positive.

$$u = u_t - c_u,\quad v = v_t - c_v \tag{22}$$

where

(u_t, v_t) : actual coordinate value

(c_u, c_v) : offset value.

As in figure 2. we set a coordinate B on .the rigid body. The velocity of point Q relative to the camera coordinate (otherwise known as the transform matrix) is

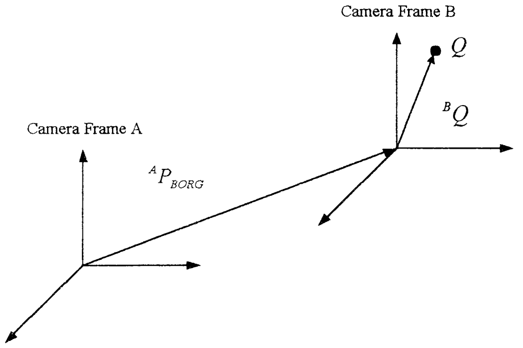


Fig. 2. Relative motion of two rigid bodies.

$$^AV_Q = {}^AV_{BORG} + {}^AR_B\,{}^BV_Q + {}^A\Omega_B \times {}^AR_B\,{}^BQ \tag{23}$$

Because coordinate B is fixed on the rigid body, the relative velocity ^VQ is zero. Equation(23) can be expressed as

$$^AV_Q = {}^AV_{BORG} + {}^A\Omega_B \times {}^AR_B\,{}^BQ \tag{24}$$

Let $^AV_{BORG}=[k_x\ k_y\ k_z]^T$ and $^A\Omega_B=[\omega_x\ \omega_y\ \omega_z]^T$

Equation(24) can then be expressed as

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \times \begin{bmatrix}x\\ y\\ z\end{bmatrix} + \begin{bmatrix}k_x\\ k_y\\ k_z\end{bmatrix} = \begin{bmatrix}k_x+\omega_y z-\omega_z y\\ k_y+\omega_z x-\omega_x z\\ k_z+\omega_x y-\omega_y x\end{bmatrix} \tag{25}$$

where

$[x\ y\ z]^T$: the position matrix of point Q relative to the camera coordinate.

$[\dot{x}\ \dot{y}\ \dot{z}]^T$: the velocity matrix of point Q relative to the camera coordinate.

Substituting equation (21) into equation(25) :

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}k_x+\omega_y z-\omega_z vS_v\left(\frac{z}{f}\right)\\ k_y+\omega_z uS_u\left(\frac{z}{f}\right)-\omega_x z\\ k_z+\omega_x vS_v\left(\frac{z}{f}\right)-\omega_y uS_u\left(\frac{z}{f}\right)\end{bmatrix} \tag{26}$$

From equation(21), $u=\left(\frac{f}{S_u z}\right)x$ and $v=\left(\frac{f}{S_v z}\right)y$.

Differentiate u and v :

$$\dot{u} = \left(\frac{f}{S_u z}\right)\dot{x} - \left(\frac{fx}{S_u z^2}\right)\dot{z},$$
$$\dot{v} = \left(\frac{f}{S_v z}\right)\dot{y} - \left(\frac{fy}{S_v z^2}\right)\dot{z} \tag{27}$$

Substitute equation (21)and equation(26) into equation(27)

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \left(\frac{f}{S_u}\right)\frac{1}{z} & 0 & \frac{u}{z} & -uv\left(\frac{S_v}{f}\right) & \left(\frac{f}{S_u}\right)+u^2\left(\frac{S_u}{f}\right) & -v\left(\frac{S_v}{S_u}\right) \\ 0 & \left(\frac{f}{S_v}\right)\frac{1}{z} & \frac{v}{z} & -\left(\frac{f}{S_v}\right)-v^2\left(\frac{S_v}{f}\right) & uv\left(\frac{S_u}{f}\right) & u\left(\frac{S_u}{S_v}\right) \end{bmatrix} \begin{bmatrix} k_x \\ k_y \\ k_z \\ \omega_x \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{28}$$

Equation (28) is the mathematical model of optical flow. The values of $\left(\frac{S_u}{f}\right)$, $\left(\frac{S_v}{f}\right)$ and $\left(\frac{S_v}{S_u}\right)$ can be got from the camera calibration results. Of course, when the target moves and the camera is fixed, we should multiply equation(28) by a negative sign.

4. VISUAL TRACKING CONTROL OF ROBOT ARM

The objective of visual tracking control of a robot arm is to track any object, even objects with uncertain motion, via image-feedback information. The difficulties of using visual tracking control of a arm are not only the control of robot arm but also image processing. In order to use real-time control, we must operate these two systems simultaneously.

We have calibrated the parameters of the camera and constructed the camera model in section 3. In order to avoid uncertainly while constructing the mathematical model of the system and to minimize the problems of noise accompanying the image signal, we adopt adaptive control, which is described in section 2, to implement visual tracking control of the robot arm. The block diagram of the visual tracking system is in figure 3[5].

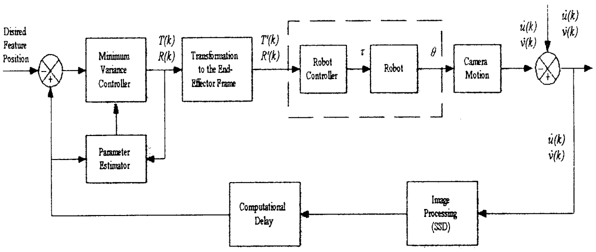


Fig. 3. Block diagram of adaptive visual tracking system.

According to figure 3, when the robot arm is tracking an object, the optical flow of the object's projection in the image plane results from the motion of the camera and the object. This change can be calculated by the SSD rule, which is introduced in section 3. When the image coordinates of the object are obtained, we calculate via the object's position in the image plane from changing significantly. In other words, we calculate the velocity of the end effector by using of the transform matrix(eq. 23) and the we use this data to drive the robot arm, thereby achieving visual tracking control.

From the block diagram for visual tracking of the robot arm, we can drive the mathematical model of the system as follows:

The object's motions in space include translation and rotation. Let

$$^{C}T_x = -k_x, \quad ^{C}T_y = -k_y, \quad ^{C}T_z = -k_z$$
$$^{C}R_x = -\omega_x, \quad ^{C}R_y = -\omega_y, \quad ^{C}R_z = -\omega_z$$

Substituting these values into equation (28) and transforming the resulting equation into the SCARA robot model, we obtain

$$\begin{aligned} A_u^{(j)}(q^{-1})u^{(j)}(k) = q^{-d}B_u^{(j)}(q^{-1})\Big\{ & -\left(\frac{f}{S_u}\right)\frac{1}{z^{(j)}(k)}\ {}^{C}T_x(k) \\ & + \frac{u^{(j)}(k)}{z^{(j)}(k)}\ {}^{C}T_z(k) + u^{(j)}(k)v^{(j)}(k)\left(\frac{S_v}{f}\right)\ {}^{C}R_x(k) \\ & - \left[\left(\frac{f}{S_u}\right) + [u^{(j)}(k)]^2\left(\frac{S_u}{f}\right)\right]\ {}^{C}R_y(k) \\ & + v^{(j)}(k)\left(\frac{S_v}{S_u}\right)\ {}^{C}R_z(k)\Big\} + C_u^{(j)}(q^{-1})n_u^{(j)}(k) \end{aligned} \tag{29}$$

$$\begin{aligned} A_v^{(j)}(q^{-1})v^{(j)}(k) = q^{-d}B_v^{(j)}(q^{-1})\Big\{ & -\left(\frac{f}{S_v}\right)\frac{^{C}T_y(k)}{z^{(j)}(k)} + \frac{v^{(j)}(k)}{z^{(j)}(k)}\ {}^{C}T_x(k) \\ & + \left[\left(\frac{f}{S_v}\right) + [v^{(j)}(k)]^2\left(\frac{S_v}{f}\right)\right]\ {}^{C}R_x(k) \\ & - u^{(j)}(k)v^{(j)}(k)\left(\frac{S_u}{f}\right)\ {}^{C}R_y(k) \\ & - u^{(j)}(k)\left(\frac{S_u}{S_v}\right)\ {}^{C}R_z(k)\Big\} + C_v^{(j)}(q^{-1})n_v^{(j)}(k) \end{aligned} \tag{30}$$

where $j = 1, 2 \cdot \cdot \cdot M$ indicates the image feature points of the object, j being a feature point and M being the total number of feature points. Because the system has six inputs $(T_x, T_y, T_z, R_x, R_y, R_z)$, we need six values to control this system's behavior entirely. Let $M = 3$, and then we can represent equation(29) · (30) in the MIMO (multi-input Multi-output) SCARA robot model. The value of each sampling time

$$\underline{y}(k) = [u^{(1)}(k)\ v^{(1)}(k)\ u^{(2)}(k)\ v^{(2)}(k)\ u^{(3)}(k)\ v^{(3)}(k)]^T$$

can be obtained according to each image feature point via the SSD rule, which is introduced in section 3.

$$\underline{\underline{A}}(q^{-1})\ \underline{y}(k) = q^{-d}\ \underline{\underline{B}}(q^{-1})\ \underline{u}(k) + \underline{\underline{C}}(q^{-1})\ \underline{n}(k)$$

where

$$\underline{y}(k) = [u^{(1)}(k)\ v^{(1)}(k)\ u^{(2)}(k)\ v^{(2)}(k)\ u^{(3)}(k)\ v^{(3)}(k)]^T$$
$$\underline{u}(k) = [^{C}T_x(k)\ ^{C}T_y(k)\ ^{C}T_z(k)\ ^{C}R_x(k)\ ^{C}R_y(k)\ ^{C}R_z(k)]^T$$
$$\underline{n}(k) = [n_x^{(1)}(k)\ n_y^{(1)}(k)\ n_x^{(2)}(k)\ n_y^{(2)}(k)\ n_x^{(3)}(k)\ n_y^{(3)}(k)]^T$$
$$\underline{\underline{A}}(q^{-1}) = \underline{\underline{I}} + \underline{\underline{a_1}}q^{-1} + \underline{\underline{a_2}}q^{-2}$$
$$\underline{A}(q^{-1}) = \underline{I} + \underline{a_1}q^{-1} + \underline{a_2}q^{-2}$$
$$\underline{\underline{B}}(q^{-1}) = \underline{\underline{b_0}} + \underline{\underline{b_1}}q^{-1}$$
$$\underline{\underline{C}}(q^{-1}) = \underline{\underline{c_0}} + \underline{\underline{c_1}}q^{-1} + \underline{\underline{c_2}}q^{-2}$$

and

$\underline{\underline{I}}$: 6×6 square matrix

$$\underline{\underline{a_1}} = -2\ \underline{\underline{I}}, \qquad \underline{\underline{a_2}} = \underline{\underline{I}}$$

$\underline{b_0} = - \underline{b_1}$

$\underline{c_0} = 0, \qquad \underline{c_1} = T^2 \underline{I}, \qquad \underline{c_2} = 0$

In this visual tracking system, we mount the camera on the end-effector of the robot arm and thus the transform matrix between the camera coordinates and end-effector coordinates is a fixed value. We use ${}^E R_C$ to indicate the rotation matrix and ${}^E T_C$ to indicate the translation vector between the camera and end-effector coordinates. We define the matrices for ${}^E R_C$ and ${}^E T_C$ as ${}^E R_C \in 3\times3$ and ${}^E T_C \in 3\times1$. ${}^C V$, ${}^C W$, ${}^E V$ and ${}^E V$ are used to indicate the translation velocity and angular velocity relative to camera coordinate C and end-effector coordinate E respectively.

Let ${}^C V_Q = [{}^C T_x \; {}^C T_y \; {}^C T_z]^T$, ${}^C W_Q = [{}^C R_x \; {}^C R_y \; {}^C R_z]^T$, ${}^E V_Q = [{}^E T_x \; {}^E T_y \; {}^E T_z]^T$and ${}^E W_Q = [{}^E R_x \; {}^E R_y \; {}^E R_z]^T$. Therefore, we get:

$${}^E W_Q = {}^E R_C \, {}^C W_Q \tag{31}$$

$${}^E V_Q = {}^E V_{CORG} + {}^E R_C \, {}^C V_Q + {}^E W_C \times {}^E R_C \, {}^C Q \tag{32}$$

Now, because the relative motion between the camera and end-effector is zero, i.e.

$${}^E V_{CORG} = {}^E \dot{T}_C = 0$$

then using above equation in equation(32), we obtain

$${}^E V_Q = {}^E R_C {}^C V_Q + {}^E W_C \times {}^E R_C {}^C Q$$

According to above equations, we can transform the velocity command to end-effector coordinates, giving the robot-arm drive the information it needs for the next step of end-effector positioning, thus accomplishing visual tracking control.

5. INTELLIGENT CONTROL SYSTEM DESIGN

A robot arm is classified as a highly complicated non-linear system. we introduce a control structure which combines the traditional computed torque method with a neural network. We use this neural network to compensate for the uncertainties of the control system.

In experimental implementation, an industrial robot arm is equipped with an additional degree of freedom, a driver-controlled mobility of the entire robot along the x-axis. Thus, the industrial robot arm has four degree of freedom. We use the computed torque method with the neural network compensation for uncertainties of the robotic manipulator. Our neural network has 12 input nodes and six output nodes. It requires a lot of calculation, which increases the sampling time period. As a consequence, it is not suitable for use in on-line learning. In order to compensate for the uncertainties during on-line building of the robot arm model, we take the couple of each axis of the robot arm into consideration. We also transform the neural network structure in figure 5 into three individual neural network structure. Each one is in charge of learning the uncertainties of one robot-arm axis in order to compensate for the insufficiencies of the computed torque method. In order to implement on-line learning, we use three neural networks, each containing a transputer, thus creating a parallel processing system which handles the calculation of each individual neural network and thus each axis simultaneously.

When we use the computed torque method, $\dot{\theta}^d$ in each sampling time can be obtained by applying the inverse Jacobean matrix on the velocity command in 3-D space. θ^d in each sampling time is computed via the following rule.

$$\theta^d(k) = \theta(k-1) + \dot{\theta}^d(k)\, T$$

where

θ^d , $\dot{\theta}^d$: the expected value of angle and angular velocity of each joint

θ^d : the current angle of each joint

T : sampling time

The entire hardware structure for the visual tracking system of our robot arm is presented in figure 4. We can divide it into three parts: SCARA robot arm system, image processing system and personal computer (PC), which communicate with the other two parts.

The SCARA robot arm was design by Samsung Electronics Co., Ltd. and requires a controller to control the robot arm's motion. For the original control system, we substituted our proposed control system. To permit the robot arm to receive commands from the controller, we use four encoders, four tachometers and four voltage commands lines, all of which are connected with tranputer's digital signal processing (DSP) cards. Then we use the transputer network to calculate and process this data in order to implement motion control of the robot arm. We keep only the power supply, motor driver, alarm system and the robot arm body. In addition, some of the simple, non-control elements of the original control system hardware, i. e. encoders, tachometer, etc. are utilized.

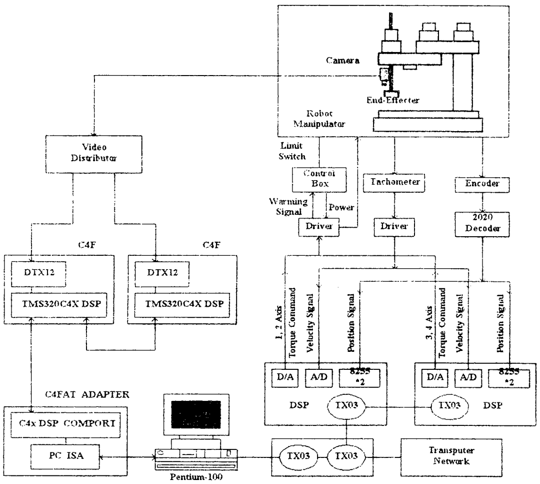


Fig. 4. Visual tracking hardware structure.

In the vision part of visual tracking system, in order to reduce the image processing time, we use a CCD camera, two C4F image processor and a video distributor to implement our vision feedback. After object is captured in the CCD camera and transformed into NTSC signals, we split and transfer this signal to the C4Fs via video distributor, and then transform the signal into a digital.

6. EXPERIMENT AND RESULTS

we implement the intelligent controller, which is mentioned in section 5, for the U-type robot arm. In order to verify the efficiency of our control rules, we test our experiment equipment in real time and 3-D space, implementing our visual tracking theory with a camera mounted on end-effector of the SCARA robot arm. In order to analyze the performance of the entire visual tracking system, we track an object that translates in a plane which is perpendicular to the camera's optical axis. Experiments that successfully included rotation were also performed but are not included herein for simplicity.

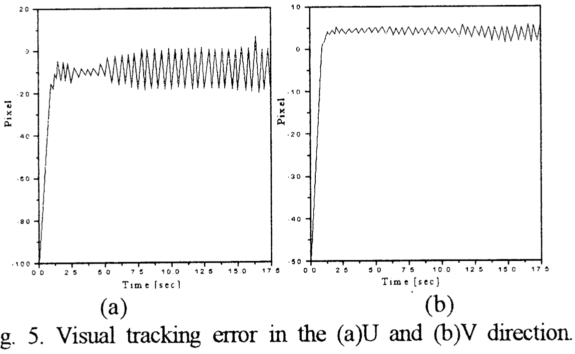
Before executing the visual tracking experiment, we test the practical implementation of the method in section 3 for calibrating the camera parameters. These parameters, as determined by real time system use, are shown in table 1.

Table. 1 Calibrated parameters of the camera.

c_u(pixel)	c_v(pixel)	f/S_u(pixel)	f/S_u(pixel)	S_u/S_v
230	460	2020	2447	1.2129

The goal of our experiment is the tracking of an object on a conveyer belt. The object moves linearly in a plane which is perpendicular to the camera's optical axis. Thus, the robot arm in this experiment needs only two degree of the freedom (x-axis and y-axis mobility) to track an object. Accordingly, this simple 2-D system requires the estimated value of only a single image feature point (u, v) to perform visual tracking. In this experiment, we choose the upper-left point of the object as the image feature point. If a different image feature point is desired, it must be specified by a change in our program. This could, of course, be made a menu option in a commercial program, but our current program is merely an experimental prototype. Before starting the program, the operator locates the object in the camera's field of view. The exact location is not critical. The operator starts the system and the rest of process proceeds automatically.

Figures 5 (a) and (b) are experimental visual tracking error figures in the U and V directions with the conveyer belt transporting the object at 2cm/sec. As can be seen form these figures, the performance of the robot-arm visual tracking system is reasonable. Although there is recurrent pixel error, if we convert this to real distance, the error is only about 5mm.



(a) (b)

Fig. 5. Visual tracking error in the (a)U and (b)V direction.

7. CONCLUSION

The main purpose of this paper is the creation of a system, using adaptive control theory with visual tracking control of a robot-arm, such that a robot arm can track any moving object in working space via a visual sensor. The robot-arm visual tracking system, which in this paper includes a camera mounted on the end-effector of the robot arm, uses adaptive control rules and on line mathematical model building. As a consequence, this system shows the following benefits:

Although system modeling needs pre-construction, we can compensate for uncertainties during system model building via adaptive control rules. Consequently, system modeling need not be exact.

In building a mathematical model of the visual tracking system, we assume noise disturbances and allow for a noise floor so that the system can get rid of noise disturbance.

Mounting the camera in the end-effector of the robot arm makes the entire visual tracking system an active system and thus shows able to do most kinds of tracking work.

REFERENCES

[1] A. C. Sanderson and L. E. Weiss, "Image-Based Visual Servo Control of Robot,"Robot Vision, 1983, pp.108-116

[2] N. P. Papanikolopoulous, P. K. Khosla and T. K. Fellow, "Visual Tracking of a Moving Target by a Camera Mounted on a Robot Vision: A Combation of Control and Vision." IEEE Trans. on Robotics and Automation, Vol. 9, No. 1, pp.14-35, 1993.

[3] Tomochika Ozaki, Tatsuya Suzuki, "Trajectory Control of Robotic Manipulators Using Neural Networks." IEEE Transactions on Industrial Electrontics, Vol.38, No.3, pp.195-202, June 1991

[4] Shigeru Okuma, Akio Ishiguro, "A-Neural Network Compensator for Uncertainties of Robotic Manipulators," IEEE Proc. 29th Conf. On Decision and Control, Honolulu, Haweii, pp.3303-3307, Dec 1990.

[5] A. J. Koivo and N. Houshangi, "Real-Time Vision Feedback For Servoing Robotic Manipulator With Self-Tuning Controller." IEEE Trans. on Systems, Man, and Cybernetics, VOL 21, NO.1, January, 1991, pp. 134-142

[6] P. Anadan, "Measuring visual motion from images sequences." COINS Dept. Univ. of Massachusetts, Tech. Rep. COINS-TR-87-21, 1987.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Dynamical Behaviors of Extended Hogg-Huberman Model

T. TANAKA[†], J. SHIBATA[†] , K. OKUHARA[†] and M. INOUE[††]

[†] Department of Management and Information Sciences,
Hiroshima Prefectural University,
Shobara, Hiroshima 727-0023, Japan.
[††] Department of Physics, Kagoshima University
Kagoshima 890-8580, Japan

Abstract

The Hogg-Huberman model is extended with respect to a time dependent reevaluation rate which relates to information about payoffs. Dynamical behaviors of an extended Hogg-Huberman model are studied numerically. We find that the change of fraction of agents using resource 1 is suppressed to small compared to the case of constant reevaluation rate.

1 Introduction

Recently, Huberman and Hogg [1] studied the dynamics of resource allocation in a model of computational ecosystems. They have shown that periodic or chaotic oscillations in the system can occur under certain conditions, and that these oscillation can effect the performance of the system. After this work, there have been done much efforts to investigate dynamical behaviors of multi-agent systems [2,3,4]. Hogg and Huberman [5] have proposed a method of controlling chaos in multi-agent systems compared of interacting agents making decisions based on imperfect and delayed information. They introduced a reward mechanism where the relative number of agents following effective strategies is increased. Hereafter we call this model the Hogg-Huberman model in this paper. Ushio and Imamori [6] have discussed dynamical properties of the discrete-time Hogg-Huberman model where the strategies are based on net bias.

In the Hogg-Huberman model, the reevaluation rate for agent's resource choice does not depend on time and is regarded as a constant even though the situation of systems is changed. Here we shall consider the case of time dependent reevaluation rate in the Hogg-Huberman model. It is natural to introduce the time-depending reevaluation rate into this system. The purpose of the present paper is to study dynamical behaviors of the Hogg-Huberman model associated with the time-dependent reevaluation rate which relates to the time development of system. After a brief summary of the discrete-time Hogg-Huberman model, we analyze the reevaluation rate with time and derive the dynamical equation of fraction of agents. Finally, the dynamical behaviors of our system are investigated numerically.

2 Discerete Time Hogg- Huberman Model

The discrete time Hogg-Huberman model

with two resorces [5,6] is given by

$$f_1(k+1) = f_1(k) + \alpha\{\rho_1(k) - f_1(k)\}, \quad (1)$$

where.

$$\rho_1(k) = \frac{1}{2}\left[1 + erf\left(\frac{z}{\sqrt{2}\sigma}\right)\right], \qquad (2)$$

$$z = G_1(f_1(k-\tau)) - G_2(f_1(k-\tau)), \qquad (3)$$

and $erf(x)$ denotes the error function of x. Here f_1 is the fraction of agents using resource 1 at discrete time k, and α is the ratio of agents which reevaluate the choice of resources to all the agent. $\rho_1(k)$ is the probability that an agent will prefer resource 1 over 2 when it makes a choice and is a function of f_1 through the payoffs. σ denotes the uncertainty of information and τ is a time delay of it. $G_1(f_1(k-\tau))$ and $G_2(f_1(k-\tau))$ are payoff functions for using resource 1 and resource 2, respectively. G_1 and G_2 are expressed by

$$\begin{aligned} G_1(f_1) &= 4 + 7f_1 - \frac{16}{3}\{f_1\}^2, & (4)\\ G_2(f_1) &= 4 + 3f_1. & (5) \end{aligned}$$

In case of $0 < \sigma \ll 1$, information is conveied precisely. When a becomes large, information cannot be conveied. Then fractions f_1 and f_2 approach to $\frac{1}{2}$ when $\sigma \to \infty$. According to the information, which resource gave agents much more benefit at time $(k-\tau)$, agents will prefer to the resource. For the sake of simplicity, we assume time delay $\tau = 1$ hereafter.

3 Time-Dependent Reevaluation Rate

Hogg and Huberman assumed that the rate α at which each agent reevaluates its choice is constant, i.e., the reevaluation rate does not depend on time. Here we introduce time-dependent reevaluation rates $\alpha_1(k)$ and $\alpha_2(k)$ for agents using resource 1 and resource 2, respectively. In our system the reevaluation rate $\alpha_1(k)$ for agents using resource 1 is different from that for agents using resource 2.

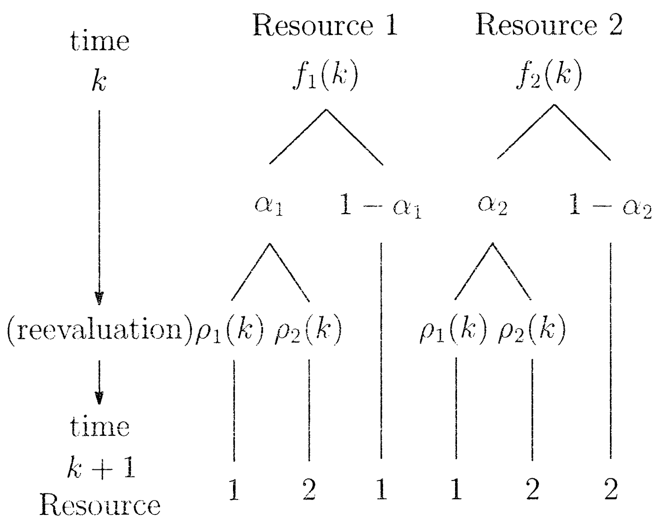


Fig.1. Rule of time development of fractions.

Thus dynamical for fraction of agents $(\rho_1(k) + \rho_2(k) = 1)$. Equation (1) can be rewritten as follows,

$$\begin{aligned} f_1(k+1) &= f_1(k) + \{\alpha_1(k) - \alpha_2(k)\} \\ &\quad \times\{\rho_1(k) - 1\}f_1(k) \\ &\quad +\alpha_2\{\rho_1(k) - f_1(k)\}. \qquad (6) \end{aligned}$$

We note that Eq. (6) becomes Eq. (1) when $\alpha_1(k) = \alpha_2(k) = \alpha$.

The reevaluation rates $\alpha_1(k)$ and $\alpha_2(k)$ are introduced as follows:

$$\alpha_1(k) = \frac{\lambda B_2(k)}{B_1(k) + B_2(k)}, \qquad (7)$$

$$\alpha_2(k) = \frac{\lambda B_1(k)}{B_1(k) + B_2(k)}, \qquad (8)$$

where λ is a factor of ecaluation, and

$$B_r(k) = f_r(k-\tau) \times G_r(f_r(k-\tau)) \quad (r = 1, 2). \quad (9)$$

$B_r(k)$ is the actual payoff received by agents using resource r at time $(k-\tau)$. The reevaluation rate $\alpha_1(k)$ for agents using resource 1 is assumed to be proportional to the actual payoff received by agents using resource 2 at time $(k-\tau)$ and vice versa. We note that $\alpha_1(k)$ and $\alpha_2(k)$ must be satisfied by the condition $\alpha_1(k) \leq 1$ and $\alpha_2(k) \leq 1$.

4 Numerical Results

In order to investigate the dynamical behaviors of the system, we calculate the bifurcation diagram of $f_1(k)$ in terms of σ. We have taken $f_1(0) = 0.5$ as a initial value and $\lambda = 0.95$. As shown in Fig.2, there are chaotic region, without of periodicity within the chaotic region and the fixed point depending on uncertainty σ. Typical behaviors for the fraction $f_1(k)$ of agents using resource 1 are calculated as a function of time k for time delay $\tau = 1$. When uncertainty is fairly small, the system shows chaotic oscillation as shown in Fig.3. The periodic oscillation within the window is shown in Fig.4. The chaotic oscillation of the Hogg-Huberman model is shown in Fig.5. As seen Fig.3 and Fig.5, we find that the amplitude of chaotic oscillation in our agent system is suppressed in comparison with that in the Hogg-Huberman model. This describes the effect of the time-dependent reevaluation rate on the system dynamics. If σ becomes large, then $f_1(k)$ converges at the equilibrium point which is given by the solution of the equation $G_1(f_1) = G_2(f_1)$. In this case we get the stable point $f_1 = 0.75$.

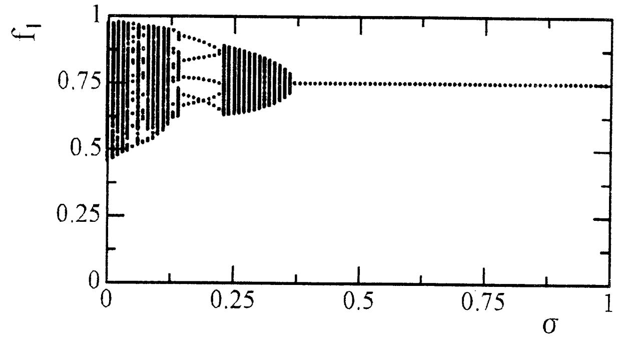


Fig.2. The bifurcation diagram of $f_1(k)$.

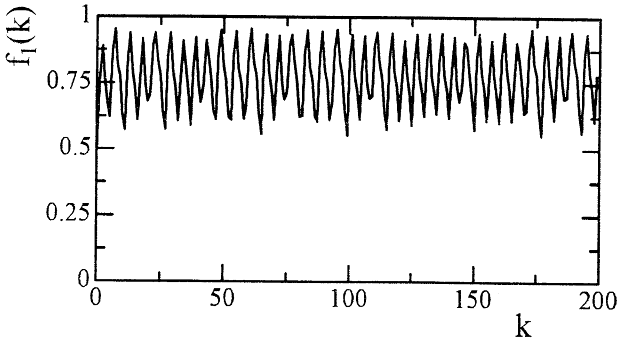


Fig.3. Chaotic oscillation ($\sigma = 0.10$).

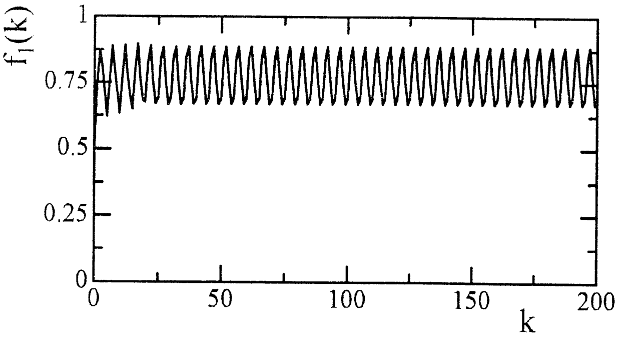


Fig.4. Periodic oscillation ($\sigma = 0.20$).

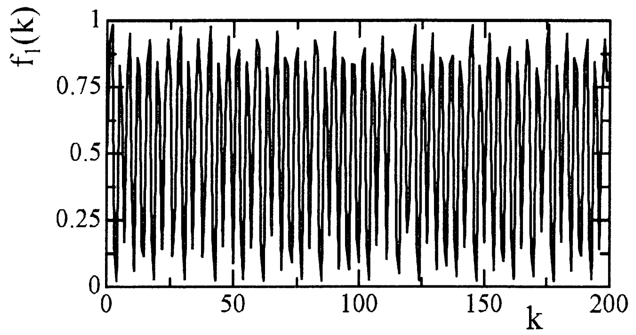


Fig.5. Chaotic Oscillation in the Hogg-Huberman model [5] ($\alpha = 0.85$ and $\sigma = 0.50$).

5 Summary

The dynamics of the discrete time Hogg-Huberman model which contains a time-dependent reevaluation rate has been investigated. We have calculated the time development of the fraction of agents using resource 1 for suitable values of parameters σ and λ in our system. We find that the amplitude of fraction of agents using resource 1 is suppressed to small compared to the case of the Hogg-Huberman model with the constant reevaluation rate.

References

[1] B. A. Huberman and T. Hogg, "The behavior of computational ecologies", in : *The Ecology of Computation*,ed. B. A. Huberman (North-Holland, Amsterdam, 1988) pp. 77-115.

[2] J. O. Kephart, T. Hogg and B. A. Huberman, "Dynamics of computation ecosystems", Phys. Rev. A40 (1989) 404-421.

[3] J. O. Kephart, T. Hogg and B. A. Huberman, "Dynamics of computation ecosystems ", Implication for DAI, in Distributed Artificial Intelligence, Vol. 2, ed. M. N. Huhns (Kaufman, Los Altos, CA. 1989).

[4] J. O. Kephart, T. Hogg and B. A. Huberman, "Collective behavior of predictive agents", Physica D42 (1990) 48-65.

[5] T. Hogg and B. A. Huberman, "Controlling Chaos in Distributed Systems", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6 (1991) 1325-1332.

[6] T. Ushio and T. Imamori, "Hogg-Huberman strategy with net bias in chaotic discrete-time computational ecosystem", Proc. 36th IEEE CDC, San Diego, CA, (1997) 389-394.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Chaos Oscillator Model of Temporal-Frequency Characteristics on Human Visual Search

○ Hiroaki MIZUHARA †, Takashi SAITO † and Jing-Long WU ‡

† Department of Mechanical Engineering, Faculty of Engineering, Yamaguchi University
2-16-1 Tokiwadai, Ube City, Yamaguchi 755-8611, Japan
E-mail: mizuhara@mina.mech.yamaguchi-u.ac.jp
‡ Department of Intelligent Mechanical Systems, Faculty of Engineering, Kagawa University
2217-20 Hayashi-cho, Takamatsu City, Kagawa 761-0396, Japan

Abstract

Human can get visual information in parallel through the vision. However, human cannot pay attention at the same time toward all the information. Human visual systems have the function of the visual search, which is the detection of unique information from many. If such a human function is realized in the artificial system that is significant to the automatic operation of the manufacturing process in the inspection. In this study, authors proposed a model, which describe the human function of the visual search in the temporal-frequency detection, using the chaos oscillator. Authors also discussed the propriety of the proposed model with comparing to the human characteristics of the visual search, measured by the psychological experiments.

Key Word: *Visual search, Cognitive Science, Chaos Engineering, Visual Attention, Early Vision*

1 Introduction

Human can get visual information in parallel through the vision. However, human cannot pay attention at the same time toward all the information. Human visual systems have the function of the "visual search"[1], which is the detection of unique information from many. If the target (a unique stimulus) is defined as distinguished from other stimuli (distracters) with only one feature, the target is found out at once, though the number of distracters are increased. Whereas, if the target is defined as distinguished from the distracters with two or more features, the target is difficult to find according with the number of distracters increased. In previous investigations, various models of visual search are proposed to describe such the function. These proposed models describe that the features are processed in parallel at the early vision[2)~4)]. However, these models do not describe the detail of the signal processing at the early vision, though describe the mechanisms of early vision.

Previous investigation reported that the chaotic signal dynamics are observed in the living brain, when the electroencephalogram was measured[5]. When the static state of the electric activity [illegible]ge to the chaotic state of that, it is reported that the process of inputted information is performed[5]. However, human visual search has been never described using such the chaotic dynamics.

In this study, we propose a model of the visual search using the chaos oscillator. At first, we measure the characteristics of the visual search in the temporal-frequency detection by the psychological experiments. Next, we propose a model describing the function of the visual search in it. We also discuss the propriety of the proposed model with comparing the characteristics of the proposed model with the human characteristics of the visual search measured by the psychological experiments. Proposed model in this study contributes to realize a function of human early vision. If the function of human vision is realized in the artificial system, this will be significant to the automatic operation of the manufacturing process in the inspection.

2 Psychological Experiment

2.1 Experimental Stimuli

In this psychological experiment, the target was defined as distinguished from the distracters with the temporal-frequency of the flickering. The subjects were required to indicate that the target was presented or was absented using the response keys pressing. Figure 1 shows an example of the experimental stimuli. A series of 32 images was presented in 1 [s]. Generally, in movies and TV broadcasting, a series of 24 ~ 30 images are presented in 1 [s][6]. Thus, number of the images in this psychological experiment is enough to make subjects to perceive the continuity of images. In this presentation, the target and distracters were flickered at 7 ~ 93 [cd/m^2] of luminance in sine wave. In this study, we used the 2 [Hz] and 4 [Hz] of the temporal frequency as the target, and the 1 [Hz] of that as the distracters.

Schematic illustrating of the detail of each stimulus is shown in Fig. 2. Each stimulus was defined as following equation:

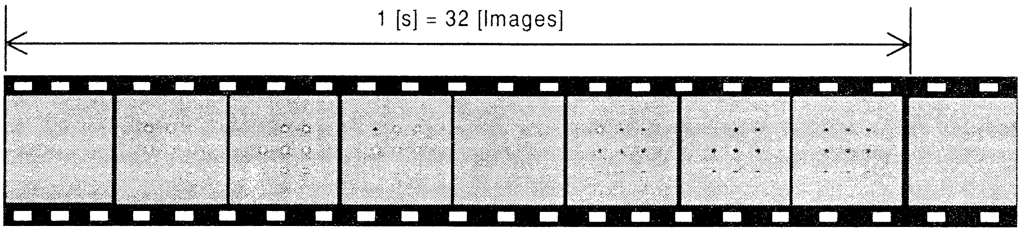


Fig. 1 An example of the experimental stimuli. A series of 32 images was presented continuously in 1 [s]

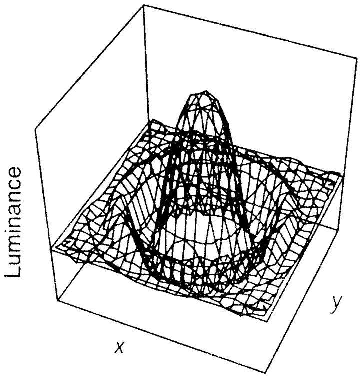


Fig. 2 Detail of the Gabor patch used in psychological experiment as the stimuli

$$L = k \times \cos\left(\frac{2\pi(x^2+y^2)}{r^2}\right) \times \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \quad (1)$$

Here, L is the luminance of the stimulus [cd/m^2], k is the revise coefficient, x and y is the spatial coordinates of each stimulus [deg], r is the radius of the stimulus [deg] and σ is the deviation of Gaussian function. In this study, these parameters were decided as the r=0.5[deg], σ=8 and L_{Max}=43[cd/m^2]. Thus, Eq. (1) show the 2D Gabor patch of 2 [c/deg] as the spatial frequency. Gabor patch is considered to process at first-order filters in human early vision and are often used in the investigations of the process in human early vision[7)-9)]. In this psychological experiment, the luminance of the stimulus [cd/m^2] was decided according to this Gabor patch, and was changed in sine wave with time.

In this psychological experiment, as mentioned above, size of each stimulus was set at 1[deg] (r=0.5[deg]). Size of the display frame was set at 10×10[deg], and distance between each stimulus was set at 1[deg]. Number of displayed stimuli in one frame (defined as 'Display Size' in this experiment) was set at 9, 16 and 25 as shown in Fig. 3. The target had two conditions that the target was flickered in 2 [Hz] (Condition 1) and in 4 [Hz] (Condition 2), and distracters were flickered in 1 [Hz]. In this psychological experiment, these conditions (Condition 1 and 2), display size and whether the target was appeared or not were presented at random order.

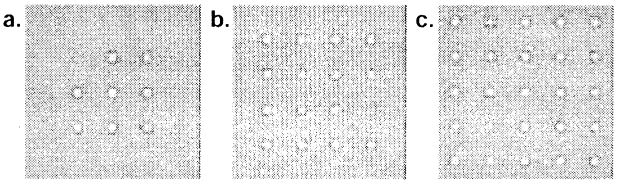


Fig. 3 Examples of the experimental stimuli after the time 225[msec] from starting the presentation. Display size was set at (a) 9, (b) 16 or (c) 25.

These experimental stimuli were created by personal computer (Sony PCG-XR9G), and presented to subjects through the CRT display (Nanao Flex Scan E35F) by the personal computer. In this experiment, three subjects (aged 23~24 year-old, two males and one female) engaged the experiments with their consent on the detailed explanation.

2.2 Results and Discussion

Experimental results were shown in Fig. 4. The horizontal and the vertical axes show the display size (i.e. number of stimuli) and the reaction time (i.e. time to need searching the target), respectively. Each of the datum point is the average of 15 measurements for one subject except the errors answered and each vertical bar indicates the standard deviation of those.

As shown in Fig. 4-(1), the reaction time does not show the significant variation with changing the display size for the subject HM (p>0.05). For the subject HM, the variation of reaction time calculated by least square method was 3.01 [ms/item] for the condition 1 (target=2[Hz]) and 1.71 [ms/item] for the condition 2 (target=4[Hz]). For the other subjects, similarly to the subject HM, the reaction times do not show the significant variation with changing the display size (p>0.05). For the subject IU, the variation of reaction time was 0.52 [ms/item] for the condition 1 and 2.25 [ms/item] for the condition 2. For the subject KY, this was -1.86 [ms/item] for the condition 1 and -1.17[ms/item] for the condition 2. Generally, when this variation of reaction time is within 5~6 [ms/item], the target is regarded to be found at once[1)]. Thus, this characteristic is typical one of what the target is distinguished from the distracters with one feature.

For the subject HM, characteristic that the reaction time of the condition 1 is larger than that of the condition 2, is observed as shown in Fig. 4-(1) (p<0.01). For the other subjects, the same

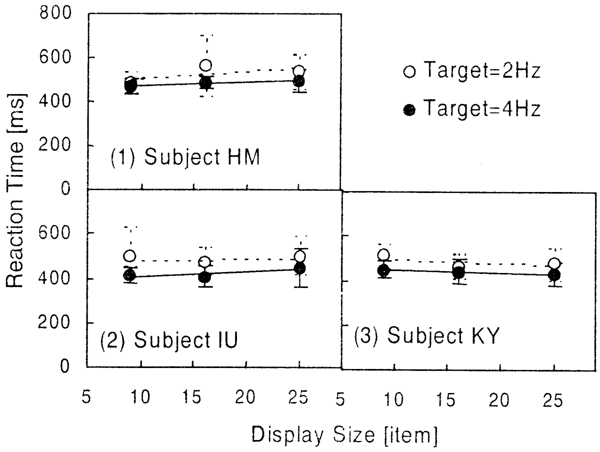


Fig. 4 Experimental results for (1) Subject HM, (2) Subject IU and (3) Subject KY. Opened circle (○) show the condition of that the target was flickered in 2 [Hz] and closed circle (●) show the condition of that the target was flickered in 4 [Hz].

characteristics were also observed, significantly (p<0.01). Thus, experimental results show that the time to find the target when the target is flickered in 2 [Hz] is larger than that when the target is flickered in 4 [Hz].

3 Chaos Oscillator Model on Human Visual Search

In order to describe the function of human visual search, we propose a chaos oscillator model of the temporal-frequency characteristics on human visual search as shown in Fig. 5. The chaos oscillators on the input layer in Fig. 5 are expressed by the logistic function with the inputted signal term as following equation (2) and move stably without the input. When the signals are inputted to these chaos oscillators, they show the oscillatory move with the transition phenomenon.

$$I_{xy}^{t+1} = a_I \times I_{xy}^{t} \times (1 - I_{xy}^{t}) + w_{SI} \times S_{xy}^{t+1} \qquad (2)$$

Here, the 'I' shows the output signal from the chaos oscillator on the input layer. The 'x' and 'y' show the spatial coordinates of the chaos oscillators, respectively. The 'a' shows the characteristic coefficient of the chaos oscillators. The 't' shows the time. The 'W' shows the connection weight between the input signal and the chaos oscillators on the input layer, and the 'S' shows the input signal (stimulus).

The chaos oscillators on the hidden layer as shown in Fig. 5 are similarly to those on the input layer. The chaos oscillators are defined by the logistic function, and move stably without the input. When the variation of the signal from the neighbor oscillators on the input layer exists, the oscillators show the oscillatory move with the transition phenomenon expressed by the following equations (3) and (4) for the x coordinate and for the y coordinate, respectively.

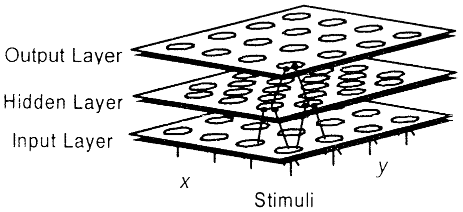


Fig. 5 A human functional model of visual search using the chaos oscillator

$$H_{(2x)(2y-1)}^{t+1} = a_H \times H_{(2x)(2y-1)}^{t} \times (1 - H_{(2x)(2y-1)}^{t}) + w_{IH} \times | I_{(x+1)y}^{t+1} - I_{xy}^{t+1} | \qquad (3)$$

$$H_{(2x-1)(2y)}^{t+1} = a_H \times H_{(2x-1)(2y)}^{t} \times (1 - H_{(2x-1)(2y)}^{t}) + w_{IH} \times | I_{x(y+1)}^{t+1} - I_{xy}^{t+1} | \qquad (4)$$

Here, the 'H' shows the output signal from the chaos oscillator on the hidden layer.

The oscillators on the output layer are also similar to that on the input and hidden layer. The chaos oscillators on the output layer are expressed by the following equation,

$$O_{xy}^{t+1} = \frac{1}{1 + n \times w_{HO}} \times \{ a_O \times O_{xy}^{t} \times (1 - O_{xy}^{t}) + w_{HO} \times (H_{(2x)(2y-1)}^{t+1} + H_{(2x-2)(2y+1)}^{t+1} + H_{(2x-1)(2y)}^{t+1} + H_{(2x-1)(2y-2)}^{t+1}) \}. \qquad (5)$$

Here, the 'O' shows the output signal from the chaos oscillator on the output layer. The 'n' shows the number of connection from the oscillators on the hidden layer. When the signals from the oscillators on the hidden layer move stably, the chaos oscillators on the output layer also show the stable move. And when the signals from the oscillators on the hidden layer move oscillatory, the chaos oscillators on the output layer also show the oscillatory move.

4 Simulation

Similarly to the psychological experiment, the intensity of the input signals (stimuli) is flickered in sine wave in this simulation experiment as shown in Figs. 6-(1). In this experiment, the target was flickered in 2[Hz] (2[cycle] of the sine wave at the 1024 [cycle] of the number of calculation, Condition 1) and in 4[Hz] (4[cycle] of the sine wave at the 1024[cycle] of the number of calculation, Condition 2). And the distracters were flickered in 1[Hz] (1[cycle] of the sine wave at the 1024[cycle] of the number of calculation). These stimuli were flickered within 0.25 ~ 0.75 of the signal intensity. In this simulation experiment, the parameters as shown in equations (2) ~ (5) were set at the following conditions: a_I=a_H=a_O=2.5, w_{SI}=0.5, w_{IH}=1.0, w_{HO}=0.6. As shown in Fig. 6-(2), at the conditions of the parameters as mentioned above, only of the chaos

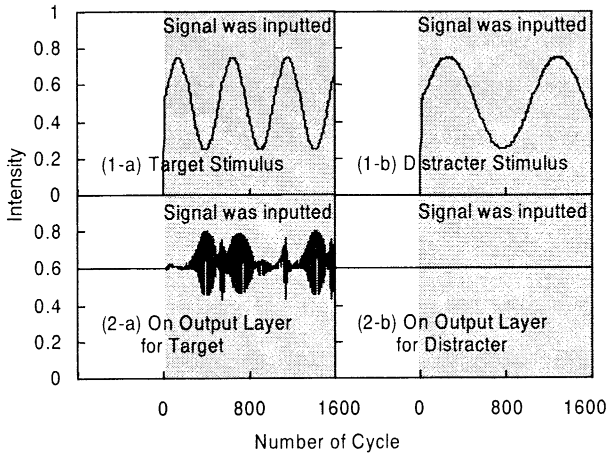


Fig. 6 Time-intensity curve of (1) the inputted signals and (2) the chaos oscillators on the output layer for (a) the target stimulus and (b) the distracter stimulus. These figures show the case of Condition 1 (Target=2[Hz] and Distracter=1[Hz]).

oscillator on the output layer for the target stimulus shows the oscillatory move, when the signals were inputted.

Next in this simulation experiment, we measured the number of calculation cycle to threshold of the output signal variation $\varDelta$ on the output layer, expressed by the following equation:

$$\varDelta = / O^{t}_{xy} - O^{t-1}_{xy} / . \qquad (6)$$

In this experiment, threshold of '$\varDelta$' was set at $\varDelta_T = 0.3$, and other parameters were set at the same as mentioned above. Experimental results as shown in Fig. 7 show the similar characteristics to the results in the psychological experiments. As shown in Fig. 7, number of cycles to the threshold does not show the significant variation with changing the display size (p<0.01). This characteristic corresponds to the characteristic in the psychological experiment, that the reaction time is independent of changing the display size. Furthermore, as shown in Fig. 7, number of cycles to the threshold at the condition of that the target was flickered in 2 [Hz] was larger than that at the condition of the target was flickered in 4 [Hz] (p<0.01). This characteristic also corresponds to the characteristic in the psychological experiment, that the reaction time at the condition of that the target was flickered in 2 [Hz] was larger than that at the condition of the target was flickered in 4 [Hz]. Thus, it is suggested that these results of simulation experiment agree with the results of the psychological experiment well.

5 Conclusion

In this study, we propose a chaos oscillator model of the temporal frequency characteristics on human visual search. The

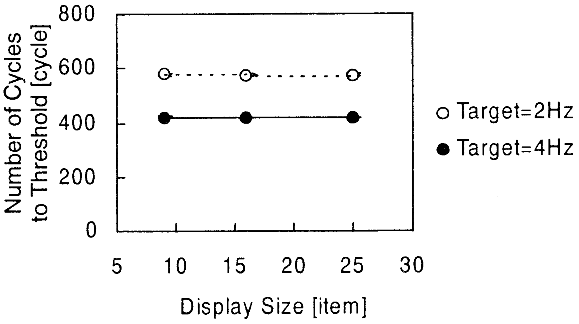


Fig. 7 Results of the simulation experiments. Opened circle (○) shows the condition of that the target stimulus was flickered in 2 [Hz] (2 [cycle] at the 1024 [cycle] of the number of calculation). And closed circle (●) shows the condition of that the target stimulus was flickered in 4 [Hz] (4 [cycle] at the 1024 [cycle] of the number of calculation).

characteristics of the simulation experiment agree with those of the temporal frequency detection on the human visual search in the psychological experiment. Thus, the proposed model describes the human characteristics of visual search. This model will be significant to the automatic operation of the manufacturing process in the inspection, when the more complex ability of human visual search is realized in the artificial system. In our future study, we will investigate the more complex ability of human visual search based on the proposed model in this study.

References

1) Yokosawa K., Kumada T. (1996), Visual Search Phenomena and Processes (in Japanese). *Transaction of Japan Cognitive Science* **3**(4): 119-138.

2) Treisman A. (1986), Feature and Objects in Visual Processing, *Scientific America* **254**(11): 114-124.

3) Humphreys G.W., Müller H.J. (1993), SEarch via Recursive Rejection (SERR): A Connectionist Model of Visual Search, *Cognitive Psychology* **25**: 43-110.

4) Yokosawa K. (1994), Multiresolutional Model for Analysis of Visual Attention and Visual Search Performance (in Japanese). *Transaction of Japan Cognitive Science* **1**(2): 64-80.

5) Freeman W.J. (1997), The Physiology of Perception, *Nikkei Science* **120**: 42-51.

6) Kitazawa M., Wu J.L., Sakai Y. (2000), A New Method of 3-D Movie Based on 2-D Photo Images for the Virtual Playing Catch System, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*: 157-162.

7) Yeshurun Y., Carrasco M. (2000), The Locus of Attentional Effects in Texture Segmentation, *Nature Neuroscience* **3**(6): 622-627.

8) Sharma V., Levi D.M., Klein S.A. (2000), Undercounting Features and Missing Features: Evidence for a High-Level Deficit in Strabismic Amblyopia, *Nature Neuroscience* **3**(5): 496-501.

9) Nishida S., Ohtani Y., Ejima Y. (1992), Inhibitory Interaction in a Split/Fusion Apparent Motion: Lack of Spatial-Frequency Selectivity, *Vision Research* **32**(8): 1523-1534.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Analysis of small-world networks with high broadcast speed

Tetsuya Maeshiro
ECM, ISD
ATR International
Kyoto, 619-0288 JAPAN

Noriko Ohi
Dep. of Applied Math. and Info.
Ryukoku University
Shiga, 520-2194 JAPAN

Katsunori Shimohara
ECM, ISD
ATR International
Kyoto, 619-0288 JAPAN

Abstract

Given a network where each element has a fixed but small number of connected elements, randomization of connections with small probability results in networks with high broadcast speed. Such networks belong to the class of so called small-world networks. Analysis indicate that the broadcast to all elements of network is fundamentally different from average transmission speed. Networks with 10% of randomness gives fastest broadcast speed with no dependence on the position of initial transmission source.

1 Introduction

Given a network with N elements where each element is connected with kp other elements, where $k \simeq \ln(N)$ or $k < \ln(N)$, we have investigated the network topology that gives the fastest broadcast speed from an arbitrary selected element, and have found that a class of networks denoted "small-world" networks is a strong candidate. Furthermore, the characteristic path length, a conventionally used parameter to evaluate networks[1], does not reflect the network topology when broadcast is under issue.

Many networks are reported to be small-world networks, which are regular networks with a slight randomness in connections among elements, thus small-world networks belong to the intermediate between regular networks and random networks. Examples are found in both Nature and artificial entities: network of neural cells of *C. elegans*, metabolic networks in unicellular organisms, World Wide Web, collaboration graph of film actors, and power grid of the western United States [1, 2, 3]. Most of these examples satisfy the condition $k \gg \ln(N)$ to a random network be fully connected [4]. The number of neighbors k in natural networks is not a constraint because Nature connects efficiently a considerable number of elements. For instance, a neuron in human brain is typically connected with tens of thousands of neurons.

This paper treats networks with small number of neighbors k compared to the total number of elements N, the networks that violate the condition $k \gg \ln(N)$.

2 Structural Properties of Networks

A graph represents a network, where a node denotes an element of the network, and a link connects a pair of nodes if the nodes are related. Nodes are allocated in a two-dimensional grid of the size $N = n \times n$. The positions of nodes in the grid reflects the distance among nodes.

Each node is connected with k other nodes, of which $k(1-p)$ nodes, $0.0 \leq p \leq 1.0$, are the nearest nodes, and kp nodes are randomly selected nodes. Consequently, networks with $p = 0.0$ are regular networks, without random connections. A gradual increase in p generates networks proportionally random, and $p = 1.0$ results in a random network.

Three parameters evaluates the structure of networks. The characteristic path length $L(p)$ is a global property, which is the average of the distance of shortest paths between all node pairs. The clustering coefficient $C(p)$, the second parameter, measures a local property, the average of relative number of links among nodes connected to each node. The third parameter, the broadcast path length $T(p)$, is a newly introduced parameter, which is the maximum of the shortest paths between all node pairs, equivalent to the number of steps necessary to broadcast information to all nodes. The first two parameters are conventionally used [1], while the third one is introduced here to measure the broadcast speed.

Regular networks ($p = 0.0$) present large $C(p)$ and $L(p)$, while random networks ($p = 1.0$) have small $C(p)$ and $L(p)$. However, the reduction manners of $L(p)$ and $C(p)$ are uncorrelated. While $L(p)$ drops sharply with small p, $C(p)$ decreases approximately inversely proportional to p, leaving a region of small p where $C(p)$ is large, like a regular network, and $L(p)$

is small, similar to a random network. Networks with p in such region are denoted as small-world networks, to which many real networks belong [1].

3 Simulation

The simulation starts from m randomly selected nodes, $m = 1, 2, \ldots$, which transmits "information" to all directly connected nodes. Subsequent nodes repeatedly transmits to all directly connected nodes until all nodes in the network receive. A node transmits to all connected nodes in a single step, and the number of steps of broadcast evaluates the networks.

Three parameters characterize the connections among nodes: k, the number of connected nodes to each node; p, the probability of random connection; and α, a parameter to define the probability distribution of distance of randomly connected nodes. The neighbor nodes are connected following a predetermined sequence (Figure 1). On the other hand, the distance of kp randomly connected nodes is given by

$$d = D \cdot q^{\alpha} \qquad (1)$$

where q is a random variable, $0.0 \leq q \leq 1.0$, and D is the distance from the considered node to the boundary on one of four directions of the network, since the two-dimensional grid is not a torus. For $\alpha = 1.0$, the distance is uniformly distributed; for $\alpha > 1.0$, further nodes are selected more frequently; and for $\alpha < 1.0$, the nearer nodes.

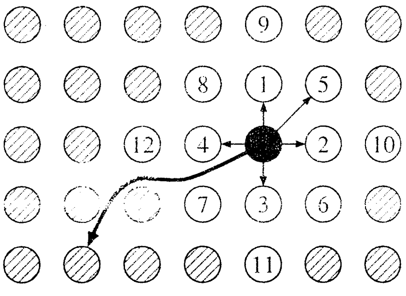


Figure 1: Order to connect neighbor nodes. Example of $k = 6$, of which 5 are neighbor nodes and 1 is randomly chosen node. Numbers in nodes indicates the order to connect. Shadowed nodes are selected randomly, according to Eq.(1)

Networks of size $10^4 = 100 \times 100$ are simulated with values in Table 1, each with five runs to calculate the mean and variance. Simulation runs that did not finish after 2,000 steps are stopped. This is based on results of pre-simulations that simulation running for more than 2,000 steps does not finish, leaving indefinitely a constant number of non-transmitted nodes.

Table 1: Values of parameters used in simulation

parameter	Values
m	1, 2, 4, 8
k	8, 12
α	0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 2.0
p	0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0

4 Discussions and Conclusions

Figure 2, 3 and 4 indicate regions of p with smallest broadcast path length $T(p)$. This region of p gives large $C(p)$ and small $L(p)$ (Figure 5), characteristic of small-world networks. The curve of $T(p)$ is similar to a quadratic function, differing from the characteristic path length $L(p)$ that decreases monotonically.

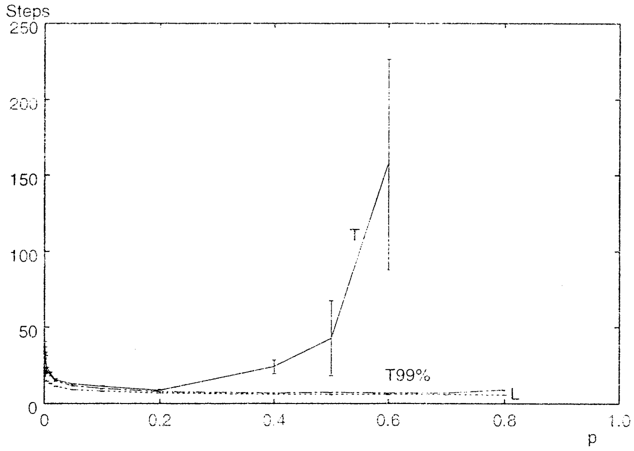


Figure 2: Broadcast path length $T(p)$, 99% broadcast path length $T_{99}(p)$, and characteristic path length $L(p)$, where $k = 12$, $m = 1$, and $\alpha = 1.0$. Mean and variance of 5 runs. For $p > 0.6$, no simulation run broadcasted to all nodes.

Simulations with values of p close to 1.0 did no finish, and the variance of broadcast path length $T(p)$ increases for larger p. For p close to 1.0, isolated groups of nodes are easily generated, impeding the broadcast to all nodes. Therefore, the number of steps

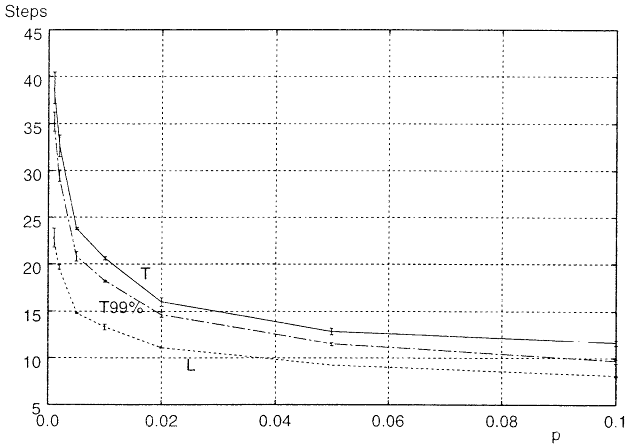


Figure 3: Magnification of Figure 2 for $0.0 \leq p \leq 0.1$

to reach 99% of nodes, denoted 99% broadcast path length $T_{99}(p)$, was also measured. Increase in variance of $L(p)$ was very small for large p, and that of $T_{99}(p)$ tends to increase (Figures 2 and fig:k8). The variance is also large for p close to 0.0, although much smaller than that for p close to 1.0. The variance is negligible for $0.05 \leq p \leq 0.2$, the range with smallest $T(p)$ that belongs to small-world networks. This suggests the existence of range of p that assures constant broadcast speed from any nodes in the network.

The behavior of $T_{99}(p)$ is similar to that of $L(p)$, but tends to increase for p close to 1.0.

Figure 6 indicates that the initial transmission speed is faster for large p, consistent with monotonic decrease in $L(p)$ (Figure 5). The curve of $T(p)$ indicates, however, that $T(p)$ for large p converge to a value near 100% and does not reach 100%, so the broadcast speed is the slowest. The curves for p in the range of small world networks is not the fastest, but one of.

The distribution of distances of randomly selected nodes also affects the broadcast path length $T(p)$ (Figure 7). The uniform distribution ($\alpha = 1.0$) gives the best performance, and bias in either nearer of further distance worsen the values of $T(p)$, true for all values of p. The same applies for both $L(p)$ and $C(p)$, suggesting the $L(p)$ has stronger influence on $T(p)$. However, the increase in $T(p)$ for large p indicates that $C(p)$ is important for the transmission to "last few nodes". Therefore, there is a crucial difference between the request to transmit to the majority of the network and to all nodes of the network.

The broadcast path length $T(p)$ decreases proportionally with the increase in the number of initial nodes (Figure 8), which is obvious.

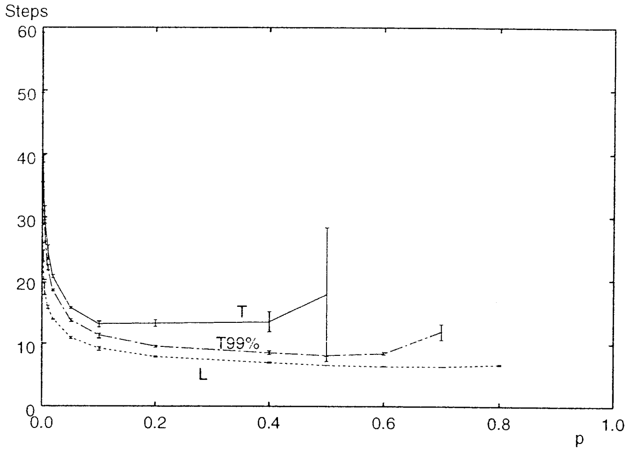


Figure 4: Broadcast path length $T(p)$, 99% broadcast path length $T_{99}(p)$, and characteristic path length $L(p)$, where $k = 8$, $m = 1$, and $\alpha = 1.0$. Mean and variance of 5 runs. For $p > 0.5$, no simulation run broadcasted to all nodes, and for $p > 0.7$, no simulation run with 99% broadcast was observed.

For small number of connections k, change in k affects only quantitatively the results, and not quantitative. The simulation of networks with small number of connections per node is useful to design artificial networks with large number of elements with the request for fast broadcast.

References

[1] D.J. Watts and S.H. Strogatz, "Collective dynamics of 'small-world' networks", *Nature*, Vol.393, pp.440–442, 1998.

[2] R. Albert, H. Jeong and A-L. Barabási, "Diameter of the World-Wide Web", *Nature*, Vol.401, p.130, 1999.

[3] H. Jeong et. al., "The large-scale organization of metabolic networks", *Nature*, Vol.407, pp.651–654, 2000.

[4] B. Ballabás, *Random Graphs*, Academic Press, 1985.

[5] N. Ohi and T. Maeshiro, "Structural analysis of networks and broadcast speed", *ATR Technical Report*, TR-IS-002, 2000 (In Japanese).

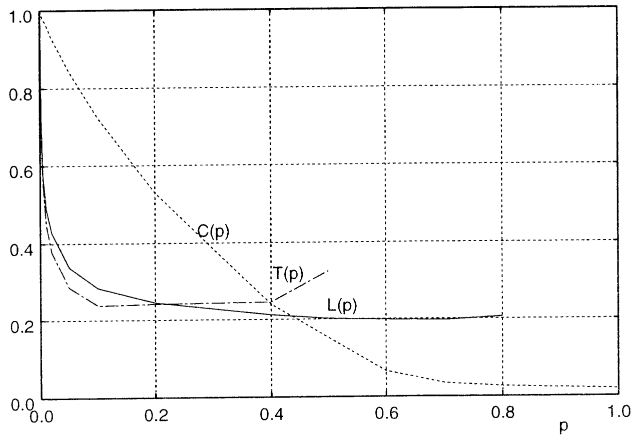


Figure 5: Relative variations of broadcast path length $T(p)$, characteristic path length $L(p)$, and clustering coefficient $C(p)$. $k = 8$, $m = 1$, $\alpha = 1.0$. All values are normalized with values of $p = 0.0$. Therefore, $T(p)$ and $L(p)$ cannot be compared in this graph.

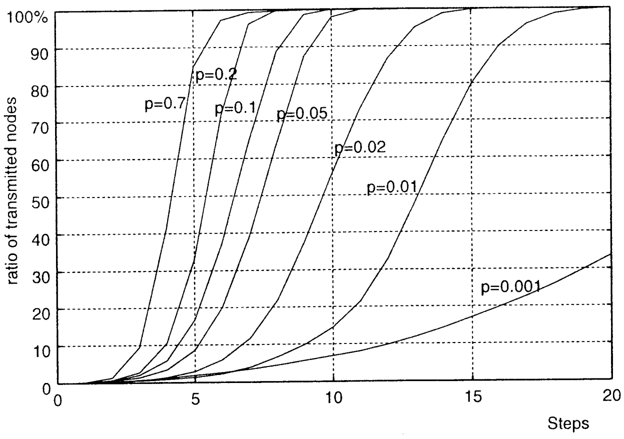


Figure 6: Transmission speed. $k = 12$, $m = 1$, $\alpha = 1.0$.

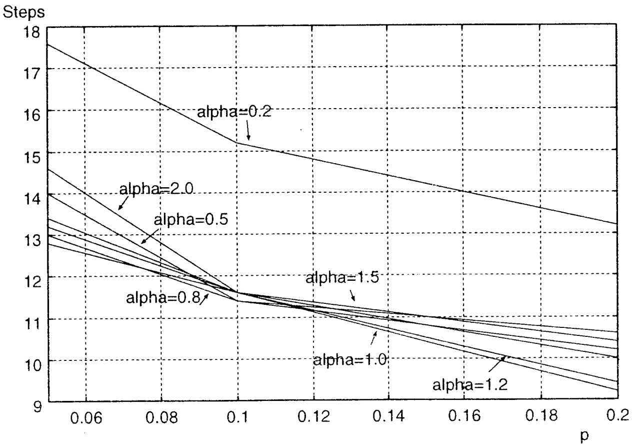


Figure 7: Effect of distribution of distances of randomly selected nodes. $k = 12$, $m = 1.0$.

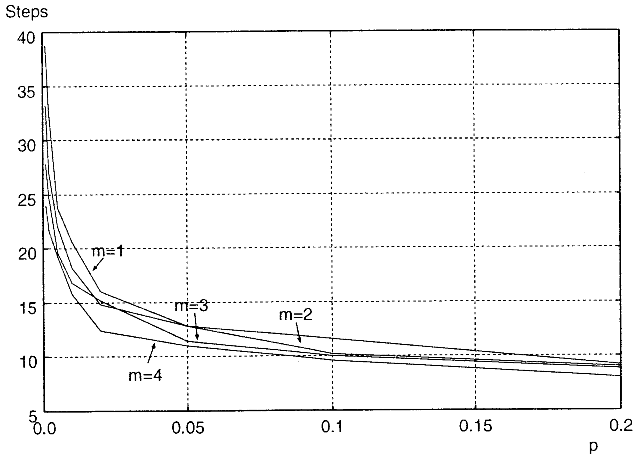


Figure 8: Dependence of broadcast path length $T(p)$ on the number of initial nodes. $k = 12$, $\alpha = 1.0$.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

On Multistage Breeding Processes

Toshio Odanaka Emeritus Professor of Tokyo Metropolitan Institute of Technology
Tsutomu Arimizu Arimizu Lab.

Abstract ; Let us assume that we are examining the operation of a two-growth complex, the "vegetative" growth and "breeding" growth. One of the reasons why it is worth whle to treat these admittedly over-simplified models in some detail is in the hope of obtaining some picture of the structure of the solution of problems of this nature.

We shall suppose, using the concept of lumped parameters which plays such an essential role, that the state of each system at any particular time may be specified by means of two quantities
(a) the amount of the vegetative growth
(b) the capacity of the vegetative organ

In short, the production of breeding will depend only upon the quantity of vegetative.

At any particular time, the vegetative may be used for either of three purposes:
(a) to produce additional vegetation
(b) to increase the existing vegetative capacity,
(c) to produce breeding growth using the existing breeding capacity.

We wish to determini allocation policies which maximize the total quantity breeding over a given period of time.

Often, processes of the type discussed below are called "bottleneck processes".

Keywords ; vegetation, breeding,
dynamic programming, bottleneck processes

1. Introduction

Let us formulate mathematicaly the adaptation in living being, then we have applicability for modern control theory.

Following, we assume that the growth is the processes allocating the nutrient for each organ.

In section 2, we shall discuss the optimal scheduling in growth.

2. Vegetative growth and breeding growth

2.1 Introduction

Before many living being are ripe and enter to the breeding season, they are sure to have the long vegetative growth time. This can be explained that they had would rather value with high birth rate by the grow body than be at the small body in a hurry breeding. We want to disscuse the growth processes that increase the total value of breeding.

2.2 A Mathematical Model

At the moment, the process will be taten to be discrete, with allocations made only at times t = 0, 1, …, T-1.

At any particular time, t = n, the state of the system is determined by the quantities.

(1) (a) $x_1(n)$= amount of the vegetative growth

(b) $x_2(n)$= capacity of the vegetative organ

In determining the allocation of available vegetation, we introduce the quantities

(2) (a) $z_1(n)$= the quantity of vegetation used to produce additional vegetation.

(b) $z_2(n)$= the quantity of vegetation used to increase vegetative capacity

(c) $z_3(n)$= the quantity of vegetation used to produce breeding

We then have the relation.

(3) $x_1(n) = z_1(n) + z_2(n) + z_3(n)$

In order to introduce some features, we impose two constraints on the Z's:

(4) (a) $z_3(n) \le a_1 x_1(n) \qquad 0 < a_1 < 1$

(b) $z_1(n) \le x_2(n)$

We are thus assuming that we have a linear model of production.

(5)

$$x_1(n+1) = a_3 z_1(n), \qquad a_2 > 1, \quad x_1(0) = c_1$$
$$x_2(n+1) = x_2(n) + a_3 z_2(n), \quad a_3 > 0, \quad x_2(0) = c_2$$

It is required to choose the quantities $z_1(n), \quad z_3(n) \quad and \quad z_2(n)$, $n = 0, 1, \cdots, T-1$, so as to maximize the total quantity of breedings produced over the entire T stage process.

2.3 Dynamic Programming Approach

Let us define, for N =1, 2, …, $c_1 \ge 0, \quad c_2 \ge 0$, the function.

(1) $f_N(c_1, c_2)$ = total quantity of breedings produced over N stages, starting with initial amount of the vegetative c_1 and initial capacity c_2 of the vegetative organ and using an optimal policy.

We have

(2) $f_1(c_1, c_2) = a_1 c_1$

and generally,

(3)

$$f_N(c_1, c_2) = \max\left[z_3 + f_{N-1}(a_2 z_1,\quad c_2 + a_3 z_2)\right]$$

for N=2, 3, …, where the maximization is over the region in z-space defined by the in equalities

(a) $z_1, z_2, z_3 \geq 0$

(b) $z_1 + z_2 + z_3 = c_1$

(c) $z_3 \leq a_1 c_1$

(d) $z_1 \leq c_2$

In the next section, we shall discuss the numerical determination of the search of vertices.

2.4 Numerical Solution

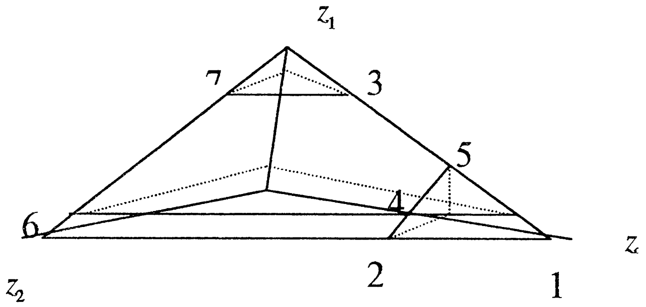


Fig.1

Region of variability of the quantities z_1, z_2, z_3 .

Table Case the breeding growth begin stage 8

stage	z_a	z_s	z_m	vertex	c_1	c_2	c_2/c_1
1	0	1	0	1	1	1	1
2	0	1	1	2	2	1	0.5
3	0	1.4	0.6	2	2	1.4	0.7
4	0	1.64	1.16	2	2.8	1.64	0.5857
5	0	2.104	1.176	2	3.28	2.10	0.6414
6	0	2.5744	1.6336	2	4.208	2.5744	0.6117
7	0	3.2278	1.9210	2	5.1488	3.2278	0.6269
8	1.2911	3.9962	1.1682	4	6.4556	3.9962	0.6190
9	1.5984	4.4634	1.9305	4	7.9924	4.4634	0.5584
10	1.7853	5.2356	1.9058	4	8.9268	5.2356	0.5865
11	2.0942	5.9979	2.3790	4	10.4712	5.9979	0.5728
12	2.3991	6.9495	2.6471	4	11.9958	6.9495	0.5793
13	2.7798	8.0083	3.1109	4	13.8990	8.0083	0.5762
14	3.2033	9.2526	3.5406	4	16.0166	9.2526	0.5777
15	3.7010	10.0768	4.1274	4	18.5052	10.6768	0.5770
			Total allocation to breeding growth 18.8522				

Bibliography

1) T. Arimizu, "Working group matrix in dynamic model of forest management", J. Japanese Forestry Soc., Vol. 40, 1958, pp. 185-190.

2) R. Bellman, "On bottleneck problems and dynamic programming," Proc. Nat. Acad. Sci. USA, Vol.39, 1953, pp. 947-951.

3) A.Bensoussan, E. Gerald Hurst, JR. and B.Naslund, 'Management Applications of Modern Control Theory", North-Holland Publishing Company, 1974.

4) T. Odanaka, "Dynamic Management Decision and Stochastic Control Processes", 1990.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Sequential Dynamical Systems

C.L. Barrett H. Mortveit C.M. Reidys
Los Alamos National Laboratory
TSA-2, MS M997, Los Alamos, NM 87545, USA

Abstract

In this paper we present the work on the special class of dynamical system referred to as discrete sequential dynamical system (sds). The definition of these systems is motivated by the generic structure of computer simulations. In computer simulations we typically find agents or entities with certain properties or states. The entities can retrieve information from other entities, and usually only from the ones in their own vicinity. Based on these states they may update their state. A schedule will take care of the update order of the entities. One possible interpretation of this is to have each entity as a vertex in a (dependency) graph where two vertices are connected if the corresponding two entities can communicate. Without loss of generality we can associate to each vertex or entity a binary state. Finally, we fix some ordering of the vertices that represent the update ordering of the entities. The above construction will be put in a strict mathematical context and leads to the concet of an sequential dynamical system (sds). In a computer simulation one typically has "perfect" knowledge about each entity and which entities can communicate or exchange information. To retrieve information on the dynamics of this "complex system" one will have to run or simulate it. The character of the results in this paper is how to extract dynamical properties from known quantities, like the dependency graph, update rules, and without actually implementing and running the system on a computer. This perspective is closely related to further work on discrete sequential dynamical systems We start by reviewing the concepts and the setting needed for the definition of a sequential dynamical system. In many cases the concept of dynamically equivalent sds is important. This framework is introduced in section 1 where we also present applications and enumeration results. In section 2 we give a full characterization of invertible sds and we show that the only homogeneous invertible sds are the ones induced by the Boolean functions Parity and its complement function.

1 Introduction

Let Y be a loop-free undirected graph with vertex set $\mathrm{v}[Y] = \{1,\dots,n\}$ and edge set $\mathrm{e}[Y]$. Let $B_{0,Y}(i)$ be the set of Y-vertices adjacent to vertex i and let $\delta_i = |B_{0,Y}(i)|$. We denote the increasing sequence of elements of the set $B_{0,Y}(i) \cup \{i\}$ by

$$B_{1,Y}(i) = (j_1,\dots,i,\dots,j_{\delta_i}) , \qquad (1.1)$$

and set $d = \max_{1\le i\le n} \delta_i$. To each vertex i there is associated a state $x_i \in \mathbb{F}_2$, and for each $k = 1,\dots,d+1$ let $f_k : \mathbb{F}_2^k \to \mathbb{F}_2$ be a given symmetric function. For each vertex $i \in \mathbb{N}_n = \{1,2,\dots,n\}$ we introduce the map

$$\begin{aligned}\mathrm{proj}_Y[i] : \mathbb{F}_2^n &\to \mathbb{F}_2^{\delta_i+1},\\ (x_1,\dots,x_n) &\mapsto (x_{j_1},\dots,x_i,\dots,x_{j_{\delta_i}}) .\end{aligned}$$

Furthermore, let S_k with $k \in \mathbb{N}$ denote the symmetric group on k letters. Let $(f_k)_{1\le k\le d(Y)+1}$ be a multiset of symmetric functions $f_k : \mathbb{F}_2^k \to \mathbb{F}_2$. Set $x = (x_1,x_2,\dots,x_n)$. For each $i \in \mathbb{N}_n$ there is a Y-local map $F_{i,Y}$ given by

$$\begin{aligned}y_i &= f_{\delta_i+1} \circ \mathrm{proj}_Y[i],\\ F_i(x) &= (x_1,\dots,x_{i-1},y_i(x),x_{i+1},\dots,x_n).\end{aligned}$$

We refer to the multiset $(F_{i,Y})_i$ as F_Y. It is clear that for each $Y < K_n$ the multiset $(f_k)_{1\le k\le n}$ induces a multiset F_Y, i.e., we have a map $\{Y < K_n\} \to \{F_Y\}$. Let $\pi \in S_n$. Now define the map $[F_Y,] : S_n \to \mathrm{Func}(\mathbb{F}_2^n, \mathbb{F}_2^n)$ by $[F_Y,\pi] = \prod_{i=1}^n F_{\pi(i),Y}$.

Definition 1 *The sequential dynamical system (SDS) over Y with respect to the ordering π is $[F_Y,\pi]$. We call an SDS homogeneous if it is induced by a set of local functions of the form $(f_k)_k = (B_k)_k$ where B is a Boolean function.*

Example. Let $Y = \mathrm{Circ}_n$, the circle graph on n vertices. The graph Circ_6 is shown in figure 1. For each vertex we have a symmetric function on 3 arguments. To be specific we pick the parity function for each vertex. The parity function (3.2) returns the sum of its

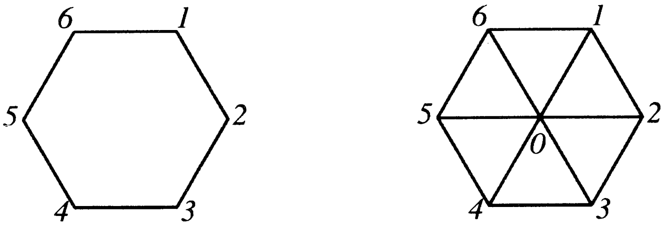


Figure 1: The circle graph on 6 vertices and the wheel graph on 7 vertices shown to the left and right respectively.

argument modulo 2. Thus for the update schedule $(1,2,3,4,5,6)$ with initial state $(1,1,1,0,0,0)$ we get

$$\begin{aligned}
F_1(1,1,1,0,0,0) &= (0,1,1,0,0,0),\\
F_2\circ F_1(1,1,1,0,0,0) &= (0,0,1,0,0,0),\\
F_3\circ F_2\circ F_1(1,1,1,0,0,0) &= (0,0,1,0,0,0),\\
F_4\circ F_3\circ F_2\circ F_1(1,1,1,0,0,0) &= (0,0,1,1,0,0),\\
F_5\circ F_4\circ F_3\circ F_2\circ F_1(1,1,1,0,0,0) &= (0,0,1,1,1,0),\\
F_6\circ F_5\circ F_4\circ F_3\circ F_2\circ F_1(1,1,1,0,0,0) &= (0,0,1,1,1,1),
\end{aligned}$$

and thus $[F_{\mathrm{Circ}_6},(1,2,3,4,5,6)](1,1,1,0,0,0) = (0,0,1,1,1,1)$.

If we remove the restriction on having symmetric functions we obtain the sequential analog of von Neumann's cellular automata. We call this class sequential cellular automata (sCA), and refer to the classical CAs as parallel cellular automata (pCA). Formally, a pCA is a pair (f_3, Circ_n) where $f_3 : \mathbb{F}_2^3 \to \mathbb{F}_2$ is the rule used to update the states associated to the vertices of Circ_n in *parallel*. An sCA is a triple $(f_3, \mathrm{Circ}_n, \pi)$. In this case the function f_3 updates the states of Circ_n sequentially in the order given by π. Note that an sCA is an SDS only in the case where f_3 is a symmetric function. We emphasize that the base graph of an SDS does not have to be a circle, but can be an arbitrary graph. Another example of a base graph is shown on the right in figure 1.

We introduce the equivalence relation $\sim_{Y,F}$ on $S_n \times S_n$ by $\pi \sim_{Y,F} \sigma$ iff $[F_Y,\pi] = [F_Y,\sigma]$ and let $\mathbb{S}_{(f_k)_k}(Y) = \{[F_Y,\pi] \mid \pi \in S_n\}$. The digraph $\Gamma[F_Y,\pi]$ is the directed graph having vertex set $\mathbb{F}_2^n$ and edge set $\{(x,[F_Y,\pi](x)) \mid x \in \mathbb{F}_2^n\}$. For $\pi = (i_1,\dots,i_n)$ write $i <_\pi j$ if $i = i_k$, $j = i_l$ and $k < l$. A point $x \in \mathbb{F}_2^n$ is periodic with period m if $[F_Y,\pi]^{(m)}(x) = x$ and $[F_Y,\pi]^{(l)}(x) \neq x$ for $1 \leq l < m$. A fixed point is a point x for which $[F_Y,\pi](x) = x$. The point x is called eventually fixed if there exists l such that $[F_Y,\pi]^{(l)}(x)$ is fixed.

2 Equivalence of SDS.

In practice one is interested in studying or prescribing a system with a given number of orbits and orbit sizes as well as a given transient behavior. Two dynamical systems can differ as maps but nevertheless they may have the same dynamical properties. To be more precise let $[F_Y,\pi]$ and $[F_Y,\pi']$ be two SDS. If there exists a bijection $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that

$$[F_Y,\pi'] = \varphi \circ [F_Y,\pi] \circ \varphi^{-1}, \tag{2.1}$$

we say that $[F_Y,\pi]$ and $[F_Y,\pi']$ are dynamically equivalent SDS. Note that (2.1) implies, e.g., that the two dynamical systems have a $1-1$ correspondence between fixed points and periodic points. If x is fixed under $[F_Y,\pi]$ we obtain $[F_Y,\pi'](\varphi(x)) = \varphi(x)$, i.e., $\varphi(x)$ is a fixed point for $[F_Y,\pi']$. With the discrete topology on $\mathbb{F}_2^n$ the definition of dynamically equivalent systems coincides with the definition of topologically conjugate systems as φ is then automatically a homeomorphism. In light of all this it is of interest to have estimates for the size of the set

$$\Sigma_Y[F_Y,\pi] = \{\mathfrak{O} \in \mathrm{Acyc}(Y) \mid \Gamma[F_Y,\mathfrak{O}^{\#}] \cong \Gamma[F_Y,\pi]\}\ . \tag{2.2}$$

In the following we will write $\mathbb{S}(Y)$ for $\mathbb{S}_{(f_k)_k}(Y)$.

Proposition 1 *Let $Y < K_n$ and define the S_n-action on $\mathbb{F}_2^n$ by*

$$\rho(x) = (x_{\rho^{-1}(1)},\dots,x_{\rho^{-1}(n)}).$$

The following holds:

1. *The map $\mathrm{Aut}(Y) \times S_n/\sim_Y \longrightarrow S_n/\sim_Y$ defined by $(\gamma,[\pi]_Y) \mapsto [\gamma\circ\pi]_Y$ is an $\mathrm{Aut}(Y)$-action on $S_n/\sim_Y$. This action induces an $\mathrm{Aut}(Y)$-action on $\mathrm{Acyc}(Y)$ given by $\{\gamma\ \mathfrak{O}\}(\{i,k\}) = \mathfrak{O}(\{\gamma^{-1}(i),\gamma^{-1}(k)\})$.*
2. *For all $\pi \in S_n$ and all $\gamma \in \mathrm{Aut}(Y)$ we have $[F_Y,\gamma\pi] = \gamma\circ[F_Y,\pi]\circ\gamma^{-1}$.*
3. *The map $\mathrm{Aut}(Y) \times \mathbb{S}(Y) \longrightarrow \mathbb{S}(Y)$ given by $(\gamma,[F_Y,\pi]) \mapsto [F_Y,\gamma\circ\pi]$ is an $\mathrm{Aut}(Y)$-action on $\mathbb{S}(Y)$ with the property $[F_Y,\gamma\circ\pi] = \gamma\circ[F_Y,\pi]\circ\gamma^{-1}$. In particular $\Sigma_Y[F_Y,\pi]$ is an $\mathrm{Aut}(Y)$-set.*

For the proof of this proposition we refer to [1]. As a consequence of Proposition 1 we derive the following upper bound for the number of nonequivalent SDS.

Corollary 1 *Let $Y < K_n$. We have*

$$|\{\Gamma[F_Y, \pi] \mid \pi \in S_n\}| \\ \leq \Delta(Y) = \frac{1}{|\operatorname{Aut}(Y)|} \sum_{\gamma \in \operatorname{Aut}(Y)} |\operatorname{Fix}(\gamma)| \,, \tag{2.3}$$

where $\operatorname{Fix}(\gamma) = \{\mathfrak{O} \in \operatorname{Acyc}(Y) \mid \gamma \circ \mathfrak{O} = \mathfrak{O}\}$.

The corollary is essentially a consequence of Burnside's theorem and the details can be found in [1]. A combinatorial interpretation of $\operatorname{Fix}(\gamma)$ is given in [2]. It can be described as follows. Let G be a group and let Y be an undirected graph. We will denote Y-automorphisms by γ. Now, G is said to act on Y if there exists a group homomorphism $u : G \longrightarrow \operatorname{Aut}(Y)$.

Definition 2 *Assume G acts on $Y < K_n$. Then $G \backslash Y$ is the graph with*

$$\mathrm{v}[G \setminus Y] = \{G(i) \mid i \in \mathrm{v}[Y]\}, \\ \mathrm{e}[G \setminus Y] = \{G(y) \mid y \in \mathrm{e}[Y]\}$$

and π_G is the surjective graph morphism given by

$$\pi_G : Y \longrightarrow G \setminus Y, \quad i \mapsto G(i) \ .$$

Proposition 2 *Let $Y < K_n$ be an undirected graph. Then we have*

$$\Delta(Y) = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} |a(\langle\gamma\rangle \setminus Y)| \ . \tag{2.4}$$

Let $(0 \otimes Y)$ denote the vertex join of 0 and Y. If there is no vertex of maximal degree in Y then

$$\pi_G(0 \otimes Y) = 0 \otimes \pi_G(Y). \tag{2.5}$$

For the proof we refer to [2]. Some remarks are in order. To begin, take $\gamma \in \operatorname{Aut}(Y)$ and write it as a product of disjoint cycles where cycles of length 1 are also included, say $\gamma = c_1 \cdot c_2 \cdots c_k$. The vertices in $G \setminus Y$ are in a $1-1$ correspondence with the cycles $c_1, c_2, \dots, c_k$. However, one should note that $G \setminus Y$ is in general not a simple graph as it may contain loops. There are two main factors making the computations of $a(\langle\gamma\rangle \setminus Y)$ relatively simple. The graph $\langle\gamma\rangle \setminus Y$ is typically of a nature well suited for computing its number of acyclic orientations. The procedure is also simplified by the fact that if $\langle\gamma\rangle \setminus Y$ has loops every orientation is necessarily cyclic.

3 Invertibility.

In this section we give a complete characterization of invertible SDS. To begin we recall some of the structure of symmetric functions.

Define $H_k : \mathbb{F}_2^k \to \mathbb{N}$ by $H_k(x) = |\{x_i \mid x_i = 1\}|$, and the equivalence relation $\sim_H$ on $\mathbb{F}_2^k \times \mathbb{F}_2^k$ by $x \sim_H y$ iff $H_k(x) = H_k(y)$. The symmetric functions $f : \mathbb{F}_2^k \to \mathbb{F}_2$ are precisely the functions that are constant on the equivalence classes of $\sim_H$. The equivalence classes will also be referred to as Hamming classes. As a consequence we note that there are 2^{k+1} such functions.
In [1] we derived the following result on invertibility:

Proposition 3 *Let $Y < K_n$, let $(f_k)_k$ be a multiset $f_k : \mathbb{F}_2^k \to \mathbb{F}_2$ and let $\mathrm{id}, \mathrm{inv} : \mathbb{F}_2 \to \mathbb{F}_2$ be the maps defined by $\mathrm{id}(x) = x$ and $\mathrm{inv}(x) = \overline{x}$. An* SDS *$[F_Y, \pi]$ is bijective if and only if for each $1 \leq i \leq n$ and fixed coordinates $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ the map $g_i : \mathbb{F}_2 \to \mathbb{F}_2$ defined by*

$$g_i = f_{\delta_{i+1}, Y} \circ \operatorname{proj}_Y[i](x_1, \dots, x_{i-1}, \qquad , \dots, x_n) \tag{3.1}$$

has the property $g_i \in \{\mathrm{id}, \mathrm{inv}\}$. Furthermore let $\pi = (i_1, \dots, i_{n-1}, i_n) \in S_n$, $\pi^* = (i_n, i_{n-1}, \dots, i_1)$ and $[F_Y, \pi]$ be a bijective SDS. *Then we have*

$$[F_Y, \pi]^{-1} = [F_Y, \pi^*] \ .$$

Consequently we have the interesting fact that the inverse of an invertible SDS is again an SDS. Define the two functions Par_k and $\overline{\mathrm{Par}}_k$ by

$$\mathrm{Par}_k : \mathbb{F}_2^k \to \mathbb{F}_2, \quad \mathrm{Par}(x_1, \dots, x_k) = \sum_{i=1}^{k} x_i, \tag{3.2}$$

and $\overline{\mathrm{Par}}_k : \mathbb{F}_2^k \to \mathbb{F}_2$ by $\overline{\mathrm{Par}}_k(x_1, \dots, x_k) = 1 - \mathrm{Par}_k(x_1, \dots, x_k)$.

Theorem 1 *Let Y be a graph an let $[F_Y, \pi]$ be an invertible* SDS *over Y. Then $F_Y = (F_i)_i$ where $F_i = \mathrm{Par}_i$ or $F_i = \overline{\mathrm{Par}}_i$.*

Corollary 2 *The only homogeneous invertible* SDS *are the ones induced by* Par *and* $\overline{\mathrm{Par}}$.

Thus there are 2^n invertible SDS for a given graph $Y < K_n$.

References

[1] H. S. Mortveit and C. M. Reidys. Discrete, sequential dynamical systems. *Discrete Mathematics*, 2000. In press.

[2] C. M. Reidys. On Acyclic Orientations and SDS. *Adv. in Appl. Math.*, 2000. Submitted.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Computational Model of the Interaction between Environmental Dynamics and Economic Behaviors

Takahito YAMAOKA Takaya ARITA
Graduate School of Human Informatics, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: {yamaoka, ari}@info.human.nagoya-u.ac.jp

Abstract

Environmental problems have been considered very important for a long time. We believe that they should be examined from an interdisciplinary view so as to reach a solution because they have been caused as the consequence of complex interactions among various factors. This paper proposes a new model termed *ColorChanger*. By using this model, we aim to explore the nature of ecological problems beyond separate discussions on specific subjects, and make the acquired knowledge available to encourage the solution of environmental problems. This paper also reports on the results of the preliminary experiments.

1 Introduction

Environmental problems have been considered very important for a long time. They have been discussed in a wide range from the fields of academic research to international arena of politics. There are many academic fields in which environmental problems are tackled. However, there seems to be an essential issue in conventional approaches.

For instance, there is a biological field termed conservation ecology which investigates environmental problems. Conservation ecology belongs to ecological science and aims at conservation of biological diversity by conducting basic/applied researches. Though the subjects of this approach range widely from genetic issues to landscape design, it rarely pays attention to the economic effects. On the other hand, environmental economics is a field in which the environment problems are discussed from an economic point of view. In environmental economics, the model of consumer and business behavior in traditional economics is evoked to explain who and why acts on environments and who and how suffers the environmental damage. However, consideration of dynamics of life is hardly at all involved in their investigations. In artificial society approach, which is a growing field, the dynamic models are investigated where both environments as resources and economic behavior are brought into view. So far, however, most of them leave environmental variation out of consideration, which must become very important when examining the environmental problem in real world from an interdisciplinary view.

Recently, Akiyama and Kaneko [1] have constructed a computational model so as to focus on the interaction between the dynamics of environment and agents' actions, and successfully analyzed the effects of the interaction on the dynamics of environment and the evolution of agents' actions. Their study doesn't necessarily cope with the environmental problems directly, but gives some indication of the possibilities that this type of constructive methodology could be very important when investigating the dynamics of the interactions between economic phenomena and ecological phenomena.

Encouraged by their results, we propose a computational model that makes it possible to discuss environmental problems from both economic and biological points of view. Our model consists of an economic activity model based on multi-agent modeling and a natural environment model based on cellular automata modeling. We focus on the interaction between economic activities and environmental variation based on the dynamics of an ecosystem surrounding human beings by conducting computational experiments.

2 The Model

2.1 Overview

There are several species of agents (players) and several game fields in *ColorChanger*. We denote a set of the agent species as $S = \{1, 2, \ldots, s\}$ and a set of the game fields as $G = \{1, 2, \ldots, g\}$. Each game field has a two dimensional space that is marked off into $i \times i$ hexagonal cells, on which players play some sort of economic games repeatedly. Each cell has a "color" which expresses the state of its biological environment and can be perceived by nearby players including the agent on it. Each cell changes its own color according to the color patterns on its nearby cells like cellular automata.

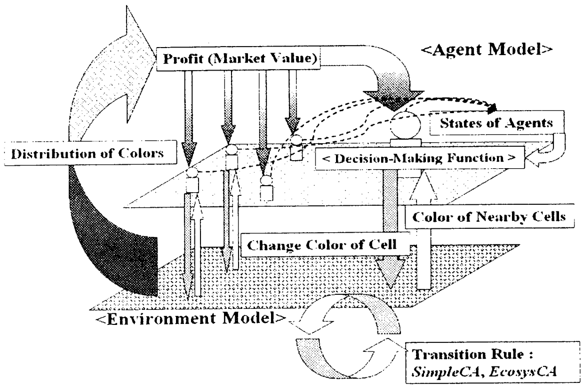


Figure 1: Conceptual diagram of *ColorChanger*: upper plane depicts the agent model, and lower plane depicts the environment model.

We define a set of players as $N = \{1, 2, \ldots, n\}$ in each game field. These n players are selected randomly from S respectively. Each agent gets a reward by affecting its environment, in other words, by changing the color of the cell on which the agent resides, which is meant to an economic action and is decided based on a species-specific decision-making function (Figure 1). An autonomous transition and a passive transition by the agent in color of each cell constitute one round. Each game consists of T rounds, and games in all game fields are played all at once. Each agent acquires its profit in every round. Genetic operations are conducted after all games in g game fields are finished, which is described in the subsection 2.4. Above described procedures are conducted again and again.

2.2 Details of the Games

The color density of the cells surrounding player i at time t is expressed as $e_i(t) = (e_i^{C(1)}(t), e_i^{C(2)}(t), \ldots, e_i^{C(c)}(t))$, where C is a set of colors which cells can be laid on. Each player has a state and a decision-making function. The state corresponds to the profit in the current round. So, the states and the decision-making functions of players are denoted by $y(t) = (y^1(t), y^2(t), \ldots, y^n(t))$ and $f = (f^{S(1)}, f^{S(2)}, \ldots, f^{S(n)})$ respectively, where the state of the player i who belongs to species $S(i)$ at time t and the decision-making function of the player i are denoted by $y_i(t)$ and $f^{S(i)}$ respectively. Player i decides its next action $a_i(t)$ based on $e_i(t)$ and $y(t)$. All of the players' actions are denoted by $a = (a^1(t), a^2(t), \ldots, a^n(t))$. Each player's individual action can be either a "wait" (doing nothing) w or a changing the color of the nearby cells into one of the set of colors $D = \{m_1, m_2, \ldots, m_d\}(m_n \in C)$. The set of these feasible actions is denoted by $A = \{w, x^{m_1}, x^{m_2}, \ldots, x^{m_d}\}$, where x^{m_i} is the action to change the color of cell into color m_i. A and D are shared among all players and is fixed through generations.

Initial state of each player is assigned a random number generated from a normal distribution with a mean of 0.10 and a variance of 0.10 before the first round of the game in each game field. Each round consists of following three steps: **1) environmental variation, 2) decision making by players,** and **3) effects of actions on cell colors and allocation of profits to players.**

1) The environmental variation consists of two steps. One is to change colors of cells autonomously, and the other is to decrease the players' states to be $y^i(t)' = u_N(y^i(t))$. We set $u_N(y) = 0.9y$ in this paper.

2) A player i's decision-making function $f^{S(i)}$ decides its action $a^i(t)$ based on the states of the environment around the player, $e_i(t)$ (density of each color of cells), and the states of all players in its game field, $y(t)'$:

$$a^i(t) = f^{S(i)}(e^i(t), y(t)'), \tag{1}$$

where $f^{S(i)}$ is the inner structure of the player i and is invisible to other players, which could vary throughout the evolution.

3) Each decision-making function of player i selects the biggest motivation in the motivation map [2] where motivations for each feasible action under the situation $\{e_i(t), y(t)'\}$ are calculated as follows:

$$max(\{mtv_r\}) : mtv_r = \eta_r e_i(t) + \sum_{l \in N} \theta_{lr} y^l(t)' + \xi_r, \tag{2}$$

where mtv_r is the motivation for action $r(r \in A)$, $\{\eta_r\}$ is a $(d+1) \times n$ real number matrix, $\{\theta_r\}$ is a $(d+1) \times c$ real number matrix, and $\{\xi_r\}$ is a $d+1$ real number vector. So $\{mtv_r\}$ means motivation map. Each element of $\{\eta_r\}$, $\{\theta_r\}$ and $\{\xi_r\}$ of the initial species of players is set random numbers generated from a normal distribution with a mean of 0.0 and a variance of 0.1. Players' actions decided in previous step can change the colors of cells. The aggregate profit R is distributed to each action as $P = \{p^{D(1)}, p^{D(2)}, \ldots, p^{D(d)}\}$, based on the state distribution (at the time before state change) calculated on randomly sampled cells as market values in reverse proportion to the frequency of the states. Each $P^{D(i)}$ is equally divided among all players who chose action $D(i)$, which increases the states of the players. Then all players pay cost Q.

2.3 Rules of Color Change in Cell and Agent Actions

A set of colors of cells is denoted by $C = \{1, 2, \ldots, c\}$, and each cell always takes one of these

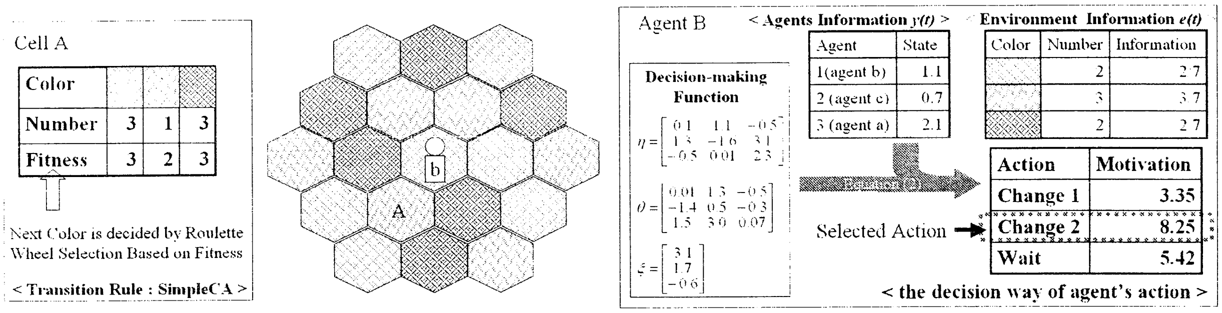

Cell A
Color
Number 3 1 3
Fitness 3 2 3
Next Color is decided by Roulette Wheel Selection Based on Fitness
< Transition Rule : SimpleCA >
b
A
Agent B
< Agents Information y(t) >
Agent State
1(agent b) 1.1
2 (agent c) 0.7
3 (agent a) 2.1
Environment Information e(t)
Color Number Information
2 2.7
3 3.7
2 2.7
Decision-making Function
η = [0.1 1.1 −0.5; 1.3 −1.6 3.1; −0.5 0.01 2.3]
θ = [0.01 1.3 −0.5; −1.4 0.5 −0.3; 1.5 3.0 0.07]
ξ = [3.1; 1.7; −0.6]
Selected Action
Action Motivation
Change 1 3.35
Change 2 8.25
Wait 5.42
< the decision way of agent's action >

Figure 2: Details of the environmental model and agent model with $n = 3$, $c = 3$, $d = 2$, $v = 2$, and the *SimpleCA* rule; Center figure expresses the situation at time t, left box shows the example of color change on cell A, and right box shows the way of decision-making by agent b.

as a state. We adopt two types of rules by which cells change their color, **1) SimpleCA** (Figure 2) and **2) EcosysCA**. The actions of players are also defined according to the adopted rule as follows.

1) The next color of each cell is decided by the roulette wheel selection based on the distribution of colors in the group of the 6 neighbor cells and itself. In counting each color, the number of its current color is multiplied by v so as to take account of an inertial effect concerning change of color. The action that the players can do is to change the color of the cell on which they exist.

2) This rule expresses a dynamics of the ecosystem. Each color of cells represents the species of animate beings on them, and is denoted by $L = \{1, 2, \ldots, l\}$. An ecological food chain is predefined by links among the species based on the method by R. J. Williams and N. D. Martinez [3], and the ecological niche can be organized under this rule. Not more than one animate beings of each species can exist on each cell. In every time step, every animate beings on each cell execute one of the following actions, prey on, bear a child or move. Prey is an action with a top priority, and is possible only when there exists a prey on the same cell. Other two actions are conducted within nearby 6 cells. If there are no possible actions, it only stays on the same cell. These actions of animate beings change the colors of cells. The color pattern on the plane of the game fields corresponds to the ecological niche. Two types of experiments concerning the player actions are being conducted. One is that players directly affect the animate beings by hunting them and they are eliminated from the cell. The players can affect d kinds of animate beings randomly selected from L. The other is that players don't directly deal with the animate beings, but change the colors of the cells on which they exist. A change of color by an player means to let the animate beings dead and/or born. The players can change d kinds of colors randomly selected from C, which contains 2^l colors that the cells can be laid on.

2.4 Evolution

Genetic operations are conducted to the agent species. The fitness of each agent species is calculated as the average profit of all agents that belong to the species during T rounds. k species with lowest k fitness are eliminated, and the surviving $s - k$ species leave their offspring which has the same decision-making function to the next generation. The eliminated species are replaced by new k species, that are k mutant species randomly selected from surviving species. Mutation happens to every coefficient of the decision-making function that the parent species has. Each coefficient in the decision-making function of the new mutant is chosen as random number from the normal distribution where the variance is 0.1 and the mean value is corresponding element of $\{\eta\}$, $\{\theta\}$, $\{\xi\}$ in the decision-making function of the parent species.

3 Preliminary Experiments

SimpleCA was adopted as a transition rule in the preliminary experiments. Following parameters were used: c (number of cell colors) $= 3$, n (number of agents in each game field) $= 3$, and d (number of colors into which agent can change)̀ $= 3$. The environmental model was initialized with equal frequencies of each color in a random spatial distribution. Other parameters were set as follows: $i = 50, g = 60, T = 400, s = 10, k = 3, R = 1.5, v = 2$, and $Q = 0.3$.

Figure 3 shows the typical transition of average fitness of each agent species. Fitness of agents shows a tendency to increase smoothly. During the first 1000 generation, the agents' strategy (decision-making function) has no clear tendency, which generates a lot of

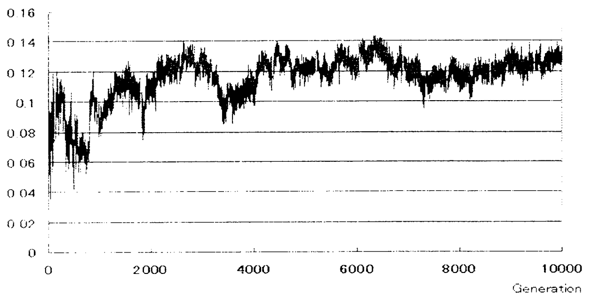


Figure 3: Average fitness of agent species (on a *SimpleCA* model, n = 3, c = 3, d = 3, i=50, g=60, T=400, s=10, k=3, R=1.5, v=2, Q=0.3).

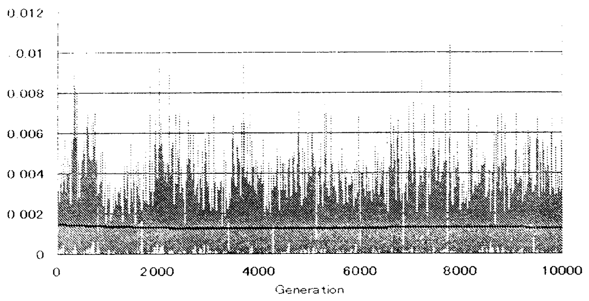


Figure 4: Degree of divergence between ideal and current environment: (gray line: the average of divergence degree of every colors, black line: the approximated curve).

"ambling" agents that behave by a hit-or-miss method. The strategy of agents comes to a relatively stable state without large oscillations beyond about the 1000th generation. At the same time, the behaviors of the agents also begin to change into the action intended to obtain assured income. The same strategy cannot always take a high fitness, because of the "noisy" situation in our model including some sort of bounded rationality. On the other hand, there is a case that a nice strategy continues to exist for a long time and similar strategies are generated by mutation, which cause a decrease in average fitness owing to decrease in diversity. The "wait" action gradually disappears as generation changes. The reason seems to be that it becomes more advantageous to do anything even though there is a risk to pay a cost rather than doing nothing.

Figure 4 shows the effect of agent activities on environments. The degree of divergence is measured as follows:

$$\frac{1}{c}\sum_{i \in C}(E_i - I_i)^2, \qquad (3)$$

where E_i is the actual existent rate of color i, and I_i is the ideal existent rate of color i, I_i is calculated as the average existent rate only with the autonomous change in environment and without agents. In this case, when *SimpleCA* is adopted, the ideal average existent rates of all colors are same because each color has same probability of existence. Figure 4 shows that they did not change remarkably though there are oscillations in any generation. There is no significant difference among the influences of agents on the environments through generations. It is a remarkable point that the evolution of agents doesn't have more effect on environments, while it brings about the increase in gains of agents.

4 Summary

ColorChanger is an attempt to throw light on the interactions between the agents that influence the environment and the dynamics of environment so as to resolve environmental problems. It is also very important not only the current issues but also to consider the problems in future which results from these interactions to solve environmental problems essentially in general. *ColorChanger* also aims to be a useful tool for that, and some results of our preliminary experiments with an environmental model based on a dynamics of the ecosystem were reported. There are some noises which influence the information in these experiments: the difference among the accurate environment information on the whole game fields, the environmental information obtained by agents, and the information on distribution of colors used for allocation of rewards. We have shown that the agents could increase their fitness in spite of these factors.

References

[1] E. Akiyama and K. Kaneko, "Evolution of Cooperation in Social Dilemma - Dynamical Systems Game Approach," *Proceedings of the Seventh International Conference on Artificial Life*, 186-195, 2000.

[2] R. Axelrod and D. Dion, "The Further Evolution of Cooperation", *Science*, 242, 1385-1390, 1988.

[3] R. J. Williams and N. D. Martinez, "Simple Rules Yield Complex Food Webs," *Nature*, 404, 180-183, 2000.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Stochastic Matching Agents.

German Hernandez
Department of Computer Science
National University of Colombia
Bogota, Colombia.

Fernando Niño
Department of Computer Science
National University of Colombia
Bogota, Colombia.

Khalid Khouri
Division of Science-Mathematics
Northwest Mississippi Community College
Senatobia, MS 38668, USA.

Yofre Garcia
Department of Mathematics
National University of Colombia
Bogota, Colombia.

Abstract

We propose and study a class of stochastic agents that solve matching problems. A Markov decision process model is developed in order to study the behavior of these agents.

1 INTRODUCTION

Matching problems arise in a wide variety of real-world situations, a well-known example is the assignment of workers to a set of jobs. Some other examples of matching problems are: 1) In the US, Canada, UK and Scotland, thousands of graduating medical students are matched yearly to residency hospital positions.[5]; 2) In Norway and Singapore, high school students are matched to universities, and primary school students are matched to secondary schools [1]; and 3) In the US, Navy personnel is relocated in new jobs each year[2].

In some cases, an automated scheme accomplishes this task, but in other cases a group of people perform the matching process. In both cases, the goal is to try to keep both the workers and the institutions offering the jobs pleased. For instance, some automated schemes are based on the "Gale-Shapley algorithm" [1], and the schemes perfomed by humans are based on their experience, negotiation and deliberations skills. In this paper, we propose an automated system based on agents, but it keeps some essential elements of human based matching schemes.

The rest of this paper is organized as follows. In the second section, the matching process and the stochastic matching agents are described. In the third section, we present the matching process is modeled based on Markov decision processes. Finally, some conclusions are presented in the fourth section.

2 MATCHING PROCESS

Let us consider the problem in which a group of experts $C_1, C_2, ..., C_N$ should assign a group of jobs $\mathcal{T} = \{T_1, T_2, ..., T_K\}$ to a group of workers (employees) $\mathcal{W} = \{W_1, W_2, ..., W_N\}$. A job is defined by a set of attributes, say the i-th job is defined by a l-tuple $(t_{i1}, t_{i2}, ..., t_{il})$, and in order to assign a worker to a job, it is necessary to take into account his/her qualifications. The qualifications of a worker are defined by a set of r attributes, represented as an r-tuple $(w_1, w_2, ..., w_r)$.

In this model, an expert will be in charge of analyzing the skills of a worker to meet the i-th requirement of a job. In other words, an expert suggests a specific worker for a job, based on only one requirement. Thus, the solution to the problem is found when all the experts C_j suggest the same worker W_i for a specific job. The problem of assigning a worker W_i to a job T_m is called *deliberation*.

The experts communicates with each other, and each one of them has an algorithmic process that creates a *scenario*, which allows to analyze the situation of assigning job T_m to worker W_i by considering the values of the worker's qualifications (attributes). The deliberation process starts when a job T_m is chosen to be analyzed. At first, let us consider the job T_m fixed. Then, all the experts analyze the assignment of the workers to such job according to their expertise to assign workers to this job based on a specific attribute, i.e., each expert C_j will choose among the workers, the one that best meets the requirement of the job attribute he is analyzing. Each expert will respond with a worker and a score that measures the fitness of the worker to meet job T_m. Thus, expert

C_j has associated a function

$$f_j^{Tm} : \mathcal{W} \rightarrow [0,1]$$

which assigns a score to each worker depending on his/her qualifications to meet job T_m. Let W_j^* the candidate with the highest score found by expert C_j, i.e.,

$$f_j^{Tm}(W_j^*) = \max_{W \in \mathcal{W}} \{f_j^{Tm}(W)\}.$$

If several workers get the same highest score, then one of the candidates will be picked at random. Therefore, at the end of this step, a set of candidates and their corresponding scores is produced by the experts, namely,

$$\{(W_1^*, s_1^*), (W_2^*, s_2^*), ..., (W_N^*, s_N^*)\},$$

where

$$s_j^* = f_j^{Tm}(W_j^*).$$

Now, among all the candidates suggested by the experts, the worker with the highest score is chosen as a candidate to perform job T_m. Let W^* be such a worker. Again, if two or more candidates have the same score, then one of them is picked at random. Clearly, this candidate might not correspond to the best assignment of a worker to job T_m, because the maximum score was obtained by analyzing only one requireement of the job. Therefore, it will be necessary to check the worker's qualifications to meet the remaining attributes of the job. In order to do this, the expert that assigned the highest score to a worker will assume the control of the system.

In the next step, the expert that has the best assignment of a worker to job T_m, denoted C^*, informs its candidate to all experts $C_1, ..., C_N$ so they can analyze the assignment of this worker, and thus produce a measure of their approval or rejection of such candidate from their own points of view. The opinion of an expert C_j about W^* (its aproval or rejection) is encoded in the score

$$s_j^{W^*} = f_j^{T_m}(W^*).$$

Notice that $s_j^{W^*}$ is the score assigned to worker W^* which takes control of the system in the following steps.

In order to make a decision about either accepting or rejecting worker W^* to perform job T_m, the expert C^* receives the opinions of all the remaining experts about such candidate. The opinions of the experts are represented as a N-tuple $(s_1^{W^*}, ..., s_N^{W^*})$. If W^* is accepted by most of the experts, then the agent will stop successfully, and will assign W^* to job T_m. Otherwise, i.e., W^* is rejected, then the agent will remove this candidate from the pool of workers and will continue the *deliberation* process. This deliberation process will be repeated until either the agent achieves an adequate assignment or it fails to do so.

The decision of either accepting or rejecting W^* for the job T_m is based on a threshold criterium, as follows:

1. W^* is rejected if $s_j^{W^*} \leq \theta^{\min}$ for some $j \in \{1, \cdots, N\}$
2. W^* is accepted if $s_j^{W^*} \geq \theta^{\max}$ for all $j = 1, \cdots, N$
3. $P(W^*$ is accepted$) = \frac{\overline{s} - \theta^{\min}}{\theta^{\max} - \theta^{\min}}$ if $s_j^{W^*} \in [\theta^{\min}, \theta^{\max}]$.

where $\theta^{\min}$, $\theta^{\max}$ are the threshold value of rejecting and accepting candidate W^*, respectively; and

$$\overline{s} = \frac{1}{N} \sum_{j=1}^{N} s_j^{W^*}$$

is the mean of the scores given to W^* by the experts. The deliberation process of assigning a job will be repeated for the remaining jobs.

3 STOCHASTIC MATCHING AGENTS

We consider a *matching system* as composed of stochastic autonomous agents each one of them representing an expert. This system can be mathematically defined as a Markov decision process, which is defined next. Specifically, the process described above is modeled here as a non-homogenous Markov Decision Process (MDP) [4].

3.1 Markov Decision Processes

Markov Decision Processes (MDP's) are a class of stochastic sequential decision processes in which the cost and transition functions depend on time, on the current state of the system x and on the current action a.

Definition. A Markov Decision Process (MDP) $\mathcal{M}$ is 4-tuple $\mathcal{M} = (\mathbb{X}, \mathbb{A}, T, C)$ where

i) $\mathbb{X}$ is a finite set called the *state space* of the process;

ii) $\mathbb{A}$ is a finite set called the *action space* of the process. It is possible that the actions available depend on the current state; in such case, the set of available actions on a particular state x will be denoted by $\mathbb{D}(x) \subset \mathbb{A}$.

iii) $\{T_t : \mathbb{A} \times (\mathbb{X} \times \mathbb{X}) \rightarrow [0,1] \mid t \in \mathbb{Z}^+\}$ is called the set of *transition probability functions.* $T_t(a, x, x')$ is the probability of passing to state x' at time t when the current state is x and the action a is executed. $T_t(a)$ will denote the transition matrix at time t due to action a. In this case, the transition probability function may be represented as a sequence of transition matrices $\{T_t(a) \mid t \in \mathbb{Z}^+, a \in \mathbb{A}\}$.

iv) $C : \mathbb{A} \times (\mathbb{X} \times \mathbb{X}) \rightarrow [0,1]$ is called the *cost function.* $C(a, x, x')$ is the expected cost associated with the transition from x to x' due to action a.

Thus, a MDP defines two stochastic processes: a discrete time Markov process X_t with values in $\mathbb{X}$, called the *state process* and a stochastic process Δ_t with values in $\mathbb{A}$, called the *action process.* Such processes have the following Markovian transition probabilities

$$\begin{aligned} T_t(a, x, x') &= P(X_t = x' | X_{t-1} = x, \cdots, X_0 = x_0; \\ &\qquad \Delta_{t-1} = a, \cdots, \Delta_0 = a_0) \\ &= P(X_t = x' | X_{t-1} = x, \Delta_{t-1} = a) \end{aligned}$$

Definition. A *policy* of a MDP is a mapping π that assigns an action (either deterministically or randomly) to each state of the process. In general, a policy is a function

$$\pi : \mathbb{X} \rightarrow \mathfrak{P}(\mathbb{A})$$

where $\pi(x)$ is a random action with probability distribution $P(\mathbb{A}) \in \mathfrak{P}(\mathbb{A})$. $\mathfrak{P}(\mathbb{A})$ is the set of all the probability distributions on $\mathbb{A}$. If the set of avaible actions depends on the state, then we have the constrain that $\pi(x) \subset \dot{\mathbb{D}}(x)$.

3.2 MDP Model

Let us consider the problem of assigning a worker from a pool of workers $\mathcal{W}$ to a particular job T. In this case, the behavior of the matching system may be modeled as a non-homogeneous MDP as follows.

i) The state space of the process is

$$\mathbb{X} = (\mathcal{W} \cup \{success\} \cup \{fail\}) \times \wp(\mathcal{W})$$

where $\mathcal{W} = \{W_1, W_2, ..., W_N\}$ is the set of workers (employees); *success* is the state of the system when a worker is successfully assigned to a job, and *fails* is the state of the system when it fails to do so. $\wp(\mathcal{W})$ is the collection of all the subsets of $\mathcal{W}$. On the other hand, if the state is a worker W_i, such worker is the current candidate for the job. Thus, a state is a pair (x, K) where x is either the current candidate W_i, *success* or *fail,* and K is the set of current workers available. Each time that a candidate is not successfully assigned to a job, it is removed from the group of available workers.

ii) The action space is

$$\mathbb{A} = \{s, d\}.$$

where s denotes ***stop*** meaning that all the agents agree in the assigment of the current proposed worker to the job; and consequently, the system stops and responds by assigning such candidate W to the job; on the other hand, the action d, *continue deliberating,* means that the system must continue deliberating because there is no agreement between the agents. In this model, we consider that when the state of the process is a worker, then the set of available actions is $\mathbb{A}$; otherwise, i.e., when the state is either $success$ or $fail$, the set of available actions is $\emptyset$, the empty set.

iii) The transition function is defined as

$$T_t(s, (x, K), (x', K')) = \begin{cases} 1 \quad \text{if} & \begin{matrix} x' = success \text{ and} \\ K' = K \end{matrix} \\ 0 & \text{otherwise} \end{cases}$$

notice that s represents the *stop* action. On the other hand, if the action is *continue deliberating,* d, then the transition function is defined as:

1. $T_t(d, (x, K), (x', K')) = P(W_j = L(R(T, K')))$ with $K' = K - \{W_i\}$ if $x = W_i, x' = W_j$ and $t = 0, 1, \cdots n - 1$;
2. $T(d, (x, K), (x', K')) = 1$ if $x' = fail$, $K' = K$ and $t = n$; and ,
3. $T_t(d, (x, K), (x', K')) = 0$ otherwise.

where

$$R : \mathcal{T} \times \wp(\mathcal{W}) \rightarrow \mathfrak{P}\left((\mathcal{W} \times [0,1])^N\right)$$

is the *ranking operator* that represents the action of the agents when they produce a set of candidates and their corresponding scores

$$\{(W_1^*, s_1^*), (W_2^*, s_2^*), ..., (W_N^*, s_N^*)\}$$

by using their functions $\{f_1^T, ..., f_N^T\}$. Such candidates are chosen from a subset of workers in $\wp(\mathcal{W})$. Thus, the resulting N-tuple of candidates and corresponding scores is a random element of $(\mathcal{W} \times [0,1])^N$ with probability distribution

$$P\left((\mathcal{W} \times [0,1])^N\right) \in \mathfrak{P}\left((\mathcal{W} \times [0,1])^N\right)$$

and it is defined by the operator R. $\mathfrak{P}\left((\mathcal{W} \times [0,1])^N\right)$ is the set of all probabilty distributions on $(\mathcal{W} \times [0,1])^N$. There are many possibilities of defining the operator R. Particularly, R may be a deterministic operator in a system in which the workers are ranked by each agent according to some particular attribute, and the best one is chosen by each agent. R could also be defined by ranking the workers and then applying random selection on them (e.g., proportional selection, tournament selection, etc.) [3]. When the action taken by an agent is continue deliberating, then the current worker will not be considered in the future, because there was no agreement. In other words, each time the deliberation action is taken, the set of available workers is reduced by one. This is the reason why K', the second argument of R on the equation above, is equal to $K - \{W_i\}$, the set obtained by removing the candidate from the set of available workers at time t.

The second operator that appears in the definition of T_t is

$$L : \mathfrak{P}\left(\mathcal{W} \times [0,1]\right)^N \to \mathfrak{P}\left(\mathcal{W}\right)$$

is the *leader operator* a random operator that represents the action of the system when picking worker W^* from the group of candidates proposed by the experts. The worker W^* is a element of $\mathcal{W}$ randomly chosen with probability distribution $P(\mathcal{W}) \in \mathfrak{P}(\mathcal{W})$ and defined by the operator L. $\mathfrak{P}(\mathcal{W})$ is the set of all probability distributions on $\mathcal{W}$. The agent that takes control of the system is the agent C_j that proposed worker W^*. C_j will be called *leader agent.* As mentioned above, the leader agent will broadcast its candidate W^* to the other agents, and it will receive some response of either accepting or rejecting W^*. Then, the system will decide whether the agents continue the deliberation process or stop the process by assigning W^* to job T. The decision process is then continued as it was explained above in the policy section of the MDP model.

iv) Finally, the cost function

$$C : \mathbb{A} \times (\mathbb{X} \times \mathbb{X}) \to [0,1]$$

is defined as $C(s,(x,K),(x',K')) = 0$ and $C(d,(x,K),(x',K')) = 1$, i.e., the value

of the cost of continuing deliberating is 1, and the cost of stopping is 0, meaning that the total cost of the process of assigning a worker to a job is given by the number of required computational steps.

3.2.1 Policy

The policy to select an action (i.e., either agreement or deliberating) on a particular state is defined as a function

$$\pi : \mathbb{X} \to \mathfrak{P}(\mathbb{A}).$$

Remember that a state is represented as a pair (x,K) and the action space is composed of the actions *stop* (s) and *continue deliberating* (d). Thus, it will only be necessary to define the probability of picking the action *stop*, $P(s)$. Clearly, since there are only two possible actions, then $P(d) = 1 - P(s)$. Therefore,

$$\pi(x,K) =$$
$$\begin{cases} P(s) = 0 \text{ if } \exists\ j \in \{1,...,N\} \text{ such that } s_j^{W^*} \leq \theta^{\min} \\ P(s) = 1 \text{ if } \forall\ j \in \{1,...,N\}, s_j^{W^*} \leq \theta^{\max} \\ P(s) = \frac{\overline{\beta}-\theta^{\min}}{\theta^{\max}-\theta^{\min}} \text{ if } s_j^{W^*} \in \left[\theta^{\min},\theta^{\max}\right] \text{ otherwise.} \end{cases}$$

with $\overline{\beta} = \frac{1}{N}\sum_{j=1}^{N} s_j^{W^*}$ and $x = W^* \in \mathcal{W}$, i.e., the state is a worker. Otherwise, if the state is either *success* or *fail*, then the set of available actions is $\emptyset$, and thus the probability distribution given by the policy is empty.

4 CONCLUSIONS

A system based on stochastic agents that solve matching problems the way human experts do was proposed. In addition, such system was modeled as a Markov decision process. In future work, we will analyze if the matching produced by this system are stable in the sense of Gale and Shapley [1].

Acknowledgments. We thank Ravikumar Kondadadi form the Conscious Software Research Group at the University of Memphis.

5 Bibliography

References

[1] Gale D. and Shapley L., College Admissions and the Stablility of Marriage. American Mathematical Monthly, 69:9-15, 1962.

[2] R. Kondadadi and S. Franklin. Deliberative Decision Making in "Conscious" Software Agents. Submited to Agents 2000.

[3] M. Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA, 1996.

[4] M.L. Puterman. Markov Decision Processes in D.P. Heyman and M.J. Sobel eds., in *Handbook in Operations Research and Management Science, Volume2 Stochastic Models,*. North Holland, 1990.

[5] R.W. Irving, Matching medical students to pairs of hospitals: a new variation on a well-known theme in Proceedings of ESA'98: the Sixth Annual European Symposium on Algorithms, vol. 1461 of Lecture Notes in Computer Science, pp. 381-392.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

What Is the Ultimate Form of Communications?

Ryohei Nakatsu
ATR Media Integration & Communications Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288 Japan
Tel: +81-774-95-1400, Fax: +81-774-95-1408,E-mail:nakatsu@mic.atr.co.jp

Abstract

This paper treats the issues of ideal communications and the way to achieve this. First, it is discussed that the technological trend of moving from logical to ideal communications is similar to a journey resulting in spiritual enlightenment in trying to find the original self. Then, it is stated that new media that have interaction capabilities and can realize immersion have the possibility of achieving the ideal communications. Also it is pointed out that there is a danger to get into the imperfect communications on the way to achieve ideal communications referring to the same phenomena in Zen.

1. Introduction

Recently, the word *kansei* has come to be used quite frequently. In addition, the communications of today's young people who use cellular telephones is sometimes called "*kansei*-transmitting communications." Also it is often alleged that the animation movies and video game contents produced in Japan is one example that the *kansei* of the Japanese is of a high caliber. However, there are also a number of adults who frown upon young people who become engrossed in long conversations on cellular telephones. A lot of voices can also be heard criticizing media, especially video games, which are said to distort the boundary between "reality" and "virtual reality" for children addicted to them. Why have such polar reactions emerged?

In this paper, first, an investigation is made on the meaning *kansei* has in communications. Next, it is pointed out that the technological trend of moving from communications handling the conventional logical information to communications handling *kansei* information is similar to a journey resulting in spiritual enlightenment in trying to find the original self. In addition, it is stated that the process resulting in spiritual enlightenment is likely to extend in the wrong direction called the "evil border" as time passes, and it is shown that this "evil border" is appropriate in explaining why children who are addicted to video games fall into a state of being unable to discriminated between "reality". and "virtual reality" as time passes.

Then it will be shown that both ideal communications and imperfect communications hava a clear distinction with the conventional logical communications in the sense of giving a feeling of immersion, but what distinguishes between the two is the existence or non-existence of interaction. Also it will be shown that active type media, with which we can positively interact, and that give a sense of immersion show the possibility of achieving the ideal communications.

2. *Kansei* Communications

2.1 State of Kansei Communications

Conventional communication technologies had come to consider that only logical information was sent and received in human communications. Recently, however, it has generally become clear that this way of thinking is insufficient. Looking at our daily lives, the exchange of sensations, emotions, and moreover, higher-level *kansei* information with one's partner plays the main role.

For example, the appearance of cellular telephones can be said to have largely changed telephone communications. This fact has been clearly shown recently particularly via the state of communications by young people using cellular telephones. The characteristic of such communications is not in the clarity of the messages these young people want to send to the other side, but in the point of continuing their evasive conversations with the other side. In other words, messages with logical mea' ings are probably not transmitted, but *kansei* message, are definitely transmitted.

However, there are a number of adults who frown at

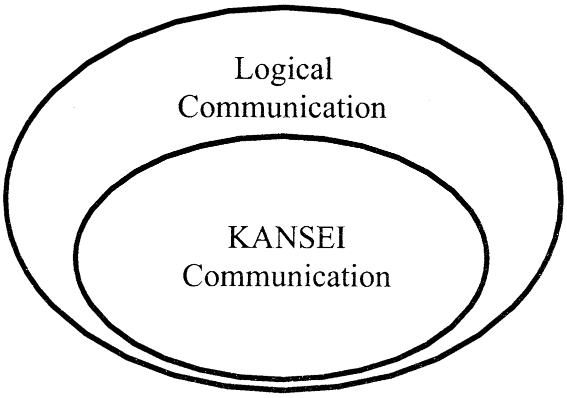


Fig.1 Communication Model

the appearance of young people having fun with long conversations on cellular telephones on roads, in trains, and so on. In addition, looking at the children that have fun with video games, there is the danger of these children being unable to discriminate the real world and the virtual world, and some people return to video games as the cause of recent crimes by such young people. Why has such a thing emerged? It is possible to explain the answer from the relationship of *satori* with communications.

2.2 Model of communications

First, an attempt is made to explain current conditions from the viewpoint of communications. Figure 1 shows a modeļ of communications. This model of communications is composed of two layers, i.e., one layer to handle logical information and one layer to handle *kansei* information.

The conventional communication technologies had primarily come to handle communications based on logic. It cannot help but be said that this was insufficient. Originally, humans basically thought that they could handle both logical communications and *kansei* communications. However, the problem here was that people stuck to the thinking that the layer performing logical communications at the surface was operating as the main layer, versus the operating of a layer performing the original logical communications and *kansei* communications inseparably connected, and that the layer performing *kansei* communications was rarely directly exposed to outside information, thereby becoming an open condition.

Incidentally, it is conceivable that the progress of recent media technologies has led to the following forms: displayed images, large-volume sounds, and very realistic computer graphics, which work directly to the *kansei* communication layer. This has made it possible for the *kansei* communication layer, which is rarely affected by information directly outside, to become overloaded, and to result in unusual responses at times. Isn't it possible that this has caused young people to commit crimes recently?

3. *Satori* and *Kansei* Communications

3.1 Process to reach *satori*

In Zen, it has been explained that the cause of human distress and worldly desires is that the original self and the self in real life are separated. It is possible to think that the above-mentioned phenomenon can be explained completely by the same point of view.

First, an explanation is given on the process to reach Buddhist *satori*. Famous pictures called "The Ten Oxherding Pictures" (Fig. 2) exist that explain the process to reach *satori* in Zen Buddhism. These pictures are utilized in the explanation.

(1) Searching for the Ox

In this picture, the ox can be considered to symbolize one's original self that was lost sight of. The picture shows the state of determination (by the person) of attempting to rediscover, by some trigger, the original self that was lost sight of, after being involved in a variety of daily complex chores.

(2) Seeing the Traces

This picture symbolizes that, through various types of disciplinary training, it is possible to interpret one's original self that was lost sight of when some clue is obtained.

(3) Seeing the Ox

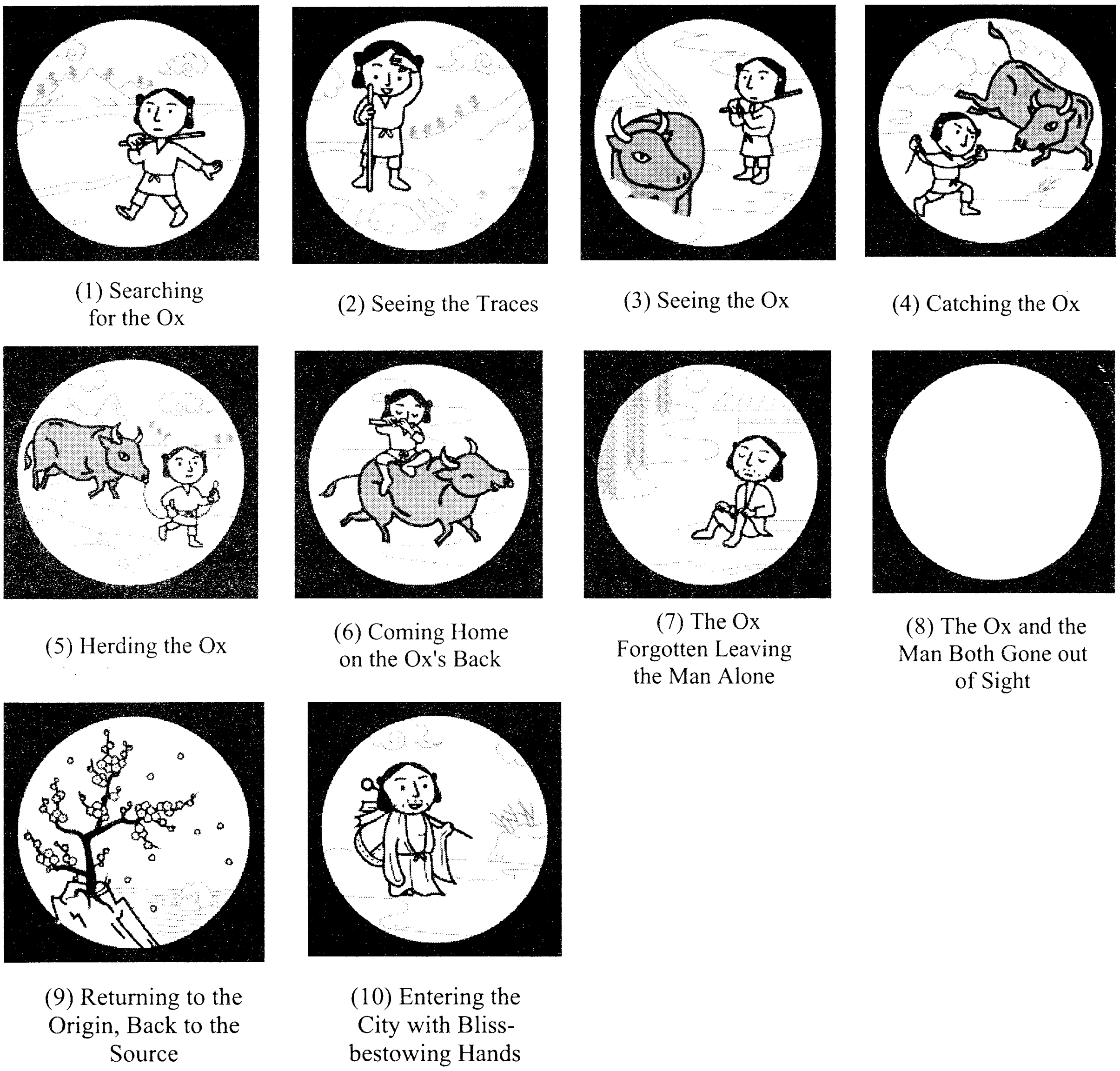
This picture symbolizes the state of finding one's original self, after various types of training and trial and error.

(4) Catching the Ox

This picture shows the state of attempting to catch the discovered ox. This picture symbolizes that it is quite difficult to accept what one thinks is one's discovered original self into one's actual self.

(5) Herding the Ox

It shows the state in which *satori* is finally obtained after a long period of disciplinary training. However, the picture shows a high possibility that the *satori* seem-



(1) Searching for the Ox

(2) Seeing the Traces

(3) Seeing the Ox

(4) Catching the Ox

(5) Herding the Ox

(6) Coming Home on the Ox's Back

(7) The Ox Forgotten Leaving the Man Alone

(8) The Ox and the Man Both Gone out of Sight

(9) Returning to the Origin, Back to the Source

(10) Entering the City with Bliss-bestowing Hands

Fig. 2 The Ten Oxherding Pictures (by **Keiko Nakao**)

ingly obtained is still a "small" *satori*, i.e., in a step along the way towards true *satori*.

(6) Coming Home on the Ox's Back

This picture shows the state of riding on the back of the ox, after taming the ox, and returning home. Accordingly, it shows the state in which one completely finds one's original self, that is, the state in which one obtains true *satori*.

3.2 Consciousness and sub-consciousness

Let us explain the relation between original self and the self in real life from a viewpoint of human consciousness and sub-consciousness. Figure 3 shows a modeling of the relationship of human consciousness and sub-consciousness. It can be considered that the human sub-consciousness is typically hidden under the activities of the human consciousness, at the base of the control of the human consciousness. *Satori* can be considered to be a process to cut off conscious behaviors and subconscious behaviors from the relationship of oppression and non-oppression, and to create conditions enabling both sides to be unified.

In Zen, a representative of the disciplinary training for reaching *satori* is Zen Meditation. Zen Meditation involves making the inside of the heart empty by sitting in silence and minimizing the exchange of information with

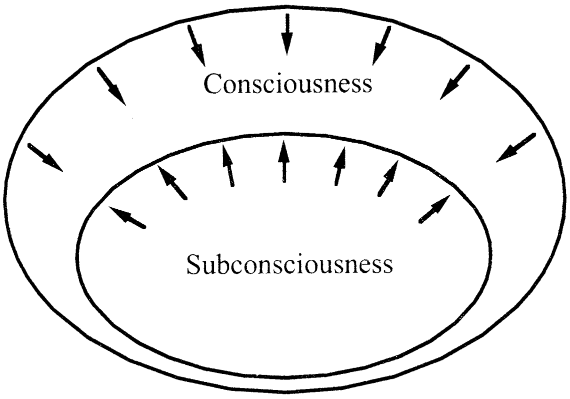


Fig. 3 Consciousness and Sub-consciousness

the outside. This can be thought to be a process to suppress the behaviors of the conscious layer and to liven the behaviors of the subconscious layer. In this way, an attempt is made to integrate the behaviors of the conscious layer and the subconscious layer by suppressing the consciousness and magnifying the sub-consciousness. This condition is the so-called true *satori*.

However, this process has the possibility of moving to a dangerous state as time elapses. The human subconsciousness is generally placed under the control of the human consciousness, i.e., placed in the suppressed state. Because of this, suppressed desires and urges sometimes spout in the step in which the human consciousness is suppressed and the behaviors of the human subconsciousness become animated. Because suppressed desires and urges can come to appear on the surface, they can appear as images taking such forms as demons or the evil one, like dreams, and can sometimes appear taking such forms as temptation by lust. In any case, this is not true *satori* but a step along the way to *satori*. Zen calls this the "evil border", and strongly rebukes this so that it is not confused with *satori*.

As can be understood from the above explanation, most of the conditions able to be achieved by communication technologies involve not true *satori*, but the "evil border". It can be seen that people who notice this intuitively are emphasizing the dangers new types of media (e.g., cellular telephones, video games, etc.) possess as well as the negative side, and are spreading this through the mass media. This itself is not wrong, but if we try changing this one-sided viewpoint, then we can believe that new types of media and communication media are readying the possibility to lead people to *satori*, and it will generally not be necessary to criticize new types of media.

4. Toward the Realization of Ideal Communications

4.1 Active immersion

When replacing the above discussion with the word of "communications", it is possible to say that the integration of logic communications and *kansei* communications is the ideal communications that is not biased towards logic, and is not biased towards *kansei* as well. In this section, it will be discussed what type of sensation people have through this ideal communications and how we can achieve the ideal communications.

Through the integration of logic communication and kansei communication, it can be said that we would have integrated experiences. Probably this integrated experiences resemble the experiences in art creation such as the creation of drama, performance, and sculpture. The typical feeling they have during this experience is fascination and uplifting. There are probably various kinds of fascination and uplifting, but here the word "immersion" is used as a general term for these. In an integrated experience, many types of things that are felt give the sensation of uplifting and fascination, but can any such sensation be useful in achieving the ideal communications we discussed above? Here, we think it is important to make a key distinction. By carefully investigating immersive situations, we find that we can classify them into "passive immersion" and "active immersion."

Does not the level of consciousness differ between the two types of immersion, although there is no difference between becoming absorbed in passive immersion and in active immersion? Is there not a big difference in the condition of fascination between active immersion and passive immersion? The condition where one forgets oneself or the condition where one loses consciousness (fascination, hypnosis, trance) becomes a key factor that explains the feeling of passive immersion.

Active immersion, in contrast, is the condition whereby one's consciousness is maintained in a normal manner while that person becomes absorbed. Conditions of immersion while clearly maintaining consciousness include doing work with concentration and immersion in the act of creating art. Even in the case of sports, such conditions are experienced.

When such passive immersion or active immersion pays attention to the processes that take place, an interesting fact becomes clear. It is the existence or non-existence of interaction. We feel passive immersion when we are only receiving information, such as when we watch movies. In other words, there is a lack of interaction here. In contrast, active immersion differs from the point of working on the object, such as creating art and participating in sports. In other words, interaction exists with active immersion. Consequently, the existence or non-existence of interaction is the key that distinguishes passive immersion and active immersion. From the above discussion, it is clear that "active immersion" and "passive immersion" correspond to ideal communication and imperfect communication respectively.

4.2 Route to *satori* communications

From the above explanation, it is possible to provide a number of routes to reach *satori*. It is natural to think that *satori* in so-called Zen is attained by disciplinary training centered on Zen Meditation. In contrast, the key point with our ideal communications, as explained above, is the existence of interaction with a sense of immersion. Media giving a sense of immersion had already existed in the past, such as movies and novels. By adding some functions, i.e., having people interact with the powers of movies and novels, and having people that are enjoying media simultaneously become senders, and having senders and receivers carry out communications from the same standpoints, there is a high possibility that the conventional entertainment, amusement, and communication media will become integrated, and proceed to higher-degree communication media. Such new media can be considered to exceed *kansei* communications, and to give a feeling of an integrated experience of interaction with a sense of immersion. In the world of communications, is it not possible to rephrase this as *satori* being achieved (Fig. 4)?

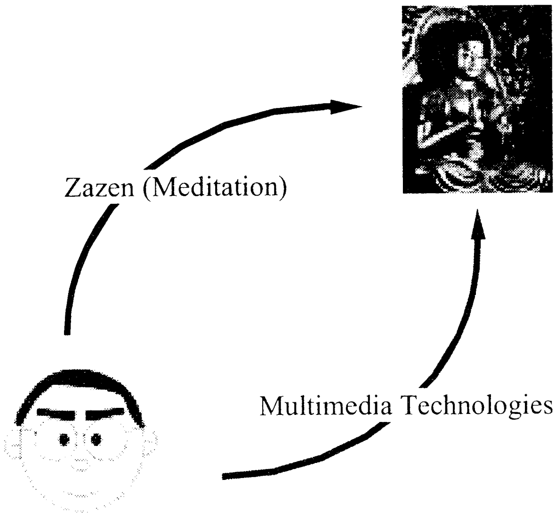


Fig. 4 Approaches toward SATORI

5. Conclusion

In this paper, it was discussed that the interactive media has the possibility of achieving high level of communications that can correspond to *satori*. *Satori* is a condition in which peace of the heart is achieved. However, there are also viewpoints that *satori* is something that is achieved only when we are isolated from general life, and that it has no relationship with our usual way of life, that is, interactive media relate only to a special part of human life, i.e., to the domains of entertainment and amusement, and do not contribute to practical parts. Responding to this argument, the author wants to point out that the latter part of "The Ten Oxherding Pictures," Fig. 2 (7) – (10), clarifies that there is a process to return once again to real life at the point of *satori* The detailed discussion on this issue will be done elsewhere.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Robovie: Communication Technologies for a Social Robot

Michita IMAI
ATR MI&C Research Lab.

Tetsuo Ono
ATR MI&C Research Lab.

Hiroshi Ishiguro
Wakayama Univ.

abstract

ATR Media Integration and Communications Research Laboratories has developed a humanoid-type robot that can work in our daily life and naturally fit into human society. This robot will be a platform for developing various types of service robots, such as cleaning robots, patrol robots, and entertainment robots, based on a rich communications ability.

1 Introduction

A new type of robot has been developed and appears to work in living spaces of humans. This new robot, named Robovie, is different from the ordinary robots working at factories. In particular, Robovie requires skills for daily communications with humans when participating in human society. It was developed as an everyday robot to investigate the necessary functions for communications. This paper describes some problems of communications between a human and a robot through an explanation of Robovie.

For a robot to participate in human society, it has to be able to communicate with people naturally while carrying out its tasks. In other words, the robot must not only obey commands from users but must also generate requests to ask people to solve problems it cannot deal with by itself. Such bidirectional communications is vital if the robot is to succeed in human society and to play some role. To date, however, no one has proposed a method of communications between humans and robots in terms of robot participation in human society.

An ordinary robot has the following features.

- Master-slave relationship where a person gives the robot commands and the robot obeys them.
- Code model communications where each command used between the person and the robot has a static meaning.
- Task oriented design where the robot's behaviors are designed only to carry out given tasks.

In short, the robot is designed as a tool and communicates with people as a tool. For example, a cleaner robot is designed with behaviors and commands only for cleaning somewhere.

From another perspective, the design theory in terms of tool robots is insufficient for developing new types of robots capable of performing activities in the living spaces of humans; it does not consider that a person might be requested by a robot to do something. Almost all people might be unaware of the robot's requests as defined by its programmer when they run into the robot for the first time by accident. In short, to work in the living spaces of humans, the robot must have an ability to communicate independent of the ordinary command set. Moreover, the robot must be able to vary its role in human society because the robot may encounter an event not related to a given task.

According to above discussion, this paper proposes an everyday robot named "Robovie." The problem of master slave relationship, code model communication, or task oriented design comes from the lack of a relationship between a human and a robot. Robovie achieves a new type of human-robot communications by developing a relationship with a person via physical interaction. Robovie has the following features.

- Peer: Robovie achieves bidirectional communications with a person by developing a relationship with him/ her dynamically.
- Mutual mind-reading: The person and Robovie can communicate with each other by inferring the other's communicative intention based on the relationship developed between them.
- Everyday behaviors: Robovie can perform everyday activities (autonomous battery charging function and autonomous behaviors) for basic interaction in human society.

The remainder of this paper is organized as follows. Section 2 describes a crucial element for human-robot interaction in human society. Section 3 describes the mechanisms of Robovie and explains the communications process between a person and Robovie. Section 4 concludes the paper with a summary.

2 Requests from a robot

Bidirectional communications between a person and a robot is an important function if the robot is to participate in human society. In the development of Robovie,

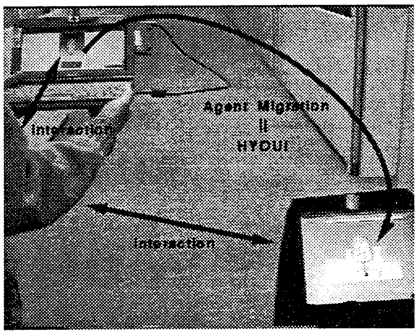


Figure 1: The migration of a relationship according to the migration of a CG character.

we conducted a psychological experiment [OI00] where the robot requested experimental subjects to perform a task. Before explaining the details of the experiment, we describe the results. The results indicated that the relationship between the subjects and the robot was a crucial factor in determining whether the subjects would obey the request from the robot. The rest of this section gives a brief explanation about the experiment. In the experiment, an autonomous mobile robot requested the subjects to perform a task upon appearing in front of them suddenly (Fig. 2). The request message was "Please move the trash can!" The message was generated with a speech synthesis system.

The experiment investigated effects of the relationship between the subjects and the robot on the request from the robot. We employed the migration of a CG character to develop the relationship between the subjects and the robot. The display of a mobile PC (upper left side of Fig. 1) showed the CG character. In addition, the robot (lower right side of Fig. 1) could also display the CG character. In the first step of the experiment, the subjects interacted with the CG character on the mobile PC. In this step, the subjects came to form a relationship with the CG character. In the next step, the CG character disappeared from the display of the mobile PC and appeared on the display of the robot. According to the migration of the CG character, we gave the subjects a relationship with the robot.

Figures 2 and 3 show results of the experiment. The subject in Fig. 2 was given a relationship with the robot according to the migration of the CG character. As a result of the migration, he obeyed the request from the robot naturally and moved the trash can. The subject in Fig. 3 was not given a relationship with the robot because her CG character did not migrate to the robot's display. As a result, without the migration, she ignored the request from the robot appearing suddenly in front of her.

These results suggest that forming a relationship between a person and a robot is crucial for the robot to request the person to do something.

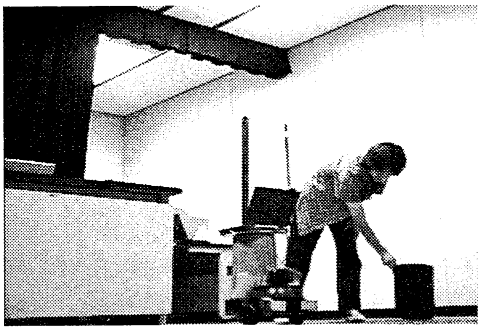


Figure 2: Subject who could understand the request from the robot.

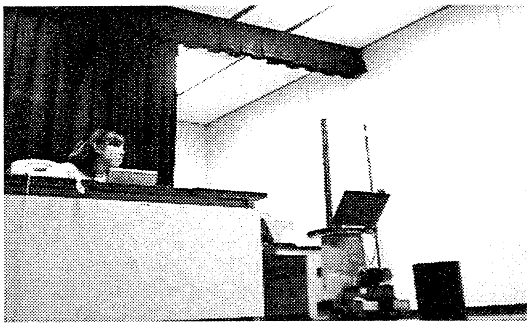


Figure 3: Subject who could not understand the request from the robot.

3 Robovie

Robovie was developed to investigate human-robot interaction in terms of the development of a relationship. The upper part of Robovie consists of one head and two arms like a human body, and Robovie develops a relationship with a person by physical representation. The lower part of Robovie is a mobile base with two wheels and one caster. Figure 4 shows an overview of Robovie. The arms of Robovie have four degrees of freedom enabling the hands to move like human hands. In addition, since the points fixed on the arms bend towards the front of Robovie, it is easy for the arms to reach the front part of Robovie. According to the movable range of the arms, Robovie can generate gestures more naturally. In addition, the head has three degrees of freedom to move like a human head.

Table 1 shows Robovie's other specifications.

Robovie moves around and avoids obstacles with ultrasonic distance sensors. In addition, it can notice when a person touches it with its skin sensors. It has two cameras in its head to find a battery station with stereo vision in most cases. The back of Robovie has an omni-directional vision sensor, which captures a panoramic scene (around Robovie) via one shot. The omni-directional vision sensor is mostly used by Robovie

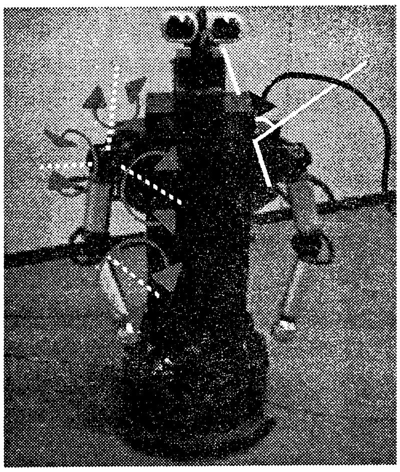


Figure 4: Everyday Robot "Robovie"

to find its location by comparing the captured current image with previously captured images. In addition, Robovie has a speech synthesis system and a recognition system to communicate with people via vocal messages.

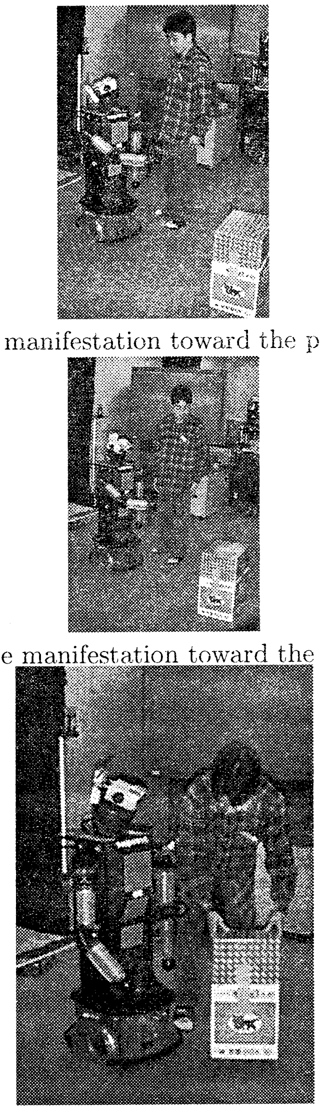
3.1 Communications based on a relationship between a human and robot

Robovie achieves communications based on the formation of a relationship with a person with the above hardware. The crucial methodology for developing the relationship is physical representation.

The most important factor for the development of the relationship is the behavior of gaze. In particular, the most important factor in the behavior of gaze is eye-contact between the person and Robovie to develop a relationship (the first picture in Fig. 5). Robovie also employs gaze for another communication function. The function is called attention manifestation and it manifests the direction where Robovie pays attention to. The second picture in Fig. 5 shows Robovie manifests the direction of attention towards the box.

Robovie can request a person to perform a task such as "Please move this box!" by eye-contact with the person and attention manifestation towards the box (the third picture in Fig. 5).

In addition, Robovie has an autonomous battery charging function. This function was implemented to allow us to investigate a process in which Robovie can acquire a role in human society dynamically. In short, Robovie can find a battery station and charge its battery automatically when the voltage level becomes low. Since Robovie is designed to work in the living spaces of humans without the aid of humans, it has its own life-cycle. As a result of this life-cycle, we can promote research that investigates the communication functions necessary for dynamical role changes.



Gaze manifestation toward the person.

Gaze manifestation toward the box.

The person move the box in response to the physical representation and vocal request.

Figure 5: Communication with gaze behavior

4 Conclusion

This paper described an everyday robot named Robovie. Before developing Robovie, we conducted a psychological experiment to acquire knowledge on the interaction between a person and a robot. Our experiment suggested that verbal interaction alone was insufficient for bidirectional human-robot communications. The important factor for achieving such communications is the development of a relationship between the person and the robot. As a result, we gave Robovie a physical representation function to develop a relationship with people. At present, interaction with Robovie

Table 1: Specifications of Robovie

Size:	H 114cm×W 52cm×D 50cm
Weight:	39kg
Max movable speed:	1.6 m/sec
Max arm speed:	200 dcgrccs/scc
Battery specification:	DC12V 21Ah
Movable time:	3.5 hours
Movable base specification:	2 wheels (drive), one caster
Minimum height from ground:	3cm
Cameras :	Sony EVI-G20×2
Ultra sonic distance sensors:	24 pieces
Omni-directional vision sensor:	1 piece
Hand sensors:	Switch type×2 pieces
Skin sensors:	Touch sensor×16 pieces
Arm motors:	Harmonic drive DC motor ×8 pieces
Head motors:	Harmonic drive DC motor ×3 pieces
Computer:	K-6-II 400MHz
	HDD 6GB
	SDRAM 128MB
	Video capture board
Communication:	Wireless LAN

is being evaluated at several exhibitions, for example, ROBODEX 2000 (Fig. 6).

```
Web site:
http://www.mic.atr.co.jp/~michita/everyday/
```

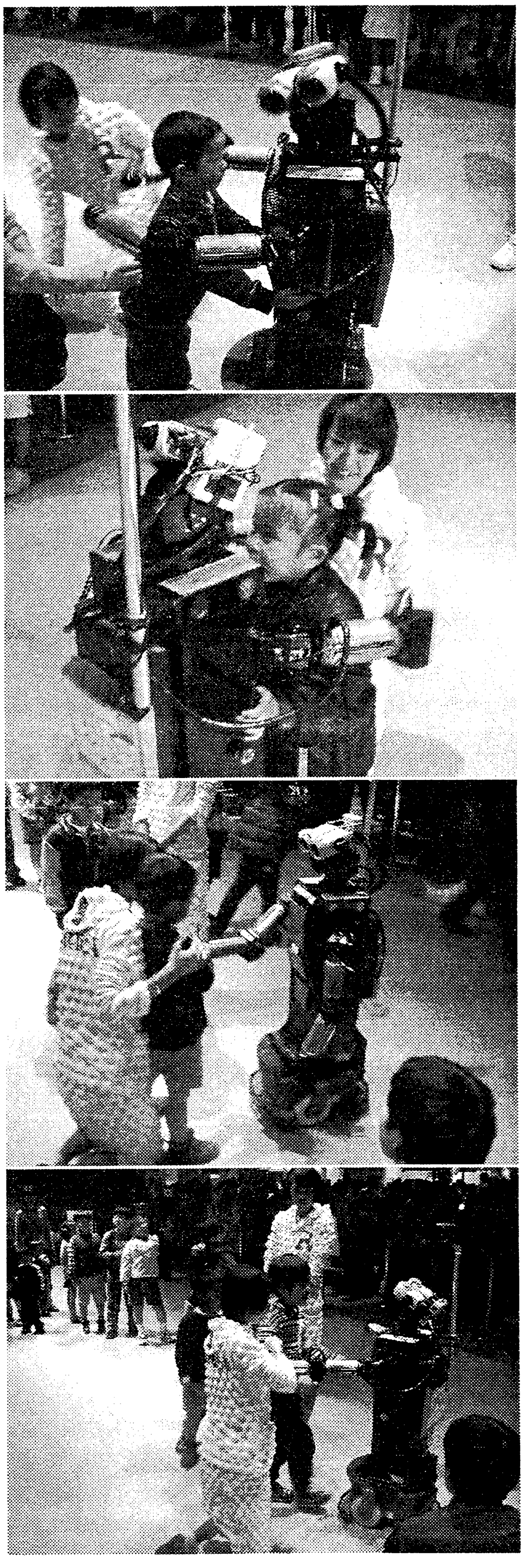


Figure 6: Exhibition at ROBODEX2000

References

[OI00] T. Ono and M. Imai. Reading a robot's mind: A model of utterance understanding based on the theory of mind mechanism. In *Proceedings of AAAI-2000*, pages 142–148, 2000.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Integration System for Emotional Recognition Using Visible and Infrared Face Images and Voice

Tatsuhiko Ikeda, Yasunari Yoshitomi, Tetsuro Kitazoe, Sung-Ill Kim
Department of Computer Science and Systems Engineering
Faculty of Engineering , Miyazaki University 1-1,
Gakuen Kibanadai Nishi , Miyazaki , 889-2192 Japan

E-mail : kitazoe@cs.miyazaki-u.ac.jp

Abstract

A new integration method is presented to recognize the emotional expressions. We attempted to use both voices and facial expressions. For voices, we use such prosodic parameters as pitch signals, energy, and their derivatives, which are trained by Hidden Markov Model (HMM) for recognition. For facial expressions, we use feature parameters from thermal images in addition to visible images, which are trained by neural networks (NN) for recognition. The thermal images are observed by infrared ray which is not influenced by lighting conditions. The total recognition rates show better performance than each performance rate obtained from isolated experiment. The results are compared with the recognition by human questionnaire.

Key words : Emotional Pattern Recognition, Emotional Speech, Prosody, Facial Expression, Thermal Image

1. Introduction

It is useful and perhaps necessary to introduce a bit of emotional taste into the course of communications between human and robots. Our future society will be more enjoyable if a robot understands the emotional state of a user. Not only human beings but also some of animals express their own emotions through voices, face expressions and body gesture in total. Therefore, it is one of the important subjects for the ultimate goal to human like robot to use those kinds of nonverbal communication methods in combination, although there have been many attempts to recognize those nonverbal characteristics separately.

In the present paper, we attempt to perform a new method modeling the emotional features and recognizing emotional states. We present an integration method of human speech as well as visible and thermal facial expressions, aiming total understanding of the human mental states.

The recognition by thermal images is, among others, stressed in the present study. The thermal facial expressions obtained by infrared ray are not influenced by lighting condition in a room, while the face images by visible ray change very much depending on the surrounding circumstance.

2. Emotional Feature Extraction

For recognizing emotional states in both voices and facial expressions, we need to extract emotional feature parameters from them. We first analyze voices which contain emotional information including four kinds of feature parameters. As well as emotional feature extraction from voice, we also extract useful feature parameters from facial expressions of both visible and thermal images.

2.1. Emotional Feature Extraction from Voice

The prosody[1,2] is known as an indicator of the acoustic characteristics of vocal emotions. In our experiments, we used four kinds of prosodic parameters, which consist of fundamental pitch signal (F0), energy, and each derivative elements. The pitch signals in the voiced regions were smoothed by a spline interpolation. In order to consider the effect of a speaking rate, furthermore, we also used a discrete duration information when training Hidden Markov Models (HMM).

We analyze the feature parameters from the speech waveform shown in Figure 1, considering only the voiced regions as data points. All speech samples were labeled at the syllable level (/Ta/ and /Ro/) by manual segmentation in order to train each HMM using separated feature parameters. Taro is one of the most popular male name in Japan like John in English, which does not have any specific emotional meaning in itself. Figure 2 and 3 show the pitch and energy signals extracted from each emotional speech of /taro/ spoken by a female announcer, respectively. In the figures, we can see that the feature signal of an emotion, anger, is the highest among all signals in each graph.

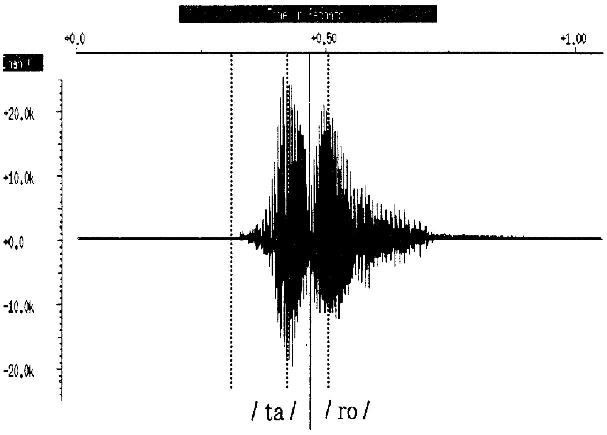


Figure 1. Speech waveform labeled by two parts /ta/ and /ro/.

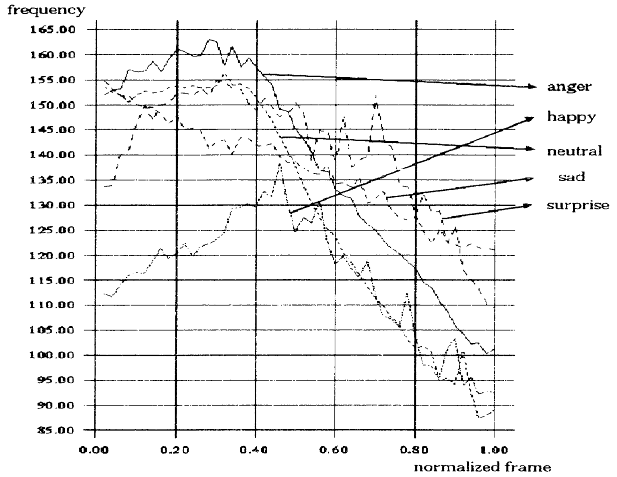


Figure 2. Pitch signals in each emotional state.

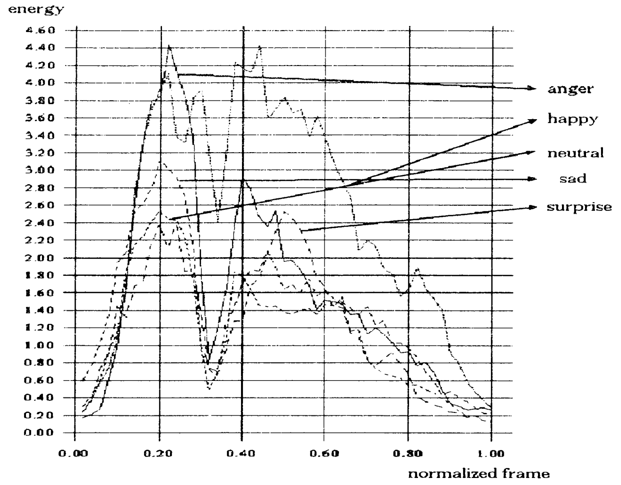
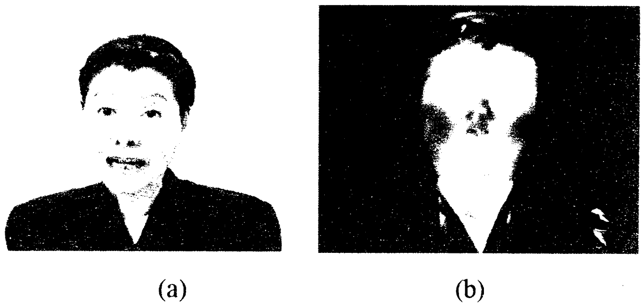


Figure 3. Energy signals in each emotional state.

2.2. Emotional Feature Extraction from Visible and Thermal Image of Face

Many studies have been performed to tackle the issues of understanding mental states of human through facial expressions, using ordinary visible camera. However, those trials still seem to be tough jobs since there is a only slight difference among various facial expressions in terms of characteristic features for the gray level distribution of input image using visible ray (VR). Thus, we have attempted to apply thermal distribution images to facial expression recognition[3,4] using an infrared ray (IR). Figure 4 illustrates the examples of female face images by VR and IR. VR image has the shortcoming that the accuracy of facial expression recognition is greatly influenced by a lighting condition including variation of shadow, reflection, and darkness. However, it is perfectly overcome by exploiting IR.

When a face image is given into recognizer, it is necessary for better recognition performance to decide when to extract face images in accordance with voiced parts and where to take IR or VR images in the face part of the subject. We take three timing positions to extract face images, shown in Figure 1 as dotted lines where the first one is just before the beginning of the voice, the second and third ones are the maximum voice parts of /ta/ and /ro/, respectively. Figure 5 and 6 show an extraction of face-parts areas which consist of three areas in the VR image and six areas in the IR image, respectively.



Figur[illegible] xamples of face images of VR(a) and IR(b)

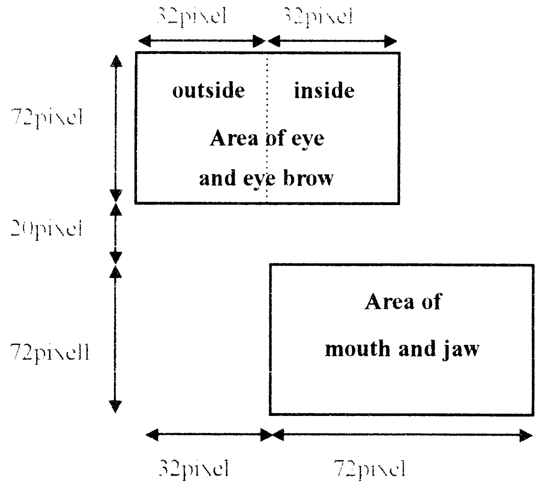


Figure 5. Extracting face-parts areas in the VR image.

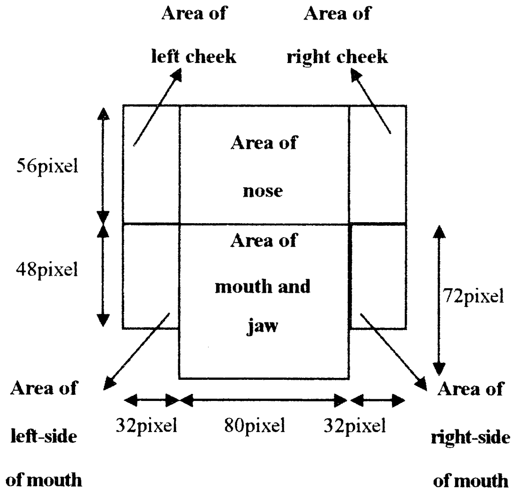


Figure 6. Extracting face-parts areas in the IR image.

In the next step, we generate differential images between the averaged neutral face image and the test face image in the extracted face-parts areas to perform a discrete cosine transformation (DCT). Figure 7 illustrates the procedure of recognizing the emotional state expressed in voice and face images.

3. Recognition of emotion

In case of processing the emotional voice, the speech is sampled in the experimental conditions illustrated in Table 1 for pre-processing of emotional voice recognition, from which four dimensional emotional features are extracted.

We then train the discrete duration continuous Hidden Makov Models (DDCHMM) by using these features with three states, using label information, and run recognition tests.

Sampling rate	16Khz , 16 Bit
Pre-emphasis	0.97
Window	16 msec. Hamming window
Frame period	5 ms
Feature parameters	pitch signal (F0), energy, delta pitch, delta energy, discrete duration information

Table 1. Analysis of speech signal

In case of processing the facial images, on the other hand, 63 and 45 bits of feature parameters are used as input data for neural network (NN) with three layers for IR and VR facial expressions, respectively. The unit number of hidden layer is decided experimentally for improving the recognition accuracy for facial expression. The unit number of output layer is the number of facial expressions which should be recognized.

Figure 7 shows the overall procedure of recognizing emotional states using integration method of three different recognition results. The integration of three separate recognition accuracies is performed by following equation.

$$S_i = \sum_{j=1}^{3} x_{i,j} \qquad (1)$$

where $x_{i,j}$ is an output value (1 or 0) for an emotional state i using a method j. Therefore, recognition results are chosen when the S_i is maximum. However, the only output value from the method with voice is accepted in case that the recognition results from isolated methods scatter. Moreover, the output from the method with image is not accepted when the output intensity of its neural network is less than 0.5.

4. Experimental Results

We first examine a human performance on three different types of questionnaire. Next, we examine how effective our integration of three kinds of recognition methods is, when we compare the simulation performance results with human performance ones.

4.1. Database

The samples consisted of semantically neutral utterance, Japanese name 'Taro', spoken and acted by an announcer. We capture voices and images simulating five emotional states such as neutral, anger, happiness, sadness, and surprise. We have simultaneously recorded voices and image sequences containing emotional states. We assembled 20 samples per each emotional expression as training data and 10 as test data.

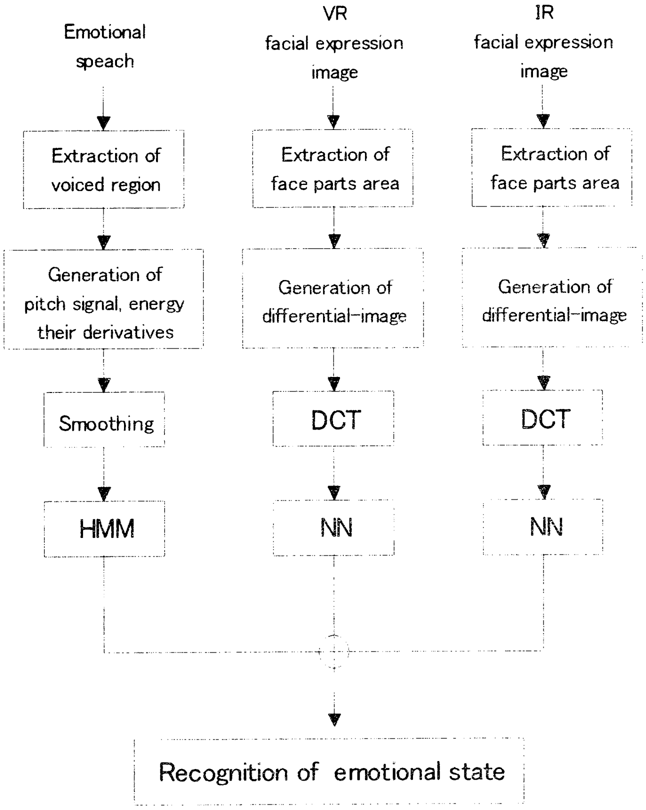

Emotional speach
VR facial expression image
IR facial expression image
Extraction of voiced region
Extraction of face parts area
Extraction of face parts area
Generation of pitch signal, energy their derivatives
Generation of differential-image
Generation of differential-image
Smoothing
DCT
DCT
HMM
NN
NN
Recognition of emotional state

Figure 7. Procedure of recognizing the emotional state expressed in voices and face images.

4.2. Questionnaire Results

The emotional states of speech, image samples, and both of them are estimated subjectively by total 21 students who consist of 17 male and 4 female students. Table 2 shows the three kinds of human performance results we obtained. As shown in this table, the average recognition rates for five emotional states are 94.2%, 99.0%, and 99.7%, when using emotional voices, facial expressions, and both of them, respectively. From these human performances, we could see that our data were included relatively correct emotional states and that the questionnaire result integrating both emotional voices and images gave better performance than those separately obtained from voices or images.

4.3. Simulation Results

We next performed recognition of mental states over the same test data used in the questionnaire test, by integrating voices and facial expressions. Table 3 shows the recognition accuracies for each emotional state in case of emotional voices, VR and IR facial expressions, and total recognition accuracies using the integration method of the three different recognition results, respectively.

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out_-put	Ang.	96.7	0.5	1	1	1
	Hap.	1.4	96.2	1	1.8	4.8
	Neu.		0.5	95.2	0.5	3.3
	Sad	0.4	0.5	0.5	96.2	2.8
	Sur.		0.5	0.5	0.5	86.7
	No ans.	1.4	1.8	1.8		1.4

(a) Human performance on emotional voices

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out-put	Ang.	98.1	1			0.5
	Hap.	0.5	96.2	0.5		
	Neu.			99.5		
	Sad				100	1
	Sur.					98.5
	No ans.	1.4				

(b) Human performance on facial expressions

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out-put	Ang.	99.5				
	Hap.		100			
	Neu.			99.5		
	Sad				99.5	
	Sur.	0.5				100
	No ans.			0.5		

(c) Human performance on both emotional voices and images

Table 2. Human performance on each emotional state

As shown in the Table 3, the average recognition rates for five emotional states are 90% 66%, and 56%, when using emotional voices, VR and IR facial expressions, respectively. In both cases of VR and IR facial expressions, the failure of recognition of emotion was mainly due to the difficulty to extract face-parts correctly because the subject changed her face-orientation. Overall results are shown in table 3(d) and the total recognition rates amount to 92% among five emotions.

5. Conclusion

This paper has described the new integration approach of recognizing human emotional states contained in voices and facial expressions. The emotional parameters were trained and recognized by HMM and NN for voices and images, respectively. The recognition results show that the integration method of recognizing emotional states in voices and images gives better performance than each isolated method in spite of the lower recognition rates compared to questionnaire counterpart.

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out-put	Ang.	100				
	Hap.		90			
	Neu.			100		
	Sad				90	30
	Sur.		10		10	70

(a) Recognition accuracy for emotional voices

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out-put	Ang.	50	20	10		10
	Hap.	10	70		20	10
	Neu.		10	90	20	
	Sad	10			40	
	Sur.	30			20	80

(b) Recognition accuracy for VR facial expressions

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out-put	Ang.	0				
	Hap.		90	20	10	20
	Neu.			80		50
	Sad	70	10		90	10
	Sur.	30				20

(c) Recognition accuracy for IR facial expressions

		Input emotion				
		Ang.	Hap.	Neu.	Sad	Sur.
Out-put	Ang.	100				
	Hap.		100		10	
	Neu.			100		
	Sad				80	20
	Sur.				10	80

(d) Total recognition accuracies using integration method

Table 3. Recognition accuracies for each emotional state

References

[1] Waibel, A (1986), Prosody and Speech Recognition. Doctoral Thesis, Carnegie Mellon Univ.

[2] C Tuerk (1990) A Text-to-Speech System based on {NET}talk. Master's Thesis, Cambridge University Engineering Dept.

[3] Yoshitomi Y, Miyawaki N, Tomita S, and Kimura S (1997), Facial Expression Recognition using Thermal Image Recognition and Neural Network. Proc. of 6th IEEE Int. Work. on Robot and Human Communication, pp.380-385

[4] Sugimoto Y, Yoshitomi Y, and Tomita S (2000), A method for Detecting Transitions of Emotional States using a Thermal Facial Image based on a Synthesis of Facial Expressions. Journal of Robotics and Autonomous Systems, Vol.31, pp.147-160.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Fractal Evaluations of Fish School Movements in Simulations and Real Observations

Tatsuro Shinchi*, Haruhiko Nishimura** , Tetsuro Kitazoe*** , Masayoshi Tabuse***, and Nobuyuki Azuma****

*Faculty of Education and Culture, Miyazaki Univ., 1-1 Gakuen-Kibanadai-Nishi, Miyazaki City, 889-2192 Japan
**Studies of Information Science, Hyogo Univ. of Education, 942-1 Yashiro-cho, Hyogo, 673-1494 Japan
***Faculty of Engineering, Miyazaki Univ.,1-1 Gakuen-Kibanadai-Nishi, Miyazaki City, 889-2192 Japan
****Faculty of Agriculture and Life Science, Hirosaki Univ., 3 Bunkyo-cho, Hirosaki City, 036-8591 Japan

Abstract

Fish schools behave like a single organism and these aggregate movements offer a considerable survival advantage. We simulate fish school movements swimming in a water tank with behavior models added rules of actions in wall neighborhood. And we compare the characteristics of simulated movements with real fish school movements. The results of comparison through fractal analyses are that movements can be quantified as characteristic fractal dimension according to its posture, such as guards against predators' attacks. And we show the validity of fractal analyses in the evaluation of fish school movements.

1 Introduction

We can watch many species of animals forming groups, such as flocks of pigeons, herds of cattle, and schools of fish. A fish school in union behaves like one body as if it has the intelligence for the entire movement[1]. However, there is not a consistent leader in many kinds of fishes. Some studies with computer simulations about the emergence of fish school movements have showed that continuations of mutual interactions among individuals develop the autonomous behavior of a fish school.

In the precedence study, we grasped the characteristics of the complex fish school movements by the fractal analyses with the time-series trail data[2]. And we quantified complexities of fish school movements, which are corresponding to the time coarsened levels as Fractal Dimension. Although the behavior model of a fish is based on the observation of real fish school movements, it is worth to examine the Fractal Dimensions in both of movements on a computer and real movements. In this paper, we quantify fish school movements seen in video-recorded pictures as Fractal Dimension, and we compare with the fractal dimension of simulated fish school movements with considering the boundary conditions of a square water tank.

2 Behavior Model and Boundary Conditions

2.1 Behavior Model of a Fish

The schooling of fish has been observed and investigated in various approaches to understand the rules of forming groups and maintaining groups. These researches revealed that each fish perceive surroundings by both of eyes and lateral lines. It is known that visual angle of fish eyes is often larger than 300° and lateral lines detect water currents, vibrations, and pressure changes. It is the fact of great interest that fish school movements emerge by only individual interactions with perceiving neighbors. In other words, fish school does not need the special fish to lead the entire movements.

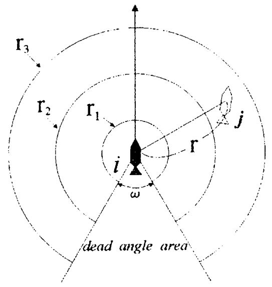


Fig. 1: Rages of the basic behavior patterns

Aoki suggested the behavior models for a fish, which emerge autonomous school movements. Fig.1 shows the geometrical drawing to illustrate parameters specifying interactions in Aoki's models. There are four styles of reactions, repulsion, parallel orientation, attraction and searching, according to different possible positions of neighbor's:[3]. A direction of the black fish(i) located in the center in Fig.1 will be determined by the distance r to the perceived white fish(j) as follows,

i) repulsion ($r < r_1$):

[If the neighbor fish(j) is too close, the fish tries to avoid a collision.],

ii) parallel orientation ($r_1 \leqq r < r_2$):
[If the neighbor fish(j) is in a certain parallel orientation area, the fish swims in the same direction as its neighbor.],

iii) attraction ($r_2 \leqq r < r_3$):
[If the neighbor fish(j) is too far away, the fish swims in the direction of its neighbor.],

iv) searching ($r > r_3$ or dead angle area):
[If the fish cannot perceive its neighbor, the fish begins to search for a neighbor fish with turning around by chance.].

In each action i)$\sim$iv) at time t, the fish(j) turns by the angle $\beta_i(t)$ as follows,

i) repulsion ($r < r_1$):
$\beta_i(t) = min\{\measuredangle(\overrightarrow{v_i(t)}, \overrightarrow{v_j(t)}) \pm 90°\}$,

ii) parallel orientation ($r_1 \leqq r < r_2$):
$\beta_i(t) = \measuredangle(\overrightarrow{v_i(t)}, \overrightarrow{v_j(t)})$,

iii) attraction ($r_2 \leqq r < r_3$):
$\beta_i(t) = \measuredangle(\overrightarrow{v_i(t)}, \overrightarrow{p_j(t) - p_i(t)})$,

iv) searching ($r > r_3$ or dead angle area):
$\beta_i(t) = 0$,

where $\measuredangle(\vec{a}, \vec{b})$ denotes an angle between two vector $\vec{a}$ and $\vec{b}$, $min\{a \pm b\}$ denotes a smaller angle($a + b$ or $a - b$), and p_j denotes a coordinate of the fish(j). The turn, $\beta_i(t)$, results that the new direction $\alpha(t + \Delta t)$ of fish(i) is

$$\alpha(t + \Delta t) = \alpha(t) + \gamma_{ij}\beta_i(t) + \sqrt{2}\beta_0, \qquad (1)$$

where the term of $\sqrt{2}\beta_0$ is a fluctuation in determining a new direction, β_0 follows a Gamma distribution N(0,1). γ_{ij} is the constant which adjusts the degree of turning angle. The neighbor fish(j) among N fishes which will be selected with the inverse proportion to the distance. r, and the nearer fish will be perceived as the fish(j). A new velocity of a fish, v_i, is determined independent of other fishes to simplify the models. The velocity v_i of every fish is computed by chance with Gamma distribution, which based on observation of fish behaviors.

$$\begin{aligned} v_i &= chance(p(v)) \\ p(v) &= \frac{A^K}{\Gamma(K)} \cdot exp(-Av) \cdot v^{K-1} \qquad (2) \end{aligned}$$

where v is velocity, K and A are constant parameters, $v \geq 0$, $K > 0$, $A > 0$, $\Gamma(K)$ is a Gamma function, and if K is an integer, $\Gamma(K) = (K - 1)!$.

We run this model based on Aoki's model which has the biological concept through the analysis of internal organization and communication process in fish schools.

2.2 Boundary Conditions Swimming in a Water Tank

Generally, it is so hard to grasp precisely all individual's behavior under the natural conditions, because animals move vary widely. As the next best, we evaluate the fish school movements swimming in a water tank. For comparisons between movements simulated on a computer and real fish school movements swimming in a water tank, actions of near walls [v), vi)] are added to actions i) $\sim$ iv). The direction of fish(i) are changed to avoid a collision against walls. The turning angle of β_i of fish(i) is determined by the distance to a wall r_w as follows,

v) avoid a collision against walls ($r_w < r_0$):
[If a perceived wall is too close, a fish swims along a wall to avoid a collision]
$\beta_i = min\{\measuredangle(\alpha_i(t), \theta_1), \measuredangle(\alpha_i(t), \theta_2)\}$,

vi) avoid a collision against corners ($r_{wa} < r_0$ and $r_{wb} < r_0$)
[If a perceived corner is too close, a fish turns to swim along one of two walls]
$\beta_i = min\{\measuredangle(\alpha_i(t), \theta_a), \measuredangle(\alpha_i(t), \theta_b)\}$,

where r_0 is the distance between walls to an inner rectangle shown as the dotted line in Fig.2, r_w is the distance between the fish(i) and wall, r_{wa} and r_{wb} denotes the distance to two walls respectively when fish(i) is in a corner, $\measuredangle(\alpha, \theta)$ denotes an angle between α and θ, min$\{a, b\}$ denotes a smaller angle(a or b). In selecting one of actions i) $\sim$ vi), actions against another fish i) $\sim$ iv) are given priority to actions against wall v), vi). And action vi) is given priority to action v).

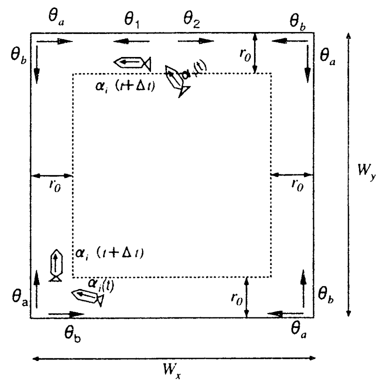


Fig. 2: Boundary Conditions in a Water Tank

3 Simulation of Fish School Movements in a Water Tank

3.1 Coarsened trail patterns

In this paper, we fixed the model with $r_1 = 0.5BL$, $r_2 = 2.0BL$, $r_3 = 5.0BL$, $\omega = 30°$(Fig.1), $\gamma_{ij} = 0.3$, and $r_0 = 0.5BL(= r_1)$. And only when action v) or vi) occurs, the new direction $\alpha(t + \Delta t)$ of fish (i) is determined with $\gamma_{ij} = 1.0$ and $\beta_0 = 0.0$. The rectangular water tank is 25BL long (W_x) and 32BL wide (W_y).

It assumed that each of individuals almost draws the same locus in the polarized schools, except for the period when it needs to the self-organization immediately after a simulation start. It can be said that the evaluation to the behavior of one tail results the analysis of a school movements. In this paper, we examine the trail pattern of a fish as the complicated fish school movements.

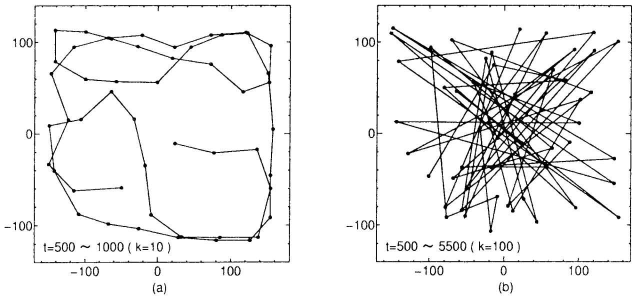


Fig. 3: The coarsened trail pattern of a fish school (N=100). (with 50 lines). (a):t=500 $\sim$ 1000 (k=10), (b):t=500 $\sim$ 5500 (k=100)

Let us gaze visually simulated fish school movements with some different time-intervals. In Fig.3, the fish school movements(N=100) are caught by links of time-series two-dimensional coordinates. In Fig.3(a), position coordinates are linked with every 10 simulation steps. We call Fig.3(a) the coarsened trail pattern with *the coarsened level of* k = 10. And Fig.3(b) can be said the coarsened trail pattern with *the coarsened level of* k = 100. On analyses the same pattern, in general, we see less number of lines in more coarsening. Therefore we need to obtain each trail pattern with same number of lines and in same area of a figure, so as not to be influenced by the number of lines in evaluating complicated movements. Then trail patterns in both of Fig.3(a) and Fig.3(b) consist of 50 lines. Fig.3 shows the complexities of a fish school movement, corresponding to the level of coarsening.

3.2 Fractal Analyses

We would like to focus on quantitative evaluations for the different complexities shown in Fig.3. Then, we apply fractal geometry in understanding of complex behavior. In this study, we obtain the fractal dimension which relates to the length of coarsened trail patterns $< L(k) >$ and the time interval k in measuring length $< L(k) >$. Then, henceforth, we call $< L(k) >$ the coarsened length. The $< L(k) >$ at the level of coarsening k is defined as follows:

$$\begin{aligned} < L(k) > \;&=\; \frac{1}{k}[\sum_{t=1}^{T-k} \sqrt{(x_{t+k}-x_t)^2 + (y_{t+k}-y_t)^2} \\ &\quad \cdot \frac{T}{(T-k)k}], \end{aligned} \tag{3}$$

where x_t and y_t are the elements of the position coordinate $P(t)$ at time t. The whole is multiplied by $1/k$, because $< L(k) >$ is found from k sets of time-series with each the different starting point in computing. T is the all time of the fish school simulation. And the term of $T/((T-k)k)$ normalize the difference of number for subsets. If then $< L(k) >$ is related to the coarsening time,k, as

$$< L(k) > \propto k^{-D}, \tag{4}$$

D is the exponent known as the fractal dimension. A school movement in a two-dimensional area gives D in range between 1.0 and 2.0.

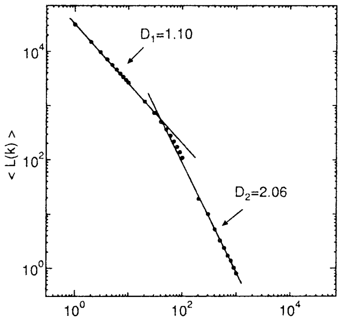
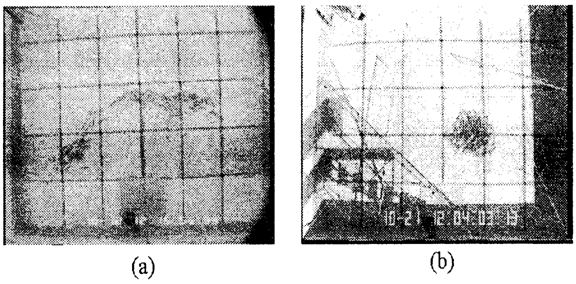


Fig. 4: The fractal analyses for a simulated fish school movement(N=100).

There are two fitted lines in Fig.4, one is fitted to the gradual inclination plot group ($1 \leqq k \leqq 40$) and another is fitted to the urgent inclination plot group ($40 \leqq k \leqq 1000$), calculated slopes of two line are $D_1 = 1.06$ and $D_2 = 2.06$ respectively. D_1 and also D_2 are fractal dimensions obtained as the characteristic of fish school movement.

4 Fish School Movements in Real Observation



(a) (b)

Fig. 5: The pictures of video-recorded fish school movements(N=100).(a):The ordinary fish school. (b):The fish school perceiving predators.

We analyze the video-recorded fish schools consists with about 100 *sardines* in a water tank. The average of *sardines'* body length is about 19.4[cm]. The water tank is 5.0[m] long, 6.0[m] width, and 0.75[m] in depth. Fig.5(a) shows the ordinary fish school. And Fig.5(b) shows the highly polarized movements perceiving the predators(*young yellowtails*).

4.1 Coarsened trail patterns

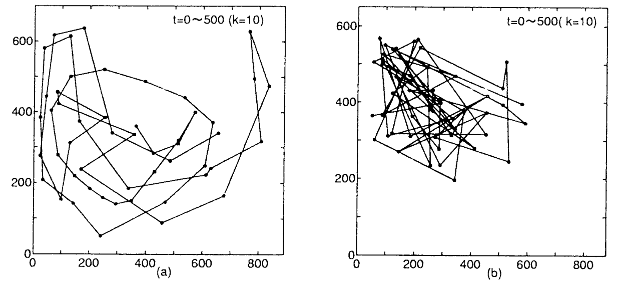


Fig. 6: The coarsened trail pattern of a real fish school(N ÷ 100).(**k=10**, with 50 lines). (a):The ordinary fish school. (b):The fish school perceiving predators.

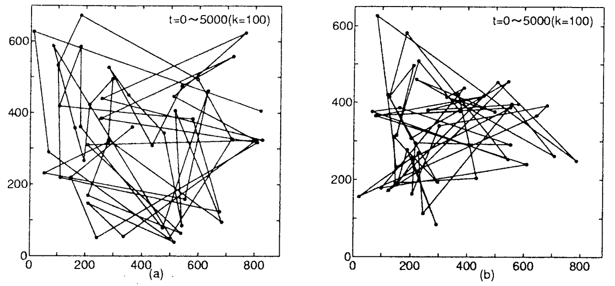


Fig. 7: The coarsened trail pattern of a real fish school(N ÷ 100).(**k=100**, with 50 lines). (a):The ordinary fish school. (b):The fish school perceiving predators.

We examine the video-recorded fish school movements with coarsening and fractal analyses as to the simulated fish school movements. In analyses of real fish school movements, we grasped the fish school movements by tracing a fish with operating a mouse. And we obtained the time-series position coordinates every 0.5[s]. Namely, $t = 0 \sim 500$ in Fig.6 means the duration of 250[s] with 500 sampling steps. The depicted patterns with 50 lines (in Fig.6 and Fig.7) definitely show that the range of fish school movements in a tank is influenced by a state of school. In a state of perceiving predators, a fish school moves in a limited small area in comparison with movements of the ordinary fish school. Fig.6 and Fig.7 also show that the direction of fish school is changed largely and frequently in a state of perceiving predators.

4.2 Fractal analyses

The result of the fractal analyses to the real fish school shown in Fig5 $\sim$ Fig.7 is presented as the Fig.8. $D_2(\div 2.0)$ obtained with steep slopes in both of Fig.8(a) and Fig.8(b) may account for the complicated behavior by the repeated direction changes against walls. These complexities are also found in fractal analyses of simulated movements(Fig.4). On the other hand, D_1 obtained with gradual slopes show the different complexities grasped in the coarsening of trail patterns. Namely, the complicated movements caused by large and frequent direction change are quantified as the large Fractal Dimension $D_2 = 1.17$ in compared with $D_2 = 1.10$ of the movement at the ordinary fish school. It is possible to interpret that Fractal Dimension can grasp the degree of tension at some special movements.

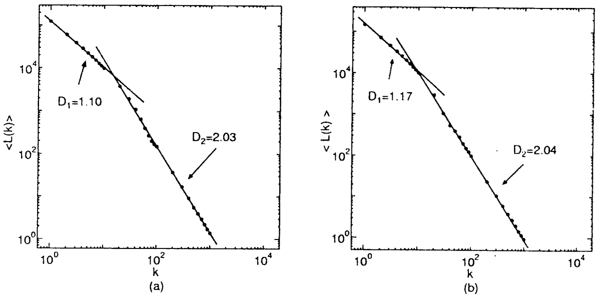


Fig. 8: The fractal analyses for a real fish school(N÷100).(a):The ordinary fish school. (b):The fish school perceiving predators.

5 Summary

As for the result of fractal analyses about the ordinary fish school movements in both of simulations and real observation, Fractal Dimension D_1 obtained with small value of k are almost in of one accord as $D_1 \div 1.10$. It can be said that fish school movements swimming sufficiently inside of a tank on a computer are similar in complexities to the real fish school movements. However, it is supposed that D_2 obtained with large value of k show is influenced by fish movements in the boundary area.

Moreover, the fish school in case of some special postures will have different rules of actions against wall. In this paper, fish school keeps long distance to walls for the guard against the attack by predators. These drive us that we need to investigate the boundary conditions in more detail and to improve simulation models. And we expect that the reliability of simulations is evaluated with Fractal Analyses.

We wish to thank Dr. I.Aoki for his valuable suggestion about the observation of real fish school movements.

References

[1] Braian L.Partridge:The Structure and Function of Fish Schools, Scientific American,vol246, pp.90-99(1982)

[2] T.Shinchi,H.Nishimura,T.Kitazoe and, M.Tabuse: Quantitative Analyses and Fractal Structures of Fish School Movements;Proc. of AROB5th, pp153 ~156(2000)

[3] Ichiro Aoki:A Simulation Study on the Schooling Mechanism in Fish, Bulletin of the Japanese Society of Scientific Fisheries,48(8), pp.1081-1088(1982)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Application of Visual Recognition Neural Network to Hearing System for Continuous Speech

Makoto Funamori ,Tetsuro Kitazoe, Tomoyuki Ichiki
Department of Computer Science and Systems Engineering
Faculty of Engineering, Miyazaki University
1-1, Gakuen Kibanadai Nishi, Miyazaki, 889-2192 Japan
e-mail: kitazoe@cs.miyazaki-u.ac.jp

ABSTRACT

The two or three layered networks 2LNN, 3LNN which originate from stereovision neural network are applied to speech recognition. To accommodate sequential data flow, we consider a window to which new acoustic data enter and from which final neural activities are output. Inside the window recurrent neural network develops neural activity toward a stable point. The process is called Winner-Take-All(WTA) with cooperation and competition. The resulting neural activities clearly showed recognition of a continuous speech of a word. The string of phonemes obtained is compared with reference words by using dynamical programming method. The resulting recognition rate amounts to 96.7% for 100 words spoken by 9 male speakers, which is compared to 97.9% by hidden markov model (HMM) with three states and single gaussian distribution. The present results which are close to those of HMM seem noticeable because the architecture of the neural network is very simple and parameters in the neural net equations are small numbered and always fixed.

1. INTRODUCTION

Since we recognize speech through neural network in the brain, many works on this line have been conducted for speech recognition. Though probabilistic acoustic models represented by Hidden Markov Model (HMM) has been widely used recognizers, it has been long standing go to let machine enable human abilities of speech recognition in the brain. Various kind of neural networks have been proposed for speech recognition such as multilayer perceptrons (MLP)[1,2], time delayed neural network (TDNN)[3], hidden control neural network (HCMM)[4], hybrid system combining HMM and MLP (HMM/MLP)[1] and fully recurrent neural network (FRNN)[5,6], notable things are that these models use more or less learning algorithms of back propagation of error and need many parameters to be adjusted.

In the previous works, we employed a new approach to the problem by applying stereo vision neural network to hearing system[7,8,9,10,11]. The neural networks are two or three layered (2LNN, 3LNN) and the parameters in the equations are fixed and not changed at any time. The learning processes are considered as that the feature parameters characteristic of each phoneme are stored or memorized in the brain in the form of probability density functions. We consider recognition processes as that the neural network equations employed from visual system process the similarities between the characteristic phonetic features stored in our memory and the input acoustic data from our ears, eventually giving stable neural activities. The resulting phoneme recognition rate was fairly good, resulting 7-9% higher than HMM model. In the present paper we are going to give an algorithms for the continuous speech recognition. The major problem in this case are how to introduce an algorithms to the real time acoustic data flows and how to employ the neural network to process the data flows, giving the recognition of continuous speech.

2. APPLICATION OF NEURAL NETWORK TO SPEECH RECOGNITION

The speech(phoneme) recognition system using stereo vision neural net equations is divided into four main processes;

(1) A number of training speech data are classified and parameterized into sequences of feature vectors for each phonemes. The feature vectors are used to form standard Gaussian PDFs which are supposed to be memorized in our brain for each phoneme.

(2) An input phonemes are referred to these memorized phoneme data and a similarity measure is obtained by comparing the input phoneme data with the memorized PDF of each phoneme.

(3) Suppose that there is a neuron activity ξ_u^a according to the similarity measure λ_a^u to a certain phoneme /a/ at the frame number u.

(4) The stereo vision neural net equations are performed to make an activity ξ_a^u move toward a stable point after the equations receive the similarity measure as an input and a recognition results are achieved when it reaches to a stable state.

The memorized standard acoustic models for each phonemes are expressed in terms of Gaussian PDF for input o .

$$\mathrm{N}(o;\mu_a,\Sigma_a)=\frac{1}{\sqrt{(2\pi)^n|\Sigma_a|}}e^{-\frac{1}{2}(o-\mu_a)^t\Sigma^{-1}(o-\mu_a)} \qquad (1)$$

where μ_a is a mean value and Σ_a is covariance matrix, of feature vectors for training data of a phoneme /a/. The normalized similarity λ_a^u of input data o_u at u-th frame to a certain phoneme /a/ is defined as

$$\lambda_u^a = \frac{\log N(o_u;\mu_a,\Sigma_a)-<\log N>}{<\log N>} \qquad (2)$$

where <logN> means an average over phonemes at the same frame.

3. TWO LAYERED NEURAL NET EQUATIONS

Since 3LNN has a similar property with 2LNN, we discuss about 2LNN which is given as

$$\tau_1\dot{\xi}_u^a(t) = -\xi_u^a(t) + f(\alpha_u^a) \qquad (3)$$

$$\tau_2\,\dot{\alpha}_u^a = -\alpha_u^a + A\lambda_u^a - B\sum_{a'\neq a} g(\xi_u^{a'}(t)) + D\sum_{u'=n-l1}^{u+l2} g(\xi_{u'}^a(t)) \qquad (4)$$

f(x) is a well known as sigmoid function and g(u) is a function given by

$$f(x) = (\tanh(w(x-h))+1)/2 \qquad (5)$$

$$g(u) = u^+ = (u+|u|)/2 \qquad (6)$$

where A,B,D,w,h are positive constants which are to be chosen appropriately .

Figure 1 shows three layered structure of the stereo vision neural network.

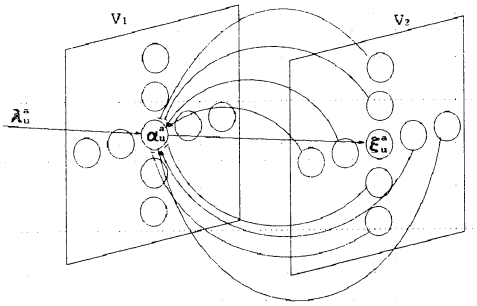


FIGURE 1. Three layered neural network(2LNN)

To understand the qualitative feature of the equations consider an equilibrium for $\alpha(\dot{\alpha}=0)$. We obtain

$$\tau_1\dot{\xi}_u^a(t) = -\xi_u^a + f(\alpha_u^a) \qquad (7)$$

$$\alpha_u^a = A\lambda_u^a - B\sum_{a'\neq a} g(\xi_u^{a'}(t)) + D\sum_{u'=u-l}^{u+l} g(\xi_{u'}^a(t)) \qquad (8)$$

Notice that the equations (7),(8) give the same solution as (3),(4) if the stable solution is unique. And simulations show this is the case. Equations (7) and (8) are understood as that the similarity measures λ_u^a are inputted into the α layer and the outputted $\alpha's$ are fed to the ξ layer. In the ξ layer, it is noticed that large (small) α gives large (small) ξ due to the sigmoid function. The new $\xi's$ thus obtained are again brought back into the α layer and the same procedure is repeated. In the α layer Winner-Take-All processes take place with competition (the second term of (8)) and cooperation (the third term of (8)). The typical dynamics is shown in figure 2 where cooperation works in the neighboring frames for the same phoneme, while competition does against other phonemes at the same frame. The Winner-Take-All processes accelerate the neuron activities toward stable points where we will get a speech recognition.

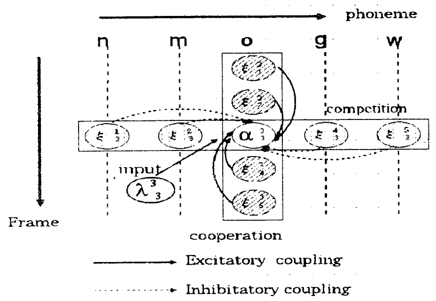


FIGURE 2. Dynamical process of neural activities with cooperation and competition

4. CONTINUOUS SPEECH RECOGNITION

For continuous speech, it is considered that input data λ_u^a are fed to the neural network (3), (4) and the activities α , ξ in the network develop their value toward a stable point. We judge as that a phoneme /a/ is recognized at a flame u if ξ_u^a goes close to 1, while it is not recognized if ξ_u^a goes close to 0. In the present study, we take equations (7),(8) to save calculation time. Differential equations (7),(8) are described as a loop of N steps in numerical calculation where time is divided by discrete span Δt . Thus, ξ develops as

$\xi(t)$, $\xi(t+\Delta t),\cdots$ to $\xi(t+N\Delta t)$.

To treat data flow λ_u^a sequentially, we consider that the data enter into a window with L frames and the neural network (7), (8) processes the L frame data for N steps. Then, the data are passed one frame forward through the window, where the initial values of ξ are set to zero before entering to the window. The procedure is stated more in detail as follows; When input data λ_{u+L-1}^a , λ_{u+L-2}^a ,..., λ_u^a are entered inside the window, equations (7), (8) develop activities $\xi(t)$ until it arrives at $\xi_{u'}^a(t+N\Delta t)$ for $u'=u,u+1,\ldots,u+L-1$. Then, new data λ^{a}_{u-L} enter into the window from the left and the old data λ^{a}_{u} get out from the window. At the same time whole $\xi^{a}_{u'}$ are replaced by $\xi^{a}_{u'-1}$, setting the initial values of $\xi^{a}_{u'-1}$ for the next N step loop calculations. At this time we have new comers ξ^{a}_{u+L} entering into the window and the final ones ξ^{a}_{u} outputted from the window. We eventually judge if phoneme /a/ at u-th frame is recognized or not according to whether the final ξ outputted from the window is close to 1 or 0, respectively. A sequence of the same processes continues until whole input data go through the window completely and whole values of final ξ are obtained.

5. EXPERIMENTAL RESULTS

We extracted a total 24 labeled phonemes from ATR Japanese speech database composed of 4000 words spoken by 10 male speakers and from ASJ speech database of 500 sentences by 6 male speakers to make Gaussian PDFs for each phoneme. For recognition test which is independent, we take 100 words of the training data. The experimental conditions are as follows

Sampling rate	16kHz, 16bit
Pre-emphasis	0.97
Window function	16ms Hamming window
Frame period	5ms
Feature parameters	10-order MFCC +10-order Δ MFCC

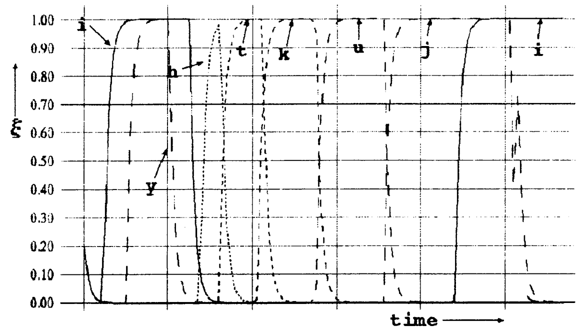


FIGURE 3. Best two of ξ are selected and plotted against frame (time). Here parameters are set as window size L=10, Δt=0.01, N=100, A=3.0, B=3.0, D=0.5, w=2.0, h=1.0, 11=6, 12=0, τ_1=1

In figure 3 the typical result for a word pronounced /i/ /k/ /u/ /j/ /i/ is shown for the best two ξ^{a}_{u} outputted from the window. In the figure the best ξ are read sequentially as /i/ /h/ /t/ /k/ /u/ /j/ /i/. It is noticed that /y/ has rather high values following to /i/, because /y/ resembles to /i/. /h/ and /t/ do not have correspondence in the reference word and may be recognized as context dependent effects between /y/ and /k.

NG	a	b	ch	d	e	f	g	h	i	j	k	m	n	o	p	r	s	sh	t	ts	u	w	y	z	_
1	.0	.2	.0	.1	.0	.0	.4	.0	.0	.2	.0	.8	.9	.0	.0	.2	.0	.0	.0	.0	.5	.0	.0	.3	.0
	1	.0	.0	.2	.2	.0	.0	.9	.0	.0	.2	.0	.0	.1	.0	.3	.0	.0	.0	.0	.1	.8	.0	.0	.0
		1	.0	.4	.0	.0	.5	.0	.0	.1	.0	.3	.3	.0	.0	.4	.1	.2	.0	.0	.3	.3	.0	.1	.0
			1	.0	.0	.0	.0	.2	.1	.0	.4	.0	.0	.0	.1	.0	.2	.4	.2	.2	.0	.0	.0	.0	.0
				1	.0	.0	.4	.0	.0	.1	.0	.1	.1	.0	.0	.2	.0	.0	.1	.1	.0	.0	.0	.3	.0
					1	.0	.0	.1	.1	.0	.0	.0	.0	.0	.0	.3	.0	.0	.0	.0	.2	.0	.5	.0	.0
						1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
							1	.0	.0	.3	.0	.3	.3	.0	.0	.4	.0	.0	.0	.0	.2	.0	.0	.1	.0
								1	.0	.0	.6	.0	.0	.4	.6	.0	.0	.0	.6	.2	.0	.3	.0	.2	.0
									1	.6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.8	.0	.0
										1	.0	.0	.0	.0	.0	.2	.0	.0	.0	.0	.0	.0	.3	.0	.0
											1	.0	.0	.0	.6	.0	.0	.0	.6	.2	.0	.0	.0	.2	.0
												1	.9	.0	.0	.5	.0	.0	.0	.0	.1	.0	.0	.0	.0
													1	.0	.0	.5	.0	.0	.0	.0	.4	.0	.0	.0	.0
														1	.0	.0	.0	.0	.0	.0	.4	.5	.0	.0	.0
															1	.1	.0	.0	.6	.2	.0	.0	.0	.0	.0
																1	.0	.0	.0	.0	.3	.0	.0	.0	.0
																	1	.3	.0	.7	.0	.0	.0	.0	.0
																		1	.2	.3	.0	.0	.0	.0	.0
																			1	.7	.0	.0	.0	.0	.0
																				1	.0	.0	.0	.0	.0
																					1	.5	.2	.4	.0
																						1	.0	.0	.0
																							1	.0	.0
																								1	.0

TABLE 1. Likelihood table among 24 phonemes for the dynamical programming method. /-/ shows missing phonemes in inputted word against a reference word.

```
input word:_aaaaaaaaaaaaaaaaaaaayyiiiiiiiiiiiiiiiiiiiiiiii
           jjjjjjjjjjjjjjjjjCjjjjjjjjjjjjooooWoooWWoooooooooo
           oooooooooooooooooooooooooooooooooooooooooohoh
```

```
 reference words

  1 aaaaaaaaaaaaaaaaaaaaaiiiiiiiiiiiiiiiiiiiiiiiiiiiii
    jjjjjjjjjjjjjjjjjjjjjjjjjjjjjooooooooooooooooooooo
    ooooooooooooooooooooooooooooooooooooooooooo   185.20

 40 kkkkkkkkaaaaaaaaaaaaaaaiiiiiiiiiiiiiiiiiiSSSSSSSSS
    SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSooooooooooooooooooo
    oooooooooooooooooooooooooooooooooooooooooooooo   93.30

 57 ggggggggggggeeeeeeeeeeeeeeeeeeeeeeeeeeekkkkkkkkkkk
    kkkiiiiiiiiiiiiiiiiiijjjjjjjjjjjjjjjjjjjjjjooooooo
    oooooooooooooooooooooooooooooooooooooooooooooooooo
    oooooo   93.00

 31 oooooooooooooooooooooooookkkkkkkkkkkkkkkkuuuuuuuuu
    uuuuuuuuuuuuuuujjjjjjjjjjjjjjjjjjjjjjjjjjjoooooooo
    oooooooooooooooooooooooooooooooooooooooooooooooooo
    oooooooo   84.50

  6 aaaaaaaaaaaaaaaaaaaaSSSSSSSSSSSSSSSSSSSSSSSSSiiiii
    iiiiiiimmmmmmmmmmmmmooooooooooooooooooooooootttttt
    ttttttttttoooooooooooooooooooooooooooooooooo   73.30
```

FIGURE 4. An example of dynamical programming where a string of phonemes recognized by the neural networks is compared with reference words and the best five words are selected. The best scored /a/ /i/ /j/ /o/ is correctly recognized in this case.

To get the recognition rate, a sequence of phonemes for a word obtained from the window are compared with a list of reference words. We make the list of reference words composed of phonemes with the mean length which is estimated from the data of the same word spoken by many different people. We take dynamical programming (DP) method to match an inputted words with the reference word, where a likelihood table between different phonemes is estimated from phoneme recognition results given by neural networks reported previously [8-11](TABLE1). An example for DP is given in FIGURE 4, where higher scores for reference words show better likelihood for inputted data.

	2LNN	HMM
mau	98	98
mht	100	98
mms	93	98
mmy	94	97
mnm	95	97
msh	94	98
mtk	99	98
mtm	97	99
mtt	100	98
average	96.7	97.9

TABLE 2. The recognition results for 100 words spoken by 9 male persons

Experiment was performed in this way and 100 words uttered by 9 male speakers were recognized with the rate of 96.7%, where HMM model with 3 states and single gaussian distribution gave 97.9% for the same sample data.

6. CONCLUSIONS AND DISCUSSIONS

The two or three layered networks 2LNN, 3LNN which originate from stereovision neural network are applied to speech recognition. To accommodate sequential data flow, we consider a window to which new outputted acoustic data enter and from which final neural activities are outputted . Inside the window recurrent neural network develops neural activity toward a stable point. The process is called Winner-Take-All (WTA) with cooperation and competition. The resulting neural activities clearly showed recognition of a continuous speech of a word. The string of phonemes obtained is compared with reference words by using DP matching. The recognition results are 96.7%.compared with 97.9% by HMM. A simple step forward will be to use multi-gaussian distribution to obtain more accurate similarity measures and further improvement of recognition is expected together with the study of more input data. The nice feature of our model is that it does not have many parameters to be adjusted and the algorithm for recognition is simple.

Reference

[1] Bourlard, C.J.Wellekens. "Link between Markov Models and Multi-layer Perceptoron" IEEE Trans. Patt. Anal. Machine Intell., Vol.12, pp.1167-1178, 1990

[2] Hung, A.Kuh. "A Combined Self-Organizing Feature Map and Multilayer Perceptron for Isolated Word Recognition" IEEE Trans. on Signal Processing, Vol.40,pp.2651-2657, 1992

[3] J.Lang, A.Waibel, G.E.Hinton. "A Time-Delay Neural Network Architecture for Isolated Word Recognition" Artificial Neural Networks, Paradigms, Applications and Hardware, 1992

[4] Martinelli. "Hidden Control Neural Network" IEEE Trans. on Circuits and Systems, Analog and Signal Processing 41(3):245-247,1994

[5] T.Robinson(1992) Recurrent Nets for Phone Probability Estimation. Proceedings of the ARPA Continuous Speech Recognition Workshop, Stanford, Sept.

[6] Williams,R.J.,Zipser.D.(1990)Gradient based Learning Algorithms for Recurrent Connectionist Networks. Tech. Rep. NU=CCS-90-9,Northeastern University, College of Computer Science, Boston

[7] T.Kitazoe, J.Tomiyama, Y.Yoshitomi, and T.Shii "Sequential Stereoscopic Vision and Hysteresis" Proc. of Fifth Int.Conf on Neural Information Processing, pp. 391-396, 1998

[8] T.Kitazoe,S-I.Kim,T.Ichiki. Speech recognition using Stereovision Neural Network Model. Fourth International Symposium on Artificial Life and Robotics, pp.576-579,Vol2, January, Beppu, Oita, Japan, 1999.

[9] Acoustic Speech Recognition Model by Neural Net Equation with Competition and Cooperation(Tetsuro Kitazoe,Tomoyuki Ichiki,Sung-Ill-Kim) ICSLP'98(The 5th International Conference on spoken Language Processing,Vol 7.pp3281-3284,30th November-4th December, Sydney,Australia,1998

[10]T.Kitazoe,S-I.Kim,T.ichiki,M.Funamori.Acoustic Models in Speech Recognition by Stereo Vision Neural Nets. International Conference on Speech Processing,pp81-86,Vol1,August,Seoul,Korea,1999

[11] T.Kitazoe, S-I.Kim, T.Ichiki, M.Funamori. Acoustic Speech Recognition by Two and Three Layered Neural Networks with Competition and Cooperation. International Workshop SPEECH AND COMPUTER, pp.111-114, October, Moscow, Russia, 1999.

Evolutionary Robot Controllers with Competitive and Cooperative Neural Network

Akinobu Todaka, Tetsuro Kitazoe, Masayoshi Tabuse, and Kei Sugihara
Department of Computer Science and Systems Engineering
Faculty of Engineering, Miyazaki University
1-1, Gakuen Kibanadai Nishi, Miyazaki, 889-2192 Japan
{gaa, tabuse,ks,kitazoe}@cs.miyazaki-u.ac.jp

Abstract

This paper describes a new approach to control systems for an autonomous mobile robot by using sandwiches of two different kinds of neural networks. One is a neural network with competition and cooperation and used for recognizing sensor information, where synaptic couplings are fixed. The second is a neural network with adaptive synaptic couplings corresponding genotype in the creature and used for self learning of wheel controls. In the computer simulative model, we are successful to obtain four types of robot with good performance when going along a wall. The model also showed robustness in real environment.

1.Introduction

Many attempts have been focused on developing autonomous robots inspired by animals and humans which have robust adaptation and stable behavior in changing environments. One approach on this line is to use neural networks between input from senores and output to controllers and to adapt synaptic couplings in the networks to the environment. Many authors have proposed evolutionary robot control systems by using evolutionary adaptation of neural network [1,2,3], genetic programming [4] and classifier system [5]. In general, evolutionary processes require the large population size and a number of generations. Thus, experiments for evolutionary robotics are usually carried out in the computer simulations which were helpful to train and test robot control systems. However, we cannot simulate the real world with a proper treatment of noise completely, so that evolved robots in the computer simulations are hard to work successfully in the real world.

Certainly, however, there are some parts in the brain which do not include learning procedure, just inherited from parent when born. Some stages of image processing do not need learning procedures. Stereovision neural networks, for example, do not have learning procedure since we have automatic focusing ability without training and their synaptic couplings are considered always fixed. The strategy of the present paper is, therefore, that the neural networks are divided into two parts. One is sensor recognition processor in which sensor input data are processed with cooperation and competition, reducing noise from environment and giving definite decision for sensor data. The network parameters in this part are always fixed. Another is processor with selftraining which processes the data from sensor recognition part and outputs motor control. The synaptic couplings are trained to adapt to the environment by means of , for instance, genetic algorithms.

For sensor recognition procedure we use the neural network with competition and cooperation as a processing unit of robot sensors. The idea of neural network with competition and cooperation originated initially from a stereovision pattern recognition. The famous neural network model for stereovision was studied by Amari and Arbib[6] which is called a primitive competitive model. Reinman and Haken [7] proposed neural network with cooperation and competition. Kitazoe et al [8] presented neural networks which were able to give stereo vision recognition for moving objects. In the neural network with competition and cooperation, competition makes only one neuron active and cooperation maintains the active state. Thus, the robot recognizes the nearest object in surrounding environments and keeps up this recognition under small fluctuations of sensor values and behaves correctly in dynamically changing environments

2. Khepera

In our experiments we use a miniature mobile robot Khepera[9]. The Khepera body is 32 mm height and 55 mm in diameter. It has two wheels, each of which is controlled by a DC motor, and can rotate in both directions.

The eight infra-red proximity sensors are installed around the Khepera body (six in front and two in rear, see Figure 1). These sensors allow two measures: the light reflected by obstacles and the ambient light, so that it detects an obstacle and a light source. The output values of each sensor are from 0 to 1023 in integer. For the measurement of the light reflection, low value means that there are no obstacles near the sensor, while high value means that an obstacle is close to the sensor. The maximum detection range of the sensor is about 3 cm. The Khepera communicates with a computer using a serial line, so that the computer obtains the sensor values from the Khepera and provides the wheel speed to the Khepera.

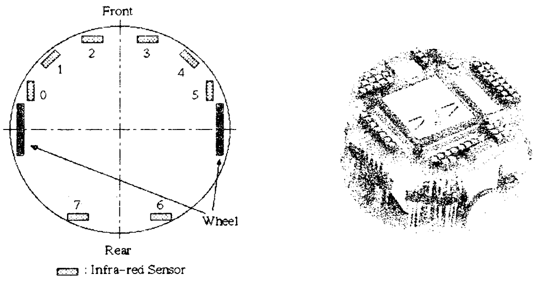


Figure 1: Mobile Robot Khepera

3. Three Layered Neural Network with Competition and Cooperation

We explain the three layered neural network[8], which controls the Khepera. The three layered neural network equations are given as

$$\frac{d}{dt}\alpha_u^a(t)=-\alpha_u^a(t)+A\lambda_u^a(t)-B\sum_{a'\neq a}g(\xi_u^{a'}(t))+D\sum_{u'=0}^{l}g(\xi_{u'}^a(t)) \quad (1)$$

$$\frac{d}{dt}\beta_u^a(t)=-\beta_u^a(t)+g(\alpha_u^a(t))+g(\xi_u^a(t)) \quad (2)$$

$$\frac{d}{dt}\xi_u^a(t)=-\xi_u^a(t)+f(\beta_u^a(t)) \quad (3)$$

where ξ_a^u *(t)* is a neuron activity, $\alpha_u^a(t)$ is a first layer activity and $\beta_u^a(t)$ is a middle layer one. λ is an input sensor value to this neural network. *f(x)* is a well known sigmoid function and *g(u)* is a function given by

$$f(x)=(\tanh(w(x-h))+1)/2 \quad (4)$$

$$g(u)=u^+=(u+|u|)/2 \quad (5)$$

where *A, B, D, w, h* are positive constants which are to be chosen appropriately. The neural network for these equations has a three layered structure as shown in Figure 2.

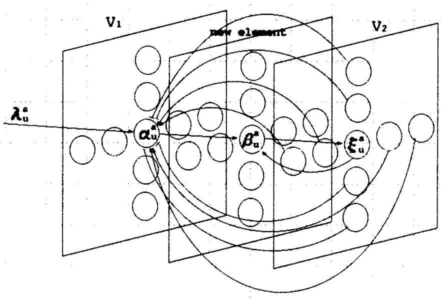


Figure 2: Three layered neural network

In equation (1), the second, third and forth terms are referred to the input, competitive and cooperative terms, respectively. The second term describes an input data, the third term represents a competition with other neuron activities $\xi_u^{a'}(a'\neq a)$ and the forth term measures a cooperation with other neuron activities $\xi_{u'}^a$.

The qualitative feature of this neural network is as follows.

1) When all λ's have small values (about zero or less), all activities ξ are not active, i.e. their values are almost zero.
2) When one λ has a large value (between zero and one), it becomes active with value almost close to one.
3) When two or more λ's have large values, ξ of the largest λ is active and others are not active be due to the competition term.
4) When two or more λ's have very large values (about one or more), ξ of these λ's are all active.
5) Even if the value of λ for the active ξ becomes somewhat smaller, this ξ keeps up the active state because of the cooperative term.
6) According to the values of λ's, each ξ gets active or not.

Let us consider the control of Khepera by using this feature. In our experiments we suppose that the neuron activities related to Khepera's sensors compete with each other and the time sequence of the neuron activity for each sensor cooperates with each other. Thus *a* and *u* indicate a sensor number (*a*=0,1,2,3,4,5,6,7) and the *u*-th past time (*u*=0,1,...,l) at a discrete time step, respectively. The input value λ is defines as

$$\lambda_u^a=2\frac{\text{sensor value of }\#a}{1024}-1 \quad (6)$$

for a light reflection. We calculate the neural network equations for *a*=0, 1, 2, 3, 4, 5, *u*=0 and keep up the *u*-th past activities (*u*=0,1,2,...,l). Fig.3 shows sensor values for each detector and Fig4 shows the neuron activities after they are processed by the neural networks eqs.(1),(2),(3). A comparison of two Fig's show how the neural network gives a decisive result for noisy low sensor data.

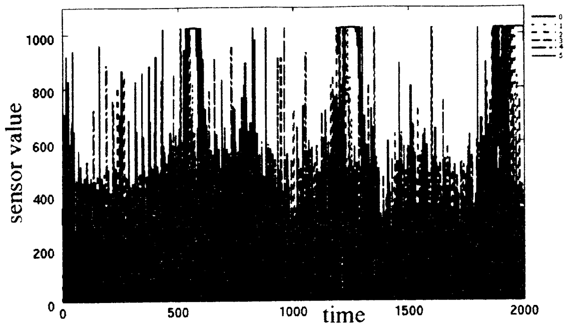


Figure 3: Sensor values of Khepera

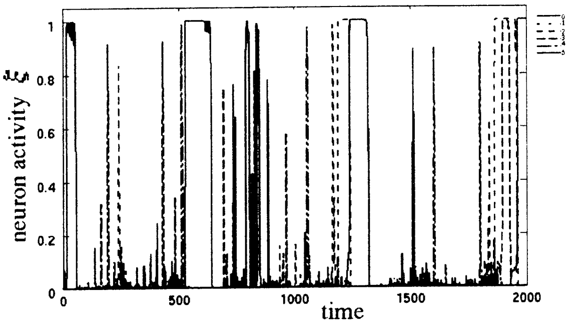


Figure 4: Neuron activity ξ where A=2.5, B=1.0, D=1.0, w=2.0, h=1.0, l=5, in eqs. (1),(2),(3).

4. Evolutional Adaptation

To save time, we perform adaptation under computer simulative model of a robot and its environment The data x_0 , x_1 , x_2 ,..., x_5 outputted from the neural networks (1),(2),(3) are fed to the second neural networks with self training abilities. The synaptic couplings are revised by genetic algorithm. The control signals to right and left wheels, R, L are given by

$$R = F\left(W_{R0} + \sum W_{Ri} x_i\right) \qquad (7)$$

$$L = F\left(W_{L0} + \sum W_{Li} x_i\right) \qquad (8)$$

where F(x) is the sigmoid function given by (4) with h=0 and w=0. Synaptic couplings, W_{Ri} and W_{Li} are to take discrete values of –1.0, -0.5, 0, 0.5, 1.0 which correspond to genotype in genetic algorithms. The evolutional algorithms are as follows.

(1) We make N1 robots with randomly generated synaptic couplings and let them run inside a box with some obstacles for a certain time period. .

(2) Make new N2 robots from old N1 robots by using genetic algorithms of mutation and crossover with a certain rate. Then, let them run for the same period as (1).

(3) Measure the whole robots N1+N2 by a given evaluation function and choose best N1 robots with high scores. If total score of N1 robots exceed a given threshold, stop the loop, otherwise goto (2).

4. Experiments

(1) Movement along a wall

We investigate the movement along a wall task in order to estimate capability of the control of Khepera by the combined use of competitive and cooperative neural network and genetic algorithms, For anti clock wise movement along a wall, #5 sensor has a crucial role: if x_5 is active, it is near a wall so that it should go along the wall, else it must go toward the wall. Here, let us take simplest choice for the variables:

$$X_1 = x_0 + x_1 + x_2 + x_3 + x_4 \qquad (9)$$

$$X_2 = x_5 \qquad (10)$$

The evaluation function is given as

$$\begin{aligned} g = &\left(\left|V_R + V_L\right|/2\right)\cdot\left(1 - \left|\left(V_R + 1\right) - V_L\right|/V\max\right)\cdot \\ &\left(1 - \textstyle\sum_{i=0}^{4} x_i/5\right)\cdot\left(\left(V_R + V_L\right)/4V\max + 1/2\right)\cdot \\ &\left(\left(sensor(5)/1023 + 1\right)/2\right) \end{aligned} \qquad (11)$$

where sensor(5) shows row sensor value at #5 and take 0 to 1023. g has high value if a robot goes along walls without avoiding obstacles and if it runs forward as fast as possible. We set N1 = N2 = 20 and take crossover among those synaptic couplings with better scores and set mutation rate to 30%. The evaluation function is calculated for 1000 steps of a robot run. Fig.5 shows gradually growing curves of evaluation function for 50 generations. Coupling values of best 8 robots are shown in table 1 together with fitness value which is a sum of g for 2000 steps. It is interesting to see these robots are classified into four types as shown in table 2. Type 1 robot tends to go straight and sometimes leave from the wall. Type 2 and type 4 have similar behavior with small frequent oscillation when going along the wall. Under the circumstance there is no obstacle around, however, type 2 goes straight, while type 4 goes forward with a little bit right curved. Type3 goes along a wall with slow oscillation.

	W_{R1}	W_{R2}	W_{R0}	W_{L1}	W_{L2}	W_{L0}	FITNESS
1	0.5	-0.5	1.0	-1.0	0.0	1.0	18610.5
2	1.0	0.0	0.5	-1.0	0.0	1.0	18408.3
3	0.5	0.0	0.5	-1.0	0.0	1.0	18368.0
4	0.0	0.5	0.5	-1.0	0.0	1.0	16785.4
5	1.0	0.5	0.5	-1.0	0.0	1.0	16661.3
6	0.5	1.0	0.5	-1.0	0.0	1.0	16653.8
7	0.5	0.5	0.5	-1.0	0.0	1.0	16513.0
8	0.0	1.0	0.5	-1.0	0.0	1.0	16481.9

Table 1: Coupling values of best 8 robots after 50 generation

X2 \ X1	0	1
0	R(Big)	L(Small)
1	Straight	L(Small)

Type 1 (4,5,7)

X2 \ X1	0	1
0	Straight	L(Small)
1	R(Big)	L(Small)

Type 2 (1)

X2 \ X1	0	1
0	R(Big)	L(Small)
1	L(Big)	L(Small)

Type 3 (6,8)

X2 \ X1	0	1
0	R(Big)	L(Small)
1	R(Big)	L(Small)

Type 4 (2,3)

Table2: Four different types of robot movement going along a wall under X_1 or X_2 active(1) or not(0). R(L)(Big(Small)) shows a robot goes right(left) with big(small) curve.

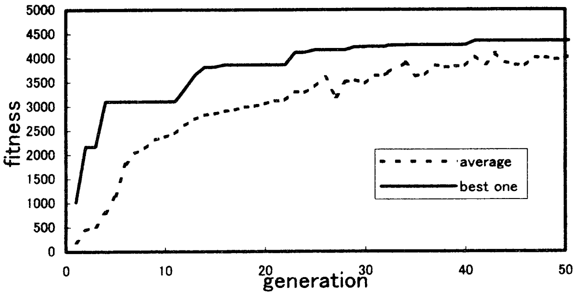
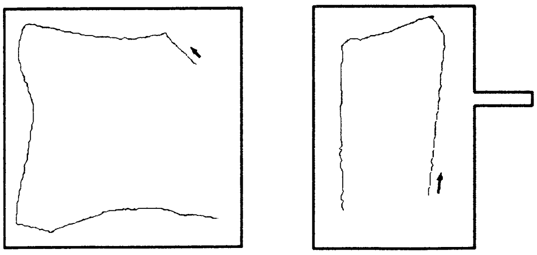


Figure 5 : Fitness value of robot for the best one and for average of N_1 robots.

(2) Movement in the real environment

In the real environment, we have to make readaptation for new situation. However, the task is very time consuming if applied in real world. Here we show an example which is successful in the real environment. We examine the Khepera's behavior in the case that a part of a wall is lacking as shown in Figure 6(b). In this case, the value of the #5 sensor decreases at the gap. The Khepera with three layered network (1),(2),(3) moves forward without turning right, because ξ keeps up the active state due to the cooperative term (see Figure 8). If we take off the neural network with cooperation and competition and let the robot move under row sensor data, the robot happens to collide the wall gap(Figure 7). This shows the three layered network has a nice robust feature under defective situations.



(a) (b)

Figure 6: Khepera's trajectory in experiment (2).

(a) Movement along a wall.

(b) Movement along a wall, which is lacking partially.

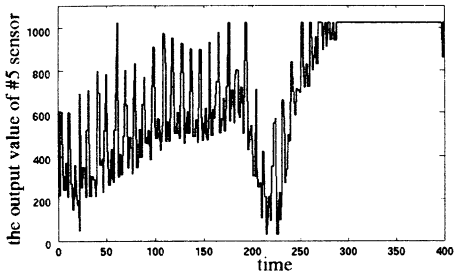


Figure 7: Output value of #5 sensor.

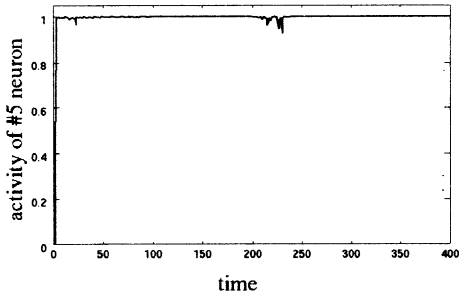


Figure 8: Neuron activity ξ related to #5 sensor.

5. Conclusion

We have presented the control system for mobile robots by using competitive and cooperative 3 layered neural network and also using self adaptive neural network. The competitive term makes only one neuron ξ get active for large input values and the cooperative term makes ξ keep up the active state for a small fluctuation of the input value. As a result, self learning neural network created four distinct types of Khepera by using the ξ outputted from the 3 layered network. We found that the Khepera controlled by the competitive and cooperative neural network had robust behaviors when going along a defective wall.

Reference

[1] Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press

[2] Floreano D, Mondada F (1998) Evolutionary neurocontrollers for autonomous mobile robots. Neural Networks 11:1461-1478

[3] Harvey I, Husbands P, Cliff D, Thompson A, Jakobi N (1997) Evolutionary robotics: the Sussex approach. Robotics and Autonomous Systems 20:205-224

[4] Brooks RA (1992) Artificial Life and Real Robots. In: Proceedings of the first European Conference on Artificial Life, pp.3-10

[5] Dorigo M, Colombetti M (1998) Robot Shaping: An Experiment in Behaviour engineering. MIT Press

[6] Amari S, Arbib MA (1977) Competition and Cooperation in Neural Nets. Systems Neuroscience:119-165

[7] Reibnab D, Haken H (1994) Stereo Vision by Self-organization. Biol. Cybern. 71:17-26

[8] Kitazoe T, Tomiyama J, Yoshitomi Y, Shii T (1998) Sequential Stereoscopic Vision and Hysteresis. In: Proceedings of Fifth International Conference on Neural Information Processing, pp. 391-396

[9] 11. K-Team SA (1995) Khepera USER MANUAL Version 4.06.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Control Systems for Real Robot using Classifier Systems

Tomofumi Jinguuji, Masayoshi Tabuse and Kei Sugihara
Department of Computer Science and Systems Engineering
Faculty of Engineering, Miyazaki University
1-1, Gakuen Kibanadai Nishi, Miyazaki, 889-2192 Japan
{ralph, tabuse, ks}@cs.miyazaki-u.ac.jp

Abstract

In this paper we consider a control system for a mobile robot Khepera using learning classifier system and estimate the capability of this system in the simulation based on the real word physics. In our researches, we find the Khepera controlled by learning classifier is an effective method in patially obserbable and noisy environments.

1. Introduction

Recently, many people have been researching autonomous robots extensively. One purpose of this research is to build the robots that are able to behave in unknown or dynamically changing environments.

Evolutionary robotics is one of the attractive methodologies for autonomous robots. In evolutionary robotics approach, developing robot systems adapt to dynamic environments, based on evolutionary algorithms[1]. One promising approach to evolutionary robot control systems is learning classifier systems.

In general, evolutionary processes require the large population size and the number of generation. Thus, experiments of evolutionary robotics are usually carried out in the computer simulations. Computer simulations may be very helpful to train and test robot control systems. In many cases, the simulations of agents controlled by classifier systems are carried out in a grid world. In the grid world, agent actions are 5 primitive ones (up, down, left, right and stay) and a state of an agent surely changes into a next state according to an agent action. In the real world, an agent moves in any direction and transition of states is not always same. Thus to maximize a fit between real and simulated environments, the simulation is based on a spatially continuous two dimensional model of underlying real world physics.

In the present paper, we simulate a mobile robot Khepera based on a spatially continuous two dimensional model and estimate capability of the control of Khepera using learning classifier systems.

2. Khepera

We consider control system for a miniature mobile robot Khepera[2]. The Khepera body is 32 mm height and 55 mm in diameter. It has two wheels, each of which is controlled by a DC motor, and can rotate in both directions.

The eight infra-red proximity sensors are installed around the Khepera body (six in front and two in rear, see Figure 1). These sensors allow two measures: the light reflected by obstacles and the ambient light. So these detect an obstacle and a light source. The output values of each sensor are from 0 to 1023 in integer. For the measurement of the light reflection, 0 means that there are no obstacles near to the sensor, while 1023 means that an obstacle is very close to the sensor. The maximum detection range of the sensor is about 3 cm. On the other hand, for the measurement of an ambient light, the output values of the sensor degrease when the intensity of a light increases. The Khepera recognizes its own environments through these sensors, so that the Khepera can understand the local environments only.

The Khepera can communicates with a computer using a serial line. So we can obtain the sensor values from the Khepera and provide the wheel speed to the Khepera. In our experiments, the Khepera is controlled by a Sun SPARCstation through a serial line.

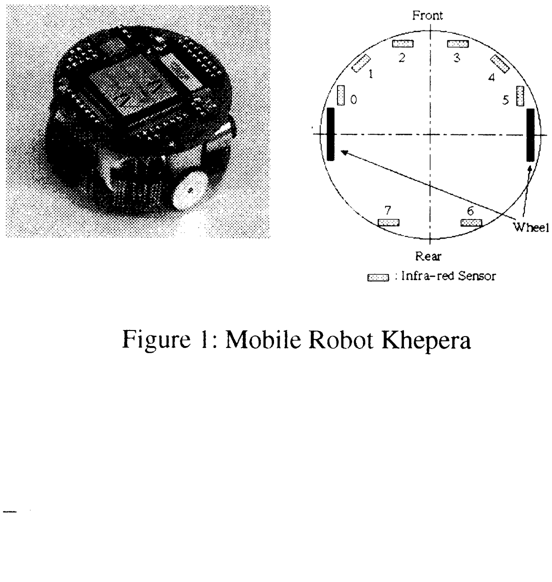
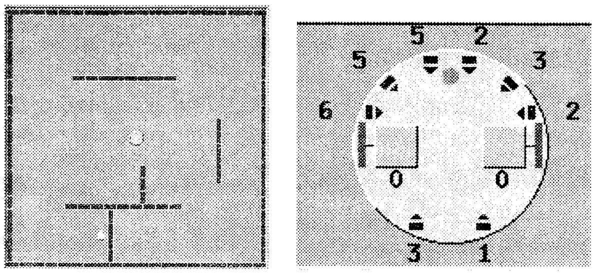


Figure 1: Mobile Robot Khepera

3. Khepera simulator

The experiments are carried out with Khepera simulator[3], which simulates a real mobile robot Khepera (See figure 2). The robot has two motors and eight sensor measuring both ambient light and distance from obstacles, which are the same ones as the real robot Khepera. To minimize a gap between real and simulated environments the simulation takes into account noise at the level of sensor measurements and motor actions.



(a) (b)

Figure 2: (a) Field of Khepera simulator. (b)Khepera's sensors and motors in Khepera simulator.

4. Learning Classifier Systems

A classifier is a condition-action rule with a scalar strength, which estimates the classifier's utility of the system. The condition part of classifier is represented by a string composed from the set {0,1}, while the action part is composed from {1,0,-1}. The strength is used a measure of utility in both performance and rule discovery.

In our experiments, the condition part corresponds to eight sensors' values, where 0 for a small sensor value (<600) and 1 for a large sensor value(≧600). The action part corresponds to left and right motor actions, where 1, 0, -1 mean positive rotation, stay and negative rotation for left and right motor, respectively. Thus Khepera has nine movements (forward, stay, back, turn-left, turn-right, spin-left, spin-right, backspin-left, backspin-right).

The genetic operations of classifiers consist of one point crossover, mutation and selection according to the strength of classifiers. In one point crossover, we select two parents, where the probability of a classifier being selected as a parent is based on its strength, and construct two offspring using one point crossover. In the selection of classifiers the genetic operations select 100 different classifiers of highest strength. Figure 3 shows the genetic operations of classifiers.

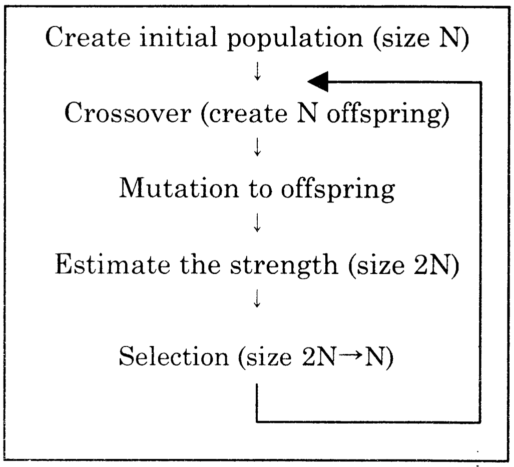

Create initial population (size N)
Crossover (create N offspring)
Mutation to offspring
Estimate the strength (size 2N)
Selection (size 2N→N)

Figure 3: Genetic operations.

5. Experiments

In order to estimate capability of the control of Khepera by classifier system, we consider the following tasks:

- Obstacle avoidance

(1) Obstacle avoidance

We consider the obstacle avoidance task, which is that the Khepera moves around and avoids a wall (see Figure 4). In the classifier system, the number of classifiers is N=100, the crossover rate is 100%, that is 100 offspring are created from 100 parents, and the mutation rate is 4%. The evaluation function for the strength of classifiers is

$$f = \left| \frac{V_L + V_R}{2} \right| \cdot \left(1 - \frac{|V_L - V_R|}{2 \cdot V_{\max}}\right) \cdot (1-i)^3 \cdot \left(\frac{V_L + V_R + 2 \cdot V_{\max}}{4 \cdot V_{\max}}\right) \qquad (1)$$

where V_L,V_R, $V_{\max}$ and i are left and right motor speeds, the maximum speed of V_L and V_R and the normalized maximum sensor value, respectively. The first component is maximized by speed, the second by straight motion, the third by distance from object and the forth by forward motion. In addition to the above rewards, when the Khepera collide with a wall, the series of selected classifiers (*rule*(t_0), *rule*(t_0-*1*), ...) suffer punishment:

$$f_1(t) = p \cdot \gamma^{t_0 - t} \quad for \quad t = t_0, t_0 - 1 \qquad (2)$$

where p=-10 is punishment, γ=0.4 is the discount rate and $f_1(t)=0$ for $t<t_0-1$.

Figure 4 shows Khepera's trajectory controlled by this classifier system, which has learned in 30 generations. The average values of reward and punishment at each generation in 4 trials presented in Figure 4. The average values increase steadily during 30 generations. Table 1 shows the 30 classifiers of highest strength obtained in this experiment.

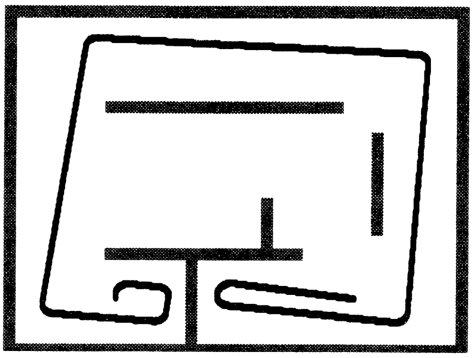


Figure 4: Khepera's trajectory controlled by the classifier system.

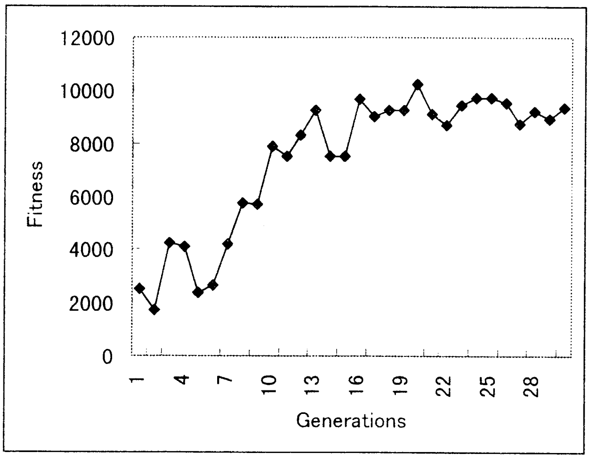


Figure 5: The average values of reward and punishment

6. Conclusion

We have presented the control system for a mobile robot Khepera by using learning classifier system and simulate the obstacle avoidance task in continuous two dimensional model based on real world. In the continuous model the transition of states is not always same, because the Khepera's observations of the environment is partial, and sensor measurement and Khepera movement are noisy and uncertain. As a result, we could find that the Khepera controlled by learning classifier is an effective method in these situations.

Condition	Action	Strength
0,0,0,0,0,0,0,0	1,1	8369.548828
1,0,0,0,0,0,0,0	1,1	966.902344
0,0,0,0,0,1,0,0	1,1	614.152344
0,0,0,0,1,1,0,0	0,1	67.132324
1,1,0,0,0,0,0,0	1,0	64.986816
0,1,0,0,0,0,0,0	1,1	29.998047
1,0,0,0,0,0,0,0	1,0	29.401855
0,0,0,0,1,0,0,0	1,1	26.851562
0,0,1,1,0,0,0,0	0,-1	20.938232
0,0,0,0,0,0,0,0	1,0	20.100098
0,1,1,1,0,0,0,0	0,-1	19.477295
0,0,0,0,1,0,0,0	0,1	15.95459
0,0,0,0,0,0,0,0	0,1	15.535645
0,0,1,1,0,1,0,0	0,1	15.268311
0,1,0,0,0,0,0,0	1,0	15.265381
0,0,1,1,1,0,0,0	-1,0	15.177246
0,0,1,1,0,0,0,0	-1,0	14.906738
0,1,1,0,0,0,0,0	1,0	14.00708
0,0,0,0,0,1,0,0	0,1	13.9104
0,0,0,0,0,0,1,1	1,1	11.029297
0,0,0,0,1,0,1,0	1,1	10.910156
0,0,0,0,0,0,1,0	1,1	10.363281
0,0,1,1,1,1,0,0	-1,0	10.116943
1,1,0,0,0,0,0,1	1,1	9.773438
0,0,1,1,1,0,0,0	0,-1	9.268555
1,0,0,0,0,0,1,1	1,1	8.865234
0,0,0,0,0,1,1,1	1,1	8.662109
0,1,0,0,0,0,0,0	0,-1	8.615723
0,1,1,1,0,0,0,0	-1,0	8.143799
0,0,0,0,1,1,0,0	1,0	6.615479

Table 1: Classifiers obtained in the experiment (1)

Reference

[1] J.H. Holland, "Adaptation in natural and artificial systems", The University of Michigan Press (1975).

[2] "Khepera USER MANUAL Version 4.06", K-Team SA (1995).

[3] O. Mitchel, "Khepera Simulator Package version 2.0", http://wwwi3s.unice.fr/~om/khep-sim.html (1996).

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Evolutionary simulations of "*SUMIWAKE*" habitat segregation in a finite and heterogeneous ecosystem

T. Oohashi [*1, *2], T. Maekawa [*3], O. Ueno [*4, *5], N. Kawai [*5, *1], and E. Nishina [*6, *1]

*1: ATR Human Information Processing Research Laboratories, Japan
*2: Chiba Institute of Technology, Japan
*3: ATR Media Integration & Communications Research Laboratories, Japan
*4: Department of Biochemistry, Gifu University School of Medicine, Japan
*5: Foundation for Advancement of International Science, Japan
*6: National Institute of Multimedia Education, Japan
Contact: oohashi@hip.atr.co.jp

Abstract

The authors, as previously reported [1], developed a general platform for simulation, called SIVA, on which life activities and their evolution can be simulated in conditions that approximate the actual terrestrial ecosystem and living organisms, based on the development and examination of SIVA-I, II, and III as prototypes of the simulator. That is, SIVA is characterized by simulated environmental conditions that are finite and heterogeneous, and by simulated individuals that are able to self-decompose as well as self-reproduce. The source program of SIVA version 1.1 in JAVA language is distributed by ATR[*1, *3]. In the present study, the authors conducted simulations on SIVA in order to examine evolutionary adaptation, and tried to clarify whether divergent proliferation of the most prolific or vital individual introduced the broad distribution of individuals observed in the result of the simulation, or whether it was due to the complications of "habitat segregation." As a result, it was found that the complications of "habitat segregation" led to the broad distribution of individuals, through a comparative analysis of existable area and that of the degree of correspondence of genetic information. This finding introduces a way to investigate the constructive mechanism of "habitat segregation" in the actual terrestrial ecosystem, and also suggests a survival strategy for human beings.

1. Introduction

Observation of living organisms in the actual terrestrial ecosystem shows that the most prolific organisms do not expand their living area uniformly or occupy the whole ecosystem. Instead, diverse kinds of organisms co-exist and share the ecosystem. Such phenomena were named as "***SUMIWAKE*** (habitat segregation)" in Japan and gained interest of many biologists since the 1930's (e.g. K. Imanishi [2]). In order to examine these phenomena, we have made the following definition: All living organisms on the earth have their own inherent positions and extents (dwellings) to live, and they apportion the positions and extents to each other or share them with each other. In other words, an ecosystem includes a complex pile of all the positions and extents. There is a special state in which plural groups of organisms that belong to different ***species*** from each other or to the same ***species***, and that have similar life styles, apportion the environment in aspects of time or space to each other, do not mix with each other, and live their own lives with little or no interaction. This state is defined as "*SUMIWAKE* (habitat segregation)." The Japanese name for this concept is used in this report because it may not precisely correspond to the concept called *habitat segregation* or *niche* in English. ***SUMIWAKE*** is an essential basis of an ecosystem. It is important to investigate its characteristics and the mechanism of its formation in order to obtain a more profound understanding of the terrestrial ecosystem.

Since a certain period of history, human beings have come to consider continuous advancement and expansion as great value. This sense of values should be a basis and a fountainhead for the advancement of modern societies and imperialism. It is thought that the authentic understandings of biological evolution also are closely connected with this sense of values, as shown that the sense was authorized by the beginnings of Darwinism as the logic of power. For example, "natural selection, " which is regarded as the most important and authentic mechanism of biological evolution, can be expressed as "the most generally prolific organisms prosper." That is, natural selection presupposes a tendency towards expansion, divergence, and competition with other organisms. It is important to examine the idea that such a viewpoint towards expansion, divergence, and competition can explain the actual terrestrial ecosystem with more complete fidelity than the viewpoint of ***SUMIWAKE*** towards convergence and co-existence.

Many studies have been conducted in the field of Artificial Life, in order to simulate the evolution or adaptation of living organisms, towards clarifying the evolutionary mechanism and its application to engineering. The design of almost all simulative platforms attached importance to ALife modeling itself or to interactions between ALives, especially the struggle for existence, but the procedure and the expression of the interactions between ALives and their environment were still relatively primitive. Therefore, these platforms were relatively useful for simulations of the proliferation, growth, and activities of ALife or for interactions between ALives, but not for simulations of ***SUMIWAKE*** in which the interactions between ALives and their environment are a main factor.

Concretely, the actual terrestrial ecosystem has finite, heterogeneous, and dynamic environmental conditions. Therefore, it is expected more effective to let a simulative environment also have finite, heterogeneous, and dynamic conditions, in order to investigate life activities including interactions with the environment. Moreover, in spite of the fact that death and decomposition as well as reproduction and growth are important aspects of life activities [3-7], the ordinarily designed simulation platforms could treat death and decomposition only in a passive way, and the dynamics of the life activities realized by them were narrow. Therefore, ALife should have the ability of autonomous death. In order to realize these requirements, we previously developed and proposed the simulation platform SIVA [1]. SIVA is a general simulation platform, which enables the simulation of self-reproductive and self-decomposable ALives in a finite, heterogeneous, and dynamic environment.

Based on the above, a simulation of evolutionary adaptation on SIVA and the following analyses were conducted in the present study. First, it was confirmed that virtual organisms expand their distribution to various areas characterized by different environmental conditions. Second, we examined whether such distribution resulted from the phenomena of ***SUMIWAKE*** of the habitat segregation type of organism, or from a uniform expansion of the divergent proliferation type of organism that appeared evolutionarily, or anything else.

2. On the composition of SIVA [1]
2.1 Overview

We have proposed a theoretical hypothesis of the programmed self-decomposition (PSD) model [3,4] and obtained some supportive results of computer simulations using SIVA-I, II, and III, the prototype series of the simulation platform, concerning merits of the PSD mechanism [4-7]. The simulation platform SIVA was designed and developed on the basis of the above concept and findings. It is a general simulator, which enables the simulation of virtual life activities based on the PSD model in a finite, heterogeneous, and dynamic environment. SIVA has the characteristics shown in Table 1. The version number of the program, used in this study, is 1.1, and we call it SIVA-v1.1 in this report.

Table 1 The characteristics of SIVA

	Characteristics
Environment	Finite (Substance, Energy, Space) Heterogeneous (Substance, Energy)
Individuals	Based on the PSD model Virtual biological molecules are classified According to their roles in the activities - Automaton - Structure automaton - Function automaton - Temporary information automaton - Information - Constitutive information - Energy Virtual biological molecules compose a multi-layer system - Polymer - Sub-polymer - Monomer - Element

2.2 Implementation of SIVA-v1.1

Implementation of virtual environment

The virtual space of SIVA-v1.1 is a finite 2-dimensional lattice of 16 x 16 pixels. The initial values of physical conditions, such as the amount of substances, amount of energy, and temperature, can be defined for each block of the lattice independently. Each block consists of a lattice of 8 x 8 pixels. One individual can occupy each pixel.

Implementation of virtual individuals

Simulative individuals consist of virtual biological molecules. The molecules are classified into Automata, Information, and Energy according to their roles (Table 1). Automata and Information have a 4-layer system (Polymer, Sub-Polymer, Monomer, and Element). One can design the diversity of the structure and the activities of the individuals in a well-ordered style. The activities of the individuals are defined using SIVA-Language, each word of which is equivalent to one Automaton-Sub-Polymer. Each Automaton is synthesized according to the series of Information. There is an inter-generational transmission of Information (sometimes mutated). Consequently, one can observe hereditary or evolutionary phenomena. There is redundancy in the correspondence between a series of Information and Automaton, as like the redundancy of the actual living organisms.

3. Simulative experiment of evolutionary adaptation
3.1 Methods

(1) Configuration of the environmental conditions

The target of the present investigation was evolutionary adaptation in a finite, heterogeneous environment. So, the initial distribution of the environmental conditions was prepared according to the default configuration of SIVA-v1.1 (Figure 1). When the upper direction of the simulative environment was called as north in figure 1, abundant element A was prepared in the west area, but poor in the east area for example. The higher temperature was given in the northwest, but lower in the southeast. Abundant energy was prepared in the center of the environment, and but poor in the marginal area.

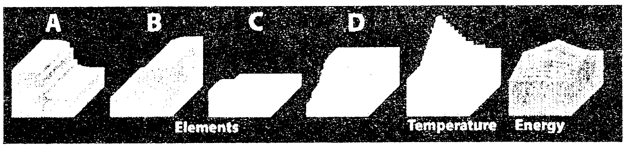


Figure 1 Initial distribution of the environmental conditions

(2) Designing of the virtual organism

The target of the present investigation was evolutionary adaptation of the self-reproductive and self-decomposable virtual organisms. So, the original individual of the virtual organism was designed according to the default configuration of SIVA-v1.1. That is, the original individual was designed to have a set of Automaton, which conducted the statements of SIVA-Language shown in table 2, and a series of Information, which corresponded to the Automaton. Consequently, the optimum temperature of this individual was set to 10 degree of temperature (ID0), and it could synthesize Automaton according to Information (ID1), could replicate Information (ID2), could divide a new individual from itself (ID3), and could decompose itself to elements (ID4).

Table 2 Statements of SIVA-Language of the original individual

Automaton Polymer 0	ID0 . opt_temperature == 50000 <<4762 .
Automaton Polymer 1	ID1 . syntha . movef .
Automaton Polymer 2	ID2 . copyi . movef .
Automaton Polymer 3	ID3 . length_AP >= max_length_AP length_IM >= max_length_IM divid .
Automaton Polymer 4	ID4 . length_IM = 0 length_AP = 1 decsf . unconformity > 2 decae decie . age > 20 decae decie .

Some essential parts of these Information was masked against mutation, in order not to damage the basic life activities (ID0 – ID4). Therefore, the optimum temperature and the ratio of the element formation, which was due to mutation of the redundant parts of Information, were changeable in the present experiment. The change of these conditions changes the conformity of the individual to the local block of the environment where the individual exists.

That is, the larger the distance between the local temperature and the optimum temperature, or the fewer the elements buried in the local block than the requirement, the more the energy for temporary adaptation of the individual is required. If the execution of a life activity was failed due to lack of energy and so on, the degree of the unconformity of the individual to the environment increases by 1. When the degree of the unconformity exceeds 2, the PSD mechanism must be triggered (ID4).

(3) Executive procedure of the simulation

In order to simulate evolutionary adaptation, one individual designed as above-mentioned was seeded in the center of the environment configured as formerly mentioned. Mutation rate at both the replication of Information and the synthesis of Automaton was 10^{-4}. Duration of the simulation was TC = 0 - 100000. The detailed conditions of SIVA-v1.1 are shown in table 3.

Table 3 The present conditions of simulation for SIVA-v1.1

Environment		Individual			
env_size	128x128	co_mask	1500	Temperature	
block_size	8x8	matter_gain	0.4	Distance	Rate
diff_matter	0.001	monomer_gain	0.0	0	1.0
diff_monomer	0.0	energy_gain	0.8	1	1.0
matter_sense	0.45	mis_copyi_rate	0.0001	2	0.95
monomer_sense	0.4	mis_syntha_rate	0.0001	3	0.9
energy_sense	0.6	mis_co_rate	0.0	4	0.8
Individual		cosmic_ray	0.0	5	0.7
energy_release	0.0	temp_adaptability	0.95	6	0.5
diff_indiv	off	temp_adapt_base	16.0	7	0.2
opt_temperature	10	temp_adapt_thrsld	0.1	8 - 20	0.1

(4) Analysis of the simulation results

Following analyses were conducted, contriving to examine whether it was the phenomena of *SUMIWAKE* of the habitat segregation type of organism, the result of a uniform expansion of the divergent proliferation type of organism, or anything else.

Observation of individuals distribution
The transition of the simulation was observed. It was confirmed that self-reproductive and self-decomposable individuals adapted evolutionarily and advanced to the new areas in the finite and heterogeneous environment during 100000TC of the simulation.

Sampling individuals and analysis of existable area
Extension of the existable areas of individuals was examined at the end of the simulation. Based on the extension of the existable area, it was determined whether the examined individual was an individual characterized by the divergent proliferation type which had non-limited existable areas, or whether it had limited existable areas. First, fifteen individuals were randomly sampled from various areas at the end of the simulation of 100000TC. Second, each 256 copied-individuals were prepared for the each sampled individual. Third, each of 256 copied-individuals was seeded in the center of the each 256 environmental blocks (environmental conditions were homogeneous in one block) after initialization of the simulation environment. Fourth, a simulation was conducted for each sampled individual under the condition as follows: mutation rate was 0.0; duration was TC0 - 1000. It had been confirmed that extension of living areas became almost stable after 1000TC of simulation if mutation rate was 0.0 by preliminary experiments. Fifth, the blocks in which any descendant of the copied-individuals survived after the simulation were determined as the existable area of the sampled individual.

Analysis of correspondence of Information
In order to examine whether the degree of correspondence of Information (what is called ***genotype***) was high or not between individuals that have similar existable areas (what is called ***phenotype***) with each other, the degree of correspondence was calculate about Information according to the following procedure. First, each pair of Information-Monomers was compared with each other, from the top of Information-Polymers of the examined two individuals to the bottom of them. Second, the number of pair ***c*** which consisted of the same kinds of Monomers (W,X,Y, or Z) was counted. Third, the degree of correspondence was calculate as the ratio of ***c*** to the total number of Monomers ***n***.

3.2 Results

(1) Changes in individual distribution

We verified the emergence of evolutionary adaptation by observing the progress of simulation. That is, individuals expand their distribution through repetitive proliferation and decomposition, and finally (TC100000) occupied one third of the whole environment (Figure 2). We conducted the following analyses to examine the state of adaptation to the heterogeneous environment.

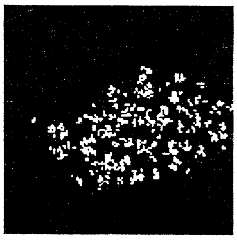
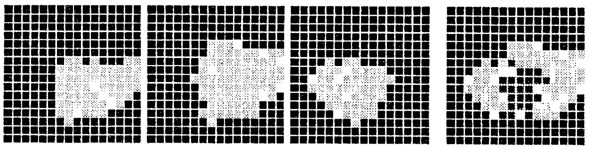


Figure 2 Individual distribution observed at TC100000

(2) Distribution of existable areas

Fifteen individuals were randomly sampled. By examining their existable areas, we determined whether they were individuals characterized by the divergent proliferation type which had non-limited existable areas, or individuals which had limited existable areas. As a result, all the 15 samples had limited existable areas, and not the divergent proliferation type. The existable areas of these 15 samples were roughly categorized into three groups shown in Figures 3(a)(b)(c).



(a)Group A (b)Group B. (c)Group C (d)A+B+C

Gray: Existable area for each group
White: Locations for random sampling of individuals
Dark Gray: Overlapped existable area among Group A, B, C

Figure 3. Existable areas

As shown in these figures, the group A had its existable areas in the southeast, when the upper direction of the simulative environment was called as north. The existable area for the group B contained that for the group A and extended to northeast. However, the individuals belonging to the groups A and B did not intermix. The existable areas for the group C extended to the southwest. Although these three groups shared the common existable area in the center of the simulated ecosystem (Figure 3(d)), the individuals belonging to these three groups never intermixed.

(3) Degree of correspondence in Information

We examined the degree of correspondence in Constitutive Information for the fifteen sampled individuals. Then, we examined relationship between the degree of correspondence in Information and the overlaps of existalbe area. As a result, each of the group A and B was divided into two groups, respectively (Figure 4). The individuals belonging to each group showed remarkably high degree of correspondence of the Constitutive Information, namely, always above 93 % and around 99 % in most cases. In contrast, the individuals belonging to the different groups showed significantly lower degree of correspondence, between 54 and 66 %. Note that all the individuals analyzed here shared 65-68% of Constitutive Information with the original individual in the evolutionary adaptation simulation.

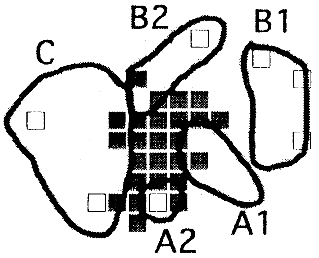


Figure 4. Segregation of groups by correspondence of Information

3.3 Discussion

First of all, we discuss that whether the composition of expansion of living areas observed in the present simulation resulted from the phenomenon of *SUMIWAKE* of the habitat segregation type of organism, or from a uniform expansion of the divergent proliferation type of organism that appeared evolutionarily, or anything else. It is expected that the divergent proliferation type of organisms would possess extensive .existable areas where they would distribute. However, no sampled individual was characterized by such divergent proliferation type. In addition, there was a clear tendency that the individuals which possessed similar existable areas lived neighborhood. Conversely, the individuals which possessed different existable areas in a certain degree never intermixed. Although these sampled individuals possessed an living ability in the areas where their ancestor (original individual) lived to some extent, they developed evolutionary adaptation so as to ensure different existable areas. When longer simulation time will be employed, it is very likely that each existable area will not have any overlaps. This direction of evolution can be judged as a progression towards *SUMIWAKE*, based on the present definition.

The analysis of degree of correspondence of constitutive information supported the above characteristics of distribution structure that each group of individuals did not diverge, but did converge and gradually segregated. Although any individuals belonging to the same group showed more than 93 % of the degree of correspondence, any individuals belonging to the different groups showed the degree of correspondence between 54 % and 66 %. Namely, similarity of gene (Constitutive Information) was clearly categorized into two groups, similar or dissimilar, and no intermediate zone. This finding indicated that individuals, which possessed similar gene (Constitutive Information), gathered together and composed each group. In other words, it is suggested that the evolution would be developed in a direction to compose groups.

It is interesting that we observed only 55% degree of correspondence in Constitutive Information between A-1 group and A-2 group, in spite that both groups had a similar existable area. The degree of correspondence between B-1 group and B-2 group was also only 66%. Namely, accumulated mutation, which has less relation to determination of existable area, caused less degree of correspondence. The accumulated genetic information develop totally different life activities at some point, resulting in generating groups that have different life activities and genetic information. M. Kimura proposed the neutral theory of molecular evolution where he argued that the evolutionary change in the molecular level is mostly explained by the incidental fixation of mutational gene, which is neutral or almost neutral in the sense of Darwinian selection [8]. The results we observed may be related to this Kimura's neutral mutation theory in its principle.

Thus, it is interesting that the evolutionary adaptation to finite, heterogeneous environment is realized by complication in state of *SUMIWAKE*, not by occupation of the divergent proliferation type of organisms. The simulation using SIVA has made it possible to examine the mechanism of *SUMIWAKE* that can explain actual terrestrial ecosystem. We would like to examine it considering its relation to community theory and non-equilibrium theory of species coexistence, etc. Present finding also suggests, for human beings living in the terrestrial ecosystem characterized by finite heterogeneous environment, living strategy more effective than divergent and expansive behaviors shown in the divergent proliferation type of organisms in the present simulation.

4. Conclusion

A simulation of the evolutionary adaptation of self-reproductive and self-decomposable virtual organisms in a finite, heterogeneous, and dynamic environment was conducted on the SIVA version 1.1 simulation platform. The results of a comparative analysis of existable area and that of the degree of correspondence of genetic information suggest that the complication of "*SUMIWAKE* (habitat segregation)" led to the broad distribution of individuals.

Acknowledgment

The authors would like to gratefully thank Mr. Seigo SHINDO for his cooperation in conducting the simulations.

References

[1] T. Oohashi *et al.*: Proposal of a general simulator "SIVA" for heterogeneous environment and self-decomposable ALife, AROB 5th '00, 45-49 (2000)
[2] K. Imanishi: ***Seibutsu no Sekai***, (in Japanese) (1941)
[3] T. Oohashi *et al.*: Programmed self-decomposition model, ***Kagakukisoron kenkyuu***, 18, 2, 79-87 (in Japanese) (1987)
[4] T. Oohashi *et al.*: Artificial Life based on Programmed Self-Decomposition Model, ATR HIP, TR-H-198 (1996)
[5] T. Oohashi *et al.*: Requirements for immortal ALife to exterminate mortal ALife in one finite, heterogeneous ecosystem, ECAL'99, 49-53, Springer (1999.9)
[6] T. Maekawa *et al.*: Diffusible immortal ALife rarely exterminate diffusible mortal ALife in one finite, heterogeneous ecosystem, AROB 5th '00, 34-39 (2000)
[7] T. Maekawa *et al.*: Evolutionary advantage of Self-Decomposition mechanism, AROB 5th '00, 40-44
[8] M. Kimura: The neutral theory of molecular evolution, Cambridge University Press, 1983

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Macro-micro Evolutionary Algorithm: Multi Agents Model for Optimization

Sang-Keon Oh, Kap-Ho Seo and Ju-Jang Lee

Department of Electrical Engineering and Computer Science
Korea Advance Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, Korea, 305-701
e-mail: sgoh@odyssey.kaist.ac.kr

Abstract

This paper introduces a new evolutionary algorithm named Macro-micro Evolutionary Algorithm (MmEA), developed for solving multimodal optimization problems. The algorithm consists of two evolutionary algorithms which controls global species and local individuals respectively. The hierarchical architecture of the proposed algorithm incorporates two contradictory aspects of evolutionary search methods: Speed and Robustness. By examining the computational results, we notice that the algorithm is effective and efficient than previous algorithms in large instances of test problems.

1 Introduction

Evolutionary algorithms (EAs) are stochastic search techniques inspired by the mechanism of natural evolution. Although they have been demonstrated to be robust in searching large spaces in a wide range of applications, they have some problems such as premature convergence and slow search speed.

In order to accelerate and improve the performance of EAs, many algorithms have been developed. In simple genetic algirthm (GA), diversity of population is controlled by genetic operators (mutation rates, crossover rates, and selection methods, etc.) [2]. De Jong suggested crowding method as a niching method. Goldberg and Richardson [2] proposed a sharing function to induce species and niches. On the other approaches, parallel subpopulation structures have been researched including many hierarchical and hybrid methods [1]. To enhance the search speed, problem-specific local search operators have been used in many practical problems [3]. Although local operators enhance the speed of search, it easily induces premature convergence and must be adopted deliberately.

In this paper, we propose a Macro-micro Evolutionary Algorithm (MmEA) to maintain the diversity of population and enhance the local search ability. The model is primarily inspired by the macorevolutionary algorithm (MA) introduced by Marin *et al* [4]. The MA model exploits the presence of links between "species" that represent candidate solutions to the optimization problem. Using the relation links, extinction and diversification of species are at work as evolutionary operators. In our work, we extend the structure of MA to two-level hierarchical model which implies many subpopulations. Each subpopulation have several individuals and performs local search operations to accelerate the search processing. On the other side, we exploit the evolution of multiple species using the relationship similar to that of MA to maintain the diversity of population.

In the next section, a brief review on MA and μGA is given. Then, we describes the proposed MmEA. In Section 3, computational experiments are presented. Section 4 concludes this paper with some remarks.

2 The Macro-Micro Evolutionary Algorithm

2.1 Macroevolutionary algorithm

The model of macroevolution was proposed by Marin and Solé and facilitates simulating the dynamics of species extinction and diversification for large time scales. The relation between species are essential to determine the existence of each species and represented by a connectivity matrix $\mathbf{W}$, where each item $W_{i,j}$ measures the influence of species on species with a continuous value. The influence $W_{i,j}$ represents the relative fitness with sharing function (eq. 1).

$$W_{i,j} = \frac{f(\mathbf{p}_i) - f(\mathbf{p}_j)}{|\mathbf{p}_i - \mathbf{p}_j|} \quad (1)$$

where, $\mathbf{p}_i = (p_i^1, ..., p_i^n)$ are input parameters of the ith individual.

At every generation, all influence $W_{i,j}$ are calculated and used for determining the existence and extinction of each species. The selection is determined by sum of relation S_i.

$$S_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j=1}^{P} W_{i,j}(t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where, t is the generation number. Species with $S_i = 0$ will be extincted from whole population P.

Then, all extinct species are replaced by varieties of existing species or randomly generated species. This replacement operator called colonization operator has two different operators based on exlorationa and exploitation. As an exploration operator, a totally new solution (p_n) will be generated to inject new alles with a given probability. Otherwise, exploitation of surviving solutions takes place through colonization using extinct solution p_i and surviving solution p_b (eq. 3).

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{p}_b(t) + \rho\lambda(\mathbf{p}_b(t) - \mathbf{p}_i(t)), & \text{if } \xi > \tau \\ \mathbf{p}_n & \text{otherwise} \end{cases} . \quad (3)$$

where $\xi \in [0,1]$ is a uniform random number, $\lambda \in [-1,+1]$ and ρ and τ is given constant.

The main idea of MA is that the system will choose, through network interactions, which are the individuals to be eliminated so as to guarantee exploration by new individuals and exploitation of better solutions by further generations. In MA model, each species have only one individual and interact with each other using network structure, so it can be categorized into fine-grained model of evoultionary algorithm.

2.2 Micro-Genetic Algorithm

One drawback of simple genetic algorithm (SGA) is the time penalty involved in evaluating the fitness functions (performance indices) for large populations, generation after generation. In Kris's work, a small population approach (named μGA) with some very simple genetic parameters was proposed, and it was shown that μGA implementation reaches the near-optimal region much earlier than the SGA implementation. The superior performance of the μGA in the presence of multimodality and their merits in solving non-stationary function optimization problems were demonstrated.

In the μGA proposed by Kris, the population size is fixed at five. It is a known fact that GA generally do poorly with very small populations due to insufficient information processing and early convergence to non-optimal results. The key to overcome the drawback is in bringing in new strings at regular intervals into the population. Based on this, a procedure for the μGA implementation is presented below.

1. Select a population of size 5 either randomly or 4 randomly and 1 good string from any previous search.
2. Evaluate fitness and determine the best string. (elitist strategy)
3. Perform selection and crossover.
4. Check for nominal convergence. If converged go to step 1.
5. Go to step 2.

2.3 Macro-micro Evolutionary Algorithm

The MmEA is proposed to balance the exploration and exploitation by incorporating local search operators and global search operators. Because EAs display inherent difficulties in performing local search for many applications, there exists a lot of approaches that use various fine local tuning methods [3] [5]. The MmEA also use those hierarchical approach in which high-level algorithm maintains the diversity of population while low-level algorithm finds local optimal solution. In high-level structure of MmEA, various species are conserved by network interaction between species and undergo diversification and extinction like as does in MA. In low-level structure of MmEA, each species has small number of individuals as its members. The extension of concept of species helps to finding local optimal solutions by giving chance to cooperate among individuals. Although it is possible to adopt various local search operators, we use a simplified quadratic optimization method as a local search operator for numerical optimization problems in our implementation.

The detailed method is as following.

A) Global Optimization Structure: Population of size P is consist of $P/3$ subpopulations (species). Competition between species is determined by eq. 2, and fitness value of i'th species $(\mathbf{p}_i)$ is calculated using best indviduals i_{best} in i'th species.

$$W_{i,j} = \frac{f(\mathbf{p}_{i_{best}}) - f(\mathbf{p}_{j_{best}})}{|\mathbf{p}_{i_{best}} - \mathbf{p}_{j_{best}}|} \quad (4)$$

Extinction and diversification of species is determined by relative fitness values of species (see eq. 2). This sharing process ensures that a spread of solutions is obtained over multi-modal fitness spaces.

B) Fine Local Tuning: Each species contains three individuals as its members, and these individuals cooperate with each other to find local optimal solution. Because the size of subpopulation is small, computation burden per each species are much smaller than standard GA and local search ability is better than other algorithm. The simplified quadratic optimization method introduced in this paper performs d one-dimensional quadratic optimization rather than solving d-dimensional quadratic optimization problem which requires gradient and hessian matrix of fitness landscape. For given three points and their function values (x_1,f_1), (x_2,f_2), (x_3,f_3), an optimal solution can be obtained under the assumption that those points are lying on same quadratic function. The optimal solution candidate $g(\cdot)$ based on three points (x_1,x_2,x_3) can be formulated as follows:

$$g(x_1,f_1,x_2,f_2,x_3,f_3) = \\ 0.5\frac{f_1(x_2^2-x_3^2)+f_2(x_3^2-x_1^2)+f_3(x_1^2-x_2^2)}{f_1(x_2-x_3)+f_2(x_3-x_1)+f_3(x_1-x_2)} \tag{5}$$

Using above results, one offspring is obtained by simplified quadratic optimization method like eq. 6.

$$p_{i,1}^k(t+1) = g(p_{i,1}^k, f(p_{i,1}), p_{i,2}^k, f(p_{i,2}), p_{i,3}^k, f(p_{i.3})) \tag{6}$$

where, $p_{i,j}^k$ is kth vector of individual j in species i and $f(p_{i,j})$ is $p_{i,j}$'s fitness value. This simplified method shows acceptable results for a range of numerical optimizaion problems with small dimensions, and easy to implement as a local search operator. In addition to the offspring candidate generated by above method, five offsprings are generated. One is generated by elitism and others are generated by standard evolutionary strategy (ES). With these six offsprings, next 3 parents are determined by 2-fold tournament selection.

The flow of the MmEA is as follows.

1. Initialize all species and individuals.
2. Generate six offsprings per species using local search and ES.
3. Select three individuals from six offsprings.
4. Repeat Step 2 and Step 3 T times. (T=5) (micro EA: Step 2, 3)
5. Calculate $W_{i,j}$ using eq. 4 to determine species to survive or to extinct.
6. Fill the vacant species by migration or random initialization. (macro EA: Step 5,6)
7. If the maximum number of generations is reached, then stop, else go to step 2.

3 Performance Comparison

Minimization experiments on the test suite, described in following Section 3.1, were carried out in order to determine the peformance of MmEA. During our experiments we compared the MmEA to GA and MA. The first is a GA, in which crossover is applied with probability 0.7 and the mutation rate is set to $1/l$, where l is the length of the bit-string. Also, MmEA is compared to the macroevolutionay algorithm of Marin *et al.*

3.1 Test Problems

The test suite that we have used for the experiments consists of eight test functions which can be seen as part of a standard test set [6].

1. Sphere: $\mathrm{f}_{sph}(x) = \sum_{i=1}^n x_i^2$
2. Rosenbrock's function: $f_{ros}(x) = \sum_{i=1}^{n-1}(100\cdot(x_{i+1}-x_i^2)^2+(x_i-1)^2)$
3. Step function: $\mathrm{f}_{step}(x) = \sum_{i=1}^n \lfloor |x_i|+0.5\rfloor^2$
4. Ellipsoid: $\mathrm{f}_{elli}(x) = \sum_{i=1}^n ix_i^2$
5. Schwefel's ridge: $f_{ridge}(x) = \sum_{i=1}^n(\sum_{j=1}^i x_j)^2$
6. Cigar: $\mathrm{f}_{cigar}(x) = x_1^2 + A\sum_{i=2}^n x_i^2$
7. Rastrigen's function: $f_{ras}(x) = \sum_{i=1}^n[x_i^2 - 10\cdot cos(2\pi\cdot x_i)+10]$
8. Ackley's function: $$f_{ack}(x) = -20exp(-0.2\sqrt{\tfrac{1}{n}\sum_{i=1}^n x_i^2}) \\ -exp(\tfrac{1}{n}\sum_{i=1}^n cos(2\pi x_i)) + 20 + e$$

3.2 Results

In our all experiments, results are averaged over 30 runs. Fig. 1 shows the effect of the simplified quadratic optimization method on sphere function. The local search operator shows good results for small dimensional problems, but the effectiveness decreases as the problem size increases. This origins from the correlation between paramters which is ignored in simplified quadratic optimization method. However, the local optimization method can be useful for lots of optimization problems with moderate complexity.

The improvement by local search opeartor is illustrated by comparing the MA and MmEA (Fig. 2). Table I summarize the comparison results on several test functions.

4 Summary

This paper presented the MmEA, an algorithm that mimics natural evolution of species and individuals. MmEA can be considered as a hierarchical parallel EAs with large number of subpopulation of small size which is characterized by hierarchical sturcture that mimics evolution of eco-systems. At population level, sharing mechanism between species controls global evolution processing and diversity of population based on the concept of MA. Within each species, both the local search operator inspired by numerical optimization methods and basic ES are used as an local evolutionary operators. By looking at the results, MmEA outperformed MA over a wide range of condition, but it docs not shows any improvement for complex problems. Application to large size problems would constitute a part of future investigation.

References

[1] E. Cantú-Paz, "A Survey of Parallel Genetic Algorithms," Univ. Illinois at Urbana-Champaign, Illinois Genetic Algorithms Lab., IlliGal Rep. 97003, 1997.

[2] D. E. Goldberg, em Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley, 1989.

[3] F. Herrera and M. Lozano, "Gradual Distributed Real-Coded Genetic Algorithms," *IEEE Trans. on Evolutionary Computation*, Vol. 4, No. 1, pp. 43-63, 2000.

[4] J. Marín and R. V. Solé, "Macroevolutionary Algorithms: A New Optimization Method on Fitness Landscapes," *IEEE Trans. on Evolutionary Computation*, Vol. 3, No. 4, pp. 272-286, 1999.

[5] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1994.

[6] R. Salomon, "Evolutionary Algorithms and Gradient Search: Similiarities and Differences," *IEEE Trans. on Evolutionary Computation*, Vol. 2, No. 2, pp. 45-55, 1998.

Table 1: Comparison Results (50000 evaluations)

Method	f_{Sph}	f_{Ros}	f_{Ridge}	f_{Ras}
GA	6.5e-19	6.5e-01	3.1e-14	2.5e+00
MA	3.7e-16	3.2e-01	1.4e-16	2.4e+00
MmEA	2.8e-21	2.4e-01	3.9e-17	2.4e+00

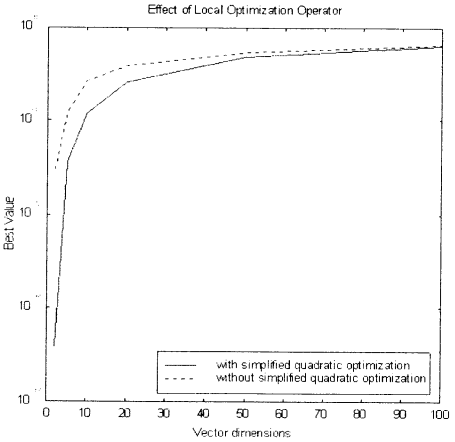


Figure 1: Effect of simplified quadratic optimization method in low-dimensional problems. (Minimization of sphere function during 100 generations)

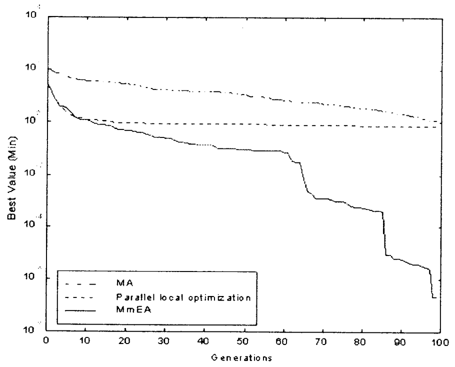


Figure 2: Comparison results of MA, MmEA, GA (f_{Ras})

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

DNA computing for shortest path problem

○ Nobuo Matsuura, Masahito Yamamoto, Toshikazu Shiba, and Azuma Ohuchi

Graduate School of Engineering
Hokkaido University
Sapporo 060-8628, Japan

E-mail address : {matsuura,masahito,shiba,ohuchi}@dna-comp.org

Abstract

DNA computing is one of computing paradigms based on biomolecules. In this field, L. Adleman presented the first research in 1994 [1]. Adleman was successful in solving a Hamiltonian path problem with 7 vertices by using some chemical reactions of DNA. Among them, the hybridization process is one of the most important reactions in Adleman's experiment, which generates many paths in a directed graph by annealing two single strands DNAs with Watson-Click complementary base pairing.

In this paper, we focus on the hybridization process because its feature has a great direct effect on the DNA computing process. We consider that defining initial DNA (input DNA) concentrations are the most critical factor for the computing process. Based on this idea, we aim to realize a new DNA computing model, showing that the concentration of target DNA can be controlled by regulating the input DNA concentrations. For the realization of this model, we propose a new encoding method based on the initial concentrations of input DNA, and we examined the possibility of quantitative analysis with the concentration. Next, we carried out laboratory experiments and simulations in order to verify that our DNA computing model is valid.

Keywords: DNA computing, chemical reaction, optimization problem, parallel computing

1 Introduction

Biomolecules (e.g., DNA, RNA, protein, and bacteria) exist in our bodies to maintain physical and metaphysical conditions. For this purpose, they have the ability to process the biological information in a cell or in an organ. In particular, it is important that DNA can process genetic information in the nucleus with a superior data processing system. The system has three main features. First is massive parallelism, which means that many DNA molecules react in parallel. Second is high efficiency, which means that DNA needs little energy to realize its chemical reaction. Third is massive storage, which means that DNA can store a massive amount of information in its double helix structure; for example, 1 g of DNA is equal to about 4,000 CD-ROMs.

DNA computing (molecular computation) is a computing paradigm based on the abovementioned features, for construction of a computer that transcends the traditional silicon-computer. In 1994, L.Adleman showed the possibility of realization of DNA computing for the first time [1]. Adleman was successful in solving a Hamiltonian path problem with 7 vertices by using a chemical reaction of DNA. Adleman's DNA computing model consists of three processes. The first process is the encoding of the problem to DNA. The vertices and edges of the Hamiltonian path are encoded in single strand DNA. The second process is hybridization, which generates many paths in the directed graph by fusing two single strands DNA with Watson-Click complementary base pairing. This process corresponds to the computing process. The third process is the detection of the target DNA path encoded as the Hamiltonian path in the graph. The methods for detection include PCR, gel electrophoresis, and affinity separation.

In this paper, we focus on the hybridization process since the hybridization process is one of the important processes in DNA computing. In particular, we consider that the input DNA concentrations is a critical parameter for annealing of DNA. Based on this idea, we propose a new DNA computing model and aim to realize this model. The model shows that concentration of the target DNA can be controlled by regulating the input DNA concentrations.

For the realization of our model, we apply the proposed model to the directed shortest path problem (SPP). First, we encode the costs (distances) of edges in the SPP into the input DNA concentrations. After the encoding process is finished, we perform the hybridization process, similar to that in Adleman's model. Then in the detection process, we find the shortest path (optimal or near-optimal solution) by analyzing the concentration of the resultant DNA paths.

Furthermore, in order to show the validity of our model, we analyze its results by using the hybridization simulator, which is a simulation model of the hybridization process based on the concentration dynamics model. We discuss the quantities of the resultant DNA paths.

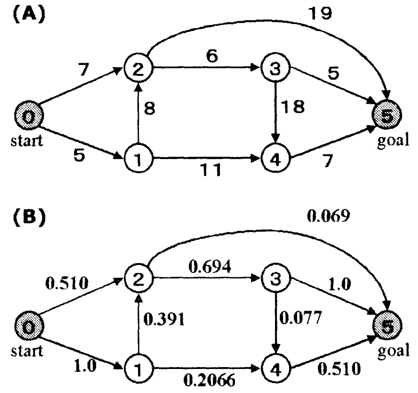


Figure 1: (A) The shortest path problem treated in this paper. (B) The problem coded by the encoding method based on the control of concentration.

2 DNA computing for shortest path problem

In this section, we present the new DNA computing model that enables the target DNA concentration to be controlled.

2.1 Encoding

In DNA computing, encoding is the process of translating a given problem we want to solve into DNA. In this paper, we are concerned with the shortest path problem with 6 vertices and 9 edges (Fig. 1 (A)). To encode the problem, we used a traditional encoding method, similar to Adleman's one. In this method, we must encode the information (e.g., vertices and edges) into nucleotide sequences or lengths of nucleotide sequences. Furthermore, we propose a new encoding method to encode the costs into the concentration of DNA edges. In our encoding method, the costs of one shortest path problem are encoded as follows:

$$D_{ij} = (Min/C_{ij})^{\alpha} \qquad (1)$$

where Min is the shortest cost among all edges in the graph, C_{ij} are the costs in each edge, and α is a parameter related to the relative concentration. The shortest path problem encoded by the above equation is shown in Fig. 1 (B):

2.2 Hybridization

The hybridization process is one of the most important chemical reactions of DNA computing. DNA has four types of bases (A, G, C, T), and a single strand DNA is composed by the combinations of these 4 bases. Annealing two single strands DNA each other based on Watson-Click complementary base pairing, is called *hybridization*, restricted that base A combines with base T and base G combines with base C. In our DNA computing model, the hybridization process functions as a computing process that generates many paths in the directed graph from the input DNA. Fig. 2 shows that DNA complexes representing a path in the directed graph are generated in the hybridization process.

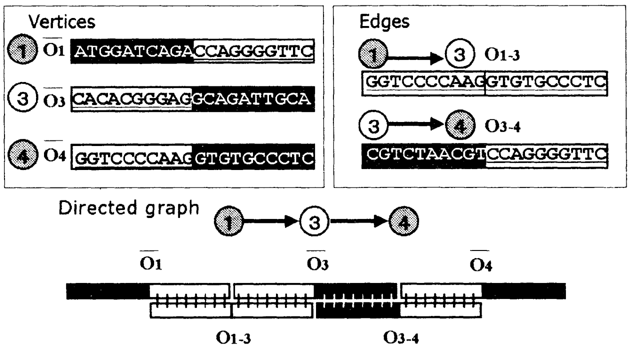


Figure 2: Design of vertices and edges, and the DNA complex representing the DNA path generated from them.

The concentrations of the resultant DNA complexes strongly depend on the input DNA concentrations.

Therefore, we try to control of the target DNA complex by altering the concentrations of initial DNAs. So we implemented the new encoding method mentioned above, in order to control the quantities of the resultant DNA complexes.

We treat a shortest path problem with 6 vertices and 9 edges (see Fig. 1), and there are 7 paths in the graph from the vertex 0 to the vertex 5. We applied our proposed DNA computing model to the graph shown in Fig. 1. In our computing model, the input DNA concentrations are calculated using equation (1). We then performed the hybridization process by using a mixture containing each single strand DNA. It is expected that more DNA paths corresponding to a shorter path in the graph will be generated than will other DNA paths. Therefore, we can find the shortest path by analyzing the concentrations of resultant DNA paths.

2.3 Detection

In order to select the DNA paths that begin with O_0 (start) and end with O_5 (goal), DNA amplification is performed by using polymerase chain reaction (PCR). PCR is a method that enables amplificaton of DNA contained specific sequence in a short time by using DNA polymerase, two primers, and fluctuation in temperature.

Next, we need to quantify in detail the concentrations of each DNA path amplified by PCR. To do this, amplified DNA paths are separated by polyacrylamide gel electrophoresis, and the DNA bands are visualized by ethidium bromide (Fig. 3). A photograph of the gel is taken, and the image is fed into a computer. By using software for analysis of DNA bands, we can analyze the histograms representing the concentrations of DNA path. As a result, we consider the band showing the highest peak to be the shortest path. To confirm the abovementioned feature, we perform the DNA sequence analysis of the most-intensive DNA path.

In order to verify that our DNA computing model is valid, we use a simulation model of the hybridization process proposed by Yamamoto et al. [2]. The simulation takes input DNA (e.g., DNA sequences, reaction velocity, and concentration) and calculates the possible DNA paths and their concentrations based on the concentration dynamics model.

3 Experiments

In this section, we show the validity of our DNA computing model by presenting results of both laboratory experiments and simulations. We then discuss the implementation of our model based on a comparison of the results of laboratory experiments and those of simulations.

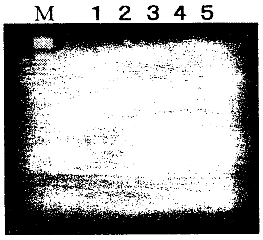


Figure 3: Confirmation of the formation of DNA paths. Gel electrophoresis was performed and the DNA bands were quantified by image analyzer. Lane M, DNA size marker (100 bp ladder); lanes 1-5, amplified DNA paths after 11 cycles of PCR; templates (hybridized DNA solution) were diluted as follows, lane 1 : 1, lane 2 : 1/5, lane 3 : 1/10, lane 4 : 1/20, lane 5 : 1/40.

3.1 Simulations

In order to determine whether the DNA path that has the highest concentration actually corresponds to the shortest path, we simulated our model using a simulator (Fig. 4).

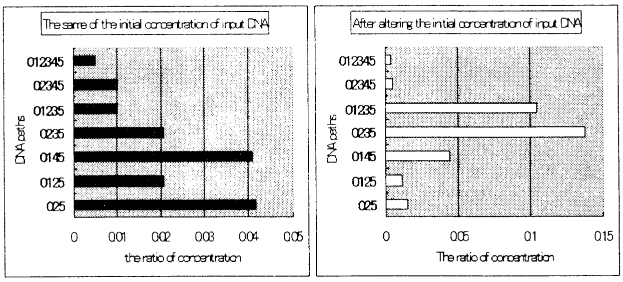


Figure 4: Comparison of the amounts of DNA paths from the simulation.

The graph on the left shows the results in the case in which the concentrations of all input DNA are the same, and the graph on the right shows the results in the case in which the concentrations of each input DNA are set to the values calculated by equation (1). These results, that the DNA path we designed to be the shortest path was generated with the highest concentration, showed the possibility of the control of concentration.

Next, in order to verify that equation (1) is effective for solving the directed shortest path problems, we performed some experiments using the simulator. In the same graph as Fig. 1, further 30 problems were generated by randomly assigning costs to all edges, the shortest path being calculated by the Dijkstra method. Comparing it with the solution obtained by using the simulator (DNA path with the highest concentration), we evaluated our computing model. We were able to find the shortest path by our model in 21 of the 30 problems (70 %). The results showed the validity of our model.

3.2 Laboratory Experiments

We compared the results of laboratory experiments with those of simulations. First, we used SSCP, which is a gel electrophoresis that enables separation of DNA complexes according to their size, base sequences, and base composition. We analyzed the concentrations of each DNA path by using SSCP and compared the results with the simulation results (Fig. 5).

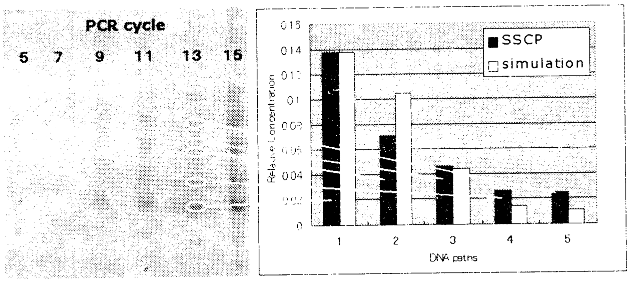


Figure 5: Comparison of the relative amounts of DNA paths obtained from an SSCP experiment and from simulation.

We found a definite dark band in the image of SSCP, which we assume is the DNA path representing the shortest path. We are currently confirming its DNA sequence. Next, in order to show the validity of our model by another experiment, we used a 6 % polyacrylamide gel electrophoresis that enables separation of DNA complexes according to their lengths. The comparison results are shown in Fig. 6.

The tendencies in both the results of laboratory experiments and those of simulations are similar. We therefore conclude that the control of concentrations of DNA paths can be achieved by using our new encoding method for solving the directed shortest path problems.

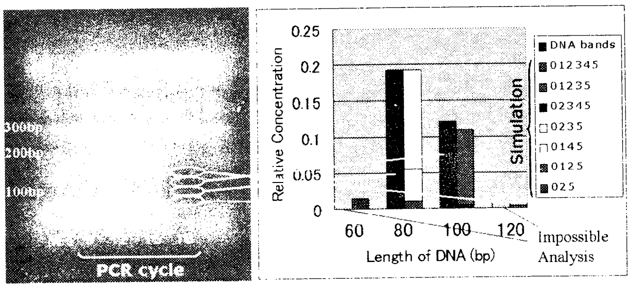


Figure 6: Comparison of the relative amounts of DNA paths obtained from 6 % gel electrophoresis and simulation.

4 Summary

In this paper, we focused on the feature related to the concentration of DNA in the hybridization process, and we proposed a new DNA computing model based on the concentration of DNA. Moreover, we presented a new encoding method that enables the concentration of target DNA to be controlled. In order to determine the validity of our DNA computing model and the reliability of our proposed encoding method, we performed laboratory experiments and simulations, and we compared their results. The results show the validity of our proposed DNA computing model based on concentration control.

References

[1] L. Adleman, "Molecular Computation of Solutions to Combinatorial Problems" *Science*, vol. 266, pp. 1021-1024, 1994.

[2] Masahito Yamamoto, Jin Yamashita, Toshikazu Shiba, Takuo Hirayama, Shigeharu Takiya, Keniji Suzuki, Masanobu Munekata, and Azuma Ohuchi, "A Study on the Hybridization Process in DNA Computing" *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, in printing, (2000).

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

The optimization of neural network structure using genetic algorithm

Masanori Sugisaka , Minoru Itou

Department of Electrical and Electronic Engineering
Oita University
700 Oaza Dannoharu , Oita , 870-1192, JAPAN
Tel: +81-97-554-7831
Fax: +81-97-554-7841
E-mail msugi@cc.oita-u.ac.jp , minoru@cc.oita-u.ac.jp

Abstract

In this paper, genetic algorithm (GA) is proposed to search for the optimal structure (i.e., the kind of neural network, the number of hidden neuron) of the neural network which are used approximating a given nonlinear function in the temperature control system in the steel plant. We consider one kind of neural network in this paper, that is general multi layer feed-forward neural network. The decision method of synapse weights of each neuron in each generation used the improved back-propagation method. In this study, we simulated nonlinear function approximation in the temperature control system.

1.Introduction

Neural network (NN) have been successfully implemented in various application fields in the last decade because of their parallel computation and complex nonlinear function mapping characteristics compared with conventional schemes. Neural networks also possess a learning ability which is the most important feature in its real applications.

The neural networks implemented in above applications may have different structures, may use the classical back-propagation training algorithm or other improved training algorithms to obtain the synapses weights of the neural networks, they all have the common features, i.e., the neural networks have the fixed structures. It means that we do not exactly know what kind of neural networks, the related training algorithms and how many inputs and hidden neurons will be mostly suitable to the special object.

To design the optimal architectures of neural network, e.g. attempt is implementing the evolutionally algorithms, e.g. evolutionaly programming (EP) and genetic algorithm.

In this paper, GA based algorithm is proposed to search for the optimal architecture of neural network which are used to approximate a given nonlinear function in the temperature control system.

2.Neural network structure and training algorithm

We mainly consider the general multi layer feed-forward neural network in this paper. The training algorithm is the improved back-propagation algorithm.

3.Genetic algorithm

3.1 Fitness function

The fitness of each fully trained network is calculated using Eq.1.

$$f = \frac{1}{E}\left\{1 + \alpha\left(1 - \frac{N_i}{N_{i\max}}\right) + \beta\left(1 - \frac{N_h}{N_{h\max}}\right)\right\}\cdots(1)$$

where f is the fitness of the neural network, E is the error of neural network, α and β are coefficients implying the influence of inputs number and hidden neuron number, N_{imax} and N_{hmax} is the maximum number of inputs and hidden neurons, respectively.

3.2 Encoding algorithm

The first step before using GA operators is encoding the neural networks into binary strings called chromosome. A chromosome's characteristic is determined by the genes which are presented by binary bits in this paper. There are numerous encoding algorithms in neural network optimization. Because we emphasis our research on the neural network architecture optimization, the encoded strings just contain the architecture information of neural network, i.e. the synapse weights of neural network will not be involved in the encoded strings. The job of obtaining suitable weights is fulfilled by using the neural network training algorithms. The general description of encoding method is shown by Fig.1.

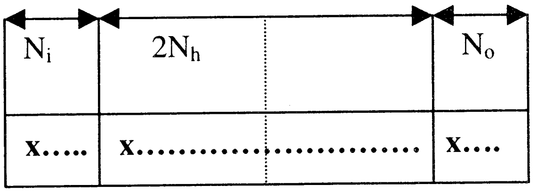


Fig.1 The chromosome presentation

Fig.1 shows one chromosome related to one neural network architecture, where x is one binary bit representing ‘0’ or ‘1’. The chromosome consists of four parts, i.e. input area, hidden layer area and output area, respectively. The length of each parts is N_i, N_h and N_o bits, respectively, where N_i, N_h and N_o is the number of inputs, hidden neurons and outputs of neural network, respectively.

In input area, one bit represents one input in input layer of neural network, with ‘1’representing the existence of the input and ‘0’representing the absence of the associated input. In hidden layer and output areas, one bits represent one neuron because we consider sigmoid function for each neuron, with ‘0’--- represents the absence of the neuron, ‘1’--- the neuron with sigmoid function, respectively.

Fig.2 shows one of chromosome generated by GA randomly, with the capacity of 4 inputs, 16 neurons in hidden layer, one neuron in output layer.

1110	**11011010**	**01011101**	**1**

Fig.2 One of chromosome of NN

3.3 GA operators

Three GA operators, i.e. reproduction, crossover and mutation are implemented one generation by generation to search for the optimal architecture of neural network.

3.3.1 Reproduction

One commonly used technique is roulette wheel reproduction. It can be regarded as allocating pie shaped slices on a roulette wheel to population members, with each slice proportional to the member's fitness. There are two drawbacks to this method: it is possible that some of the best individuals may not be reproduced at all, and thus their genes may be lost. Also, it is possible that the genetic operators alter the best chromosome's gene so that whatever was good about them is destroyed.

To improve the properties of roulette wheel reproduction, an alternative called steady state reproduction is used in this paper. In this method, P_r percent of the best individuals will just be copied into the new generation, while the remained ($100-P_r$) percent individuals will carry out the crossover and mutation process. This method removes the drawbacks in roulette wheel, the best individuals will always reproduce because they are simply copied into the new generation and their genes are not changed by GA operators. The factor P_r is set between 5%-10% to guarantee the number of remaining individuals large enough and to yield enough new individuals to compete in the new generation.

3.3.2 Crossover

Crossover is one of important operators in GA unlike the case in evolutionary programming, in which crossover is not implemented and only mutation is carried out.

There are different possible crossovers in GA literature, e.g. one-point crossover and two-point crossover. For one-point crossover, one part of the parent chromosomes are exchanged at the randomly selected point, another part is kept the same as before. The simplicity of one-point crossover makes it be widely used in GA operators, but its drawback is obvious, that is the possible searching space is limited because only one point is chosen. In this paper, we use the two-point crossover to over come the shortage of the previous one.

3.3.3 Mutation

Mutations are important for keeping a bit of wildness and random search flavor to optimization process. In this paper, we use the general random mutation method, that means each gene in a chromosome changes its value (called allele) from '0' to '1', or from '1' to '0' at the given probability P_m, where P_m is set to 0.003 in the following simulation.

4.Simulation

4.1 Training and testing data

The nonlinear function which will be approximated is multi input single output function. It is extracted from a temperature control system in a steel plant. The input and output are illustrated in Fig.3.where, x_1, x_2, x_3, x_4, x_5 are inputs, d is the output of the real system. There are 666 sets of input and output data as the real inputs and output sets. Two parts of these sets are separated, i.e. 333 training and 333 testing data. The training data is used to train the neural network and the testing data is used calculate the fitness of the same neural network to sort the networks and find the best one in one generation. Every two real data in time sequence compose of the training sets, with the remaining 333 real data compose the testing data. Each generated neural network by GA operator or randomly at start stage will be fully trained using the trained using the training data and them, its fitness will be calculated and the GA operators are applied to these chromosomes.

To consider the influence of neural network based on the training data and testing data, a combined fitness is employed in estimating the feature of evolved neural network, see Eq.2.

$$f = \frac{\mu_1 f_{train} + \mu_2 f_{test}}{2} \cdots (2)$$

where f_{train}, f_{test} is the fitness of neural network based on the training data and testing data, respectively. μ_1 and μ_2 are the influence coefficients for training and testing data, respectively.

4.2 Normalization

Data normalization is important for the neural network training. In this paper, we employ the general dimensionless normalization technique to pre-process the real data and form the training and testing data. The data normalization is performed by Eq.3.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \cdots (3)$$

where x', x, x_{min} and x_{max} is the variable after normalization, the original variable, the minimum of x and the maximum of x, respectively. After the data normalization, the value of data will be limited to the range from 0 to 1, referring to Fig.3.

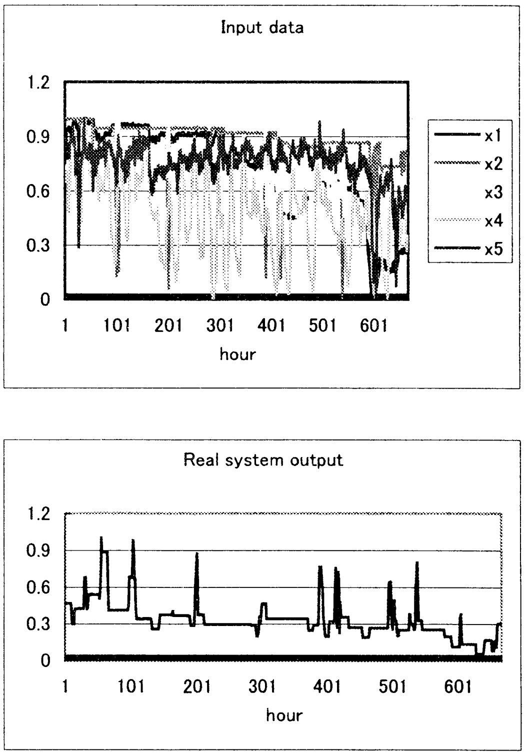


Fig.3 The nonlinear function

4.3 Simulation

The simulation was carried out using the real data illustrated by Fig.3.Each generated neural network by GA operator or randomly at start stage will be fully trained using the trained using the training data and them, its fitness will be calculated and the GA operators will be implemented to carry out the GA process.

The system parameters are set as following. Population size N_p=30, generation number N_g=100, the input factor α=0.025, the hidden neuron factor β =0.1, reproduction factor P_r=0.05, the mutation probability P_m=0.003, using two point crossover method, the maximum number of hidden neurons N_{hmax}=20, the maximum number of inputs N_{imax}=5, the initial random weights are limited between ± 0.5, the learning rate is 0.1, the momentum coefficient is 0.08. The final results are shown by Fig.4.

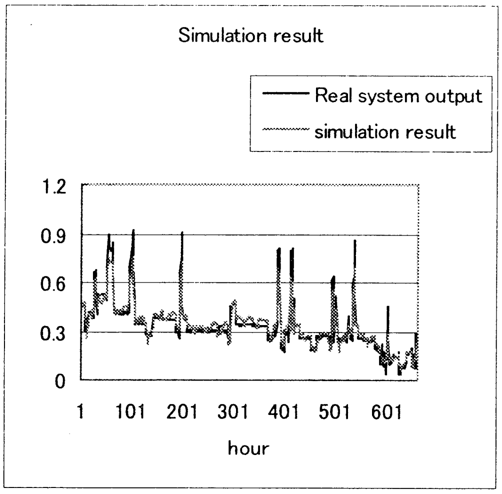


Fig.4 The final result

5.Conclusion

We made some attempts to automatically search for the optimal architecture of neural networks employing genetic algorithm. The simulation results show that the presented method is suitable in nonlinear function approximation application. In the near future, we will add other kinds of neural networks and training algorithms to expand the searching ability of this method.

Also, we are considering to use this method to other applications.

References

[1]M.Sugisaka and Z.Liu (2000) "Optimization of Neural Network Structure Based on A Hybrid Method" Proc. of the fifth Int. on Artificial Life and robotics, vol.1, pp212-215

[2]M.Sugisak and Z.Liu (1999) "Neural Network Architecture Optimization and Application Using Genetic Algorithm" Proc. of The Forth Int. on Artificial Life and Robotics, vol.2, pp767-770

[3]M.Sugisaka and Z.Liu (1999) "A Genetic Algorithm Approach Used to Generate the Neural Network Structures" Proc.of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.2, pp763-768

[4]M.Sugisaka and Z.Liu (1998) "Artificial Neural Network and Application in Temperature Control System" Proc. of the 13th Korea Automatic Control Conference, pp260-264

[5]M.Sugisaka and M.Itou (2000) "The nonlinear function approximation based on the neural network application" Proc. of the Korea Automatic Control Conference 2000

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Self-learning Probabilistic Neural Network Hardware Using Reconfiguable LSIs

○ Noriyuki Aibe*, Moritoshi Yasunaga†, Ikuo Yoshihara‡

*Graduate School of Sciences and Electronics, University of Tsukuba, Tsukuba,Ibaraki.305-8573 Japan.
†Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba,Ibaraki.305-8573 Japan.
‡Faculty of Engineering, University of Miyazaki, Miyazaki,Miyazaki.889-2192 Japan.
E-mail: *susu@bforc.co.jp, †yasunaga@is.tsukuba.ac.jp

Key words: Pattern Recognition, Image processing, Neural Networks, PNN, FPGA, Reconfigurable System.

Abstract

The probabilistic neural networks (PNN) is one of the promising neural networks because of its theoretical background of the Bayesian statistics and rather simple learning algorithm. However, it requires long computational time both in the recognition and the learning process. Especially, there is no generalized algorithm to choose the network parameter (kernel size) in the learning phase. Therefore, the learning must be repeated changing the parameter within a wide range. The purpose of this paper is to propose novel neural network hardware based on the PNN, and the goal of our project is to demonstrate the high performance of newly proposed hardware under the real world applications.

1 Introduction

The pattern recognition/classification is widely used in various applications such as security systems, criminal investigation, biology, automatic motion control, computer vision, etc. The probabilistic neural networks (PNN) is one of the promising neural networks and is expected to achieve high recognition accuracy for those practical applications because the PNN is based on the Bayesian statistics. However, a large number of sample patterns are required for each category, thus, some applications containing many categories such as fingerprint recognition or face recognition cannot be finished within the required time. Furthermore, in the PNN, there is no generalized algorithm to choose the learning parameter (i.e., kernel size). Therefore, a set of learning procedure using all sample patterns must be repeated changing the parameter within a wide range to choose the best one.

Because of this background, specialized hardware of the PNN both for the high speed recognition and for the high speed learning is required. In order to meet this requirement, we propose newly PNN hardware with a built-in learning circuit. Moreover, reconfiguable LSIs such as FPGAs or CPLDs are effectively used in order to design the optimized circuits for each application in terms of the computational precision. Because of this flexibility, we can choose the optimal design for each application in the trade-off between the circuit complexity (computational precision) and recognition accuracy.

2 Probabilistic Neural Network

The probabilistic neural network(PNN) is one of the promising neural networks, and its hardware implementation is expected much easier than the backpropagation and other widely used neural networks[1] [2] [3]. The PNN is one of the non-parametric clas-

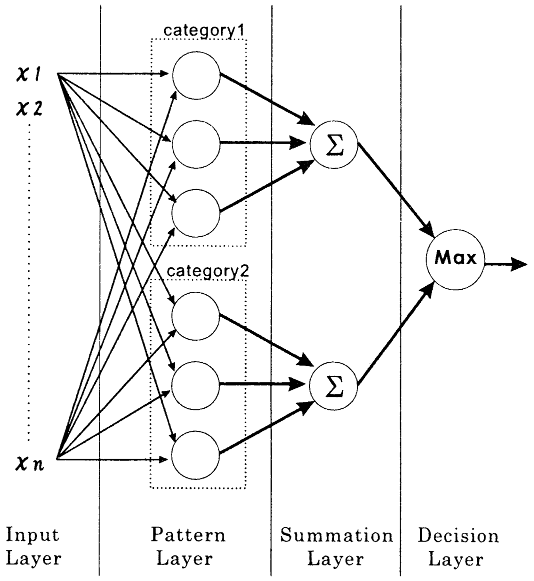


Figure 1: Structure of Probabilistic Neural Network.

sification algorithms based on the Bayes classification rule. One neuron corresponds to a kernel function the center of which is at each sample pattern. In the network, superposition (, or summation) of the neurons' outputs in the same category forms the conditional probabilistic distribution of each category, so that a large number of sample patterns (, or neurons) are required to estimate the distribution precisely. The recognition accuracy thus tightly depends on the number of sample patterns, that is, the number of neurons. The PNN's structure is shown in Fig.1. Fig.1 shows an example using only 2 categories in the pattern layer. In the practical applications much more than hundred categories are frequently used. However, the same structure shown in Fig.1 in universally used no matter how the number of categories increases. The are unknown pattern(x1,x2,...,xn) at the input layer, is input to all neurons at the pattern layer. The Pattern layer consists of groups of neurons for each category. The kernel size which the neurons represents is the distance from the sample pattern to the border of the kernel function. The kernel size is the only parameter in this classification, and it is decided in the learning phase. The summation layer sums up the number of the neurons into which the unknown pattern falls. Finally, the detector in the decision layer picks up the category the summation of which is the largest and it is estimated as the category of the unknown pattern.

3 System Configuration

The PNN hardware architecture is shown in Fig.2. The unknown pattern is compared with the sample patterns in the node processor. The node processor outputs dégree of similarity with binary number of 24-bit precision, where each set of 8-bit in 24-bit is sent sequentially to the max detector. Then, the max detector picks up the node processor outputting the largest number, and the category of the node processor is chosen as the unknown pattern's category. The central calculation in the PNN is the superposition of the kernel functions which are regarded as neurons in the PNN. The kernel function, or neuron is defined to each learning datum (sample patterns) with a kernel size σ (Fig.3). The learning phase in the PNN is to determine the optimal kernel size σ previously before the recognition phase. The best σ detector(Fig.2) detects the σ with which the highest accuracy of recognition is achieved by using the max detector's output, test patterns' categories, changing the kernel size σ. This calcuration is based on the leave-one-out method. It uses only sample patterns in the memory, and takes out a test pattern from sample patterns, and calculates the recognition accuracy. Then, another set of sample patterns and a test pattern is used. This procedure is repeated changing the combination of sample and test patterns within a wide range of σ. The Multiplexer (MUX) in Fig.1 is switched to S0 at the learning phase, and the 4-bit binary counter generates the variable σ from 0 to 255. In the recognition phase, MUX is switched to S1, and each node processor uses the detected best σ.

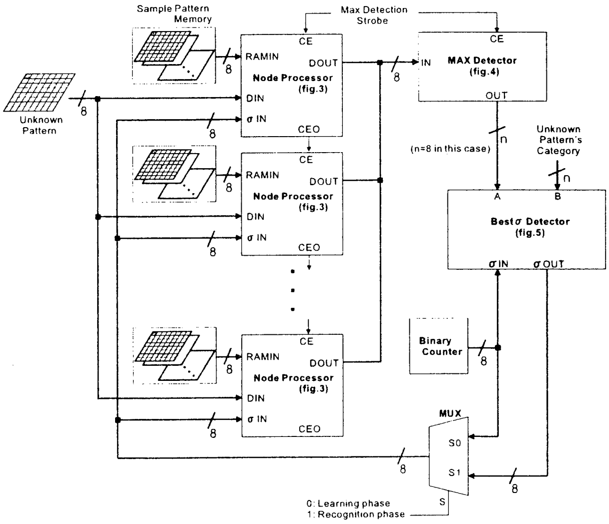


Figure 2: PNN Hardware Configuration.

Node Processors:

The node processor is shown Fig.3. The $\sigma/2$ is added to and subtracted from an pixel's brightness in each sample pattern from the memory in the kernel size adjuster. Then, it is compared with the corresponding pixel's brightness in the unknown pattern in the kernel size comparator. The kernel size comparator outputs "1" if the unknown pattern is inside the kernel, while "0" if not. Then, the counter in Fig.3 counts the number of "1"s for each category. The shift register converts 24-bit to the sequential three sets of 8-bit, and its output is connected to the external bus through a try-state buffer.

Max Detector:

The category having the largest count is chosen by the max detector(Fig.4). The each data sent from the node processors is reconstructed to 24-bit data in the shift register, and compared with each other. The index number to the category of the largest one is stored in the output register in Fig.4. In the recognition phase, this register's output is the answer of recognition.

Best σ Detector:

In the learning phase, the best σ detector(Fig.5) de-

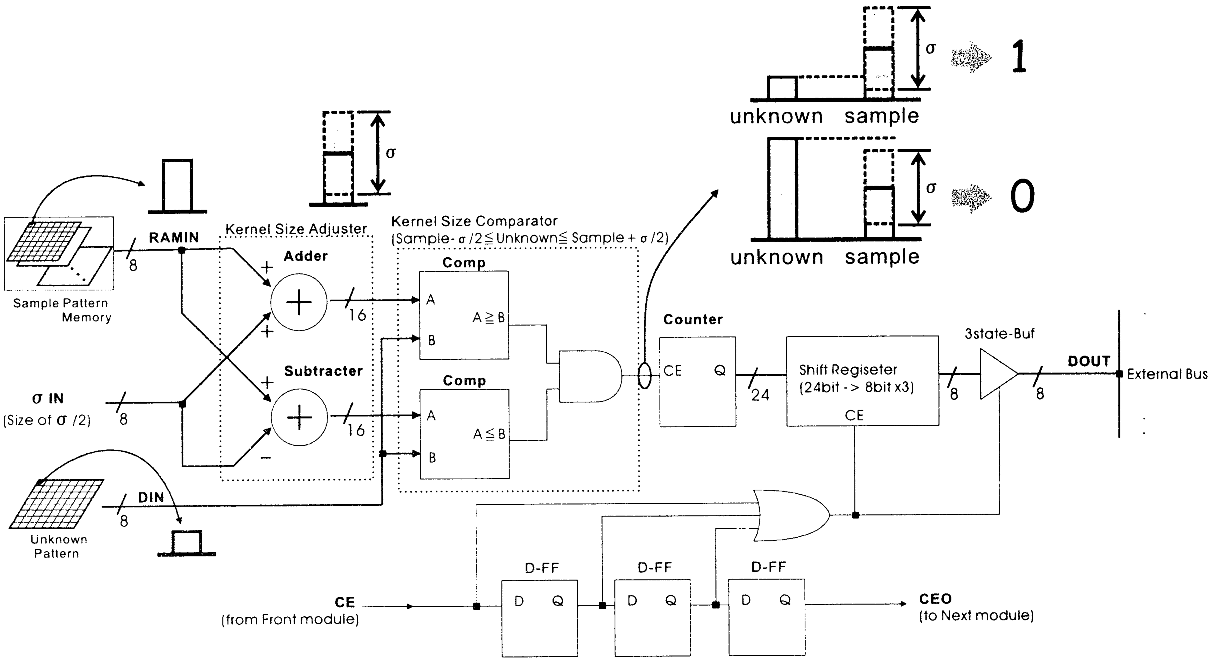


Figure 3: PNN Node Processor

tects σ which achieves the highest recognition accuracy. The predicted category output from the max detector is input to A while its corresponding answer is input to B. The binary counter counts how often A matches B. This number of times is compared with the previous one, and this comparison goes on sequentially changing σ little by little. Finally, the best one is stored in the register. The recognition accuracy depends on the kernel size σ considerably, thus, the kernel size needs to be chosen precisely within the large search space in the learning phase.

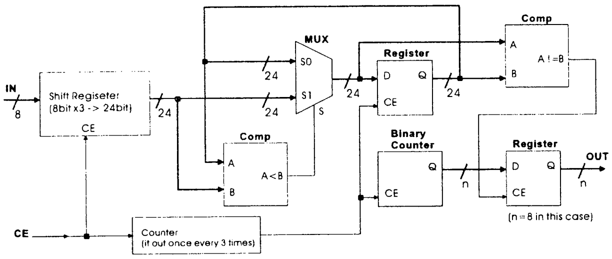


Figure 4: Max detector

4 Implementation

The photograph of hardware prototype is shown in Fig.6. One FPGA chip(Xilinx XCS30XL) contains 8 node processors (neurons), and 8 memory chips (4Mbit SRAM chips with 25ns access time) are connected to the FPGA. The current base clock frequency is 25MHz. The speed is especially limited by the memories' read access time. The max detector and the best σ detector are implemented onto another FPGA that is located on the end of external bus, and it contains 8 node processors too. This architecture also has high scalability, because it needs a few control lines(such as clk, CE, etc.) and only four common bus(Fig.2): the unknown pattern's input, the σ's input, the node processor's output, and memory address bus(it's not shown in Fig.2). And, a slight increment of the number of pins does not impede the scalability. Therefore, it is possible to connect the node processors in parallel until the clock delay bounds.

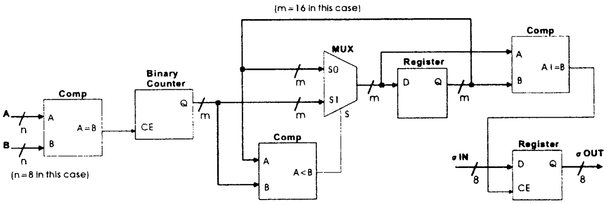


Figure 5: Best σ Detector

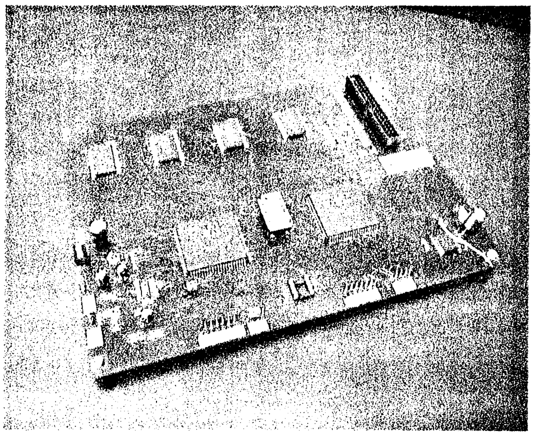


Figure 6: Photograph of Prototype Board.

5 Theoretical Estimation of Performance and Simulation

The part of the circuits are implemented onto FPGAs. Thus, the processors and their peripheral circuits can be designed flexibly and optimally. On the other hand, external memories' capacity, band width, and base clock speed are fixed. Therefore this system's performance is limited by these properties. For example, if the unknown picture and sample patterns size are 16 × 16pixels, and each pixel's depth is 8-bit, the one picture's total size is computed to $16 \times 16 \times 8 = 2048$bits. If the memory can be read with 30MHz, and the band-width to it is 8-bit, it is possible to read about 3,900 pictures at 33ms. Thus, the total capacity is counted to $2048 \times 3900 \simeq 1MB$, therefore, by using the memory of 1MB capacity, 8-bit data bus, and read access time shorter than 33ns it is possible to process 3,900 sample pictures in 33ms. And, it is possible to execute it in parallel, in realizing high scalability. In the case of a picture size is 16 × 16pixels, and each pixel's depth is 8-bit, one picture has $16 \times 16 \times 8 = 2048$bits. The unknown pattern and the all sample patterns have this size. These pictures are made generated from original pictures taken by the NTSC video camera through 8-bit A/D converter and a preprocessor. The 30,000 sample pictures are compared with one unknown pattern, and the system can process them in about 33ms. The same process on the PC(PentiumII-400MHz,on Linux with gcc) requires about 1~2s. By using the proposed hardware, it is possible to process pictures at the real-time video speed(≃33ms).

Memory capacity[MB]	pictures
1	3,906
2	7,812
4	15,625
8	31,250
16	62,500
32	125,000
64	250,000

Table 1: Scalability

6 Conclusions

We have proposed a novel hardware for the probabilistic neural networks. The hardware contains self-learning circuits to choose the best parameter. Therefore, it can accelerate not only the recognition speed but the learning speed. By using FPGAs with memory chips connected to them to implement the hardware, the optimal design in the trade-off between the circuit complexity (computational precision) and recognition accuracy.

Acknowledgements

This research is supported in part by the 1999 and 2000 grand of the Ministry of Education, Science, Sports and Culture is Japan (Grant No.10450131, No.12044204). The authors would like to thank N.Kadoya for his help with the benchmark programs on PC and all members of the ISLab. for helpful discussions.

References

[1] D.F.Specht, "Probabilistic Neural Networks," *J,Neural Networks*,Vol. 3, pp. 109-118,1990.

[2] D.F.Specht, "Probabilistic Neural Networks and the Polynomial Adeline as Complementary Techniques for Classification," *IEEE Trans. Neural Networks*,Vol. 1, No.1, pp.111-110 1990.

[3] Bin Tian, Mahmood R.Azimi-Sadjadi, Thomas H.Vonder Haar, and Donald Reinke, "Temporal Updating Scheme for Probabilistic Neural Network with Application on Satellite Cloud Classification," *IEEE Trans. Neural Networks*,Vol. 11, No.4, pp.903-920 2000.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

New Camera Calibration Method for Stereo Vision using Neural Network

Yousun Kang[o] and Yoonjeong Shin
Division of Computer, Electronic & Communication Engineering, Kwangju University
592-1 Jinwoldong, Namgu, Kwangju, Korea, 503-703
e-mail : {yskang, yjsin} @hosim.kwangju.ac.kr

Namjoong Kim
Dept. of Computer Science, Chosun College of Science & Technology
290 Seuseukdong, Donggu, Kwangju, Korea, 501-744
e-mail : njkim@mail.chosun-c.ac.kr

Abstract

In this paper, we propose a new method of camera calibration for stereo vision using neural network. Multilayer perceptrons of neural network are developed to transform the coordinates of each image point to the real world coordinates. Experimental results, the new method of calibration for stereo vision is theoretically and experimentally proven to be viable for depth extraction with stereo images and it is shown to be efficient, accurate, and simple to implement in camera calibration for stereo vision.

KeyWord : Camera calibraton, Stereo vision, Multilayer perceptrons

1 Introduction

Camera calibration is the process of determining the coordinate relationship between a camera image and its real world space. Accurate calibration of a camera is necessary for the applications that involve quantitative measurement of camera images. Camera calibration in the context of three dimensional (3D) machine vision is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and the 3D position and orientation of camera frame relative to a certain world coordinate system (extrinsic parameters), for the inferring 3D information from computer image coordinates and inferring 2D computer image coordinates from 3D information[5].

It is difficult to calibration that complex camera model to consider of constraints and nonlinear optimization. In addition, Tsai algorithm need that the calibration plate is sufficiently tilted with respect to the image plane (at least 30). So we need to new method of camera calibration that regardless of tilt of image plane and external, internal parameter of camera model. We intended to extract depth information (3D World Coordinate) from 2D image plane. But we are impossible to extract from only one image plane. So we need to a pair of image with stereo vision.

Calculating the distance of various points in the scene relative to the position of the camera is one of the important tasks for a computer vision system. A common method for extracting such depth information form intensity images is to acquire a pair of images using two cameras displaced from each other by a known distance. Two images of the same scene taken from a slightly different view-point convey information about the distance of the object to the camera pair.

The displacement between the locations of the two features in the image plane is called the *disparity*. 3-D information can be estimated indirectly from 2-D intensity image using principles of triangulation. The depth at various scene points may be recovered by knowing the disparities of corresponding image points. Thus if we have a pair of image with stereo vision, we can extract 3D depth information from stereo image with disparity.

We propose a new method of stereo camera calibration using neural network that have input to stereo images and output to 3D depth information. The result shows that the new method is proven to be efficient, simple to implement in calibration of stereo camera .

2 Camera calibration using neural network

In this section, we describe the structure of proposed neural network and learning mode and testing mode in The study showed that neural networks were able to model these simple calibration of stereo vision, illustrating the feasibility of using neural networks to perform calibrated object extracting of 3D depth information on stereo vision.

2. 1 Structure of neural network

Main concept of proposed camera calibration algorithm is based upon the multilayer percpetrons. Multilayer perceptrons are developed to transform the coordinates of each image point to the world coordinates. It has two hidden layer and input layer and output layer.

The structure of neural network is given in Fig. 1. Input layer has four node and output layer has three node and hidden layer has four node per one layer. These node is connected in each layer entirely. Transfer function is used to sigmoid function in first hidden layer, and used to linear function last hidden layer.

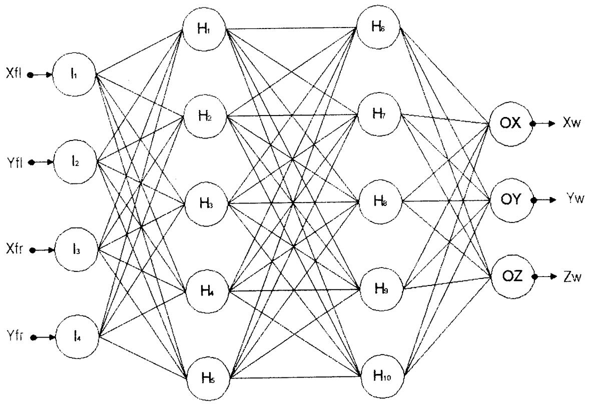


Fig. 1. structure of multilayer perceptrons for stereo camera calibration

In Fig. 1, (x_{fl}, y_{fl}) of input layer means pixel value of target points in left image and (x_{fr}, y_{fr}) means pixel value of target points in right image. (x_w, y_w, z_w) of output layer output 3D world coordinate in pixel value of target points.

2. 2 Training and testing the neural network

We have presented the new method of calibration using neural network that is build to learning mode and testing mode. The network is provided with a set of pixel value of a stereo pair of input images and corresponding output pixel value of target points.

First, learning mode (training mode) procedure for determining the weight factors. The weight factor produce correct real world coordinate to response for a given set of pixel value of target points. We use error backpropagation learning algorithm that correct weight factors using pixel value of target point value.

After the network is trained to give the desired 3D world coordinate provided with the pixel value of a stereo pair of input images, its performance with tested. A new target point of input image is provided and the output is compared with the desired 3D world coordinate (x_w, y_w, z_w).

2.3 divide boundary with lens distortion

In training mode, the input pixel value of target point is divided boundary to lens distortion. Lens distortions include two components: radial distortion and tangential distortion. Lens distortion can be measured by $D_x = k(r^2)$ as Fig. 2. Radial distortion bends the light rays by more or less than the correct amount. We measured change of radial distortion of distance from image center. D_x is radial distortion and r is distance from center of input image.

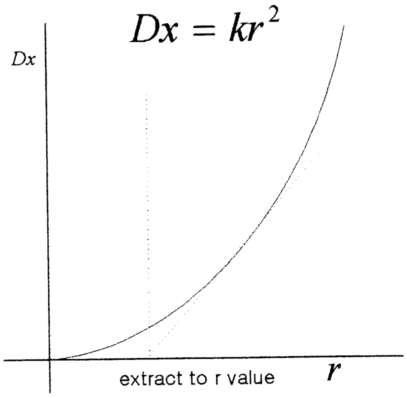


Fig. 2. Extract r value from image center

Fig. 3 shows that the target points of input image are classified with radial distortion from image center. Classified data is trained the classified neural network in each one.

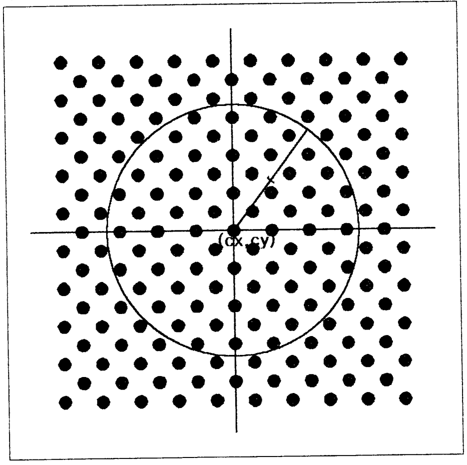
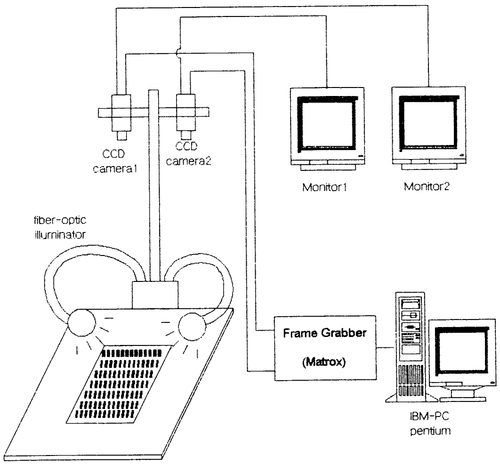


Fig. 3. Data classification with lens distortion of input image

3 Experiment of camera calibration

In this section, we will describe the procedure and analyze the result of two different tests of the two-stage camera calibration: 1) single-plane calibration : calibrate with no transformation of z axis, 2) multiplane calibration : calibrate with transformation of z axis. Fig. 4 shows that overall of system.



CCD camera : PULNIX TM-7AS
Lens : 12mm lens × 2
Frame grabber Matrox Meteor Board

Fig. 4. Configuration of overall system

3.1 Single plane calibration

The validity of the approach is tested with calibration target points which is covered the whole 2D space concerned in coplanar. The center of the 180 circles are treated as calibration point, 100 points were used in learning data and 81 point were used in test data of neural network. Diameter of target points is 2mm and interval of target points is 6mm.

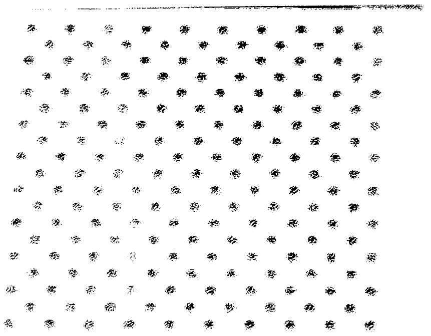


Fig. 5. right input image before image processing

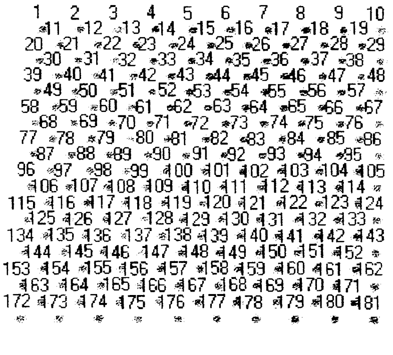


Fig. 6. right input image after image processing

The stereo pair of input image which is processed various image processing is showed in Fig. 6. Input image is processed in threshold that is a method to convert a gray scale image into a binary image so that target points are separated from background. A blob analysis algorithm finds all target points in an input image and a labeling algorithm assigns a unique label to all target points for correspondence in the stereo pair of images.

Note that error of calibration points is comparable to Tsai's camera calibration in Fig. 7 and they are giver in the following :

$$error_i = \sqrt{(x_{wi} - Ox_i)^2 + (y_{wi} - Oy_i)^2}$$

: calibration error of target point i

(Ox_i, Oy_i)

: output value of mutil layer perceptron

$Xe_i = |x_{wi} - Ox_i \ |$

: error of target point i in x axis

$Ye_i = |y_{wi} - Oy_i \ |$

: error of target point i in y axis

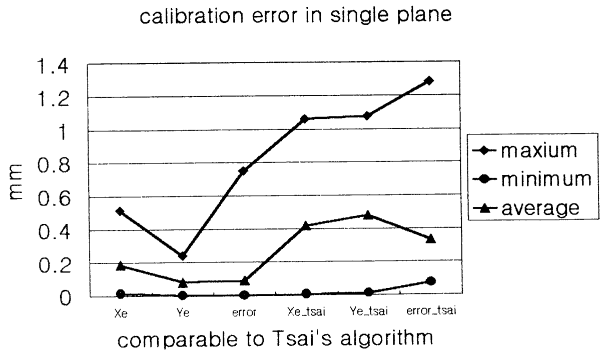


Fig. 7. error of calibration points is comparable to Tsai's camera calibration

Using neural network			
Error	Xe	Ye	error
Max	0.514	0.2400	0.752
Min	0.015	0.0002	0.273
Average	0.187	0.0805	0.089

Table 1. calibration error using neural network

As the result, the experiment is provide error of calibration in output of multilayer perceptrons within 1.0mm as shown in Table 1 but, in case of Tsai's method with intenal parameter and external parameter in camera model, the following error table is obtained.

Tsai's calibration method			
Error	Xe	Ye	error
Max	1.059	1.077	1.285
Min	0.006	0.012	0.072
Average	0.418	0.480	0.333

Table 2. calibration error using Tsai's method

3.2 multiplane calibration

In multiplane case, Zw of output pixel value is changed as input image which has 3D depth information to differ in z axis. This means that we have to train the neural network with Zw set of pixel value of a stereo pair of input images.

The neural network is trained for determining the depth information. When error The same pattern used in multiplane case can be used, except that Zw of output value is changed. To obtain error of depth information, the following equation are necessary :

$$error_i = \sqrt{(x_{wi} - Ox_i)^2 + (y_{wi} - Oy_i)^2 + (z_{wi} - Oz_i)^2}$$

: calibration error of target point i

(Ox_i, Oy_i, Oz_i)

: output value of mutil layer perceptron

$Xe_i = |x_{wi} - Ox_i|$

: error of target point i in x axis

$Ye_i = |y_{wi} - Oy_i|$

: error of target point i in y axis

$Ze_i = |z_{wi} - Oz_i|$

: error of target point i in z axis

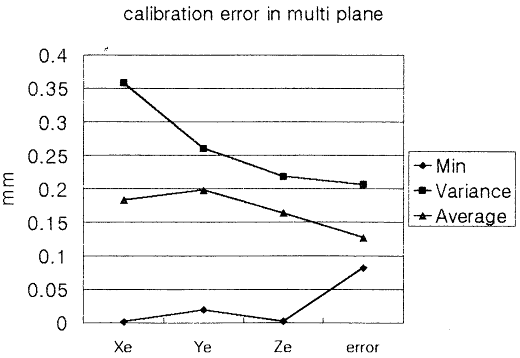


Fig. 8. error of calibration in multiplane

Using neural network in multiplane				
Error	Xe	Ye	Ze	error
Min	0.00104	0.01911	0.001446	0.08202
Variance	0.35823	0.26075	0.218340	0.206330
Average	0.18322	0.19814	0.163921	0.127123

Table 3. calibration error in multiplane

The error in case of multiplane is shown in Fig. 8, and in Table 3. That has minimum of calibration error and variance and average of error to extract depth information and the output from trained neural network data is available.

4 Conclusion

In this paper, the new method of calibration is experimentally proven to be viable for stereo vision. The contribution of this work can be summarized as follows:

- We have proposed new method of camera calibration for a stereo pair of images using neural network. The network can be trained in target points of 2D input image and tested to output target point with 3D world coordinate.

- The network can be applied in single plane with same value of z axis and multiplane with difference value of z axis. This method is to be used when the input images of stereo camera exist in the same plane(coplanar).

- The advantage of proposed calibration method are versatile-stereo camera can be used for a variety of automation applications, availability-regardless of tilt of image plane and calibration plate, simple-we have no need to be solved extrinsic and intrinsic parameter included camera model.

Experimental results show that, 3D world coordinate computed from multilayer perceptron in neural network. The new technique is theoretically and experimentally proven to be viable for stero vision and it is shown to be efficient, accurate, and simple to implement in camera calibration.

Reference

[1] W. E. L. Grimson (1985) Computational Experiments with a Feature Based Stereo Algorithm. IEEE Trans. on Pattern Analysis and Machine Intelligence. 7(1):17-34

[2] J. Weng, P. Cohen and M. Herniou (1992) Camera Calibration with Distortion Models and Accuracy Evauation. IEEE Trans. on Pattern Analysis and Machine Intelligence. 14(10):965-980

[3] Rumelhart, D. E., Hinton. G. E., and Williams. R. J (1986) Learning internal representation by error backpropagation. Parallel Distributed Processing. 1: 318-362

[4] Y. C. Kin and J. K. Aggarwal (1987) Positioning Three-Dimensional Objects Using Stereo Images. IEEE Journal of Robotics and Automation. 3(4): 361-373

[5] Roger Y. Tsai (1987) A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Journal of robotics automation. RA-3(4): 323-344

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Structure Minimization using Impact Factor in Neural Networks

Kap-Ho Seo, Jae-Su Song, and Ju-Jang Lee

Dept. of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology,
373-1, Kusong-dong, Yusong-gu, Taejon, 305-701, Korea
Tel: 82-42-859-3432/5432; Fax: 82-42-869-3410;
E-mail: khseo@odyssey.kaist.ac.kr jssong@odyssey.kaist.ac.kr jjlee@ee.kaist.ac.kr

Abstract

The problem of determining the proper size of an neural network is recognized to be crucial, especially for its practical implications in such important issues as learning and generalization. Unfortunately, it usually is not obvious what size is best; a system that is too small will not be able to learn the data while one that is just big enough may learn very slowly and be very sensitive to initial conditions and learning parameters. One popular technique is commonly known as pruning and consists of training a larger than necessary network and then removing unnecessary weights/nodes. In this paper, a new pruning mcthod is developed, based on the penalty-term methods. This method makes the neural networks good for the generalization and reduces the retraining time after pruning weights/nodes.

1. Introduction

Despite many advances, for neural networks to find general applicability in real-world problems, several questions must still be answered. One such open question involves determining the most appropriate network size for solving a specific task. There is dilemma that stems from the fact that both large and small networks exhibit a number of advantages. When a network has too many free parameters, not only is learning fast, but local minima are more easily avoided. Large networks can also form as complex decision regions as the problem requires and should exhibit a certain degree of fault tolerance under damage conditions. On the other hand, both theory and experience show that networks with few free parameters exhibit a better generalization performance, and this is explained by recalling the analogy between neural network learning and curve fitting. Moreover, knowledge embedded in small trained networks is easier to interpret and thus the extraction of simple rules can hopefully be facilitated.

To solve the problem of choosing the right size network, two different incremental approaches are often pursued. The first starts with a small initial network and gradually adds new hidden units or layers until learning takes place. Well-known examples of such growing algorithms are cascade correlation and others. The second, referred to as pruning, starts with a large network and excises unnecessary weights and/or units. This approach combines the advantages of training large networks (i.e., learning speed and avoidance of local minima) and those of running small ones (i.e., improved generalization). However it requires advance knowledge of what size is "large" for the problem at hand, but this is not a serious concern as upper bounds on the number of hidden units have been established. Among pruning algorithms there are methods that reduce the excess weights/nodes during the training process, such as penalty term methods and the gain competition technique.

Various techniques such as optimal brain surgeon (OBS) and optimal brain damage (OBD) have been proposed in literature to prune a fully connected feedforward artificial neural network (FANN). These techniques, however, require considerable additional computation as they require calculation of the Hessian matrix of the system. These post-training pruning procedures do not interfere with the learning process, but they usually require some retraining to maintain the performance of the original network.

In this paper, a novel post-training pruning method for arbitrary feedforward networks is proposed, which aims to select the optimal size by gradually reducing a large trained network. The method is based on the simple idea of calculating sensitivity term called impact factor (IMF) in order to make the output of the each neuron over the all training input have the same value or low variance. So the output of the neuron that has the low variance can be constant value. We can easily prune this neuron or weight. IMF can easily select the unnecessary weight to be pruned and reduce retraining effort to maintain the performance of the original network. The connections that have the smallest IMF will be pruned and the FANN will be retrained. The IMF's are then recalculated and the weight connections corresponding to the lowest IMF will be pruned again. The whole process of "training - IMF calculation - pruning" will be continued until no further improvement in the FANN model is noticed, i.e., no further decrease of global um squared error (SSE).
We define the concept of IMF in the second section. And the section 3 introduces the algorithm. The related equations are scripted and the detail explanation is added. In section 4, the computer simulation results are included. In last section, we discuss and conclude paper.

2. Impact Factor

As mentioned in section one, the present studies are based on a two layer FANN with nonlinear sigmoidal activation function in the hidden layer neurons and linear activation function output. The connections between the input and the hidden layer are represented by "hidden layer weights" and the connections between the hidden layer and the output layer are represented by "Output Layer Weights."

2.1 Definition of IMF

Most significant component to determine the importance of synaptic weight is the magnitude of weight. And the component of determining the importance of the neuron using IMF method is the variation of the neuron's output. If the output of some neuron varies largely for the input patterns compared with other neurons, we can rightly think that this neuron is a significant one for this network. On the contrary, if varies insignificantly, we can think that this neuron is not important one. So we also think that this neuron's output is constant and this neuron can be transmuted into next layer's bias term and the weights related to this neuron's output layer could be pruned.

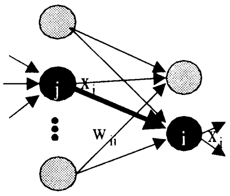


Fig. 1 Single-Layer Perceptrons

Like Fig. 1, the output equation of neuron *i* is described as follows

$$\begin{aligned} x_i &= f\left(\sum_j w_{ij}x_j + w_{j0}\right) \\ &= f\left(\sum_j w_{ij}(x_j - \bar{x}_j) + \sum_j w_{ij}\bar{x}_j + w_{j0}\right) \end{aligned} \qquad (1)$$

Where x_i is the output vector of *i*th layer, $f(\bullet)$ is activation function, w_{ij} is the weight vector between *i*th layer and *j*th layer, w_{j0} is the bias vector of *j*th layer, $\bar{x}_j$ is the average output value of *j*th layer

In the above nonlinear function $f(\bullet)$, the second and third term is constant. So the value of the *i*th neuron's output is determined by the first term that is calculated by the product of the weight and the deviation of the *j*th neuron's output. As the same reason, the importance of a synapse's weight, IMF is defined as follows

$$IMF = w_{ij}^2 \times \sigma \qquad (2)$$

where σ is variance.

Using this factor, there is two methods to calculate the error term; weight-decay and weight-elimination. The detail representation equation and differential equation is explained in next section.

3. Algorithm

3.1 Weight Decay

Weight decay is a method that after training a neural network and calculating IMF of all the synaptic weights, the weight with most small IMF is pruned and network is re-calculated. To obtain IMF, the calculating time and memory is more needed than normal training method.

Also, as the number of neuron in each layer is different, so IMFs of layer with many numbers of weights are larger than those of other layer. As this reason, IMF must be normalized.

$$E^m = E_{MSE}^m + E_{IMF}^m = E_{MSE}^m + \frac{\lambda}{M}\sum_{w \in C} w_{ij}^2 (x_j^m - \bar{x}_j)^2 \quad (3)$$

where E_{MSE} is the sum squared error between the target and the actual output of the network, E_{IMF} is the sum of all the weight's IMFs.

Therefore, the total sum of error can be written as

$$\begin{aligned} E_{Total} &= E_{MSE} + E_{IMF} \\ &= \frac{1}{M}\sum_m E_{MSE}^m + \frac{\lambda}{M}\sum_{w \in C}\sum_m w_{ij}^2 (x_j^m - \bar{x}_j)^2 \end{aligned} \qquad (4)$$

By the differentiation of above equation's second term with respect to each synaptic weight, newly proposed learning law for 'IMF can be shown. In this paper, we will apply this new learning law to two-layer perceptron and can expand to neural network that has more layers.

3.1.1 Weight connected to output part of synapse

The differentiation of the second term E_{IMF} of eqn. (4) with respect to w_{ij} is as follow

$$E_{IMF}(w_{ij}) = \frac{\lambda}{M}\sum_m w_{ij}^2 (x_j^m - \bar{x}_j)^2 + E'_{IMF}(w_{ij}) \qquad (5)$$

$$\therefore \frac{\partial E_{IMF}}{\partial w_{ij}} = \frac{2\lambda}{M} w_{ij} \sum_m (x_j^m - \bar{x}_j)^2 \qquad (6)$$

That is, the differentiation of IMF with respect to weight is the product of variance of hidden neuron's output and weight.

3.1.2 Weight connected to input part of synapse

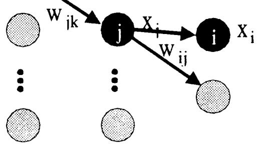


Fig. 2 Multi-Layer Perceptrons

The structure is like figure 2. The differentiation of E_{IMF} of eqn. (4) must be calculated. During the signal transmitting, the weight w_{jk} of the rear layer affects on the weight w_{ij} of the front layer. The total error E_{IMF} is composed of the input part error $E_{\text{IMF},1}$ and the output part error $E_{\text{IMF},2}$.

As E_{IMF} is a function of the input part weights, we can describe this as follow,

$$E_{IMF}(w_{jk}) = E_{IMF.1}(w_{jk}) + E_{IMF.2}(w_{jk}) + E'_{IMF}(w_{jk}) \quad (7)$$

$$E_{IMF.1}(w_{jk}) = \frac{\lambda}{M} w_{jk}^2 \sum_m (x_k^m - \bar{x}_k)^2 \quad (8)$$

$$E_{IMF.2}(w_{jk}) = \frac{\lambda}{M} \sum_m \sum_i w_{ij}^2 (x_j^m - \bar{x}_j)^2 \quad (9)$$

As eqn. (8) is as same as the case of previous output part weight, we can obtain the differentiation with same procedure. However, by chain rule, the differentiation of eqn. (9) with respect to w_{jk} is given by

$$\frac{\partial E_{IMF.2}}{\partial w_{jk}} = \sum_m \frac{\partial E_{IMF.2}^m}{\partial x_j^m} \cdot \frac{\partial x_j^m}{\partial w_{jk}} \quad (10)$$

Each part is calculated as follow,

$$\begin{aligned}\frac{\partial E_{IMF.2}^m}{\partial x_j^m} &= \frac{\lambda}{M}\sum_i w_{ij}^2 \cdot \frac{\partial}{\partial x_j^m}(x_j^m - \bar{x}_j)^2 \\ &= \frac{\lambda}{M}\sum_i w_{ij}^2 \cdot \frac{\partial}{\partial x_j^m}\left[(x_j^m)^2 - \frac{1}{M}\left(\sum_m x_j^m\right)^2\right] \\ &= \frac{\lambda}{M}\sum_i w_{ij}^2 \cdot \left(2x_j^m - \frac{2}{M}\sum_m x_j^m\right) \\ &= \frac{2\lambda}{M}\sum_i w_{ij}^2 \cdot \left(x_j^m - \bar{x}_j\right) \quad (11)\end{aligned}$$

and

$$x_j^m = f\left(\sum_k w_{jk} x_k^m + w_{j0}\right) \quad (12)$$

therefore

$$\frac{\partial x_j^m}{\partial w_{jk}} = f'\left(\sum_k w_{jk} x_k^m + w_{j0}\right) \cdot x_k^m \quad (13)$$

from eqn (11) and (13) we can calculate eqn. (10) as follows

$$\frac{\partial E_{IMF.2}}{\partial w_{jk}} = \frac{2\lambda}{M}\sum_m\left(\sum_i w_{ij}^2 (x_j^m - x_j) \cdot f'\left(\sum_k w_{jk} x_k^m + w_{j0}\right) \cdot x_k^m\right) \quad (14)$$

Therefore, the total weight learning law is given from eqn. (6) and eqn. (14).

3.2 Weight Elimination

Like the case of the weight decay problem, a new learning law for weight elimination can be obtained by defining the term with IMF as error term and differentiating it. First, we apply this learning law to two-layer perceptron and expand to more-than-three-layer perceptron.

3.2.1 Weight connected to output part of synapse

$$\begin{aligned}E_{Total} &= E_{MSE} + E_{IMF} \\ &= \frac{1}{M}\sum_m E_{MSE}^m + \frac{\lambda}{M}\sum_{w \in C}\sum_m \frac{w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0}{1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0} \quad (15)\end{aligned}$$

The differentiation of the second term E_{IMF} of eqn. (15) with respect to w_{ij} is written by

$$E_{IMF}(w_{ij}) = \frac{\lambda}{M}\sum_m \frac{w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0}{1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0} + E'_{IMF}(w_{ij}) \quad (16)$$

$$\therefore \frac{\partial E_{IMF}}{\partial w_{ij}} = \frac{2\lambda}{M} w_{ij} \sum_m \frac{(x_j^m - \bar{x}_j)^2 / IMF_0}{(1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0)^2} \quad (17)$$

3.2.2 Weight connected to input part of synapse

The structure is like figure 2. The differentiation of E_{IMF} of eqn. (15) must be calculated. During the signal transmitting, the weight w_{jk} of the rear layer affects on the weight w_{ij} of the front layer. The total error E_{IMF} is composed of the input part error $E_{\mathrm{IMF,1}}$ and the output part error $E_{\mathrm{IMF,2}}$.

As E_{IMF} is a function of the input part weights, we can describe this as follow,

$$E_{IMF}(w_{jk}) = E_{IMF.1}(w_{jk}) + E_{IMF.2}(w_{jk}) + E'_{IMF}(w_{jk}) \quad (18)$$

$$E_{IMF.1}(w_{jk}) = \frac{\lambda}{M}\sum_m \frac{w_{jk}^2 (x_k^m - \bar{x}_k)^2 / IMF_0}{1 + w_{jk}^2 (x_k^m - \bar{x}_k)^2 / IMF_0} \quad (19)$$

$$E_{IMF.2}(w_{jk}) = \frac{\lambda}{M}\sum_m \sum_i \frac{w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0}{1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0} \quad (20)$$

As eqn. (18) is as same as the case of previous output part weight, we can obtain the differentiation with same procedure. However, by chain rule, the differentiation of eqn. (20) with respect to w_{jk} is given by

$$\frac{\partial E_{IMF.2}}{\partial w_{jk}} = \sum_m \frac{\partial E_{IMF.2}^m}{\partial x_j^m} \cdot \frac{\partial x_j^m}{\partial w_{jk}} \quad (21)$$

Each part is calculated as follow,

$$\begin{aligned}\frac{\partial E_{IMF.2}^m}{\partial x_j^m} &= \frac{\lambda}{M}\sum_i w_{ij}^2 \cdot \frac{\partial}{\partial x_j^m}\left[\frac{(x_j^m - \bar{x}_j)^2 / IMF_0}{1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0}\right] \\ &= \frac{\lambda}{M}\sum_i w_{ij}^2 \left[\frac{2(x_j^m - \bar{x}_j) / IMF_0}{(1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0)^2}\right] \quad (22)\end{aligned}$$

and

$$x_j^m = f\left(\sum_k w_{jk} x_k^m + w_{j0}\right) \quad (23)$$

therefore

$$\frac{\partial x_j^m}{\partial w_{jk}} = f'\left(\sum_k w_{jk} x_k^m + w_{j0}\right) \cdot x_k^m \quad (24)$$

from eqn (22) and (24) we can calculate eqn. (21) as follows

$$\frac{\partial E_{IMF.2}}{\partial w_{jk}} = \frac{2\lambda}{M}\sum_m\left(\sum_i w_{ij}^2 \left[\frac{(x_j^m - \bar{x}_j) / IMF_0}{(1 + w_{ij}^2 (x_j^m - \bar{x}_j)^2 / IMF_0)^2}\right] \cdot f'\left(\sum_k w_{jk} x_k^m + w_{j0}\right) \cdot x_k^m\right) \quad (25)$$

Therefore, the total weight learning law is given from eqn. (17) and eqn. (25).

4. Simulations and Results

We applied the new pruning method to the function approximation. The given function is as fig. 3. The number of training data is 225 and the number of testing data is 10000.

Fig. 3 Testing Function

Before testing our pruning algorithm, it must be predetermined that how many weight is needed. Therefore, Using Quasi-Newton method, we had trained this function. We used 2-input layer, n-hidden layer and 1 output-layer full-linked network.

Through the preprocessing, we have known that the 20~30 hidden neurons are suitable for this function. Therefore through pruning algorithm, the network size must converge to this size.

Table.1 shows the results of pruning with the proposed pruning algorithm.

Table 1. The results of pruning with weight-decay and elimination using IMF

Method	# of hidden neurons	# of pruned weight	ratio	Performance improvement
Weight	40	51	31.7%	5.6%
Decay	50	60	29.8%	8.5%
Weight	40	66	40.9%	5.0%
Elimination	50	72	35.8%	3.4%

The table 1 was made after 50 trials and averaging. The "training–pruning–retraining" procedure was iteratively done. This result shows that the proposed pruning algorithm is doing well. Both weight-decay and weight-elimination methods make network size 20~30. In addition, generalization performance improved by 3~8%. Next two figures show graphically the change of generalization performance. After each pruning, the performance was re-calculated.

Fig 4. The change of generalization performance Using weight-decay and weight-elimination

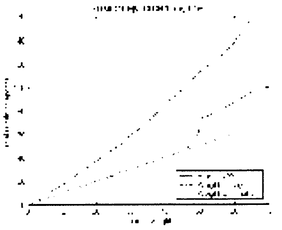


Fig 5. Result of Retraining Time of each algorithm

As the value of weight-decay method is larger than the value of weight-elimination method, so the variation of generalization performance and the epoch for re-training of weight-decay are larger than these of weight-elimination.

Weight-elimination method helps more small weight with small IMF.

5. Conclusions

In this paper, a formal pruning selection scheme based on a penalty term was proposed. During the training procedure, the insignificant neuron will make the same output for all input pattern. So we can easily select the weight that should be pruned.

Using IMF pruning algorithm, we can more easily select insignificant weights and more easily re-train the network.

This paper shows the improvement of generalization performance. This algorithm improves the network by the 8.5% for weight-decay and 3.4% for weight-elimination.

Reference

[1] Andreas S. Weigend, David E. Rumelhart, and Bernardo A. Huberman, "Generalization by Weight-Elimination applied to Currency Exchange Rate Prediction," *IEEE International Joint Conference on Neural Networks*, vol.3, pp. 2374 -2379, 1991

[2] Russell Reed, "Pruning-Algorithm – A survey," *IEEE Trans. on Neural Networks*, vol. 4, pp. 740 –747, 1993

[3] Sevki S. Erdogan, Geok-See Ng, and Khue-Hiang Chan Patrick, "Measurement Criteria for Neural Network Pruning," *IEEE TENCON – Digital Signal Processing Application*, pp. 83-89, 1996

[4] Giovanna Castellano, Anna Maria Fanelli, and Marcello Pelillo, "An Iterative Pruning Algorithm for Feedforward Neural Networks," *IEEE Trans. on Neural Networks*, vol. 8, pp. 519-531, 1997

[5] P. V. S. Ponnapalli, K. C. Ho, and M. Thomson, "A Formal Selection and Pruning Algorithm for Feedforward Artificial Neural Network Optimization," *IEEE Trans. on Neural Networks*, vol. 10, pp. 964-968, 1999

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Dynamical associative memory model consists of structurally unstable oscillators and learning rule based on the internal state of the network

Kazuhiro Kojima　Koji Ito

Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
4259 Nagatsuta Midori-ku, Yokohama 226-8502
Tel:+81-45-924-5654, Fax:+81-45-924-5654
kojima@ito.dis.titech.ac.jp　ito@dis.titech.ac.jp

Abstract

In this paper, we have propose autonomous learning algorithm based on the internal state of the associative memory. The proposed associative memory model consists of structurally unstable oscillators and a common field such as chemical concentration.

In computer simulations, when the pattern memorized in the network is given to the network from the environment, the state of the common field becomes a periodic state. On the other hand, when the pattern has not been memorized is given to the network, the state becomes an intermittently chaotic and the output of the network travels around the input and some memorized patterns. This chaotic state is regarded as "I don't know" state. Further, when the proposed autonomous learning algorithm is applied to the proposed network, the network can learn only the novel patterns automatically without destroying the previously memorized patterns.

Keywords: autonomous learning algorithm, associative memory, time-dependent internal state.

1 Introduction

There are many neural networks which model some information process in the brain. The associative memory model proposed by Hopfield[1] is one of them. In this model, the memories are embedded in the network such as each synaptic weight J_{ij} between the neuron i and neuron j in advance. For recalling the memory, an initial state which is a point inside of the basins is given to the network as the stimulus. Then, the state of the network converges to the relevant point attractor, that is, the network dynamics is based on the Liapunov stability theory. This stability is robust to a noise. However, for this robustness, if novel patterns are given to the network as the continuous external input, the network must converges to 1)one of the embedded patterns, 2)the input pattern, or 3)a mixed pattern of them. Hence, it is necessary that the informational process of the network is divided into the the associative and learning phases.

Freeman and coworkers[2] showed that when the novel odor stimuli are given to the rabbit, the olfactory bulb responses chaotically, which is known as "I don't know" state. Further, if the rabbit has learned the novel odor, the olfactory bulb changes to response periodically. These physiological facts suggest that the brain and neural systems utilize a chaotic dynamics and a phase transition or a bifurcation for the recognition process including the learning process.

From this point of view, we propose a dynamical memory network constructed from chaotic dynamics and an new learning algorithm based on the internal state of the network. There are many network based on a chaotic dynamics[3]~[6]. These studies focused on the dynamical associative process on each network. On the other hand, in this paper we focus the learning process of the network.

2 Proposed Dynamical System

We need a dynamical memory system not based on the Liapunov stability in order to realize "I don't know" state. We proposed the new network based on an analogy between associative memory and Lorenz system[7].

The proposed network is shown in Fig. 1 and the evolution equations are represented by a set of ordi-

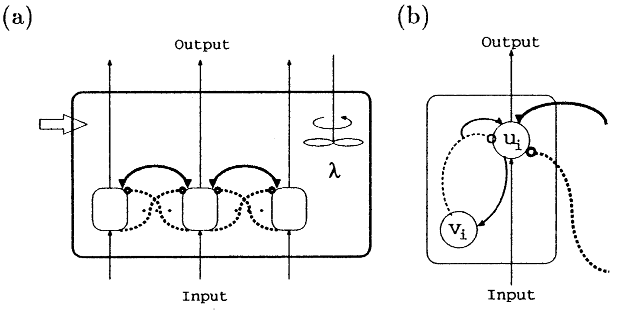


Figure 1: Schematic illustration of the network architecture.

nary differential equations as follows.

$$\dot{u}_i = -u_i + \alpha \left(\lambda - \frac{1}{N}\sum_{j=1}^{N} v_j^2 \right) v_i + h_i \quad (1a)$$

$$\dot{v}_i = -\sigma(v_i - u_i) \quad (1b)$$

$$\dot{\lambda} = b\left(1 - \lambda - \frac{\gamma}{N}\sum_{j=1}^{N} v_j^2\right) \quad (1c)$$

where $\gamma = 2\sigma/b - 1$.

This network consists of the neural oscillators and the common field which stands for the concentration of a certain kind of chemical substances. Further, these oscillators are soaked in this common field. The schematic illustrations of the proposed network are shown in Fig. 1.

Each neural oscillator consists of the excitatory u-neuron and the inhibitory v-neuron shown in Fig.1 (b). There are two type intra-connections from the inhibitory v-neuron to the excitatory u-neuron. These are the first order excitatory connection and the third order inhibitory connection. In Fig.1 (b), the former and the latter corresponds to the solid and the broken lines, respectively. Further, the intension of the first order excitatory connection depends on the common filed λ.

On the other hand, there are two types of inter-connections among oscillators, one is the mutual inhibitory connections from the $v_j (j \neq i)$ neurons to the u_i neuron, the other is the mutual connections among u neurons which corresponds to the thick broken and the thick solid line in Fig.1 (a) and (b), respectively.

The mutual connections among u neurons induces the local filed h_i defined as follows.

$$h_i = \varepsilon_1 \sum_{j \neq i}^{N} J_{ij} s_j + \varepsilon_2 I_i \quad (2)$$

$$s_i = \tanh(\beta u_i). \quad (3)$$

where J_{ij}, I_i and β is coupling weight, input and stepness parameter, respectively.

In Eq.(1a), when the parameter α is zero, the evolution equations are equivalent to the conventional Hopfield model[1]. Then, the proposed model is an expanded model of the conventional Hopfield model.

The common filed is uniform for the space. Further, it is consumed by the oscillators and supplied from the outside. Hence, this system is an open system.

3 Simulations

We use binary patterns that each pixel consists of $\{-1, 1\}$ and its size is 49. These binary patterns are no-orthogonal set.

In order to investigate the network dynamics when known and unknown patterns are given to the network, the patterns $\boldsymbol{\xi}^{(1)}$,$\boldsymbol{\xi}^{(2)}$,$\boldsymbol{\xi}^{(3)}$ and , $\boldsymbol{\xi}^{(4)}$ are embedded in the network in advance. Namely, the synaptic weights J_{ij} between u_i and u_j neurons are given by the superposition of auto-correlation matrix of the each pattern as follows.

$$J_{ij} = \frac{1}{N}\left(\sum_{\mu=1}^{4}(1-\delta_{ij})\xi_i^{(\mu)}\xi_j^{(\mu)}\right) \quad (4)$$

where δ_{ij} is Kronecker's δ. It is simple auto-correlation matrix. we assume no special structure on the synaptic coupling weights J_{ij}.

Each parameter of the network was selected as follows: $k = 10.0$, $b = \frac{3}{8}$, $\beta = 50.0$, $\varepsilon_1 = 0.2$ and $\varepsilon_2 = 0.048$.

4 Dynamical Associative Process

Next, we show the dynamical behavior of λ_t and the output pattern of the network around this critical region. The parameter α was selected as $\alpha = 200$. For measuring the output pattern of the network, we calculated Hamming distances defined as follow.

$$H_t^{(\mu)} = \frac{1}{2}\left(1 - \frac{1}{N}\sum_{i=1}^{N}\xi^{(\mu)} s_{i,t}\right) \quad (5)$$

When Hamming distance $H_t^{(\mu)}$ is 0 or 1, the output pattern is pattern $\boldsymbol{\xi}^{(\mu)}$ or the inverse pattern of it, respectively.

These result are shown in Fig. 2. In this Figure, when the Hamming distance is less than 0.2 or more than 0.8, dot is plotted.

When the known pattern $\boldsymbol{\xi}^{(1)}$ is given to the network, λ_t oscillated periodically with cycle 4, and the

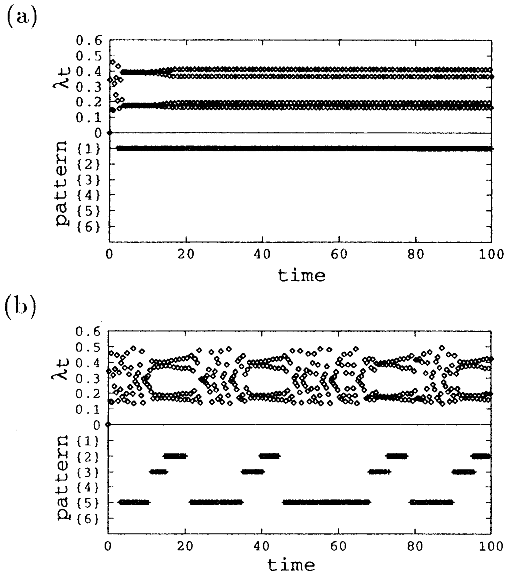


Figure 2: Temporal Association Process. In this Figure, when the Hamming distance is less than 0.2 or more than 0.8, dot is plotted.

output pattern of the network oscillated between the relevant embedded pattern and it's inverse as shown Fig. 2 (a). On the other hand, when the unknown patterns $\xi^{(5)}$ is given to the network, λ_t became a turbulent state from oscillatory state with cycle 4 intermittently, and the output pattern of the network changed with the intermittent turbulence as shown Fig. 2 (b). Therefore, this turbulent state is considered as "I don't know" state.

5 Learning Process

In the above section, we showed the dynamical behaviors of the proposed network, that is, when the known pattern is given to the network, the state of the network is oscillatory state. On the other hand, when the unknown pattern is given to the network, the state of it is chaotic state. Now, under the phase of "I don't know", a learning rule was applied to the network. Then, what will be occurred in the network? In this section, we investigate the effect of a learning process for the proposed network. Further, we propose a new learning rule that depends on the internal state of the network.

5.1 Constant Learning

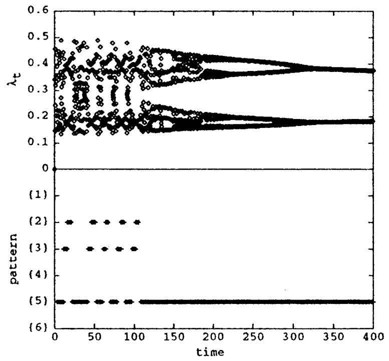


Figure 3: Inverse period-doubling bifurcation under the constant learning.

First, we applied the conventional Hebbian learning rule as following equation.

$$\Delta J_{ij} = \eta s_i s_j \tag{6}$$

Where η is the learning rate parameter.

The result of the constant Hebbian learning is shown in Fig.3. In this case, η is 1.0×10^{-4} and the initial synaptic coupling weights are given by Eq.(4).

The unknown pattern $\xi^{(5)}$ is given to the network. Then, we could observe the inverse period-doubling bifurcation of λ_t responses with the progress of the learning as shown Fig.3. It is very natural result, because the inverse period-doubling bifurcation smoothly connects from the disorder to the order.

But when the learning process progresses, the internal state of the network changes from the limit cycle to the fix point, so that previously memorized pattern are destroyed. Therefore, there is only the input pattern in the network as the memorized pattern. This state is the over learning. Next subsection, we propose a solution for avoiding the over learning.

5.2 Proposed New Learning Rule

As shown in Fig.3, the inverse bifurcation of λ_t corresponds to the index of the progress of the learning. Hence, we notice the periodicity of λ_t. In general, the calculation algorithm of the periodicity needs the memory or long sampling time, e.g. FFT and Lyapunov spectrum method. But the learning process is an on-line process.

We consider the history of the variable λ, that is the time average of it. It can be obtained to solve the following differential equation.

$$\tau \dot{\lambda}_m = -\lambda_m + \lambda \tag{7}$$

where τ is a time constant. When the known and the unknown pattern is given to the network, the tempo-

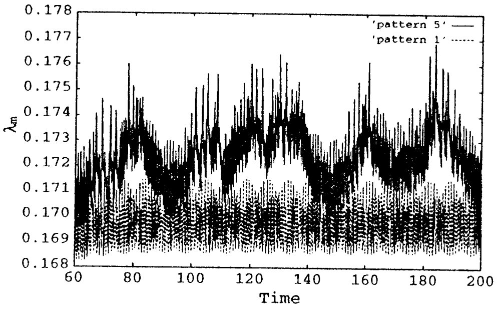


Figure 4: Temporal behavior of the time average $\lambda_m(t)$

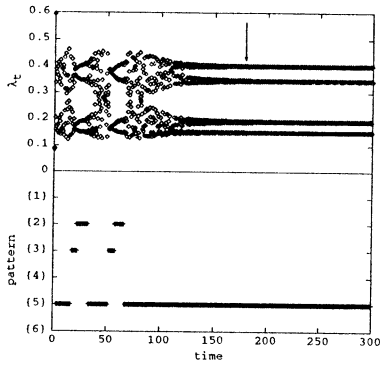


Figure 5: Inverse period-doubling bifurcation under the proposed learning rule.

ral behavior of the time average $\lambda_m(t)$ is represented in Fig.4, where the dash line and the solid line corresponds to the know and the unknown pattern, respectively. From this result, we can find an appropriate threshold that divides the state between the order and the disorder.

According to the threshold property, we propose a new learning rule that depends on the state of the network.

$$\Delta J_{ij} = \eta(\lambda_m) s_i s_j \tag{8}$$

$$\eta(\lambda_m) = \begin{cases} 0 & : x < V_L \\ \eta_0(\tanh(\lambda_m - V_L) + 1) & : x \geq V_L \end{cases} \tag{9}$$

where V_L is the threshold.

The result of the proposed learning rule is shown in Fig.5. In this simulation, we set the threshold parameter V_L to 0.172.

The unknown pattern $\xi^{(5)}$ is given to the network. Then, λ_t inversely bifurcates like as the above constant learning case. But the learning process is reduced gradually. Further, it completely stop around the arrow that is shown in Fig.5, so that we can avoid the over learning.

6 Conclusion

In this paper, we proposed an autonomous learning algorithm based on the internal state of the associative memory. The proposed memory model have a common field such as chemical concentration, that has mutual interaction with each oscillator. In computer simulations, when the pattern memorized in the network is given to the network from the environment, the state of the common field becomes a periodic state. On the other hand, in the case where the pattern has not been memorized is given to the network, the state becomes an intermittently chaotic and the output of the network travels around the input and some memorized patterns.

Moreover, when the proposed autonomous learning algorithm is applied to the proposed network, the network can learn only the novel patterns automatically without destroying the previously memorized patterns.

Acknowledgments

A part of this research was supported by the Scientific Research Foundation of Ministry of Education (10450165) and "Research for the Future" Project (JSPS-RFTF96100105) from the Japan Society for the Promotion of Science.

References

[1] J.J.Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proceedings of Natl.Acad.Sci.USA, Vol. 79, 1982, pp. 2554-2558.

[2] W.J.Freeman and W.Schneider, "Changes in spatial patterns of rabbit olfactory EEG with conditioning to odors", Psychophysiology, Vol. 19, 1982, pp. 44-56.

[3] S.Nara, P.Davis and H.Totsuji, "Memory Search Using Complex Dynamics in a Recurrent Neural Network Model", Neural Networks, Vol. 6, 1993, pp. 963-973.

[4] K.Aihara, Y.Takebe and M.Toyoda, "Chaotic neural networks", Phys. Lett 144A, 1990, pp. 333-340.

[5] I.Tsuda, "Dynamic Link of Memory - Chaotic Memory Map in Nonequilibrium Neural Networks" Neural Networks, Vol. 5, 1992, pp. 313-326.

[6] M.Dauce, M.Quoy, B.Cessac, B.Doyon and M.Samuelides, "Self-organization and dynamics reduction in recurrent networks stimulus presentation and learning", Neural Networks, Vol. 11, 1998, pp. 521-533.

[7] E.N.Lorenz, "Deterministic nonperiodic flow", J.Atmos.Sci., Vol. 20, 1963, pp. 130-141.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Network-Based Human Assist Robotic System Using CORBA

○ Songmin Jia Kunikatsu Takase
Graduate School of Information Systems
University of Electro-Communications
1-5-1 Chofgaoka, Chofu-City, Tokyo 182-8585

Abstract

In order to support the aged and disabled, we proposed the network-based human assist robotic system. In this paper, we detail system concept, system hardware, services provided, Web user interface and recognition strategy for tableware. For real-time recognition of the tableware, color mark-based and correlation calculation technique are used. The proposed system used CORBA to implement networking connections between a client and a remote robotic system. The human caregivers can control the telerobotic system to retrieve and manipulate the desired tableware across Internet by using the intuitive user interface.

keywords: CORBA, remote system, Internet, assist

1 Introduction

Many Internet robotic systems have been developed on the Web over the last few years. Until now, Internet telerobotic systems have typically been operated using two ways communication links between a client and a remote robotic system. KhepOnTheWeb [1] uses CGI (Common Gateway Interface), which allows the user to control a Khepera robot in a static environment. Another CGI system is the WebPioneer [2], in which the user drives a Pioneer robot in an unknown room. The speed of the CGI system is limited because CGI circumvents the problems of static pages by creating a new page for each HTTP request. The USA PumaPaint project [3] is a Web site allowing any user with a $Java^{TM}$ compatible Web browser to control a PUMA760 robot to paint on an easel with real brushes and paint. Mobile robot on the Web project [4] uses Java to establish a modular framework for mobile robots on the Web. Java is executable within a Web page, so Java programs can be run as applets within the browser and supply an interactive interface instead of a static one. But Java client must know the location of all servers to which it wants to connect and Java allows applets to connect only to the host they were served from. In this paper, we use CORBA (Common Object Request Broker Architecture) for distributed, object-oriented applications to develop and implement our system.

2 System overview

Our primary goal was to assist the aged or disabled in their homes when their caregivers are away because the elderly population is growing while the number of people to take care of them is declining. The primary task of the system would be to supply food and a cup of water or juice to aged and disabled persons. To accomplish these tasks, the human caregivers can control the telerobotic system to retrieve and manipulate the desired tableware, by viewing live image feedback from the robot's site through a cellular phone and notebook computer or a common computer.

The proposed system is a CORBA-based, which uses an ORB (Object Request Broker) as the middleware that establishes a client-server relationships between objects, and is an object orientated extension of RPC (Remote Procedure Calls).

3 Hardware base

The hardware configuration of this system includes robot, camera, as server computers and the other workcells. The Mitsubishi Movemaster Super RV-E3J was used in this system. It consists of a manipulator with five degrees of freedoms, a controller, and a teaching box. In order to implement soft grasp, the robot hand with force sensors being affixed to the robot fingers has been designed. The grasp force and speed can be controlled by CPU control circuit.

Two VC-CI MKII cameras are used. One is positioned at the place where it can provide user an feedback image easy to understand, the other is uesed to take an image of the tableware scattered on the table in this system. A high-speed image processing board, IP5000, was used for processing the obtained images. The resolution of the vision system is 512 x 512 pixels, and it takes 3.6 milliseconds to process each image.

The second is the server computer that receives client's command from world wide network, does various kinds of processing and then sends commands to the special devices. A high-performance computer is necessary if we want to provide our services. A workstation with a Pentium III processor running Windows NT 4.0 was used in this system.

The third is the user's computer. Linking to the Internet is the only requirement on user's computer. World Wide Web users can use any computer platform and operating system .

The relationship between hardware of the proposed system is shown in Fig. 1. The communications between server and robot arm's controller and the robot hand's controller are through RS232-C links. User's computers in the world communicate with the server via world wide Internet.

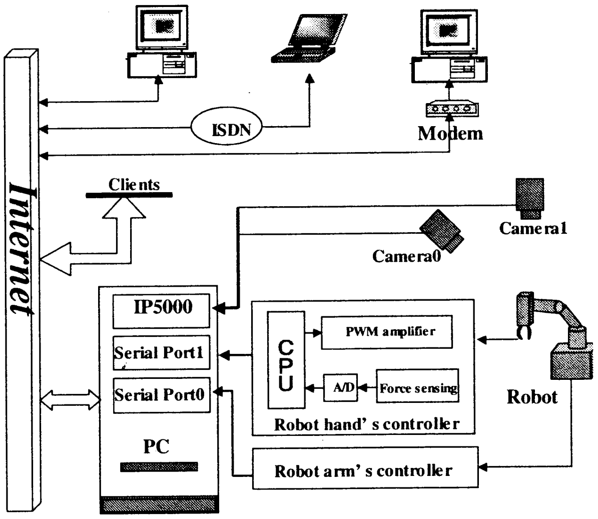


Figure 1. Configuration of the system hardware

4 Services provided by the system

4.1 Web user interface

The Web user interface designed is shown in Fig. 2. It consists of a live feedback image part and a robot commands part and a message text box. To cope with time delays on the communication path, we provide user task-level commands to control the robot. These commands include "Grasp the green cup," "Grasp the spoon," etc.. The live feedback image lets the user see the robot working in the other site. When the user push the button "grasp the spoon", this command will be sent to server and the result retrieven message will be feedback to the client by popping one message text box on the Web user interface. Once one task-level button is pushed, the other task-level button will be invalid until this task is finished for safety. Besides that, the user can also use the emergency button to stop the robot in emergency case.

If the user has a cellular phone and notebook computer or a common computer, he or she can access the robotic system anywhere using this intuitive user interface.

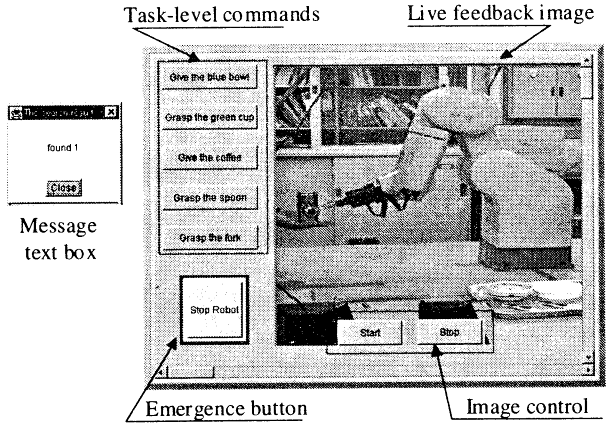


Figure 2. Web user interface

4.2 Task-level robot control

In our system, we have implemented robot control server by the optimal programming language C++ because C++ is the best language for the core functions of the system, such as imaging processing, robot control. The assignments of the server is as follows:

- Receive commands from the remote World Wide Web users.
- ORB interceps the "call" and is responsible for finding an object that can implement the request, passes it the command.
- Find the location and orientation of the tableware scattered on the table by the vision system.
- Generate the task implementation plan and send the commands to the special devices.
- ORB returns the feedback results to the remote clients.

This server gives user the ability to control the robotic system remotely at a task-level. The user can invoke the methods on this application server across the Internet to recognize and manipulate the tableware that they want.

4.3 Live image feedback

This task is provided by the image video feedback server implemented by C++. This server allows live image feedback from the camera and continuously sends live images of the working robot to the client according to the users's requests. It works as:

- Receive image control commands from the remote users.
- ORB intercepts the commands and transfers the requests to live image server.
- Get the new image by camera0 and change the image into a BMP format by IP5000 image processing board.
- Compress this image into JPEG format.
- Return the new image with the last information to the client by ORB.

4.4 Communication between the server and the remote clients

We use CORBA to implement network connections between client and a remote robotic system. In CORBA, we use OMG interface definition language (IDL) to define the object interface and specify the types and operations that the object support. IDL has features like language independence, distributed service specification, definition of complex data types, and hardware independence. By compiling this interface with an IDL compiler for mapping to client/server source code in a specific programming language, we can get client stub and server skeleton source code for communication between client and server. By introducing CORBA communication into a Java applet, a Java client is capable of directly accessing CORBA application servers via IIOP instead of static, it can run on different platforms, and it can transparently invoke methods on the application servers across the network.

5 Recognition strategy for tableware

For long tableware, if we want to handle them exactly, we must know not only the location but also its orientation. In order to implement real-time recognition of the tableware, the color-based technique was used. In this techique, we assume that spoons and forks consist of two different color. Using this technique, long tableware recognition can be replaced by the simple mark recognition because there is one-to-one correspondence between tableware and different combination of color marks. We can creat the tableware 's geometric model and entry them beforehand. By doing this we can recognize the tableware easily and quickly.

For circular tableware, the quick correlation calculation was used. Correlation calculation is a method that evaluates the matching between the template image registered beforehand and the input image obtained from the camera. It is better that the illumination of the template gotten is the same as that of the applied condition. In spite of doing that, the template is also circumscribed the other problems. We have done some experiments such as changing the position and the angle seen from the camera coordinate frame at the same position. The experiment results are shown in Fig. 3.

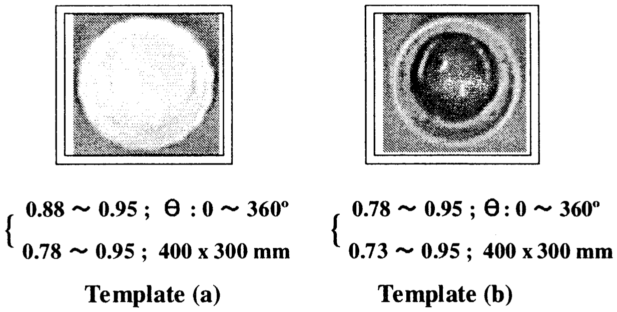

0.88 ~ 0.95 ; θ : 0 ~ 360°
0.78 ~ 0.95 ; 400 x 300 mm
Template (a)
0.78 ~ 0.95 ; θ: 0 ~ 360°
0.73 ~ 0.95 ; 400 x 300 mm
Template (b)

Figure 3. Templates and its experiment results

According the results of our experiments, we know not only a different object has a different correlation value , but also the same object will has a different correlation value in different condition. For example, template (a)'s correlation value will change from 0.88 to 0.95 if we rotate the template at different rotation angle , and correlation value will change from 0.78 to 0.95 if we move it to different position under world coordinate frame. This results from tableware's nonuniform distribution and nonuniform illumination under the application environment. So, we must measure the minimum correlation value and it should be bigger enough than the one which can distinguish itself from the other tableware and the environment.

6 Experiments

6.1 Manipulation of tableware

In this system, "Grasp" of the tableware is very necessary. Generally, We divides the "grasp" into two stages, the transport stage and the manipulation stage. In the transport stage, viewer-relative , extrinsic properties of the object such as location and orientation are used to guide hand orientation and a ballistic reach toward the object. In the manipulation stage, object-relative , intrinsic properties such as shape and size are used to preshape the hand in anticipation of the grasp. Grasp configurations are chosen based on the kinds of tableware. For cup and can, we use "grip", for big bowl, spoon and fork, we use "pinch".

6.2 Remote accesses to the system

This system have been accessed by the operators from Tama City, Saitama, Kyoto University, Japan and Chicago, America etc.. These user's computers have different specifications and network connections, and work on the different operating systems. Users operate this remote robotic system to retrieve and manipulate the tableware that they want and they can also monitoring the state the robot is executing the task by viewing the live imgae feedback coming from the robot site. Fig. 4 shows the scene of one user accessing this system. Fig. 4(1), Fig. 4(2), Fig. 4(3) are some of the live feedback images sent from the server. In order to know time delays on the communication path, some intervals between the users pushing the "start" button and the new image with the latest information showing have also been recorded. The average time of 10 times changed from 6.8 seconds to 26 seconds. These values can be acceptable by users and may vary according to the specifications of the user' s computer, the communication path and the time the operators access the system.

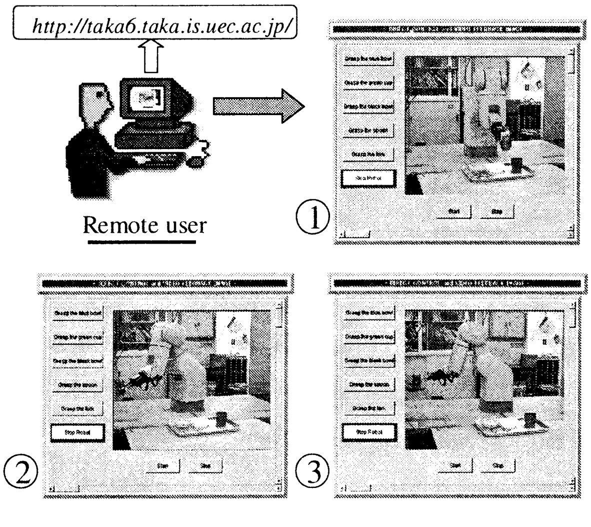


Figure 4. Remote access to the system

7 Conclusions

We used CORBA to implement networking connections between a client and a remote robotic system. This allows a client to invoke a method on a server across a network transparently, and to not have to know where the application servers are located, or what programming language and operating system are used. This lets system overcome the shortcomings of the other Web-based telerobotic systems. In order to support old and disable persons, our system provides services that caregivers can control the telerobotic system to give some necessary help across Internet by using the intuitive user interface. For future work, we will develop the mobile robotic system with arm to implement a real tele-assist robotic system to the aged and disabled.

References

[1] Patric Saucy (1998), KhepOnTheWeb: One Year of Acess to a Mobile Robot on the Internet, Proc. of 1998 IEEE/RSJ Conference on Intelligent Robots and Systems; Workshop on Web Robots, Victoria, B.C. Canada, October 12-17, 1998, pp.23-31.

[2] WebPioneer Website, ActivMedia, Inc., *http://webpion.mobilerobots.com*, 1998.

[3] Matthew Stein (1998), Painting on the Web: The PumaPaint Project, Proc. of 1998 IEEE/RSJ Conference on Intelligent Robots and Systems ; Workshop on Web Robots, Victoria, B.C. Canada, October 12-17, 1998, pp.37-43.

[4] Roland Siegwart, Cedric Wannaz, Patrick Garcia et al. (1998), Guiding Mobile Robots through the Web, Proc. of 1998 IEEE/RSJ Conference on Intelligent Robots and Systems ; Workshop on Web Robots, Victoria, B.C. Canada, October 12-17, 1998, pp.1-7.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Cooperative Behavior of Multi Robot System with Simple Interaction

Ken Sugawara[†][‡], Masaki Sano[*], Ikuo Yoshihara[+], Kenichi Abe[§], and Toshinori Watanabe[‡]
[†]PRESTO, Japan Science and Technology Corporation(JST)
[‡]Graduate School of Information Systems, Univ. of Electro-Communications
[*] Graduate School of Physics, University of Tokyo
[+] Dept. Information Engineering, Miyazaki University
[§] Graduate School of Electrical Engineering, Tohoku Univ.

Abstract

Researches of multi-robot system are active in these days. The most remarkable characteristic of multi-robot system is that the robots work cooperatively and achieve the task which a single robot cannot do. This characteristic is important and it is essential to investigate number effect of multi-robot system. There are some tasks which are suitable for multi-robot system and we chose foraging task. Here we introduced multi-robot with simple interaction and fractal distribution of pucks to deal with their behavior quantitatively. In this paper, we investigated the performance of homogeneous multi-robot system at first, and clarified there is an optimum interaction duration. Secondly we investigated the performance of heterogeneous system which is composed of interacting robots and isolated robots, and we clarified there is an optimum ratio.

1 Introduction

The research of multi-robot system is active in these days. Many researchers have been studied the behavior of multi-robot system [1, 2, 3, 4, 5]. One of the best example of multi-robot system is a Robo-Cup, which is a soccer tournament by autonomous robots. The most important point of multi-robot system is "they can work cooperatively." In other words, they have a possibility to achieve the task which single robot cannot do. Most researchers have focused on this characteristic of multi-robot system and applied the system to concrete problems from the engineering viewpoint. But most of them tend to focus on qualitative aspect of multi-robot system. In this paper, we discuss the efficiency of the system quantitatively focusing on the relation between the number of robots and their performance by simple task.

2 Foraging Behavior of Multi-robot

Remarkable characteristic of multi-robot system is that the robots work cooperatively. It means the work performed by a system of N interacting robots can be larger than the plain sum of N individuals without interaction. Tasks which are suitable for multi-robot system are considered as follows[6].
(1) Parallel searching
(2) Cooperative transportation
(3) Mutual observation
(4) Structure formation

In our research, we choose foraging behavior as a task (Figure 1). Forage is the act of searching or hunting for food. This task is composed of parallel searching and cooperative transportation, and it is easy to extend another kind of task such as mutual observation or structure formation. The most popular phenomenon of cooperative foraging behavior by simple interaction is observed in social insects such as ants and bees[7, 8].

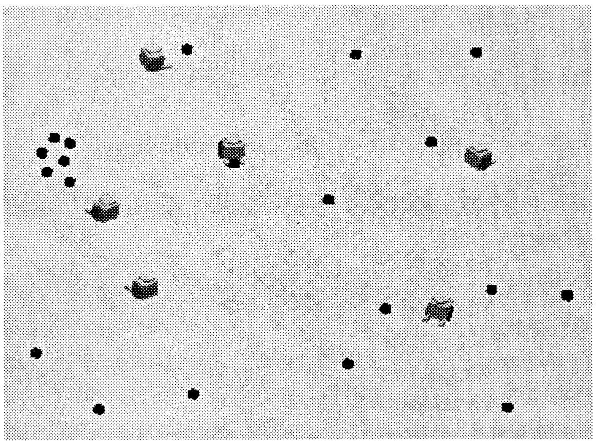


Figure 1: Schematic of foraging task

3 Experiments and Results

3.1 Simulation Condition

3.1.1 Movement of Robots

Cooperative foraging behavior can be described by the combination of the following basic behaviors: Wandering, Broadcasting, Attracted, Homing, Staying and Avoiding.

- Wandering: to move straight at constant velocity.

$$\vec{v} = v \cdot \vec{s_i}, \tag{1}$$

where v,$\vec{s_i}$ are a constant and heading unit vector of robot i respectively ($\vec{s_i} = (\cos\theta_i, \sin\theta_i)$).

- Broadcasting: to radiate isotropic signal light. The condition is described as

$$l_i = \begin{cases} 1 & \text{(radiating)} \\ 0 & \text{(otherwise)} \end{cases} \tag{2}$$

- Attracted: to move toward the signal source. Angular velocity obeys the following equation.

$$\frac{d\theta_i}{dt} = \tanh(c \cdot \sum_{j \neq i}^{n}((\vec{s_i} \times \frac{\vec{r_{ij}}}{||\vec{r_{ij}}||})_z \cdot \frac{l_j}{||\vec{r_{ij}}||^2})) \cdot \Omega, \tag{3}$$

where $\vec{r_{ij}}$ is a vector directed from robot j to i, c and Ω are constants, and $()_z$ implies z component of the vector.

- Homing: to move toward home. Angular velocity obeys the following equation.

$$\frac{d\theta_i}{dt} = \tanh(c \cdot (\vec{s_i} \times \frac{\vec{r_{ih}}}{||\vec{r_{ih}}||})_z) \cdot \Omega, \tag{4}$$

where $\vec{r_{ih}}$ is a vector directed from home to robot i.

- Staying: to stay there.

- Avoiding: to avoid collision with other robots and walls. This behavior is described as follows: $\theta_i \leftarrow \theta_i + \Delta\theta$, where $\Delta\theta$ is a random variable uniformly distributed within $(0, -2\pi)$. It takes a constant time $\frac{\Delta\theta}{\Omega}$ for avoiding.

Values used in computer simulation are as follows: $v = 10, \Omega = \frac{2}{3}\pi$. These values are derived from our actual robot.

3.1.2 Simulation Field

We chose a spatially discrete model to accelerate computation. We assume a square field, with home located at the center. The field is partitioned into 800×800 cells. Each cell can have three states: "empty", "robot", or "puck." The position and 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 robot is one cell.

3.1.3 Puck Distribution

The task of these robots is to collect pucks distributed in the field. There could be various types of puck distribution. Here we choose a fractal distribution because we can discuss the characteristics of the field quantitatively[9]. At first, 800 pucks are distributed by the Levy flight method. The Levy flight method gives us an arbitrary dimension D of distribution. The process of Levy flight is described below.

P1) The first puck is positioned randomly.

P2) The position of the $i + 1$th puck is determined by the distance r, from the ith puck and the directional angle θ, where r is uniformly random within $(0, \infty)^{-D}$ and θ is uniformly random within $(0, 2\pi)$.

P3) P2) is iterated until 800 pucks are distributed.

Figure 2 is some example of puck distribution generated by this method. You can find the degree of the localization can be controlled by one parameter.

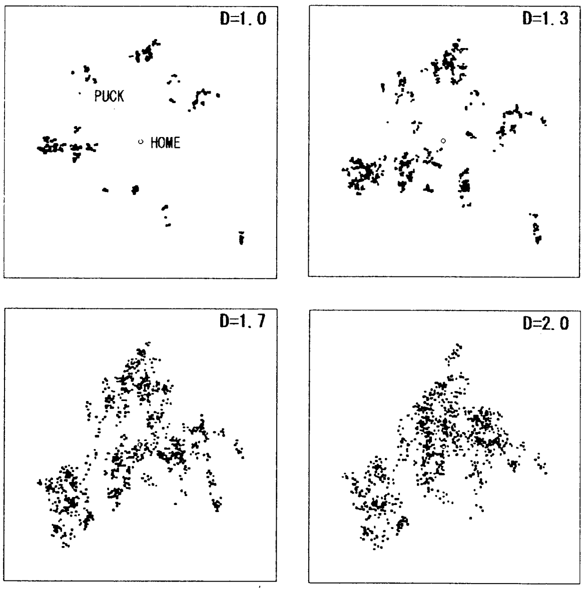


Figure 2: Some examples of fractal puck distribution.

3.2 Foraging Behavior of Homogeneous Robots

Herc, wc discuss foraging behavior of homogeneous robots. We measured the relation between the number of robots N and the number of collected pucks during a constant time P. The relation between N and P is not simple because of the finite number of pucks and the frequent collisions, but we can quite roughly approximate that it forms a power law relation, that is, $P \sim N^{\beta}$. It is known the value of β changes from 0.0 to 2.0 as a function of interaction duration and there is an optimum value of β(denoted as β_{opt})[10]. The value of β_{opt} depends on the dimension of the puck distribution. Figure 3 shows the relation between the dimension of puck distribution and the value of β_{opt}.

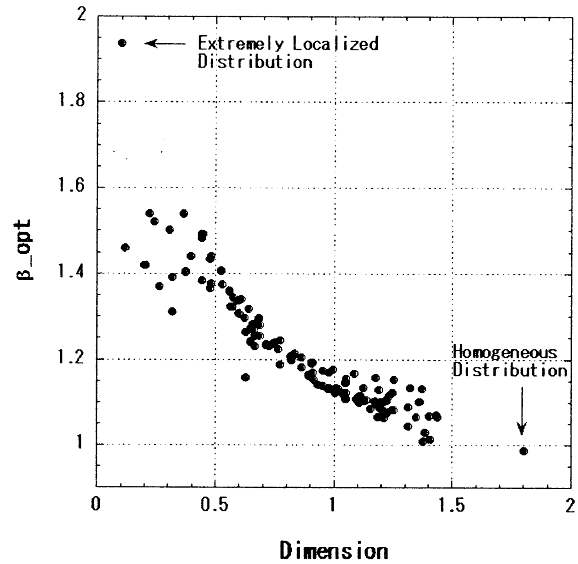


Figure 3: The relation between the interaction duration and β_{opt}.

The behavior of robots can be described by state transition diagram and we can get the following equations from the diagram[9].

$$\begin{array}{rcl} \dot{S} & = & -\alpha S + \frac{1}{\tau}R - l(x)\cdot S\cdot B + bM \\ \dot{B} & = & -\frac{1}{x+1}B + \alpha S + \gamma\cdot\frac{1}{1+cA}A \\ \dot{R} & = & \frac{1}{x+1}B - \frac{1}{\tau}R \\ \dot{M} & = & -\frac{v}{d}M + l(x)\cdot S\cdot B - bM \\ \dot{A} & = & \frac{v}{d}M - \frac{\gamma}{1+cA}A, \end{array} \qquad (5)$$

where S: the number of robots in searching mode, B: the number of robots in broadcasting mode, R: the number of robots in returning mode, M: the number of robots that move to signal source, A: the number of robots that avoid collision.

Meaning of parameters in the equation is as follows: α: probability to find a puck independently, τ: time to return home, x: interaction duration, d: average distance between interacting robots, v: velocity of robot, γ: probability to find a puck by following other robots. b: probability to lose the direction to signal source. $l(x)$ describes a probability to turn to the broadcasted signal source.

Figure 4(a) shows the result of robot simulation and figure 4(b) shows static feature of the equations. It is clear the equations describe their behavior well.

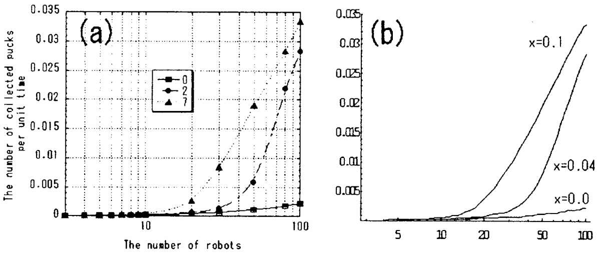


Figure 4: (a) The result of robot simulation. Parameter is an interaction duration. (b) Static feature of the equations. α=0.0002, b=0.01, d=200, v=10, γ=0.02, τ=100, c=0.05.

3.3 Foraging Behavior of Heterogeneous Robots

Here, we discuss foraging behavior of heterogeneous robots. In previous section, we assumed the characteristics of all robots are same. In case that the density of pucks is high, the concentration of robots increases their efficiency, but in case that the density is low, the concentration decreases the efficiency. It means that the effectiveness of interaction depends on the puck distribution. This fact suggests a heterogeneous robot system which consists of interacting robots and independent robots might work better than homogeneous system. In fact, some previous studies reported the efficiency of heterogeneous multi-robot system[11, 12].

We measured the number of collected pucks by changing the ratio of interacting robots and independent robots. Interacting robots behave as described before, and independent robots behave independently without reacting to other robots' signals. The interaction duration is 10, which is the optimum value obtained in previous section. Figure 5 shows the result. The number of robots is 20 ~ 100. X-axis is the ratio of interacting robots and isolated robots. Here 0 means all robots behaves independently, and 100 means all robots interacts. Y-axis is the number of collected pucks during a constant period. In case that the number of robots is small(figure 5(a)), their performance is better by interaction and it does not depend on the dimension of puck distribution. But in case that the number of robots is large, heterogeneous system works well and we can find out there is an optimum ratio depending on the dimension of puck distribution(figure 5(d)).

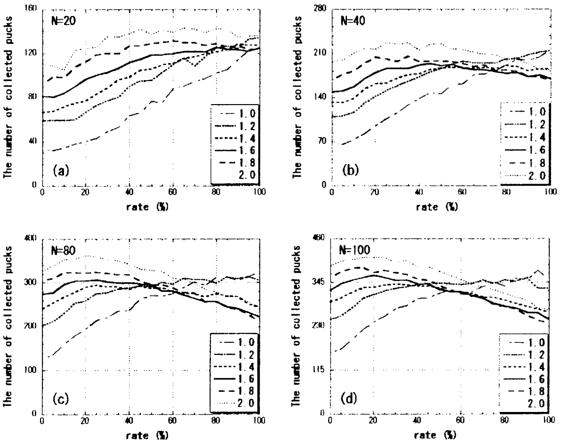


Figure 5: The relation between the ratio of interacting robots and the number of collected pucks. Parameters in the graph are dimension of puck distribution. (a)Total number of robots is 20 (N=20). (b)N=40. (c)N=80. (d)N=100.

4 Conclusion

We investigated the collective behavior of multi-robot system through the puck gathering task. Here, we introduced fractal distribution of pucks by Levy flight method. Introduction of fractal dimension of puck distribution enables us to deal with the field quantitatively.

The efficiency of the system was studied by computer simulations and mathematical model. At first, we investigated the performance of homogeneous multi-robot system and showed there is an optimum interaction duration. Next, we investigated the performance of heterogeneous multi-robot system which is composed of interacting robots and independent robots. We clarified there is an optimum ratio in each dimension of puck distributions.

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Encouragement of Young Scientists, 12750399, 2000.

References

[1] T. Sato, et al., *J. SICE*, 31, (1992)(in Japanese).

[2] K. Asama, et al., Eds. *Distributed Autonomous Robotic System 2*, Springer-Verlag, (1996).

[3] T. Balch and R.C. Arkin, "Communication in Reactive Multiagent Robotic Systems," *Autonomous Robots*, 1, (1994), pp. 27.

[4] M.J. Mataric, "Interaction and Intelligent Behavior," Ph.D Thesis, MIT, (1994).

[5] L. E. Parker, "The Effect of Action Recognition and Robot Awareness in Cooperative Robotic Teams," *Proc. of 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (1995) , pp. 212.

[6] J. JSAI, vol.5, No.4 (1990).

[7] B. Hölldobler and E. O. Wilson, *JOURNEY TO THE ANTS*, Harvard University Press, (1994).

[8] T. D. Seeley, *THE WISDOM OF THE HIVE*, Harvard University Press, (1995).

[9] K. Sugawara, I. Yoshihara and K. Abe, "A Scaling Law between Number of Multi-robots and Their Task Performance," *Proc. 3rd Int'l Symp. on Artificial life and Robotics*, (1998) pp.145.

[10] K. Sugawara and M. Sano, "Cooperative Acceleration of Task Performance: Foraging Behavior of Interacting Multi-Robots System," *Physica D*, 100, (1997), pp.343.

[11] L.E. Parker, "Multi-robot Team Design for Real-World Applications," *Distributed Autonomous Robotic System 2*, Springer-Verlag, (1996), pp.91.

[12] D. Kurabayashi, et al., "Efficiency of a Group Consisted of Robots with Heterogeneous Motion Algorithms," *Trans. of SICE*, Vol.35, No.11, (1999), pp.1377.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Resolved Motion Rate Control of a Free-Floating Underwater Robot with Horizontal Planar 2-Link Manipulator

S. Sagara T. Danjoh M. Tamura R. Katoh
Department of Control Engineering
Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan
E-mail: sagara@cntl.kyutech.ac.jp

Abstract

Through experiments of a free-floating underwater robot with 2 dimensional and horizontal planar 2-link manipulator, we have shown that a Resolved Acceleration Control with the manipulator's end-tip position and velocity feedback has a good control performance in spite of using an inaccurate hydrodynamic model. In this paper, for the free-floating underwater robot manipulator we construct a control system using a Resolved Motion Rate Control and verify the effectiveness of the control system by experiments.

1 Introduction

This paper is concerned with a Resolved Motion Rate Control (RMRC) of a free-floating underwater robot with horizontal planar 2-link manipulator. Since underwater robots are necessary to ocean exploration, many studies have been done about Underwater Robotic Vehicle (URV) including Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) [1]. To get resources some URVs have the small manipulators whose dynamics can be ignored. However, the larger manipulators of URVs, especially AUVs, are required to construct underwater structures and the dynamics cannot be ignored. Therefore, high accurate dynamic model and/or control methods of manipulator are dominant to maintain a good control performance of AUV. Recent years some dynamic models of underwater manipulator have been proposed [2, 3, 4, 5, 6]. However, the dynamic model of underwater manipulators contains nonlinear hydrodynamic forces that are added mass force and drag force, and the complete hydrodynamic model cannot be obtained. Through experiments of a free-floating underwater robot with 2 dimensional and horizontal planar 2-link manipulator, we have shown that a Resolved Acceleration Control (RAC) with the manipulator's end-tip position and velocity feedback has a good control performance in spite of using an inaccurate hydrodynamic model [7]. In this paper, for the underwater robot manipulator shown in Fig. 1 we construct a control system using the RMRC and verify the effectiveness of the control system by experiments. The RMRC uses a Generalized Jacobian Matrix (GJM) that is an extended and generalized form of manipulator Jacobian implying a reaction dynamics of free-floating systems in space [8]. Similar one for underwater systems exists [9], but the GJM cannot be used for control systems. So the GJM for space systems is used here and the end-tip position feedback is added to the RMRC to compensate the influence of the hydrodynamic forces. To verify the effectiveness of the control method, the experiments are done. From the experimental results, we show that the control method is effective to the free-floating underwater robot with manipulator and the control performance is almost equivalent to the case of the RAC.

2 Modeling and Dynamics

2.1 Modeling

The underwater robot model used in this paper is shown in Fig. 2. It has a robot base and 2-DOF manipulator that can move in a plane.

Symbols used in this paper are defined as follows:
[Symbols]

Σ_U : inertial coordinate frame

Σ_i : ith link coordinate frame ($i = 0, 1, 2$; link 0 means base)

${}^U\boldsymbol{R}_i$: coordinate transformation matrix from Σ_i to Σ_U

$\boldsymbol{r}_0$: position vector of origin of Σ_0 with respect to Σ_U

l_i : length of link i

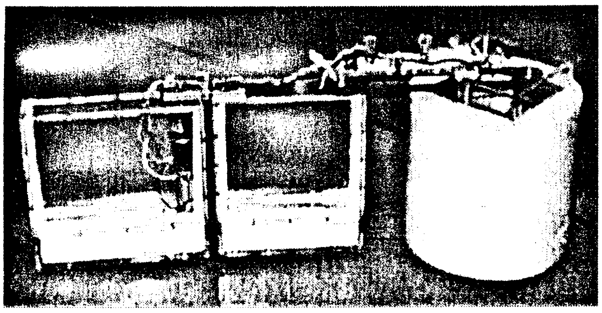


Fig. 1: Floating 2-link underwater robot manipulator

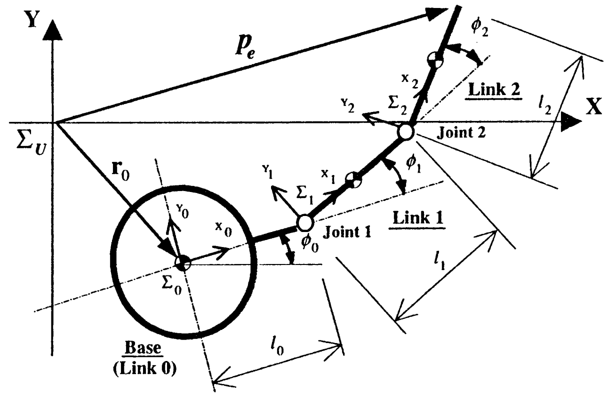


Fig. 2: 2-link underwater robot manipulator model

$\boldsymbol{v}_i$: velocity vector of link i with respect to Σ_i
ϕ_i : relative joint angle
ω_i : joint angular velocity ($= \dot{\phi}_i$)
θ_i : absolute joint angle
$\boldsymbol{p}_e$: position vector of end-tip with respect to Σ_U
$\boldsymbol{p}_0$: position vector of center of gravity of base with respect to Σ_U
$\boldsymbol{X}_e$: position and attitude vector of end-tip with respect to Σ_U ($= [\boldsymbol{p}_e^T,\ \theta_2]^T$)
$\boldsymbol{X}_0$: position and attitude vector of center of gravity of base with respect to Σ_U ($= [\boldsymbol{p}_0^T,\ \theta_0]^T$)
$\boldsymbol{\phi}$: joint angular velocity vector ($= [\phi_1,\ \phi_2]^T$)
m_i : mass of link i (link 0 means the robot base)
$\boldsymbol{M}_{a_i}$: added mass inertia of link i
$\boldsymbol{I}_i$: inertia tensor of link i
$\boldsymbol{I}_{a_i}$: added inertia tensor of link i
$\boldsymbol{E}$: unit matrix

2.2 Equation of motion

Considering the hydrodynamic forces and using Newton-Euler formulation, the following equation of motion can be obtained [7]:

$$(\boldsymbol{M}+\boldsymbol{M}_a)\ddot{\boldsymbol{q}}+\boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{F}_D=[0,\ 0,\ 0,\ \tau_1,\ \tau_2]^T \quad (1)$$

where $\boldsymbol{q}=[\boldsymbol{X}^T,\ \boldsymbol{\phi}^T]^T$, and $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the vector of Coliolis and centrifugal forces, $\boldsymbol{M}_a$ is the matrix consisting of the added mass and inertia, $\boldsymbol{F}_D$ is the vector consisting of the drag force and moment, τ_i is the joint torque.

$\boldsymbol{M}$ and $\boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}})$ are the same as free floating space robot manipulators [8]. In other words, if the robot is not affected by the hydrodynamic forces and moments, i. e. $\boldsymbol{M}_a=\boldsymbol{0}$ and $\boldsymbol{F}_D=\boldsymbol{0}$, Eq. (1) represents the free-floating space robot with 2-link manipulator. Therefore, considering the $\boldsymbol{M}_a$ and $\boldsymbol{F}_D$ as the distutbance, the RMRC method for space manipulators can be applied to the underwater robot manipulator.

3 Resolved Motion Rate Control

To design a position control system of the end-tip, a resolved motion rate control method is used in this paper.

If $\boldsymbol{M}_a=\boldsymbol{0}$ and $\boldsymbol{F}_D=\boldsymbol{0}$, the following relation can be obtained [8]:

$$\dot{\boldsymbol{p}}_e=\boldsymbol{J}^*\dot{\boldsymbol{\phi}} \quad (2)$$

where $\boldsymbol{J}^*$ is called GJM for space manipulators. And using Eq. (2) and the desired value of the end-tip $\boldsymbol{p}_d$, the desired rotational velocity of the joint $\dot{\boldsymbol{\phi}}_d$ can be obtained:

$$\dot{\boldsymbol{\phi}}_d=\boldsymbol{J}^{*-1}\dot{\boldsymbol{p}}_d \quad (3)$$

Since $\boldsymbol{M}_a\neq\boldsymbol{0}$ and $\boldsymbol{F}_D\neq\boldsymbol{0}$, instead of $\dot{\boldsymbol{p}}_d$ the following modified desired value of the end-tip $\dot{\boldsymbol{p}}_d^*$ is utilized:

$$\dot{\boldsymbol{p}}_d^*=\dot{\boldsymbol{p}}_d+K_p(\boldsymbol{p}_d-\boldsymbol{p}_e) \quad (4)$$

where $\boldsymbol{p}_d$ is the desired position of the end-tip and K_p is the position feedback gain.

4 Experiment

In this section, to verify the effectiveness of the RMRC method, the control experiments are done.

4.1 Experimental system

Fig. 3 shows a configuration of experimental system. A robot that has a 2DOF manipulator is floating water. Each joint is actively rotated by a DC servo actuator consisting of DC servo motor, harmonic drive gear and incremental type encoder but no actuator for attitude control of the robot base. Physical parameters of the underwater floating robot are shown in Table 1.

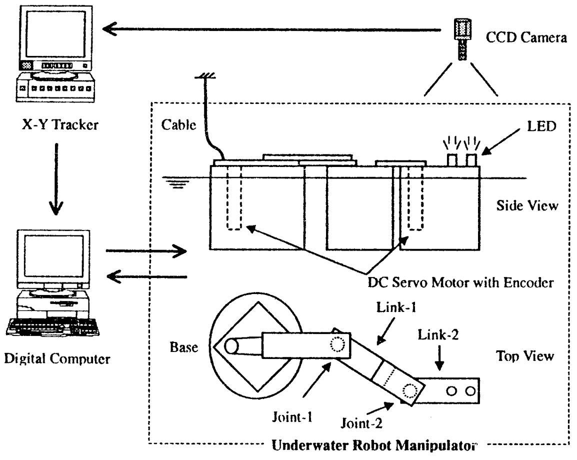


Fig. 3: Configuration of underwater robot system

Table 1: Physical parameters of the underwater robot

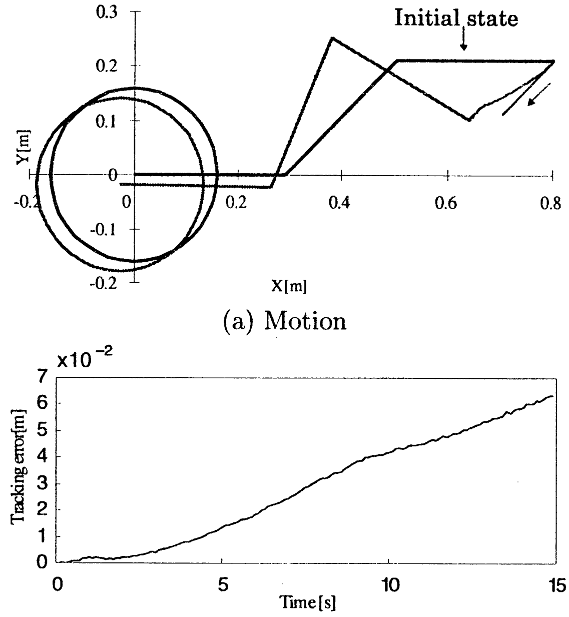
	Base	Link 1	Link 2
Mass [kg]	31.72	7.48	9.68
Moment of inertia [kg m^2]	0.41	0.20	0.37
Link length [m]	0.16	0.30	0.30
Height [m]	0.30	0.30	0.30
Added mass [kg]	48.26	32.82	32.82
Added moment of inertia [kg m^2]	0	0.15	0.33

Measurement and control system is consisting of CCD camera, Video Tracker and 32bit personal computer. Two light emitting diodes (LED) are attached to the manipulator, and their motion is monitored by a CCD camera hanging from the ceiling. Video signals of the LED markers are transformed into the position data by Video Tracker, and put into a personal computer via a GPIB communication line. Using the position data and rotational angle of each joint measured by the incremental type encoder, the positions and attitude angles of the robot base and manipulator are computed in the personal computer that is also used in controller.

4.2 Experimental result

The desired end-tip position $\boldsymbol{p}_d$ was set up along a straight path from the initial end-tip position $\boldsymbol{p}(0)$ to the target calculated from trapezoidal velocity pattern.

Fig. 4 and Fig. 5 show the typical RMRC experimental results using the feedback gains $K_p = 0$ and $K_p = 10$, respectively.



(a) Motion

(b) Tracking error

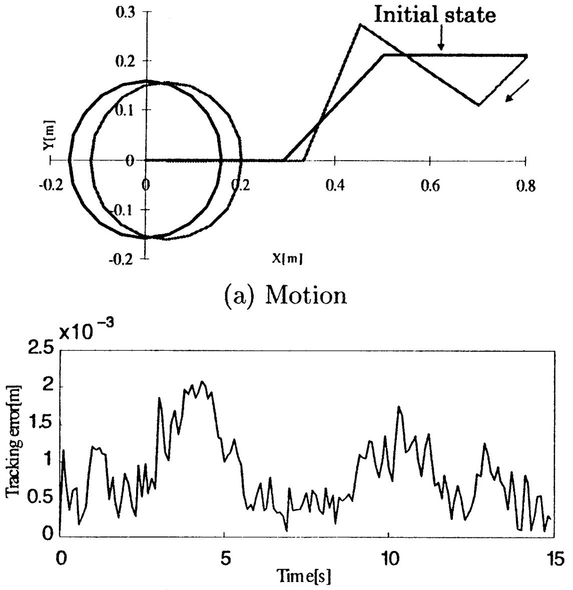
Fig. 4: Experimental result I (RMRC, $K_p = 0$)

From Fig. 4 we can see that the end-tip cannot follow the desired trajectory without the end-tip position feedback. On the other hand, from Fig. 5 it can be seen that the end-tip follows the desired trajectory in spite of the influence of the hydrodynamic forces and the RMRC with the end-tip position feedback can be applied to the underwater robot manipulators.

Furthermore, Fig. 6 shows the result of the case of RAC [7]. From Fig. 5 and 6, the control performance of the method of RMRC with the end-tip position feedback is almost equivalent to the case of the RAC.

5 Conclusion

In this paper, for a free-floating underwater robot with horizontal planar 2-link manipulator we constructed a control system using a Resolved Motion Rate Control (RMRC) with the end-tip of manipulator position feedback and verified the effectiveness of the control system by experiments. The RMRC method used a Generalized Jacobian Matrix (GJM) in space. Adding the end-tip position feedback to the RMRC to compensate the influence of the hydrodynamic forces the control experimental results showed that the control method was effective to the free-floating underwater robot with manipulator and the control performance was almost equivalent to the case of the RAC.



(a) Motion

(b) Tracking error

Fig. 5: Experimental result II (RMRC, $K_p = 10$)

Acknowledgments

This research is performed at Satellite Venture Business Laboratory, Kyushu Institute of Technology.

References

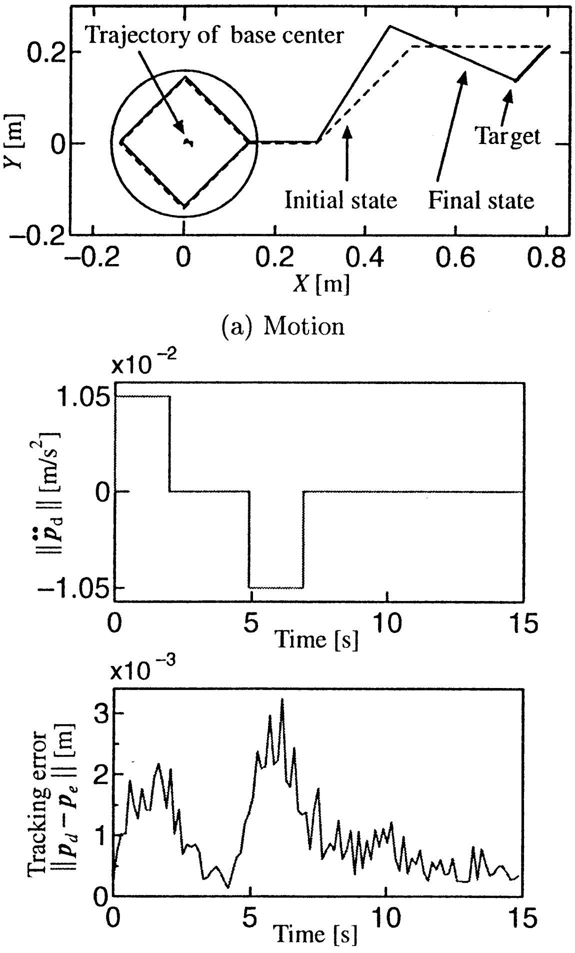
[1] J. Yuh ed., *Underwater Robotic Vehicles:Design and Control*, TSI Press, 1995.

[2] T. W. McLain, S. M. Rock and M. J. Lee, "Experiments in the Coordinated Control of an Underwater Arm/Vehicle System", *Autonomous Robots 3*, Kluwer Academic Publishers, pp. 213 - 232, 1996.

[3] T. W. McLain and S. M. Rock, "Development and Experimental Validation of an Underwater Manipulator Hydrodynamic Model", *Int. J. Robotics Research*, Vol. 17, No. 7, pp. 748 - 759, 1998.

[4] S. McMillan, D. E. David and R. B. McGhee, "Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 25, No. 8, pp. 1194 - 1206, 1995.

[5] B. Lévesque and M. J. Richard, "Dynamic Analysis of a Manipulator in a Fluid Environment", *Int. J. Robotics Research*, Vol. 13, No. 3, pp. 221 - 231, 1994.



(a) Motion

(b) Tracking error

Fig. 6: Experimental result III (RAC)

[6] T. J. Tarn, G. A. Shoults and S. P. Yang, "A Dynamic Model of an Underwater Vehicle with a Robotic Manipulator", *Autonomous Robots 3*, Kluwer Academic Publishers, pp. 269 - 283, 1996.

[7] S. Sagara, T. Tanikawa, M. Tamura and R. Katoh, "Experiments of a Floating Underwater Robot with 2 Link Manipulator", *Proc. of 5th Int. Symp. on AROB*, Oita, Japan, pp. 367 - 370, 2000.

[8] K. Yoshida and Y. Umetani, "Control of Space Manipulators with Generalized Jacobian Matrix," in Y. Xu and T. Kanade ed., *Space Robotics : Dynamics and Control*, Kluwer Academic Publishers, pp.165 - 204, 1993.

[9] Y. Adachi and K. Yoshida, "Dynamic Generalized Jacobian Matrix for Non-Conservative Unconstrained Robot Manipulator", *Trans. of SICE*, Vol. 32, No. 2, pp. 207 - 214, 1996. (in Japanese)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Legged Robot Control using Multi-agent Robot Language

Makoto Obayashi, Hiroyuki Nishiyama and Fumio Mizoguchi
Faculty of Sci. and Tech.,
Science Univ. of Tokyo
2641 Yamazaki Noda, Chiba, Japan 278

Abstract

This paper proposes a method of applying Multi-agent Robot Language to legged-robot control. In general, a robot system requires many actuators and sensors to recognize the real environment and monitor its own condition. In particular, a legged robot that has many degrees of freedom and must achieve highly accuracte control requires many more variable sensors to realize robust movement than does a mobile robot using wheels. In many cases, a robot can detect its own condition from integrated sensor data. With this in mind, we adopted the following approach to efficiently deal with many sensors within the robot. (1) Monitor all sensors concurrently. (2) Integrate data from several sensors efficiently. (3) Construct a network of various sensors. We use multi-agent robot language (MRL) to perform the above. MRL enables us to implement a large-scale multi-agent system with ease. We assigned an agent to an each sensor and leg, and applied the multi-agent system to control a legged robot. We then examined the feasibility of this system.

1 Introduction

In general, an intelligent robot is one that has the necessary ability to make decisions in real time using feedback from many sensors and its own internal information. To realize the above, it is necessary to implement image processing using one or more cameras, planning, and various sensors to avoid obstacles and so on. Many researchers have reported robotic factors.

An internal software system for an intelligent robot must efficiently integrate information about robot factors and sensors. Also, an intelligent robot has to communicate with other robots in the real world. Brooks produced the subsumption architecture[3] for constructing an intelligent robot using a network of behavior modules. Some study of robotics internal system using multi-agent has been proposed[4]. It assigns each behavior module as an agent and constructs a brain for the robot.

These systems require concurrent processing of each module and asynchronous communication between processes. Also, if a robot consists of many computing machines, each robotics factor may be implemented on a different computer. In this case, the robotics internal system is considered to be composed of not only concurrent processing on a computer but also distributed computing via a local area network. In this study, we construct an internal software system for a robot using MRL[1]. An agent implemented using MRL performs asynchronous communication with other agents, making it possible to process data asynchronously and concurrently. We then apply this multi-agent system to a legged robot and examine its potential.

2 Multi-agent Robot Language

The Multi-agent Robot Language (MRL) is a concurrent logic language. It enables us to implement a large-scale multi-agent system with ease[2]. It was used in various applications, such as a multi-robot control system, control of a robotic room, and a computer security system. MRL allows a simple description for control of a single robot or multiple robots. Each agent described using MRL is processed asynchronously and concurrently and is able to perform asynchronous communication with other agents. MRL can describe behavior of robots and sensors, such as a set of logical expressions, and has the following features. (1) Concurrent processing of agents. (2) Communication mechanism between agents. A program described in MRL is translated into a C program by the MRL compiler. The C compiler then creates an executable file. MRL can be run on the various Unix operating systems. MRL employs stream, synchronous and asynchronous communication mechanisms that are based on broadcasting between agents. Therefore, it enables concurrent processing and communication between agents.

Also, MRL is able to open communication channels with other agents dynamically. Communications channels are then created for emergencies or cooperation between agents.

An MRL program consists of guard horn clauses such as the following.

```
Head :- Guard | Body
```

The guard and body consist of atomic formulas. The above clause means "Head is implied by Guard and Body." In operation, the clause indicates that "to prove Head, prove Guard and Body." Such semantics are just logic programming, except for the operational semantics of guard. Guard is used for clause selection to prove an atomic formula. Once a clause is selected, no other clauses are selected. Another use of guard is in the synchronization rule "execute Body when sufficient information to entailed Guard is obtained; otherwise wait until Guard is entailed." Thus, guard has a blocking statement that is checked in an active manner. If there are numerous atomic formulas in a body, they are executed concurrently. MRL exploits these operational semantics to concurrently control multiple robots.

2.1 Multi-agent Construction using MRL

A multi-agent system using MRL is constructed as a root-agent centered structure. The following describes creating an agent.

#<*agent name*>:new(*arg0, ...*)

where <*agent name*> is the newly created agent's name and *arg* are arguments set to constructor in the agent module. A new agent is to be a sub-agent of the agent that creates it. Hence, a multi-agent system using MRL forms a hierarchical structure.

2.2 Communication between agents

The MRL communication model between a super-agent and sub-agent is based on broadcast transmission. The communication structure in MRL allows asynchronous and stream communication. A message from a super-agent is transmitted to all sub-agents. An agent receiving a message from another agent decides its behavior according to the message contents.

MRL allows a simple description of message handling as shown below.

- ^<*Message*>
- *<*Message*>

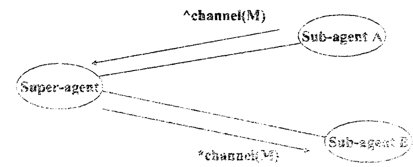


Figure 1: Dynamic channel creation

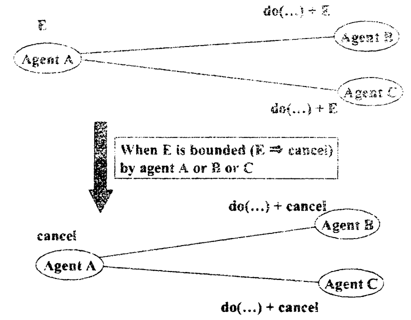


Figure 2: Emergency processing

Based on the guarded Horn clause semantics, the above messages have different meanings according to their described position. For example a message in guard may mean to receive a message, but the same message in the body means send a message to other agents.

2.3 Dynamic creation of the agent-communication channel

MRL allows dynamic creation of communication channels between agents, but not communication between a super-agent and a sub-agent. A communication channel is implemented as a shared variable between agents. Dynamic creation of communication channels is outlined in Fig.1. When agent A wants to establish a communications channel with agent B, agent A sends a variable to agent B via a super-agent, establishing the communications channel between agents A and B.

2.4 Emergency processing

Generally, a robot system's behavior must accurately reflect changing conditions in the real world. In particular, emergency processing is required in situations involving safety. In our study, the robot internal system consists of a multi-agent network. Hence, the agent that detects an emergency is often different from the agent that executes the emergency response. If multiple agents must execute an emergency process, messages must be transmitted immediately to the appropriate agents.

An outline of emergency processing in our multi-agent system using MRL is shown in Fig.2. In our system, the shared variable for emergency processing is sent to agents involved in the same emergency. The shared variable is sent to each agent via the normal agent communications channel. At first, the shared variable is unbound, but it is bound when an agent detects an emergency situation. All agents that have the same shared variable recognize the emergency at the same time and are immediately able to avoid the emergency.

2.5 Conflict resolution

A conflict resolution structure is required when behavioral conflicts between agents are possible. In MRL implementation, a conflict is resolved by providing an agent that checks and monitors the behavior of other agents.

3 Implementation

3.1 Experimental equipment

In our study, we apply our multi-agent system using MRL to a legged robot that has four legs with three degrees of freedom. The legged robot is equipped with 16 sonar sensors, a pan-tilt camera, and a range LAN.

3.2 System configuration

The multi-agent system configuration for legged-robot control is shown in Fig. 3.

The internal system of the legged-robot configuration consists of a supervisor agent and sensor agents, leg agents, and a state estimation agent. The supervisor agent only monitors and checks the behavior of other agents. Sensor agents watch each sensor and distribute sensor data to other agents demanding it. State estimation agents estimate the state of the robot using various sensor data. Leg agents watch the a state of and control each leg.

3.3 Legged-robot walking using communications between agents

This section describes the implementation of static walking of a legged robot using agents communica-

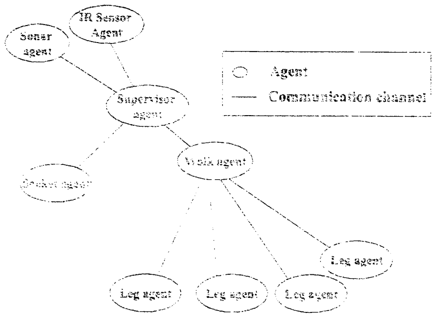


Figure 3: Agent system configuration

tion in a multi-agent system. This walking code using MRL corresponds to the behavior of a robot walking, considering where the leg is placed. In the walking process, one leg starts to swing, depending on the position of the legs. In this case, when one leg swings, other legs are prevented from swinging by conflict resolution through the super-agent. In walking, all legs share the information of the swinging leg. All supporting leg agents move according to the state of the swinging leg agent. When an emergency is detected, a flexible response is possible.

In this system, agent status is shared by the following method.

(a) Communication for conflict resolution

(b) Communication for collaboration behavior between legs

(c) Communication for emergent processing

Communication (a) is realized via the communications channel between the super-agent and sub-agent. In communication (b), a communications channel is shared by all leg -agents, and only the swinging leg agent updates its position according to its movement. In communication (c), a communication channel is shared by all leg -agents, just as in (b). All leg agents are able to bound the communication channel when they detect an emergency.

The sample code is as follows. The sample code, including the description of communications functions (a)-(c) for static walking of a legged robot, is as follows.

(% : comment out)

```
:- agent(leg).    % Create a leg agent
 new(Name) :-
    State = null,
    run(Name,State).
 %(1) Check conflict befor swinging a leg
 run(Name,null) :- swingable location |
    ^before(Name,Ret),
    NewState = before(Ret),
    run(Name,NewState).

 %(2) In the case of there is no
conflictœB!œ(Bestablish a communication channel
with
 % other agents and start to swing a leg
 run(Name,before(Ret)) :- Ret = ok |
    ^cooperate(Name, Com)+E,
    do swing,
    NewState = swing(Com)+E,
    run(Name,NewState).

 %(3) Write position data to communication
channel
 run(Name,swing(Com)+E) :- swinging |
    get location data -> Point
    Com<=Point,
    run(Name,swing(Com)+E).

 %(4) Establish communication channel from other
agents
 run(Name,null) :-  ^cooperate(Other, Com) |
    NewState = cooperate(Other, Com)+E,
    run(Name, NewState).

 %(5) Receive data from other agents
 run(Name,cooperate(Other,Com)+E) :- Com=>Point|
    do cooperate(Point),
    run(Name,cooperate(Other,Com)+E).
```

An MRL program consists of the predicate **new**, which is a constructor of an agent, and predicate **run**, which is described in the behavior code. Predicate **new** above is called by a super-agent, which sets **null** in the variable **state** and calls predicate **run**. Consequently, a walk agent is the super-agent of leg agents, and four leg agents are created. Predicate **run** in (1) avoids conflict using communication through the super-agent. In (1), the agent sends the **before** message to a super-agent. In this clause, variable **Ret** is a communications channel with a super-agent that detects and reports conflicts. In predicate (2), if there is no conflict between agents as the result of predicate (1), the agent requests other agents to cooperate and starts to swing the leg. In this clause, it sends a **cooperate** message to all leg agents through the super-agent for cooperation between agents. This message contains two communications channels, **E** and **Com**. **Com** is for cooperation, and **E** is for emergency processing. The cooperate message **E** is received by other leg agents in the predicate (4), and the leg agent swinging the leg updates its own position via communication channel **Com**. In predicate (5), all leg agents cooperate with other agents according to their states.

Predicate **run**, which enables emergency processing via communications channel **E** must be written before predicate (1).

```
 %(0) Emergent processing
 run(Name, _+E) :- E = emergency |
    do stop,
    run(Name, null).
alternatively.
```

When **alternatively** is available for priority processing, predicate **run** described before **alternatively** is processed in preference to other predicates. Hence, this system enables emergency processing during motion.

4 Conclusion

This paper has described the control of a four-legged robot using multi-agent robot language (MRL). In this method, control of each leg and sensor is assigned to an agent, and agents collaborate via a communications channel to realize static walking and emergency processing. A unique feature of this system is that the same agent can be defined as a leg agent or sensor agent. Finally, we have demonstrated that a multi-agent system using MRL enables flexible development for extension modules and addition of sensors.

References

[1] H. Nishiyama, H. Ohwada and F. Mizoguchi: A Multiagent Robot Language for Communication and Concurrency Control, *International Conference on Multiagent Systems(ICMAS)*, 1998.

[2] H. Nishiyama, H. Ohwada and F. Mizoguchi: Logic Specifications for Multiple Robots based on a Concurrent Programing Language, *Intelligent Robots and Systems(IROS)*, 1998.

[3] R.A. Brooks: A Robust layerd Control System for a Mobile Robot, *IEEE Journal on Robotics and Automation*, Vol.RA-2, No.1, pp.14-23, 1986.

[4] Ren C. Luo and Tse Min Chen: Multiagent and Event Driven Based Dynamic Collision Avoidance for an Autonomous Mobile Robot, *Intelligent Robots and Systems(IROS)*, 1998.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Control of Manipulator Mounted on Floating Underwater Robot while Maintaining Vehicle Attitude

M. Tamura S. Sagara R. Katoh
Department of Control Engineering
Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan

Abstract

In this paper, formulation of kinematics and dynamics, including the hydrodynamic forces, of free floating underwater robot (FFUR) model are treated. Furthermore, maintenance of robot vehicle attitude using active mass mounted on the robot vehicle is discussed. Computer simulation results show the possibility of the vehicle attitude maintenance by driving active mass.

Key Words: Floating underwater robot, Active mass, Vehicle attitude maintenance, Kinematics, Dynamics

1 Introduction

Underwater robots with manipulators have been expected to make important roles in ocean exploration. However, its development speed is not high, although recent years some underwater manipulators have been proposed [1,2,3,4]. For the development of this type of robots, fundamental and continuous researches must be done, and useful system model of this robot must be constructed. It is not easy to develop useful control methods of the manipulators, because the nonlinear forces, that is, drag forces, hydraulic forces, and so on, act on both the manipulators and the robot vehicle. Especially, as gravity forces and buoyancy forces continue to act on the robot system, it is suggested the robot system with manipulators may becomes unstable. In these meanings, it can be said that the control of manipulators mounted on floating underwater robot is more difficult than that of space manipulators.

This paper proposes a formulation of kinematics and dynamics of the free floating underwater robot with manipulators. And, the possibility that the attitude of the robot vehicle, which is influenced by gravity forces, buoyancy forces, and manipulator's motions, will be kept constant by motion of the active mass, while the manipulators complete the desired task.

2 Kinematics and Dynamics of FFUR

2.1 Assumptions

The underwater robot model shown in Fig. 1 is used in this paper. It has a robot base and multiple n-DOF manipulators, which can move in a static fluid space. Following assumptions are used in this paper:

(1) Robot system is in the static fluid.
(2) Robot system is independent of external system.
(3) Fluid forces are forces related to added mass tensor, added inertia tensor, buoyancy force, and fluid drag force [5].
(4) Fluid drag force is proportional to a squared link velocity and is perpendicular on the link.
(5) Total mass of the robot system including manipulators is equal to removal fluid mass.
(6) Robot base is a sphere and manipulator's link is a cylinder
(7) Degree of freedom of active mass, which is moved on a vehicle reference frame, is one.

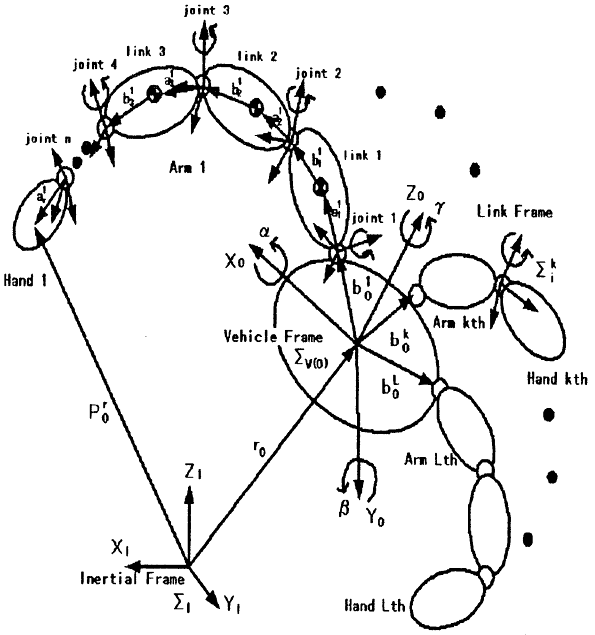


Fig. 1 Schema of Underwater Vehicle-Manipulator

Symbols used in this paper are defined as follows:

Σ_I : Inertial frame work

$\grave{\mathbf{E}}_V \in \mathbf{R}^6$: State vector of a robot vehicle body.

$\grave{\mathbf{E}}_M \in \mathbf{R}^{(n \times L)}$: State vector of manipulator's link.

$\mathbf{r}_i^k \in \mathbf{R}^3$: Position vector from an origin of Σ_I to center of mass of link i of arm k with respect to Σ_I.

$\bar{\mathbf{r}}_i^k \in \mathbf{R}^3$: Position vector from a center of mass of link i of arm k to a center of added mass of the link with respect to Σ_I.

$\mathbf{r}_r^k \in \mathbf{R}^3$: Vector denoting position and attitude of end-tip of arm k with respect to Σ_I.

$\mathbf{l}_i^k \in \mathbf{R}^3$: Length of link i of arm k.

$\mathbf{v}_0 \in \mathbf{R}^3$: Velocity vector of a mass vehicle with respect to Σ_I.

$\grave{\mathbf{u}}_0 \in \mathbf{R}^3$: Angular velocity vector of a mass center of the vehicle

$\mathbf{v}_i^k \in \mathbf{R}^3$: Velocity vector of a mass center of link i of arm k with respect to Σ_I.

$\grave{\mathbf{u}}_i^k \in \mathbf{R}^3$: Angular velocity vector of a mass center of link i of arm k with respect to Σ_I.

m_i^k : Mass of the link i. (link 0 means the vehicle)

$\mathbf{M}_i^k \in \mathbf{R}^{3\times3}$: added mass tensor of link i of arm k.

$\mathbf{I}_i^k \in \mathbf{R}^{3\times3}$: Inertia tensor of link i of arm k.

$\mathbf{I}_{ai}^k \in \mathbf{R}^{3\times3}$: Added inertia tensor of link i of arm k.

$\mathbf{F}_i^k \in \mathbf{R}^3$: Inertial force of link i of arm k.

$\mathbf{N}_i^k \in \mathbf{R}^3$: Inertial moment of link i of arm k.

2.2 Fluid forces

The drag force and moment of the joint i can be represented as follows:

(1) Fluid drag force $\mathbf{f}_{Di}^k$ and moment $\mathbf{n}_{Di}^k$ are described by

$$\mathbf{f}_{Di}^k = \frac{1}{2}\mathbf{C}_{Di}^k \rho \mathbf{D}_i^k \int_{-a_i^k}^{b_i^k} \mathbf{v}_{xi}^k \,|\, \mathbf{v}_{xi}^k \,|\, d\mathbf{x}_i^k \qquad \cdots(1)$$

$$\mathbf{n}_{Di}^k = \frac{1}{2}\mathbf{C}_{Di}^k \rho \mathbf{D}_i^k \int_{-a_i^k}^{b_i^k} \mathbf{x}_i^k \times \mathbf{v}_{xi}^k \,|\, \mathbf{v}_{xi}^k \,|\, d\mathbf{x}_i^k \qquad \cdots(2)$$

where $\mathbf{v}_{xi}^k = \dot{\mathbf{p}}_i^k + \grave{\mathbf{u}}_i^k \times \mathbf{x}_i^k$, $\tilde{n}$ is the fluid density, $\mathbf{C}_{Di}^k$ is the drag coefficient, and $\mathbf{D}_i^k$ is a height of the link i.

(2) Buoyancy force

The buoyancy force $\mathbf{F}_{bi}^k$ .can be written as follows:

$$\mathbf{F}_{bi}^k = \tilde{n} V_i^k \mathbf{g}. \qquad \cdots(3)$$

where V_i^k is a volume of the link i and $\mathbf{g}$ is an acceleration of the gravity.

2.3 Force and Moment acted between link i and link i+1 of arm k

(1) Force acted to link i , $\mathbf{F}_i^k$:

$$\mathbf{F}_i^k = \mathbf{f}_i^k - \mathbf{f}_{i+1}^k + \mathbf{f}_{Di}^k - \mathbf{m}_i^k \mathbf{g} + \mathbf{F}_{bi}^k \qquad \cdots(4)$$

(2) Moment acted to link i, $\mathbf{N}_i^k$:

$$\begin{aligned}\mathbf{N}_i^k = {} & \mathbf{n}_i^k - \mathbf{n}_{i+1}^k - \mathbf{l}_i^k \times \mathbf{f}_{i+1}^k - \mathbf{a}_i^k \times \{\mathbf{F}_i^k - \mathbf{f}_{Di}^k + \mathbf{m}_i^k \mathbf{g} - \mathbf{F}_{bi}^k\} \\ & + \mathbf{n}_{Di}^k + (\mathbf{l}_{bi}^k - \mathbf{a}_i^k) \times \mathbf{F}_{bi}^k\end{aligned} \qquad \cdots(5)$$

2.4 Momentum

Linear and angular momentum, $\mathbf{P}_i^k$ and $\mathbf{L}_i^k$, of the link i, which include hydrodynamic added mass and added inertia, are represented by

$$\mathbf{P}_i^k = m_i^k \mathbf{v}_i^k + \mathbf{M}_i^k (\mathbf{v}_i^k + \grave{\mathbf{u}}_i^k \times \bar{\mathbf{r}}_i^k) \qquad \cdots(6)$$

$$\begin{aligned}\mathbf{L}_i^k = {} & (\mathbf{I}_i^k + \mathbf{I}_{ai}^k)\grave{\mathbf{u}}_i^k + \mathbf{r}_i^k \times m_i^k \mathbf{v}_i^k \\ & + (\mathbf{r}_i^k + \bar{\mathbf{r}}_i^k) \times \{\mathbf{M}_i^k (\mathbf{v}_i^k + \grave{\mathbf{u}}_i^k \times \bar{\mathbf{r}}_i^k)\}\end{aligned} \qquad \cdots(7)$$

, and whole momentum of the system can be written by

$$\begin{bmatrix}\mathbf{P}\\ \mathbf{L}\end{bmatrix}=\begin{bmatrix}\mathbf{H}_{VPT} & \mathbf{H}_{VPR}\\ \mathbf{H}_{VLT} & \mathbf{H}_{VLR}\end{bmatrix}\begin{bmatrix}\mathbf{v}_0\\ \grave{\mathbf{u}}_0\end{bmatrix}+\begin{bmatrix}\cdots & \mathbf{H}^k_{MP} & \cdots\\ \cdots & \mathbf{H}^k_{ML} & \cdots\end{bmatrix}\check{\mathbf{E}}_M\,. \qquad \cdots(8)$$

2.5 Kinematics of the robot system

The end-tip position vector is

$$\mathbf{r}^k_r=\mathbf{r}_0+\mathbf{b}^k_0+\sum_{i=1}^{n}\mathbf{l}^k_i\,. \qquad \cdots(9)$$

A time derivative of the end-tip position vector of the manipulator and its attitude vector are derived using a conventional Jacobi Matrix as follows;

$$\begin{bmatrix}\mathbf{v}^k_r\\ \grave{\mathbf{u}}^k_r\end{bmatrix}=\begin{bmatrix}\mathbf{E}_3 & -\tilde{\mathbf{r}}^k_{0r}\\ \mathbf{0}_3 & \mathbf{E}_3\end{bmatrix}\begin{bmatrix}\mathbf{v}_0\\ \grave{\mathbf{u}}_0\end{bmatrix}+\begin{bmatrix}\mathbf{J}^k_{MT}\\ \mathbf{J}^k_{MR}\end{bmatrix}\check{\mathbf{E}}_M\,. \qquad \cdots(10)$$

3 Extended Kinematics

By differentiating Eqs.(8) and (10) with respect to time and arranging the momentum equation(8), the following relation can be obtained.

$$\mathbf{H}_{VPT}\dot{\mathbf{v}}_0+\tilde{\mathbf{H}}_{MP}\ddot{\check{\mathbf{E}}}_M=\dot{\tilde{\mathbf{P}}} \qquad \cdots(11)$$

$$\tilde{\mathbf{H}}_{VLR}\ddot{\breve{\mathbf{u}}}_0+\tilde{\mathbf{H}}_{ML}\ddot{\check{\mathbf{E}}}_M=\dot{\tilde{\mathbf{L}}} \qquad \cdots(12)$$

where

$$\tilde{\mathbf{H}}_{VLR}\equiv\mathbf{H}_{VLR}-\mathbf{H}_{VLT}\mathbf{H}^{-1}_{VPT}\mathbf{H}_{VPR}$$

$$\tilde{\mathbf{H}}_{ML}\equiv\mathbf{H}_{ML}-\mathbf{H}_{VLT}\mathbf{H}^{-1}_{VPT}\mathbf{H}_{MP}$$

$$\tilde{\mathbf{H}}_{MP}\equiv\mathbf{H}_{MP}-\mathbf{H}_{VPR}\tilde{\mathbf{H}}^{-1}_{VLR}\tilde{\mathbf{H}}_{ML}$$

$$\dot{\tilde{\mathbf{P}}}\equiv\dot{\hat{\mathbf{P}}}-\mathbf{H}_{VPR}\mathbf{H}^{-1}_{VLR}\dot{\tilde{\mathbf{L}}}$$

$$\dot{\tilde{\mathbf{L}}}\equiv\dot{\hat{\mathbf{L}}}-\mathbf{H}_{VLT}\mathbf{H}^{-1}_{VPT}\dot{\hat{\mathbf{P}}}$$

$$\dot{\hat{\mathbf{P}}}\equiv\dot{\mathbf{P}}-(\dot{\mathbf{H}}_{VPT}\mathbf{v}_0+\dot{\mathbf{H}}_{VPR}\grave{\mathbf{u}}_0+\dot{\mathbf{H}}_{MP}\check{\mathbf{E}}_M)$$

$$\dot{\hat{\mathbf{L}}}\equiv\dot{\mathbf{L}}-(\dot{\mathbf{H}}_{VLT}\mathbf{v}_0+\dot{\mathbf{H}}_{VLR}\grave{\mathbf{u}}_0+\dot{\mathbf{H}}_{ML}\check{\mathbf{E}})$$

Moreover, by substituting Eqs.(11) and (12) to Eq.(10), we derived the following relation

$$\tilde{\mathbf{J}}^k_M\ddot{\check{\mathbf{E}}}_M=\dot{\tilde{\mathbf{v}}}^k \qquad \cdots(13)$$

where

$$\tilde{\mathbf{J}}^k_M\equiv\mathbf{J}^k_M-\begin{pmatrix}\mathbf{H}^{-1}_{VPT}\tilde{\mathbf{H}}_{MP}-\tilde{\mathbf{r}}^k_{0r}\tilde{\mathbf{H}}^{-1}_{VLR}\tilde{\mathbf{H}}_{ML}\\ \tilde{\mathbf{H}}^{-1}_{VLR}\tilde{\mathbf{H}}_{ML}\end{pmatrix}\in\mathbf{R}^{6\times(n\times L)}$$

$$\dot{\tilde{\mathbf{v}}}^k\equiv\dot{\hat{\mathbf{v}}}^k-\begin{pmatrix}\mathbf{H}^{-1}_{VPT}\dot{\tilde{\mathbf{P}}}-\tilde{\mathbf{r}}^k_{0r}\mathbf{H}^{-1}_{VLR}\dot{\tilde{\mathbf{L}}}\\ \mathbf{H}^{-1}_{VLR}\dot{\tilde{\mathbf{L}}}\end{pmatrix}\in\mathbf{R}^{6}$$

Eq.(13) represents the relationship between the acceleration of the end-tip and the angular acceleration of the joint angles.

Now, it is very important to keep vehicle attitude for communicating with some ships on the sea. Here, it is considered how active mass should be driven to keep the vehicle attitude about roll and pitch axes. This problem should be considered based on the following equations;

$$\begin{bmatrix}\mathbf{0} & \tilde{\mathbf{J}}^k_M\\ \tilde{\mathbf{H}}'_{VLR} & \tilde{\mathbf{H}}_{ML}\end{bmatrix}\begin{bmatrix}\ddot{\breve{\mathbf{u}}}_{free}\\ \ddot{\check{\mathbf{E}}}_M\end{bmatrix}=\begin{bmatrix}\dot{\tilde{\mathbf{v}}}^k\\ \dot{\tilde{\mathbf{L}}}\end{bmatrix}-\begin{bmatrix}0\\ \tilde{\mathbf{H}}'_{VLR\omega}\ddot{\breve{\mathbf{u}}}_{target}\end{bmatrix} \qquad \cdots(14)$$

Therefore, vector $[\ddot{\breve{\mathbf{u}}}_{free}\ \ \ddot{\check{\mathbf{E}}}_M]^T$ is determined using the following equation.

$$\begin{bmatrix}\ddot{\breve{\mathbf{u}}}_{free}\\ \ddot{\check{\mathbf{E}}}_M\end{bmatrix}=\begin{bmatrix}\mathbf{0} & \tilde{\mathbf{J}}^k_M\\ \tilde{\mathbf{H}}'_{VLR} & \tilde{\mathbf{H}}_{ML}\end{bmatrix}^{+}\left(\begin{bmatrix}\dot{\tilde{\mathbf{v}}}^k\\ \dot{\tilde{\mathbf{L}}}\end{bmatrix}-\begin{bmatrix}0\\ \tilde{\mathbf{H}}'_{VLR\omega}\ddot{\breve{\mathbf{u}}}_{target}\end{bmatrix}\right). \qquad \cdots(15)$$

where symbol + denotes a pseudo-inverse(+) transformation of a matrix [*], and

$\tilde{\mathrm{H}}'_{VLR}$, $\ddot{\breve{\mathrm{u}}}_{free}$: Inertia matrix and angular velocity about yaw axis of the vehicle.

$\tilde{\mathrm{H}}'_{VLR\omega}$, $\ddot{\breve{\mathrm{u}}}_{target}$: Inertia matrix and angular velocity vector about roll and pitch axes of the vehicle

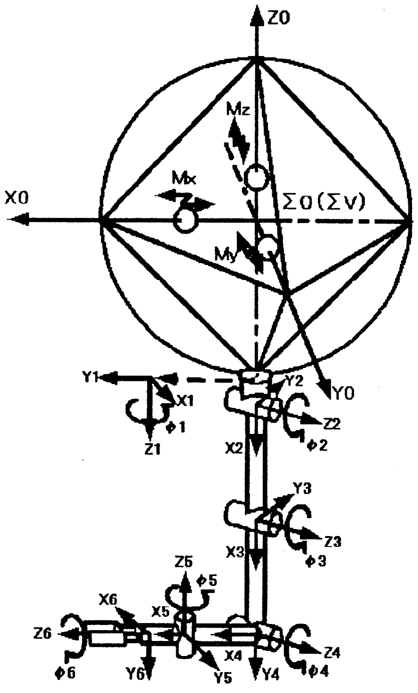


Fig.2 Simulation model of FFUR

4 Computer simulations

FFUR model used for computer simulation and its physical parameters are shown in Fig.2 and Table 1, respectively. Sampling time for simulation is 10ms.

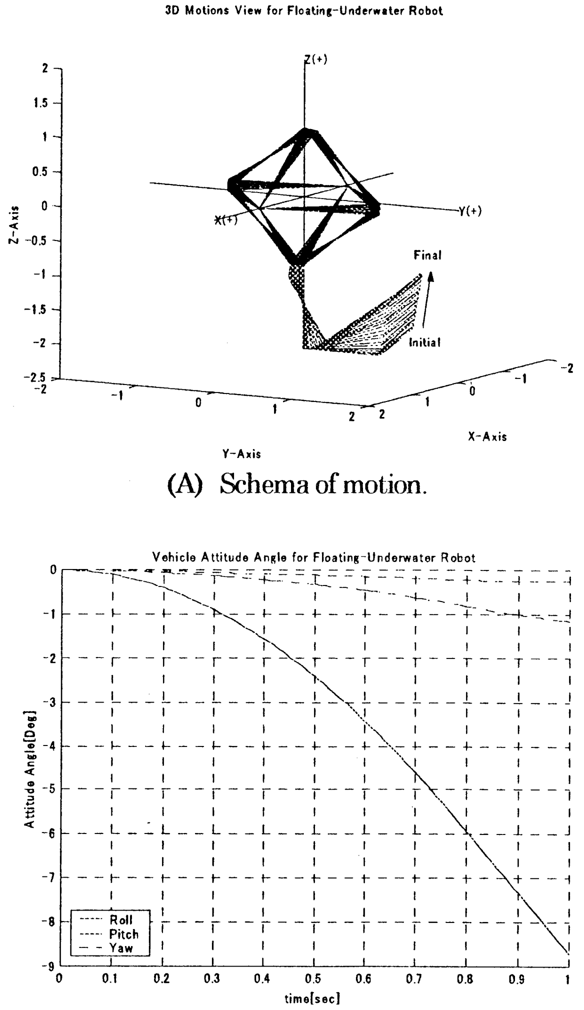
Table 1: Specification of FFUR model used for Computer Simulation

Link	Shape	L (m)	D (m)	M (kg)	Added M(kg)
0	Sphere		2.0	4109.8	2094.4 (x,y,z)
1	Cylinder	0.2	0.4	5.0	25. 1 (x,y)
2	Cylinder	1.0	0.1	50.0	7.9 (y,z)
3	Cylinder	1.0	0.1	50.0	7.9 (y,z)
4	Cylinder	0.2	0.02	5.0	0.06 (y,z)
5	Cylinder	0.2	0.02	5.0	0.06 (y,z)
6	Cylinder	0.2	0.02	5.0	0.06 (x,y)
7(X)	Particle			200.0	
8(Y)	Particle			200.0	
9(Z)	Particle			0.001	

Inertia (kgm^2)	Added inertia (kgm^2)	Drag
1644 1644 1644		0.44
0.067 0.067 0.1	0.34 0.34 0.5	1.1
0.06 4.2 4.2	0.0098 0.66 0.66	1.1
0.06 4.2 4.2	0.0098 0.66 0.66	1.1
0.00025 0.017 0.017	0.0 0.0002 0.0002	1.1
0.00025 0.017 0.017	0.0 0.0002 0.0002	1.1
0.017 0.017 0.00025	0.0002 0.0002 0.0	1.1

Fig.3 shows a motion of the FFUR whose base attitude is not constrained, while the manipulator is driven along a desired (straight) trajectory. It is clear that end-tip of the robot manipulator successfully reaches to the desired final position, while attitude change of the vehicle is not small. When this simulation is continued for long time span, robot manipulator often reaches to his singular configuration. This phenomena is caused by the gravity forces and the buoyancy forces acted in the robot. Then, some control methods should be developed to deduce this phenomena

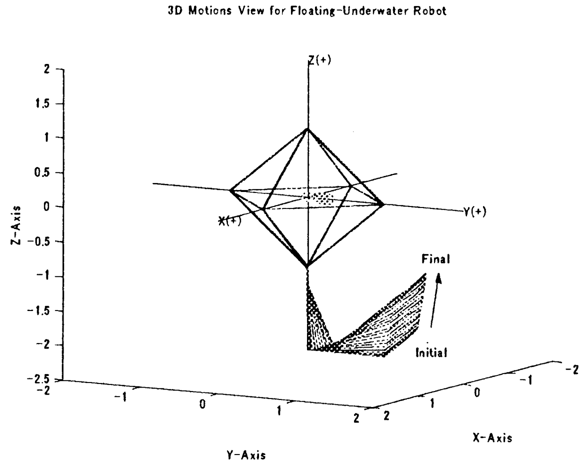
For this purpose, a method to make constrain the rotations about roll axis and pitch axis by the motion of the active mass is discussed. Fig.4 shows a typical simulation result. These figures make it clear that a tip-end of the manipulator reaches to the desired position, attitude angles, roll and pitch angles, are kept almost constant, and the manipulator doesn't reach to his singular configuration. Fig.4(B) and (C) show the time histories of the active mass position. In this simulation, the position is vibrated strongly. This vibration must be decreased by some feedback control.



(A) Schema of motion.

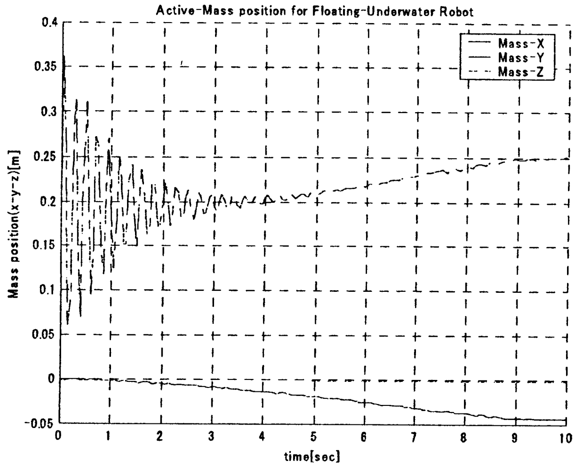
(B) Time history of vehicle attitude

Fig.3 Vehicle attitude free (without active mass).

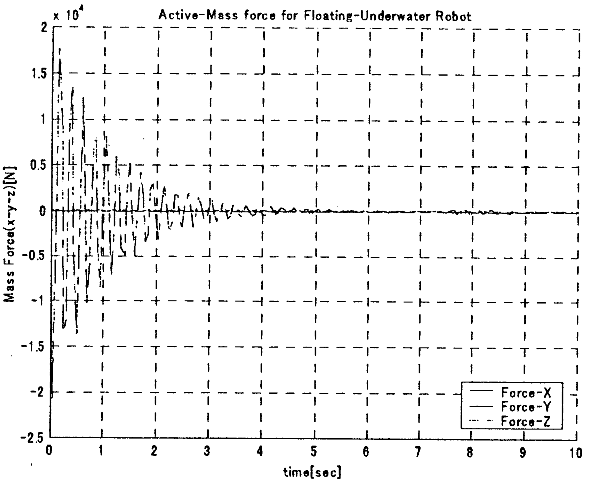


(A) Schema of motion.

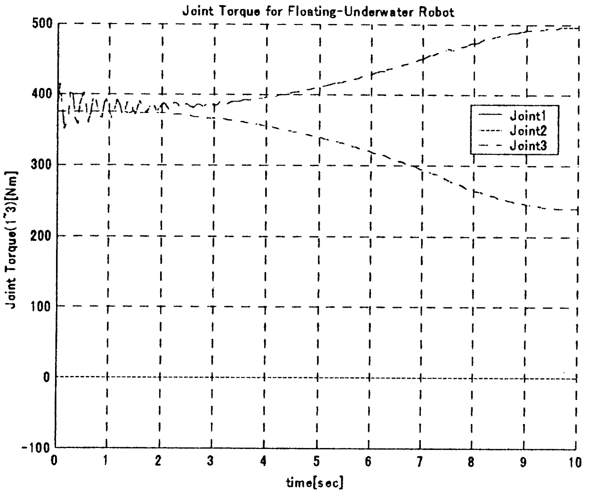
Fig.4 Vehicle attitude with active mass



(B) Time history of active mass position.



(C) Time history of active mass driving forces



(D) Time history of driving torques

Fig.4 Vehicle attitude maintenance.

5 Conclusion

In this paper, dynamic model of free floating undersea robot (FFUR) with active mass mounted on the vehicle was constructed, and inverse kinematic solutions of the robot, when some rotational motion about arbitrary axces were shown. Moreover, the dynamic motion of the robot system was simulated by computer simulation.

Control methods to control this type of robot with manipulators should be developed. To deduce the vibration of the active mass position is one of problems which are remained.

References

[1] K.Ioi and K.Itoh:"Modeling and Simulation of an underwater manipulator", Advanced Robotics, Vol.4, No.4, pp.303-317, 1990.

[2] McMillan,Scot, Orin,David E., and McGhee, Robert B. : “Efficient dynamic simulation of an underwater vehicle with a robotic manipulator”, J. of IEEE Trans. On System, Man and Cybernetics, 25, pp.1194-1206, 1995.

[3] Taurn Kanti Podder : “Dynamics and Control of Kinematically Redundant Underwater Vehicle - Manipulator Systems (for qualifying examination)” ME699 Report,Autonomous Systems Laboratory, University of Hawaii at Manoa, 1998.

[4] Sagara,Katoh,Tamura,Tanikawa:”Experiments of a Underwater Robot with 2 Link Manipulator”, Proc. Of AROB 5th, pp.367-370, 2000.

[5] Newman, J. N.: “Marine Hydrodynamics”, The MIT Press‘1977,9thPrinting 1999.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Achieving Synergy Through Acquisition of Human Skill

Thrishantha Nanayakkara*, Keigo Watanabe**, Kazuo Kiguchi** and Kiyotaka Izumi***
*Graduate School of Science and Engineering,
Faculty of Engineering Systems and Technology,
**Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
***Department of Mechanical Engineering,
Faculty of Science and Engineering,
Saga University, 1-Honjomachi, Saga 840-8502, Japan
†E-mail: watanabe@me.saga-u.ac.jp

Abstract

A concept is proposed to bring about synergy in work, by acquiring human skill and strategically mixing them to form robotic manipulation skills. The human skill is analyzed into situation - action pairs from many skilled workers. Synergy is expected to emerge from effective selection of actions from among them.

1 Introduction

The future of manufacturing systems will see a better degree of independence and intelligence that will substitute human workers specially in jobs with danger and boredom. This will require robots to demonstrate a higher degree of intelligence characterized by the ability to improve continuously and the ability to share knowledge to evolve synergy. Achieving this goal by the robots is impaired by the complexity and prohibitive processing capacity involved in motor control learning of humans. The control actions of humans for the same situation can vary across executions suggesting strong stochastic characteristics.

Many researchers have tried to model human control actions using hidden Markov Models (HMMs) [1], [2]. Yet HMMs are weak in the modeling of notions like satisfaction or motivation of human workers, that plays an important part in transition from one state to another or selection of actions based on the situation. In [3], the idea of temporal segmentation is explained. The term temporal segmentation refers to the idea that humans generate complex movements by piecing together shorter or simpler movements. One of the examples shows that in mental rotation of 3-D images, the time spent is proportional to the number of rotations. The idea of modularity in human motor control learning is discussed in [4]. These two ideas of modular formation of skill primitives and sequential execution of them, forms a basis for analyzing and transferring of many skilled motions to robots.

The motivation behind developing this approach is to enable a model to capture stochastic information of human control actions efficiently. The model reflects some basic characteristics of human decision making such as multi-stage processing and sub-conscious execution of learned actions for a given situation. Some important parts of human skill are analized and new definitions are established. An attempt is made to develop a model for acquisition of skills of many experts and achieve sŷnergy by a learning synergy network.

Rest of the paper is organized as follow. Section 2 gives an analysis of the essential features of human skillful motions. A method to achieve synergy is explained in section 3. Finally a discussion and some future proposals are made in section 4.

2 Analysis of Skilled Motions

It is very interesting to notice how human experts simplify complex tasks. First any task is divided into sub-tasks that can be executed independently and sequentially. This contracts the domain of attention at any given time to a minimum. Within a given task, there can be some set of movements. These movements may also be defined as behaviors.

We have examined one example of making a clay cup, which needs a considerable amount of skill. The skilled workers had divided the job into 11 independent tasks. We observed that in each task, the worker had to concentrate on a simple objective and one task had relatively similar set of movements and they varied with tasks. For example, the task to make clay tubes of 25cm length, involved a rolling motion, and the task of flattening the tube involved a pressing motion. Another important fact is that the skilled workers seemed to be unconsciously accounting for the qualitative progress of each task and made a transit from one task to another once they felt satisfied with the state of the work. In artificial skill acquisition,

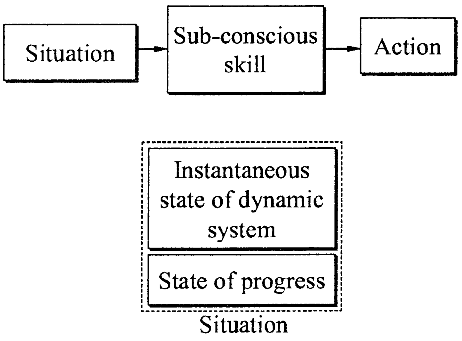


Figure 1: The basic idea of skillful action selection.

this notion of satisfaction is very important, because the final evaluation of the skill depends on the finish of the whole task that in turn depends on the finish of each task. The quality of the finish is strictly governed by the measure of satisfaction of the concerned skilled worker.

Based on the above observations, we would like to lay down the following characteristics of a skillful manipulation.

- A given job is analyzed into a optimum set of consecutive sub-tasks.
- Task execution consists of a sequence of compliant motions.
- A task is learnt through conscious effort and practice.
- There is a sub-conscious progress plan built through practice.
- Each person has his/her own measure of satisfaction and based on this abstract notion, they make transits form one task to another within a given job.
- A set of action variables can be defined for each control action. The values of action variables change during the sub-conscious execution of one control action. The pattern of variation of motion variables over time, follows some rules developed by experience, conscious thinking and aptitude.
- For a given situation, the variation of the motion variables in the resulting action can have stochastic variations specific to the worker.

Therefore a skilled motion is defined in space-time domain, and it is absolutely task specific.

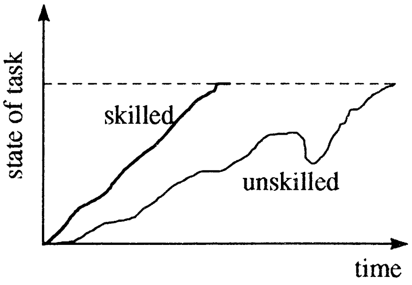


Figure 2: The concept of the *internal expected progress plan*.

2.1 Internal expected progress plan

The first important notion is the formation of an *internal expected progress plan* (IEPP). The progress of a given task can be defined as the variation of the essential observable characteristics of the job being done over time. For example in making the 25cm long clay tube, the characteristics can be the observation of length and the distribution of diameter along the tube. The IEPP of the worker is the expected variation of these characteristics over time. A skilled worker may achieve the threshold of these measures in a shorter time than the others. It is pictorially depicted in **Fig. 2**. It is very important to notice that the IEPP of a worker motivates corrective actions to keep the progress of the task. Therefore, in skill transfer to robots, it is of paramount importance to capture the IEPP of the worker concerned, through visual data.

2.2 Action variables

In conventional controllers, the control action is decided after considering state feedback in each sampling step. Sometimes this leads to control input chattering and actuator saturation problems. Yet in human control actions, it seems that the actions are far smoother due to the fact that the actions are not modified so fast. Once the situation is sensed, a predetermined action is executed so that the action variables change over an arbitrary period of time unconsciously. During execution the sub-conscious control action, skill primitives dominate over conscious attention. For example a swimmer's hand motion during one stroke is a predetermined motion than a constantly adapted action. Virtually no feedback is taken into consideration during action execution. The following **Fig. 3** shows a hypothetical variation of the action variables of force, position and velocity over time in a given situation.

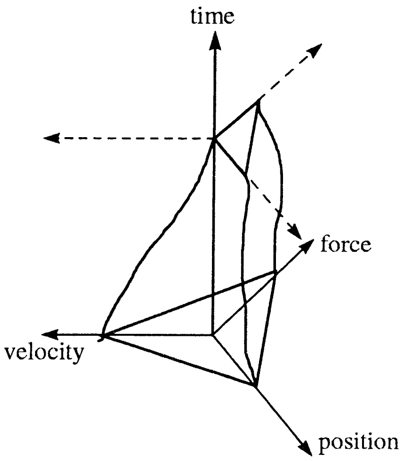


Figure 3: The concept of motion variables.

For the same situation, if the worker is asked to make a control action over and over again, the pattern of variation of motion variables seem to differ each time, yet belong to a specific stochastic distribution. For example in an expert swimmer's swimming action may show variations but there is an observable pattern that is different from another. The basic stochastic nature of human control strategy is discussed in [2].

2.3 Multi-stage decision making

We propose to use multi-stage fuzzy neural networks (FNNs) for action selection [5]. The multi-stage FNNs mimic the process of human decision making by considering the important factors first and based on these conclusions fine tuning on later stages. The basic pictorial view is given in **Fig. 4**. It is clear from Fig. 4 that the first stage decides the duration of the action. Finally the stage 2 of the FNNs will produce a variation of action variables similar to that shown in Fig. 3. The second stage generates actions for the time duration decided by the stage 1. The input for stage 2 consists of the situation, error of current situation with the IEPP, and the action feedback. Note that the information of situation and the progress error is given only once and that information is used together with the action feedback for iteration of actions over the time span decided by stage 1. The FNNs can be trained using the sensor data obtained during human demonstrations and the human linguistic rules. An evolutionary approach is proposed for guaranteed global convergence [6], [7]. This FNN architecture allows incorporation of the stochastic nature of human control actions and further improvement of the performance by modifying the IEPP.

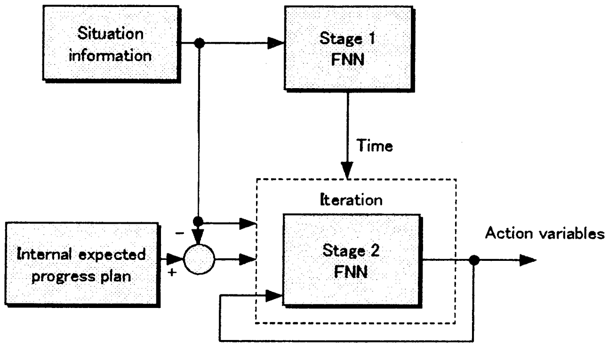


Figure 4: The concept using multi-stage FNNs for action selection.

3 Achieving Synergy

The concept of synergy can be stated as emergence of an extra ability to perform a certain job due to some combined effort of two or more elements with a hope of achieving a common objective. This is one important rationale behind making work groups to achieve a common objective. It is fair enough to argue that synergy emerges due to filling the weaknesses of each other in a group with the strengths of others and superimposing strengths to achieve a certain objective. This process allows extracting the best of strengths and exploiting the best of opportunities by reducing the weaknesses and threats to a minimum.

In the case of making control actions, the quality of the job done as a result of the control actions depend on the right variation of the individual action variables. Some workers may be good in the variation of position of the tool but weak in force exertion. Others may have other weaknesses and strengths. The proposed idea is to have a number of FNNs trained for many workers and try to select the optimum variation of action variables using a synergy network. The basic architecture of a synergy network is shown in **Fig. 5**, in which a^j_{vi}, $i = 1, \cdots, n$; $j = 1, \cdots, m$ is the ith action variable of jth expert in a given situation. In the classifier, these action variables of a number of experts are classified into groups of identical action variables. For example, if the action variables consist of force and position, the force variables are classified together and same is done for position variables. Then these classified information is given to neural networks (Qnets) assigned to select optimum action variables, that learn synergy using Q-learning. This problem is quite suitable for discrete Q-learning because the action variables from different experts can be considered as discrete actions [8].

The most challenging part is to design an effective reward function for Q-learning. This may involve a multi-objective linguistic evaluation function designed

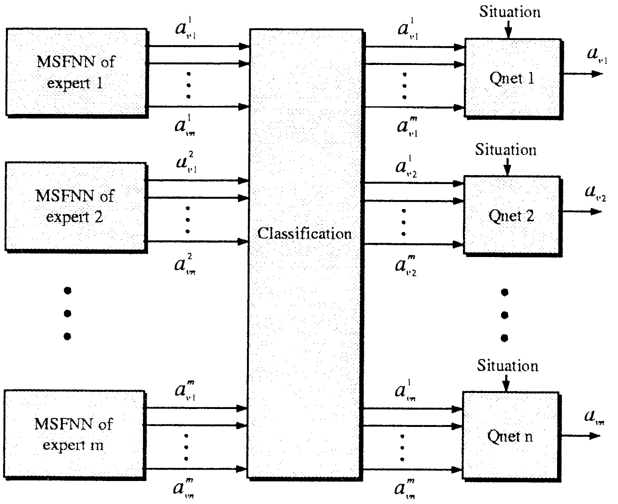


Figure 5: The concept of a synergy network.

using fuzzy sets. The information to design a fuzzy set based objective function can be obtained from the experts.

4 Discussion

The main difference of human control actions with conventional system controllers is that human control actions are started with conscious attension but executed for an arbitrary period of time without conscous attension. During the period of sub-counscious execution, the control action can be analysed into a set of action variables. Therefore we can define an action by the variation of action variables over the arbitrary period of time of sub-conscious execution. Acquisition of human skill involves learning how the action variables change and for how long they change, given a particular situation. For a given expert, these action variables may vary stochastically across executions, for the same situation. We have proposed multi-stage fuzzy neural networks as a method to capture the skilled variation of action variables of a particular expert.

After learning the control skills of sevaral experts, synergy networks could be trained based on reinforcement learning, to select optimum action variables. This is aimed at achieving synergy by selecting the best action variables among alternatives given by the experts.

Learning can be extended to the MSFNNs also, to improve the actions generated by them, while gradually modifying the internal expected progress plans for each MSFNN. This will allow continous development of the skills.

References

[1] M. Hiratsuka and H. H. Asada, "Detection of Human Mistakes and Misrepresentation for Human Perceptive Augmentation: Behavior Monitoring Using Hybrid Hidden Markov Models," in *Proc. of the IEEE conference on Robotics and Automation, San Fransisco, CA,* pp. 577–582, 2000.

[2] N. C. Nechyba and Y. Xu, "Stochastic Similarity for Validating Human Control Strategy Models," *IEEE Trans. on Robotics and Automation,* vol. 14, no. 3, pp. 437–451, 1998.

[3] J. A. Doeringer and N. Hogan, "Serial Processing in Human Movement Production," *Neural Networks*, vol. 11, pp. 1345 - 1356, 1998.

[4] D. M. Wolpert and M. Kawato "Multiple Paired Forward and Inverse Models for Motor Control," *Neural Networks*, vol. 11, pp. 1317 - 1329, 1998.

[5] F.-L. Chung and J.-C. Duan, "On Multistage Fuzzy Neural Network Modeling," *IEEE Trans. on Fuzzy Systems,* vol. 8, no. 2, pp. 125–142, 2000.

[6] T. Nanayakkara, K. Watanabe and K. Izumi, "Evolving in Dynamic Environments Through Adaptive Chaotic Mutation," in *Proc. of Fourth International Symposium on Artificial Life and Robotics,* vol. 2, 1999, pp. 520–523.

[7] T. Nanayakkara, K. Watanabe, and K. Izumi, "Evolving Runge-Kutta-Gill RBF Networks to Estimate the Dynamics of a Multi-Link Manipulator," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics,* vol. 2, 1999, pp. 770–775.

[8] C. J. C. H. Watkins, "Learning from Delayed Rewards," Ph.D. thesis, King's College, Cambridge, 1989.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

On the Fuzzy Model-Based Control for an Acrobot

Keigo Watanabe
Dept. of Adv. Syst. Cont. Eng.
Saga University
1-Honjomachi, Saga 840-8502
watanabe@me.saga-u.ac.jp

Kiyotaka Izumi
Dept. of Mech. Eng.
Saga University
1-Honjomachi, Saga 840-8502
izumi@me.saga-u.ac.jp

Kazuo Kiguchi
Dept. of Adv. Syst. Cont. Eng.
Saga University
1-Honjomachi, Saga 840-8502
kiguchi@me.saga-u.ac.jp

Abstract

In this paper, we describe a new control method for the Acrobot. In particular, we poropose a fuzzy model-based controller as a second-stage controller to extend the attractive domain where the second-stage controller can be applied, whereas a computed torque controller is used as the first-stage controller.

Keywords—Acrobatic robots, Underactuated robots, Computed torque control, Fuzzy models, Fuzzy control

1 Introduction

The study of underactuated robotic systems has received significant attention over the last fewer years. The underactuated robots have fewer actuators than degrees of freedom, so that a conventional control strategy is not directly applied to control or stabilize such robots. Also, such robotic systems can be interpreted as systems with second-order nonholonomic constraints, instead of first-order nonholonomic constraints [1]. Nevertheless, the challenge of solving control problems associated with such robots will stimulate new results in robot control theory, because such robotic systems include saving weight and energy by using fewer actuators and gaining fault tolerance to actuator failure.

The Acrobot for acrobatic robot is a simple example of an underactuated robot manipulator, which consists of a double pendulum with an actuator at only the second joint. This was originally realized as a simple model of a human gymnast on a high bar, where the underactuated first joint modeled the degree of freedom of a gymnast's hands on the bar, while the second joint modeled the gymnast's hip. The similar mechanism can be found in the study of so-called brachiation robot [2, 3]. Although there exist several control approaches to control this type of robot [4], a representative approach is a result of Spong [5], in which a two-stage control method has been proposed: the first is the use of a nonlinear controller called as the "computed torque control," and the second is the use of a LQR controller. For a swing up control of Acrobot, the first link can be moved to the vertical by the computed torque control which achieves the linearization of the first link, and an LQR controller is switched to capture and balance the Acrobot about the vértical if the second link can rotate more than 2π and the Acrobot rotates close to the vertical. However, note that it is not easy to tune the control parameters to accomplish the capture and balance phase successfully, because the attractive domain of the LQR controller is very small. From this point of view, Xin and Kaneda [6] proposed a method for the second stage controller, in which they treated the non-zero speed of the second link crossing the vertical as an uncertainty under the assumption that the position of the second link has been already closed to the vertical sufficiently, and designed a robust controller to cope with such uncertainty. They showed that their proposed method can increase the attractive domain for applying the second-stage controller, compared to the LQR method.

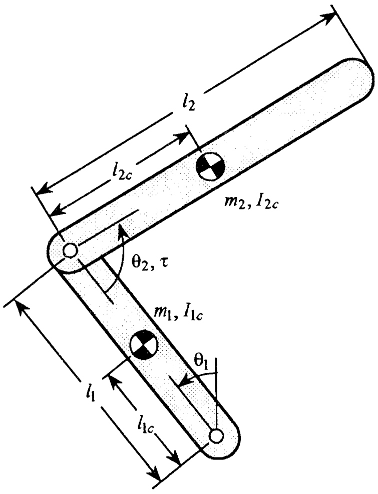


Figure 1: Acrobot parameters

In this paper, following the line of the above approaches, we propose a fuzzy model-based controller as a second-stage controller to extend the attractive domain that the second-stage controller can be applied. The present fuzzy controller is designed by using the Takagi-Sugeno (T-S) fuzzy models [7], whose linear models are derived through sector approximations of nonlinear terms, which are included in the dynamic model after achieving that the first link moves to the vertical by the computed torque control. As a result, it is shown that the attractive domain for applying the second-stage controller is much wider than those of the LQR and a robust control method in [6]: in particular, the joint position of the second link can be attractive within $\pm\pi/2$ [rad] in maximum about the vertical.

2 Acrobot Model

Figure 1 shows the notation and conventions used for the Acrobot parameters and variables. The equations of motion are written in the form below. $M(\theta)$ is symmetric and positive definite, and $C(\theta, \dot{\theta})$ has been chosen such that $\dot{M} - 2C \cdot$

is skew symmetric.

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta) = \begin{bmatrix}0\\ \tau\end{bmatrix} \tag{1}$$

$$\begin{aligned}
M(\theta) &= \begin{bmatrix} c_1 + c_2 - 2c_3\cos\theta_2 & c_2 - c_3\cos\theta_2 \\ c_2 - c_3\cos\theta_2 & c_2 \end{bmatrix} \\
C(\theta,\dot{\theta}) &= \begin{bmatrix} c_3\dot{\theta}_2\sin\theta_2 & c_3(\dot{\theta}_1+\dot{\theta}_2)\sin\theta_2 \\ -c_3\dot{\theta}_1\sin\theta_2 & 0 \end{bmatrix} \\
N(\theta) &= \begin{bmatrix} -c_4\sin\theta_1 + c_5\sin(\theta_1+\theta_2) \\ c_5\sin(\theta_1+\theta_2) \end{bmatrix}
\end{aligned}$$

where

$$\begin{aligned}
c_1 &= m_1 l_{1c}^2 + m_2 l_1^2 + I_{1c}, \quad c_2 = m_2 l_{2c}^2 + I_{2c} \\
c_3 &= m_2 l_1 l_{2c} \\
c_4 &= (m_1 l_{1c} + m_2 l_1)g, \quad c_5 = m_2 g l_{2c}
\end{aligned}$$

with

$$\theta^T = [\theta_1 \;\; \theta_2].$$

Now, write the Acrobot equations of motion in the following form:

$$\ddot{\theta} = -M^{-1}(\theta)\left\{C(\theta,\dot{\theta})\dot{\theta} + N(\theta)\right\} + M^{-1}(\theta)\begin{bmatrix}0\\1\end{bmatrix}\tau \tag{2}$$

with

$$M^{-1} = \frac{1}{D}\begin{bmatrix} m_{22} & -m_{12} \\ -m_{12} & m_{11} \end{bmatrix}, \quad D = m_{11}m_{22} - m_{12}^2 \neq 0 \tag{3}$$

where

$$\begin{aligned}
m_{11} &= c_1 + c_2 - 2c_3\cos\theta_2 \\
m_{12} &= c_2 - c_3\cos\theta_2, \quad m_{22} = c_2
\end{aligned} \tag{4}$$

Letting the following state variables,

$$\boldsymbol{x}^T = [x_1 \;\; x_2 \;\; x_3 \;\; x_4] = [\theta_1 \;\; \theta_2 - \pi \;\; \theta_3 \;\; \theta_4] \tag{5}$$

we obtain

$$\begin{aligned}
\dot{x}_1 &= x_3 && (6)\\
\dot{x}_2 &= x_4 && (7)\\
\dot{x}_3 &= -\frac{m_{22}}{D}(\bar{c}_1 + n_1) + \frac{m_{12}}{D}(\bar{c}_2 + n_2) - \frac{m_{12}}{D}\tau && (8)\\
\dot{x}_4 &= \frac{m_{12}}{D}(\bar{c}_1 + n_1) - \frac{m_{11}}{D}(\bar{c}_2 + n_2) + \frac{m_{11}}{D}\tau && (9)
\end{aligned}$$

with

$$\begin{aligned}
m_{11} &= c_1 + c_2 + 2c_3\cos x_2 \\
m_{12} &= c_2 + c_3\cos x_2, \quad m_{22} = c_2 \\
\bar{c}_1 &= -2c_3 x_3 x_4 \sin x_2 - c_3 x_4^2 \sin x_2 \\
\bar{c}_2 &= c_3 x_3^2 \sin x_2 \\
n_1 &= -c_4\sin x_1 - c_5\sin(x_1 + x_2) \\
n_2 &= -c_5\sin(x_1 + x_2)
\end{aligned} \tag{10}$$

because $\cos(x_2 + \pi) = -\cos(x_2)$ and $\sin(x_2 + \pi) = -\sin(x_2)$.

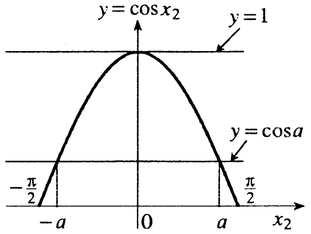


Figure 2: Approximation of $\cos x_2$

3 Computed Torque Control of Link 1

Considering the computed torque control of $\ddot{\theta}_1$ (i.e., $\ddot{x}_1$) directly:

$$\ddot{x}_1 = -\frac{m_{22}}{D}(\bar{c}_1 + n_1) + \frac{m_{12}}{D}(\bar{c}_2 + n_2) - \frac{m_{12}}{D}\tau \tag{11}$$

we have

$$\tau = -\frac{D}{m_{12}}\left[\ddot{x}_1^* + \frac{m_{22}}{D}(\bar{c}_1 + n_1) - \frac{m_{12}}{D}(\bar{c}_2 + n_2)\right] \tag{12}$$

where

$$\ddot{x}_1^* = \ddot{x}_1^d + k_p(x_1^d - x_1) + k_v(\dot{x}_1^d - \dot{x}_1) \tag{13}$$

Then, it follows that

$$\begin{aligned}
\ddot{e}_1 &+ k_v\dot{e}_1 + k_p e_1 = 0 && (14)\\
e_1 &\overset{\triangle}{=} x_1^d - x_1 && (15)
\end{aligned}$$

and $\lim_{t\to\infty} e_1(t) = 0$ if $k_v, k_p > 0$. Furthermore, since $\dot{e}_1 = \dot{x}_1^d - \dot{x}_1 = x_3^d - x_3 \overset{\triangle}{=} e_3$, it is found that

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_3\end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_v \end{bmatrix}\begin{bmatrix}e_1\\ e_3\end{bmatrix} \tag{16}$$

whose roots $\lambda_{1,2} = (-k_v \pm \sqrt{k_v^2 - 4k_p})/2$ assure that $\lim_{t\to\infty} e_3 = 0$, because $\mathrm{Re}(\lambda_{1,2}) < 0$. Note that if we want to obtain the critical damped response then it is sufficient to give a condition such that $k_v^2 = 4k_p$.

4 Fuzzy Modeling of Nonlinear Terms

4.1 Approximations of nonlinear terms

It is assumed that

$$x_2 \in [-a, a], \quad x_4 \in [-b, b]$$

where $|a| \leq \pi/2$ and b are positive values. Approximate the nonlinear terms $\cos x_2$ and x_4^2 by using two linear models, respectively, as shown in Figs. 2 and 3. That is, they can be represented by

$$\begin{aligned}
\cos x_2 &= F_1^1(x_2)\cdot 1 + F_1^2(x_2)\cdot\cos a && (17)\\
x_4^2 &= F_2^1(x_4)\cdot b^2 + F_2^2(x_4)\cdot 0 && (18)
\end{aligned}$$

where

$$\begin{gathered}
F_1^1(x_2), F_1^2(x_2), F_2^1(x_4), F_2^2(x_4) \in [0,1] \\
F_1^1(x_2) + F_1^2(x_2) = 1, \quad F_2^1(x_4) + F_2^2(x_4) = 1
\end{gathered}$$

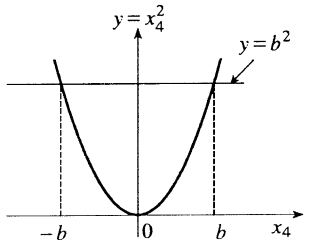


Figure 3: Approximation of x_4^2

Solving the above equations gives

$$F_1^1(x_2) = \frac{\cos x_2 - \cos a}{1 - \cos a}$$

$$F_1^2(x_2) = 1 - F_1^1(x_2) = \frac{1 - \cos x_2}{1 - \cos a}$$

$$F_2^1(x_4) = \frac{x_4^2}{b^2}, \quad F_2^2(x_4) = 1 - F_2^1(x_4) = 1 - \frac{x_4^2}{b^2}$$

It should be note that if $x_1 \approx 0$, then

$$\begin{aligned} \sin(x_1 + x_2) &= x_1 \cos x_2 + \sin x_2 \\ &\equiv x_1 \cos x_2 + \mathrm{sgn}(x_2)\sqrt{1 - \cos^2 x_2} \end{aligned}$$

where

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{19}$$

4.2 Fuzzy rules in model parameters

When $x_1 = x_3 \approx 0$, the fuzzy rules for the model parameters are as follows, depending on the positive or negative sign of x_2:

A) For $x_2 > 0$, we have

Model Parameter Rule 1:
IF $x_2(t)$ is F_1^1 and x_4 is F_2^1
THEN $m_{11} = c_1 + c_2 + 2c_3, \quad m_{12} = c_2 + c_3,$
$m_{22} = c_2, \quad \bar{c}_1 = 0, \quad \bar{c}_2 = 0, \quad n_1 = -c_4 x_1 - c_5 x_1,$
$n_2 = -c_5 x_1$

Model Parameter Rule 2:
IF $x_2(t)$ is F_1^1 and x_4 is F_2^2
THEN $m_{11} = c_1 + c_2 + 2c_3, \quad m_{12} = c_2 + c_3,$
$m_{22} = c_2, \quad \bar{c}_1 = 0, \quad \bar{c}_2 = 0, \quad n_1 = -c_4 x_1 - c_5 x_1,$
$n_2 = -c_5 x_1$

Model Parameter Rule 3:
IF $x_2(t)$ is F_1^2 and x_4 is F_2^1
THEN $m_{11} = c_1 + c_2 + 2c_3 \cos a, \quad m_{12} = c_2 + c_3 \cos a,$
$m_{22} = c_2, \quad \bar{c}_1 = -c_3 b^2 \sqrt{1 - \cos^2 a}, \quad \bar{c}_2 = 0$
$n_1 = -c_4 x_1 - c_5[\cos a x_1 + \sqrt{1 - \cos^2 a}]$
$n_2 = -c_5[\cos a x_1 + \sqrt{1 - \cos^2 a}]$

Model Parameter Rule 4:
IF $x_2(t)$ is F_1^2 and x_4 is F_2^2
THEN $m_{11} = c_1 + c_2 + 2c_3 \cos a, \quad m_{12} = c_2 + c_3 \cos a,$
$m_{22} = c_2, \quad \bar{c}_1 = 0, \quad \bar{c}_2 = 0$
$n_1 = -c_4 x_1 - c_5[\cos a x_1 + \sqrt{1 - \cos^2 a}]$
$n_2 = -c_5[\cos a x_1 + \sqrt{1 - \cos^2 a}]$

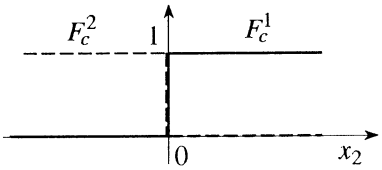


Figure 4: Crisp sets representing the sign of x_2

B) For $x_2 < 0$, it follows that $\sqrt{1 - \cos^2 a}$ in Models 3 an 4 of $x_2 > 0$ should be replaced by $-\sqrt{1 - \cos^2 a}$. Moreover, it should be noted that the above Models 1 and 2 are independent of x_4, and they are also independent of sign of x_2.

In order to represent all the above rules without using a separation logic, introduce a new variable transformation such as

$$r(t) = \mathrm{sgn}(x_2) \tag{20}$$

Then, define a crisp set that respresents the positive region of x_2 such that

$$F_c^1(r(t)) = \delta(r(t) - 1)$$

and its complement

$$F_c^2(r(t)) = 1 - F_c^1(r(t))$$

which is shown in Fig. 4. Here, $\delta(\cdot)$ is the Dirac delta function defined by

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

4.3 T-S fuzzy models

Unifying the separation logic with a variable $r(t)$, we have the following T-S fuzzy models:

System Model Rule 1:
IF $r(t)$ is (F_c^1 or F_c^2) and $x_2(t)$ is F_1^1 and x_4 is (F_2^1 or F_2^2)
THEN $\dot{\boldsymbol{x}}(t) = A_1 \boldsymbol{x}(t) + B_1 \tau(t) + \boldsymbol{d}_1$

System Model Rule 2:
IF $r(t)$ is F_c^1 and $x_2(t)$ is F_1^2 and x_4 is F_2^1
THEN $\dot{\boldsymbol{x}}(t) = A_2 \boldsymbol{x}(t) + B_2 \tau(t) + \boldsymbol{d}_2$

System Model Rule 3:
IF $r(t)$ is F_c^1 and $x_2(t)$ is F_1^2 and x_4 is F_2^2
THEN $\dot{\boldsymbol{x}}(t) = A_3 \boldsymbol{x}(t) + B_3 \tau(t) + \boldsymbol{d}_3$

System Model Rule 4:
IF $r(t)$ is F_c^2 and $x_2(t)$ is F_1^2 and x_4 is F_2^1
THEN $\dot{\boldsymbol{x}}(t) = A_4 \boldsymbol{x}(t) + B_4 \tau(t) + \boldsymbol{d}_4$

System Model Rule 5:
IF $r(t)$ is F_c^2 and $x_2(t)$ is F_1^2 and x_4 is F_2^2
THEN $\dot{\boldsymbol{x}}(t) = A_5 \boldsymbol{x}(t) + B_5 \tau(t) + \boldsymbol{d}_5$

where

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ [m_{22}(c_4 + c_5) - m_{12}c_5]/D & 0 & 0 & 0 \\ [m_{11}c_5 - m_{12}(c_4 + c_5)]/D & 0 & 0 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ -m_{12}/D \\ m_{11}/D \end{bmatrix}, \quad \boldsymbol{d}_1 = \boldsymbol{0}$$

with $m_{11} = c_1 + c_2 + 2c_3, m_{12} = c_2 + c_3, m_{22} = c_2$, and $D = m_{11}m_{22} - m_{12}^2$;

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ [m_{22}(c_4 + c_5 \cos a) - m_{12}c_5 \cos a]/D & 0 & 0 & 0 \\ [m_{11}c_5 \cos a - m_{12}(c_4 + c_5 \cos a)]/D & 0 & 0 & 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ -m_{12}/D \\ m_{11}/D \end{bmatrix}$$

$$\boldsymbol{d}_2 = \begin{bmatrix} 0 \\ 0 \\ [m_{22}(c_3 b^2 + c_5) - m_{12}c_5]\sqrt{1 - \cos^2 a}/D \\ [m_{11}c_5 - m_{12}(c_3 b^2 + c_5)]\sqrt{1 - \cos^2 a}/D \end{bmatrix}$$

with $m_{11} = c_1 + c_2 + 2c_3 \cos a, m_{12} = c_2 + c_3 \cos a, m_{22} = c_2$, and $D = m_{11}m_{22} - m_{12}^2$;

$$A_3 = A_2, \quad B_3 = B_2$$

$$\boldsymbol{d}_3 = \begin{bmatrix} 0 \\ 0 \\ (m_{22} - m_{12})c_5\sqrt{1 - \cos^2 a}/D \\ (m_{11} - m_{12})c_5\sqrt{1 - \cos^2 a}/D \end{bmatrix}$$

with the same parameters as those in Model 2;

$$A_4 = A_2, \quad B_4 = B_2$$

$$\boldsymbol{d}_4 = \begin{bmatrix} 0 \\ 0 \\ [m_{12}c_5 - m_{22}(c_3 b^2 + c_5]\sqrt{1 - \cos^2 a}/D \\ [m_{12}(c_3 b^2 + c_5) - m_{11}c_5]\sqrt{1 - \cos^2 a}/D \end{bmatrix}$$

with the same parameters as those in Model 2;

$$A_5 = A_2, \quad B_5 = B_2$$

$$\boldsymbol{d}_5 = \begin{bmatrix} 0 \\ 0 \\ (m_{12} - m_{22})c_5\sqrt{1 - \cos^2 a}/D \\ (m_{12} - m_{11})c_5\sqrt{1 - \cos^2 a}/D \end{bmatrix}$$

with the same parameters as those in Model 2.

4.4 Reduction of T-S fuzzy model rules

In the above rules, it is found that the information on $r(t)$ and x_4 contribute only the modeling error $\boldsymbol{d}_i$. From this fact, we can finally consider only the following two rules:

System Model Rule 1:
IF $x_2(t)$ is F_1^1
THEN $\dot{\boldsymbol{x}}(t) = A_1\boldsymbol{x}(t) + B_1\tau(t) + \boldsymbol{d}_1$

System Model Rule 2:
IF $x_2(t)$ is F_1^2
THEN $\dot{\boldsymbol{x}}(t) = A_2\boldsymbol{x}(t) + B_2\tau(t) + \boldsymbol{d}_2$

5 Conclusions

We have considered a new control method for the Acrobot, which consists of a double pendulum with an actuator at only the second joint. We poroposed a fuzzy model-based controller as a second-stage controller to extend the attractive domain where the second-stage controller can be applied, whereas a computed torque controller is used as the first-stage controller. It was shown that the attractive domain for applying the second-stage controller is much wider than those of the LQR and a robust control method in [6].

References

[1] Kolmanovsky, I. and McClamroch, N. H. (1995), Developments in Nonholonomic Control Problems. *IEEE Control Systems Magazine* 15(6): 20–36

[2] Saito, T., Fukuda, T., and Arai, F. (1994), Swing and Locomotion Control for a Two-Link Brachiation Robot. *IEEE Control Systems Magazine* 14(1): 5–12

[3] Nakanishi, J., Fukuda, T., and Koditschek, D. E. (2000), A Brachiating Robot Controller. *IEEE Trans. on Robotics and Automation* 16(2): 109–123

[4] Berkemeier, M. D. and Fearing, R. S. (1999), Tracking Fast Inverted Trajectories of the Underactuated Acrobot. *IEEE Trans. on Robotics and Automation* 15(4): 740–750

[5] Spong, M. W. (1995), The Swing Up Control Problem for the Acrobot. *IEEE Control Systems Magazine* 15(1): 49–55

[6] Xin, X. and Kaneda, M. (2000), Robust Control Design for the Swing up Control Problem for the Acrobot. In *Preprints of The 43rd Japan Joint Automatic Control Conference*, Matsuyama, Japan, November 2000, pp. 131–134.

[7] Tanaka, K., Ikeda, T., and Wang, H. O. (1998), Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Designs. *IEEE Trans. on Fuzzy Systems* 6(2): 250–265

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Control for a Rings Gymnastic Robot Using Fuzzy Reasoning and Genetic Algorithms

Takaaki Yamada*, Keigo Watanabe*, Kazuo Kiguchi*, and Kiyotaka Izumi**
*Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
**Department of Mechanical Engineering,
Faculty of Science and Engineering,
Saga University, 1-Honjomachi, Saga 840-8502, Japan
†E-mail : 00td53@edu.cc.saga-u.ac.jp

Abstract

To the best of our knowledge there is no work done on ring exercises. The ring exercises have free-floating characteristics of gripping point. By clarifying a motion control method in a robot suspended by an unformed object such as a rope, it is expected that the applicable range of robot will be expanded, if the robot is realized in practice. Therefore, this paper proposes a novel "rings gymnastic robot" and acquires fuzzy rules to realize performances on the rings considering torque minimization by genetic algorithms (GA), because the rules are considered to be useful for a coaching of ring exercises. The effectiveness of the obtained controller is illustrated by a simulation.

1 Introduction

Recently, making a control method of skillful motions in human beings and animals clear, the realization of such motions using robots has been studied actively. Horizontal bar gymnastic robot (sometimes called Acrobot) [1, 2] and brachiation robot [3] are the representative studies. In particular, there are various studies on the horizontal bar gymnastic robot.

On the other hand, as regards rings known as one event of men's apparatus gymnastics (see **Fig. 1**), although it has been adopted as an event since the first Olympic Games in Athens 1896, there are no studies on its motion analysis and realization as a robot to our knowledge. The ring exercises have the characteristics that the apparatus (namely rings) can move in all directions freely (i.e. free-floating characteristics of gripping point) [4]. This is a primary difference from the horizontal bar exercises that have the constraint between hands and the horizontal bar. By clarifying a motion control method in a robot suspended by an unformed object such as a rope, it is expected that the applicable range of robot will be expanded, if the robot can be realized in practice. Therefore, we propose a novel "rings gymnastic robot" that performs gripping the rings hung by a rope.

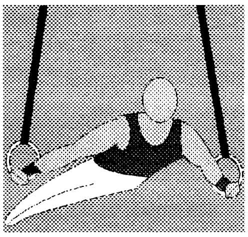


Figure 1: Ring exercises

Also, it is considered that gymnast's performance, which is acquired by day-to-day practices, is highly efficient without using too much power. In this paper, we acquire fuzzy rules to realize a back giant that is minimum joint torque as well as suitable for transition to a handstand. After deciding the rules for the back giant by GA using three kinds of evaluations including torque minimization, fuzzy rules for the handstand are determined by GA. The effectiveness of the obtained fuzzy controller is illustrated by a simulation.

2 Equations of Motion

Although the ring exercises are complex three-dimensional motions essentially, it is assumed that the rings gymnastic robot moves in two-dimensional plane. Furthermore, using a two-link robot as the simplest model, a double physical pendulum suspended by a rope as shown in **Fig. 2** can be considered.

Letting the generalized coordinate be $\boldsymbol{q} = [x_1\ z_1\ \theta_1\ \theta_2]^T$, the equations of motion derived using the Lagrangian formulation are given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Q} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is a 4×4 symmetric mass matrix, $\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is a 4×1 vector of centrifugal and Coriolis terms, $\boldsymbol{g}(\boldsymbol{q})$ is a 4×1 vector of gravity terms, and $\boldsymbol{Q}$ is a 4×1 vector of generalized force terms. Elements of the matrix and vectors are represented as follows:

$$M_{11} = M_{22} = m_1 + m_2, \quad M_{12} = 0$$
$$M_{13} = m_2 l'_{g1} \cos\theta_1 + m_2 l_{g2} \cos\theta_{12}$$

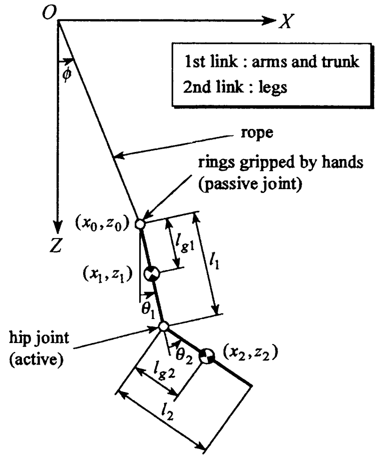


Figure 2: Model of two-link rings gymnastic robot

$$
\begin{aligned}
M_{14} &= m_2 l_{g2} \cos\theta_{12} \\
M_{23} &= -m_2 l'_{g1} \sin\theta_1 - m_2 l_{g2} \sin\theta_{12} \\
M_{24} &= -m_2 l_{g2} \sin\theta_{12} \\
M_{33} &= m_2 l'^2_{g1} + m_2 l^2_{g2} + 2 m_2 l'_{g1} l_{g2} \cos\theta_2 + I_1 + I_2 \\
M_{34} &= m_2 l^2_{g2} + m_2 l'_{g1} l_{g2} \cos\theta_2 + I_2 \\
M_{44} &= m_2 l^2_{g2} + I_2 \\
h_1 &= -m_2 l'_{g1} \dot\theta_1^2 \sin\theta_1 - m_2 l_{g2} \dot\theta_{12}^2 \sin\theta_{12} \\
h_2 &= -m_2 l'_{g1} \dot\theta_1^2 \cos\theta_1 - m_2 l_{g2} \dot\theta_{12}^2 \cos\theta_{12} \\
h_3 &= m_2 l'_{g1} l_{g2} (2\dot\theta_1 \dot\theta_2 + \dot\theta_2^2) \sin\theta_2 \\
h_4 &= m_2 l'_{g1} l_{g2} \dot\theta_1^2 \sin\theta_2 \\
g_1 &= 0, \quad g_2 = -(m_1 + m_2) g \\
g_3 &= m_2 g l'_{g1} \sin\theta_1 + m_2 g l_{g2} \sin\theta_{12} \\
g_4 &= m_2 g l_{g2} \sin\theta_1 \\
Q_1 &= F_r \sin\phi, \quad Q_2 = F_r \cos\phi \\
Q_3 &= F_r l_{g1} \sin(\theta_1 - \phi), \quad Q_4 = \tau_2 + T_2
\end{aligned}
$$

where

$$l'_{g1} = l_1 - l_{g1}, \quad \theta_{12} = \theta_1 + \theta_2$$

The first joint is passive and τ_2 is an input torque to the second joint. The tension F_r of rope and torque T_2 caused by an impedance in the second joint are expressed as

$$F_r = \begin{cases} -k_r(r - l_r) - b_r \dot r & \text{for } r > l_r \\ 0 & \text{for } r \le l_r \end{cases} \tag{2}$$

$$T_2 = -k_2 \theta_2 - b_2 \dot\theta_2 \tag{3}$$

where

$$
\begin{aligned}
r &= \sqrt{x_0^2 + z_0^2}, \quad \dot r = \dot x_0 \sin\phi + \dot z_0 \cos\phi \\
\phi &= \mathrm{atan2}(x_0, z_0) \\
x_0 &= x_1 - l_{g1} \sin\theta_1, \quad z_0 = z_1 - l_{g1} \cos\theta_1 \\
\dot x_0 &= \dot x_1 - l_{g1} \dot\theta_1 \cos\theta_1, \quad \dot z_0 = \dot z_1 + l_{g1} \dot\theta_1 \sin\theta_1
\end{aligned}
$$

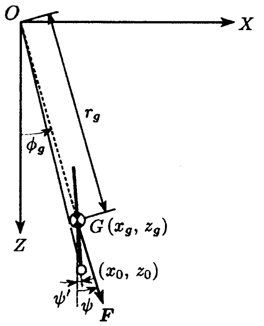


Figure 3: Definition of ψ and ψ'

3 Fuzzy Control for Performances

To realize a series of exercises shifting from a back giant to a handstand, the exercises are divided into two basic exercises, i.e., the back giant and handstand. A fuzzy controller for each basic exercise is selected according to the situation.

3.1 Control for back giant

The back giant is controlled using a simplified fuzzy reasoning and the rules are described as follows:

Rule 1: If θ_1 is ${}^1A$ then $\tau_2 = {}^1c$
Rule 2: If θ_1 is ${}^1\bar{A}$ then $\tau_2 = 0.0$

where ${}^1\bar{A}$ is the complement of ${}^1A$. Antecedent membership functions are Gaussian type represented by

$$\mu_{{}^1A}(\theta_1) = \exp\{\ln(0.5)(\theta_1 - \alpha)^2 \beta^2\} \tag{4}$$

where α and β denote the center value and the reciprocal number of the standard deviation respectively.

3.2 Control for handstand

We first consider a stability condition for the handstand. Assuming there is a lumped mass $m (= m_1 + m_2)$ on the resultant mass center G of the gymnastic robot, X- and Z-directional components of the force vector $\boldsymbol{F} = [F_x\ F_z]^T$ acting on the G are obtained by

$$F_x = -m\ddot x_g \tag{5}$$

$$F_z = -m\ddot z_g + mg \tag{6}$$

where $\ddot x_g$ and $\ddot z_g$ are calculated by difference calculus. Then, the angle ψ between Z-axis and vector $\boldsymbol{F}$ is expressed as

$$\psi = \mathrm{atan2}(F_x, F_z) \tag{7}$$

On the other hand, the angle ψ' between Z-axis and a straight line connecting the coordinate $(x_0,\ z_0)$ of the first joint to the coordinate $(x_g,\ z_g)$ of the G is represented by

$$\psi' = \mathrm{atan2}(x_0 - x_g, z_0 - z_g) \tag{8}$$

Since the gymnastic robot can keep its balance if ψ and ψ' are equal, the handstand is controlled by making ψ

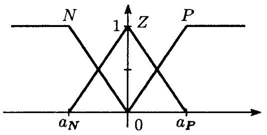


Figure 4: Antecedent membership functions for handstand

Table 1: Parameters of antecedent membership functions for handstand

Input	a_N	a_P
$\psi - \psi'$	$-\pi/9$	$\pi/9$
ϕ	-0.22	0.22
$\dot{\phi}$	-1	1

and ψ' equal (see **Fig. 3**). Therefore, $(\psi - \psi')$ is given as one of the inputs to the fuzzy reasoning. A simplified fuzzy reasoning is used for controlling the handstand and the rules are obtained as follows:

Rule l: If $(\psi - \psi')$ is ${}^2A_l$ and ϕ is ${}^2B_l$
and $\dot{\phi}$ is ${}^2C_l$ then $\tau_2 = {}^2c_l$

Here, l indicates 1 to 27. For each input, three labels with a triangular and two trapezoidal functions are assigned to the antecedent membership functions as shown in **Fig. 4**. **Table 1** presents the parameters of the antecedent membership functions for each input. Note here that the antecedent grade is given as the product of the grade of each membership function.

4 Acquisition of Fuzzy Rules

The fuzzy rules for the back giant and handstand are determined by GA respectively. First the rules for the back giant are acquired by GA, and then the rules for the handstand through the back giant are acquired by GA. In the GA, the number of individuals is $N = 60$, a uniform crossover is used with a crossover rate of 0.6, a mutation rate is 0.01, a tournament strategy with 3 individuals is adopted in the selection, and an elite strategy of 6 individuals is used in an alternation of generation.

4.1 Fitness function for back giant

In the fuzzy rules for the back giant, the consequent constant value ${}^1c$ and the parameters α and β of the antecedent membership function are determined by GA. Searching range is 600 to 1100 in ${}^1c$, $-\pi/12$ to $\pi/12$ in α, and $\pi/180$ to $\pi/6$ in β. To acquire the rules for the back giant that is appropriate for the transition to the handstand and has a minimum torque, the population is divided into 3 subgroups that has independent evaluation in each generation as shown in **Fig. 5**.

To shift to the handstand, it is required that there exists a state satisfying the stability condition for the handstand $\psi(k) - \psi'(k) = 0$ [rad]. Here, k is a discrete-time. Thus the first condition

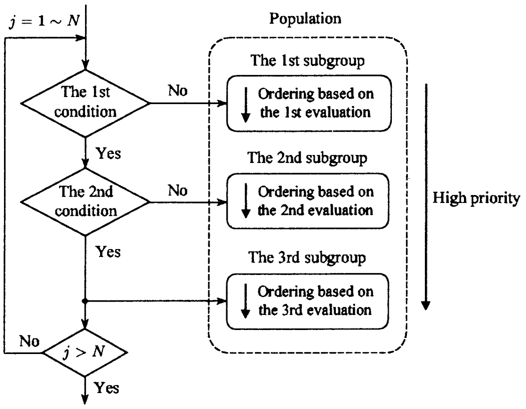


Figure 5: Three subgroups divided by two conditions

$$\theta_1(k) > \pi \text{ and } \{\psi(k) - \psi'(k)\} < 0$$
$$\text{and } \{\psi(k-1) - \psi'(k-1)\} > 0$$

judges whether it exists or not every discrete-time. Furthermore, even in the case satisfying the condition, it is impossible to shift to the handstand, unless the time rate of change of $\psi(k_s) - \psi'(k_s)$ in the discrete-time k_s is small to some degree. Therefore the magnitude of the value is restricted by the second condition

$$f_{b2} = |\dot{\psi}(k_s) - \dot{\psi}'(k_s)| < 0.1$$

How to divide the population into 3 subgroups and evaluation in each subgroup are described below (see Fig. 5). First an individual is judged by the first condition, and if all of the conditions are not satisfied, then the individual is assigned to the first subgroup which maximizes the first evaluation

$$f_{b1} = \max\{\theta_1(0), ..., \theta_1(k_{fb})\} \tag{9}$$

Here, k_{fb} denotes the predefined final discrete-time of the performance. The individuals satisfying the first condition move on to the second condition. If the second condition is not satisfied, the individual is classified into the second subgroup which minimizes the second evaluation f_{b2} and its calculation is stopped. The individuals satisfying the second condition can realize the back giant which is suitable for the transition to the handstand and are assigned to the third subgroup which minimizes the third evaluation described by the sum total of torque

$$f_{b3} = \sum_{i=0}^{k_s} |\tau_2(i)| \tag{10}$$

and the calculation is stopped. Note that the priority of evaluation is high in the order of the third, second and first subgroups, and a percentage of the third subgroup is expected to be increased with alternation of generations.

4.2 Fitness function for handstand

The consequent constant is set to zero when all labels are Z in the fuzzy rules to do the handstand, the

Table 2: Spring constant and viscous damping coefficient for the second joint

Range	Joint 2	
	k_2 [Nm/rad]	b_2 [Nms/rad]
$\theta_2 > \theta_{2u}$	1000	200
$0 \leq \theta_2 \leq \theta_{2u}$	150	9
$\theta_{2l} \leq \theta_2 < 0$	350	21
$\theta_2 < \theta_{2l}$	4000	200

other 26 values (${}^2c_1 \sim {}^2c_{26}$) are determined by GA. Searching range of these consequent constants is set to 100 from -200 respectively. In case of the handstand, the fuzzy controller for the back giant is selected at the beginning of the simulation, and then the controller is switched to that for the handstand when both $\theta_1(k) > \pi$ [rad] and $\{\psi(k) - \psi'(k)\} < 0$ [rad] are satisfied. The GA is evaluated while the fuzzy controller for the handstand is selected.

Fitness function represents a degree of stability of the handstand, which is given by

$$f_h = f_{h1} + f_{h2} \tag{11}$$

During $|\psi(k) - \psi'(k)| \leq \pi/90$ [rad]

$$f_{h1} = \sum_{i=k_h}^{k} |\psi(i) - \psi'(i)| \tag{12}$$

is used. In case of $|\psi(k) - \psi'(k)| > \pi/90$ [rad], a penalty

$$f_{h2} = k_{fh} - k \tag{13}$$

is given and the calculation of the individual is stopped. Note that k_h denotes the discrete-time at which the controller for the handstand is selected and k_{fh} indicates the prespecified final discrete-time in the simulation. By solving the minimization problem of this fitness function, the fuzzy rules to realize the most stable handstand can be acquired.

5 Simulation

In the rings gymnastic robot, link parameters shown in [5] are used. Parameters for rope are shown as follows:

Natural length	: $l_r = 2.95$ [m]
Spring constant	: $k_r = 20000$ [N/m]
Viscous damping coefficient	: $b_r = 2000$ [Ns/m]

Range of motion for the second joint [2] is given by

Upper limit : $\theta_{2u} = 120.3$ [deg]
Lower limit : $\theta_{2l} = -14.32$ [deg]

Table 2 presents a spring constant and a viscous damping coefficient set in accordance with an angle of the second joint. Simulation time is 4 [s] in the back giant and 6 [s] in the handstand, a sampling period is 1 [ms], and the initial states of the gymnastic robot are set as follows:

$$\theta_1 = -179\pi/180, \quad \theta_2 = 0, \quad x_1 = l_{g1} \sin\theta_1$$
$$z_1 = l_r + (m_1 + m_2)g/k_r + l_{g1}\cos\theta_1$$
$$\dot{q} = [0 \ \ 0 \ \ 0 \ \ 0]^T$$

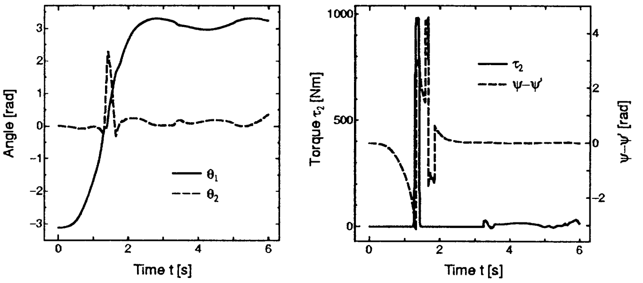


Figure 6: Simulation results

Simulation results using the fuzzy rules acquired by GA are presented in **Fig. 6**. From these results, it is confirmed that a series of exercises shifting from the back giant to the handstand considering torque minimization can be realized.

6 Conclusions

In this paper, we have proposed a novel "rings gymnastic robot." The fuzzy rules for realizing the back giant efficiently like a gymnast's skillful performance were acquired. In this case, three kinds of evaluations were introduced because it is needed that the rules are also suitable for the transition to the handstand. Consequently, such a multiobjective was achieved and the performance minimizing torque was acquired.

References

[1] S. Takashima, "Dynamic Modeling of a Gymnast on a High Bar," *Procs. of IROS'90*, Vol. 2, pp. 955–962, 1990.

[2] D. Nakawaki, S. Joo and F. Miyazaki, "Multi-model Based Analysis of a Gymnastic Movement on the Horizontal Bar," *J. of the Robotics Society of Japan*, Vol. 18, No. 1, pp. 59–65, 2000. (in Japanese)

[3] T. Fukuda, F. Saito and F. Arai, "A Study on the Brachiation Type of Mobile Robot (Heuristic Creation of Driving Input and Control Using CMAC)," *Procs. of IROS'91*, Vol. 2, pp. 478–483, 1991.

[4] A. Kaneko, *Coaching of Gymnastics Competition*, Taishukan-Shoten, pp. 79–81, 1974. (in Japanese)

[5] A. Imadu and K. Ono, "Optimum Trajectory Planning Method for a System that Includes Passive Joints (1st report, Proposal of a function approximation method)," *Trans. on the Japan Society of Mechanical Engineers (JSME)*, C, Vol. 64, No. 618, pp. 516–522, 1998. (in Japanese)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Fuzzy Behavior-Based Control of Mobile Robot in Dynamic Environments Using Modules Learned in Static Environments

Kiyotaka Izumi Keigo Watanabe Kazuo Kiguchi
Dept. of Mechanical Eng. Dept. of Advanced Systems Control Eng.
Saga University
1-Honjomachi, Saga 840-8502, Japan
{izumi, watanabe, kiguchi}me.saga-u.ac.jp

Abstract

Recently, intelligent control is being widely studied in the filed of robotics. A fuzzy behavior-based approach, which is a construction method of intelligent control system, decomposes a controller into each elemental behavior like a subsumption architecture, where each elemental behavior is realized by a fuzzy reasoning. In this paper, a learning method and a behavior fusion method are proposed for such a behavior-based control system to obtain flexible elemental behaviors. In particular, the module learning method and the behavior fusion method are applied for an obstacle avoidance problem of a mobile robot. The effectiveness of the present method is illustrated through some simulations.

Keywords—Behavior-based control, Fuzzy reasoning, Behavior Fusion, Genetic algorithm, Subsumption architecture

1 Introduction

A behavior-based control system, which can intelligently construct a controlled system for a mobile robot or manipulator, is now being attractive in the filed of robotics [1]-[8]. One very interesting approach is known as a subsumption architecture [1]. In this method, the controller are decomposed into each elemental behavior, and then several results from elemental behaviors are competed or cooperated. The important technical terms are the structure of a controller, the structure of each elemental behavior and the behavior fusion units.

The present authors have already developed a fuzzy behavior-based control system [6], [7], and verified the effectiveness by some experiments and simulations using a mobile robot and a manipulator. This system is fundamentally similar to one due to the subsumption architecture, but has a competition or cooperation unit consisting of a saturation function to generate a suitable consequent result. This approach needs no a priori information on the environmental region, as well as the behavior network approach [2] and genetic programming approach [3]. That is, the simplified fuzzy reasoning is assigned to each elemental behavior consisting of a single input-output relation. The mean value and the inverse value of standard deviation for Gaussian membership function, and the constant value in the conclusion part are learned by a genetic algorithm (GA). The total parameters associated with a fuzzy controller are encoded within one chromosome, like a Pittsburgh approach. Thus, the proposed behavior-based control system is realized as a soft-computing technique.

In this paper, we discuss the module learning approach for the fuzzy behavior-based control system in order to make the system robust against for a change of environment. Furthermore, the behavior fusion method for the module learning approach is discussed. The effectiveness of the present learning approach and the behavior fusion method is illustrated with some simulation examples for controlling a mobile robot.

2 Fuzzy Behavior-Based Control

Consider a terminal control problem in which a mobile robot with two independent driving wheels travels to a goal point as shown in **Fig. 1**. Here, D is the distance between the goal point and the mobile robot, Ψ is the relative angle between the azimuth of the robot and the goal point, v is the forward speed of the mobile robot, ϕ is the forward azimuth of the mobile robot, and $\dot{\phi}$ is the rate of ϕ. Such states are assumed to be obtained as sensor information. To solve this problem, we construct a fuzzy behavior-based control

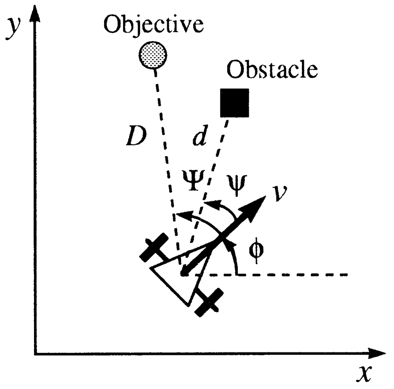


Figure 1: A fuzzy behavior-based control system

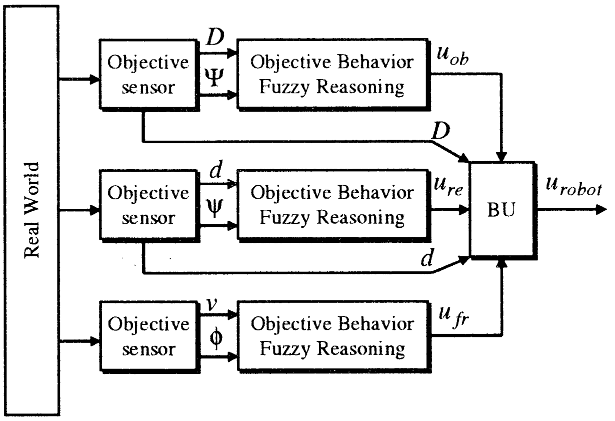


Figure 2: A fuzzy behavior-based control system

system which has an objective behavior group, a free behavior group and a reactive behavior group as shown in **Fig. 2**.

In Fig. 2, u_{ob} denotes the objective behavioral output to travel to the goal point, u_{re} denotes the reactive behavioral output to avoid obstacles, and u_{fr} denotes the free behavioral output to search the goal point. The symbol BU denotes a behavior fusion unit that implies the cooperation and competition for the output results of behavioral elements. Finally, u_{robot} denotes the robot's behavior output.

3 Sensor Based Behavior Fusion Unit

In practical, the area in which a robot can recognize is restricted depending on measuring ability of each sensor. There are two kind of sensors in section 2. One is to detect an objective point. The other is to detect obstacles. In this paper, these sensors are called the objective sensor and the obstacle sensor. The objective sensor is possible to measure between 0 to $sensor_{ob}$, and the obstacle sensor is possible to measure between 0 to $sensor_{re}$. Introducing the degree of dangerous which is given by

$$s = \frac{d}{sensor_{re}} \tag{1}$$

and the activity of each sensor, the behavior fusion unit is defined by

$$u_{robot} = \begin{cases} su_{ob} + (1.0 - s)u_{re} \\ \quad \text{if both sensors are active} \\ u_{ob} \\ \quad \text{if only objective sensor is active} \\ u_{re} \\ \quad \text{if only obstacle sensor is active} \\ u_{fr} \\ \quad \text{if both sensors are nonactive} \end{cases} \tag{2}$$

4 Module Learning

In order to obtain flexible knowledge, the robot has to learn through a lot of experience. For example, in the preceding problem statement, the experience of the robot heavily depends on the differences of conditions between a robot and a training field. A module learning method is discussed in this section by using the structure of the fuzzy behavior-based control system, where the training is assumed to be performed at each behavior group with some static conditions.

4.1 Learning of Objective Behavior Group

The role of the objective behavior group is to control the mobile robot to arrive at a goal point. The fitness of GA is defined by

$$\begin{aligned} fitness_{ob} &= \sum_{i=1}^{n_o} (D_{fit} + time_{fit} + road_{fit} \\ &\quad + B_{d_{fit}} + B_{a_{fit}}) \qquad (3) \\ D_{fit} &= w_{od} \frac{sensor_{ob} - D_{end}}{sensor_{ob}} \\ time_{fit} &= w_{ot} \frac{t_{\max} - time_{end}}{t_{\max}} \\ road_{fit} &= w_{or} \frac{sensor_{ob} - road_{end}}{sensor_{ob}} \\ B_{d_{fit}} &= w_{obd} \frac{\tau_{\max} - \sum_{j=1}^{n_{ro}} |B_d|}{\tau_{\max}} \\ B_{a_{fit}} &= w_{oba} \frac{\tau_{\max} - \sum_{j=1}^{n_{ro}} |B_a|}{\tau_{\max}} \end{aligned}$$

Table 1: Parameters of fitness for objective behavior

n_o	w_{od}	w_{ot}	w_{or}	w_{obd}
4	100	10	10	1
w_{oba}	$sensor_{ob}$	$t_{\max}$	$\tau_{\max}$	n_{ro}
1	0.3	10	15	5

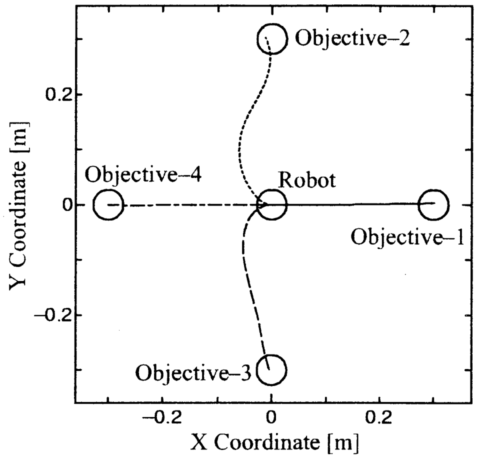


Figure 3: Trajectory of a mobile robot on objective learning environment

where n_o is the number of training conditions, D_{fit} is the evaluation of distance D_{end} between the final position of robot and the objective point, $time_{fit}$ is the evaluation of convergence time $time_{end}$ using the maximum moving time $t_{\max}$, $road_{fit}$ is the evaluation of moving distance, $B_{d_{fit}}$ and $B_{a_{fit}}$ are the evaluation of the consequent part of fuzzy rules in which the number of fuzzy rules is n_{ro} and w_* are weighting parameters of evaluations.

The training is performed with **Table 1**. The training result is illustrated with **Fig. 3**. The mobile robot can reach the objective point in whole training conditions.

4.2 Learning of Reactive Behavior Group

The role of the reactive behavior group is to avoid an obstacle such that the mobile robot keeps an distance away from an obstacle. The fitness of GA is defined by

$$\begin{aligned} fitness_{re} &= \sum_{n=1}^{n_r} (d_{fit} + time_{fit} + road_{fit} \\ &\quad + B_{d_{fit}} + B_{a_{fit}}) \end{aligned} \tag{4}$$

Table 2: Parameters of fitness for reactive behavior

n_r	w_{rd}	w_{rt}	w_{rr}	w_{rbd}
4	100	10	10	1
w_{rba}	$sensor_{re}$	$t_{\max}$	$\tau_{\max}$	n_{rr}
1	0.2	5	15	5

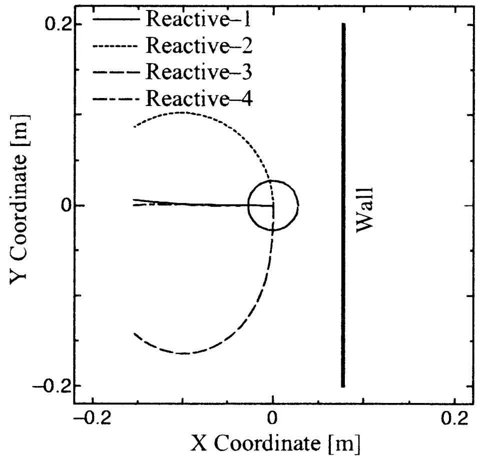


Figure 4: Trajectory of a mobile robot on reactive learning environment

$$\begin{aligned} d_{fit} &= w_{rd} \frac{d_{end}}{sensor_{re}} \\ time_{fit} &= w_{rt} \frac{t_{\max} - time_{end}}{t_{\max}} \\ road_{fit} &= w_{rr} \frac{sensor_{re} - road_{end}}{sensor_{re}} \\ B_{d_{fit}} &= w_{rbd} \frac{\tau_{\max} - \sum_{l=1}^{n_{rr}} |B_d|}{\tau_{\max}} \\ B_{a_{fit}} &= w_{rba} \frac{\tau_{\max} - \sum_{l=1}^{n_{rr}} |B_a|}{\tau_{\max}} \end{aligned}$$

where n_r is the number of training conditions, d_{fit} is the evaluation of minimum distance d_{end} between the final position of robot and the obstacle, $time_{fit}$ is the evaluation of convergence time $time_{end}$ using the maximum moving time $t_{\max}$, $road_{fit}$ is the evaluation of moving distance, $B_{d_{fit}}$ and $B_{a_{fit}}$ are the evaluation of the consequent part of fuzzy rules in which the number of fuzzy rules is n_{rr} and w_* are weighting parameters of evaluations.

The training is performed with **Table 2**. The training result is illustrated with **Fig. 4**. It is seen that the mobile robot can avoid an obstacle in whole training conditions.

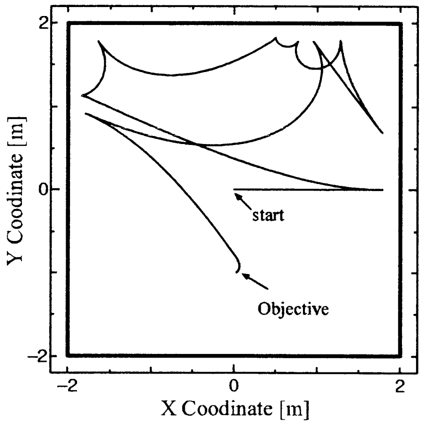


Figure 5: Trajectory of a mobile robot in a static environment

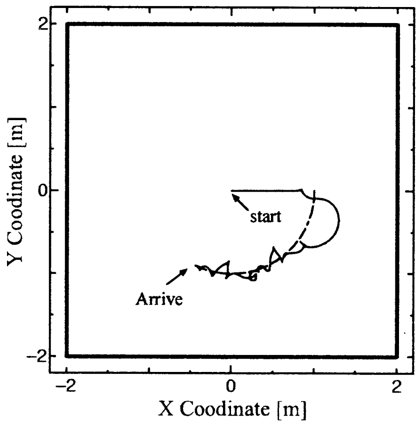


Figure 6: Trajectory of a mobile robot in a dynamic environment

5 Simulations

The moving area is 4 × 4 [m^2] with surrounding by walls. The objective point is set to be (0, -1). Initial state variables of the mobile robot are set to be $[v(0)\ \phi(0)\ \dot{\phi}(0)]^T = [0\ 0\ 0]^T$. The initial position of the mobile robot is set to be (0, 0). The simulation result is illustrated in **Fig. 5**. If the robot can not recognize the objective point and obstacles, then the robot goes straight on using the output of the free behavior. When the robot recognizes obstacles, the robot avoids obstacle. Finally the robot recognizes the objective point, and therefore the robot arrives at the objective point.

In the second simulation, conditions are almost same as the case of static environment fundamentally. But, the objective point is moving on the circle with the radius 1 [m] and center (0, 0). The simulation result is illustrated in **Fig. 6**. The first behavior is the same as the static environment. When the robot recognizes the objective point, the robot tracks the objective point. However, when the robot lost the objective point, the robot goes straight on. Thus, the robot arrives at the objective point, whenever the robot recognizes the objective point.

6 Conclusions

The module learning approach and the behavior fusion unit for the fuzzy behavior-based control system have been discussed with the obstacle avoidance problem of a mobile robot. Although the robot was trained only in the local learning, it arrived at the objective point without any collision with obstacles in the dynamic environment. Therefore it was proved that the fuzzy behavior-based control system was flexible against the change of environments.

References

[1] Brooks, R.A. (1986), A Robust Layered Control System for a Mobile Robot. *IEEE J. of Robotics and Automation* 2(1): 14–23

[2] Maes, P. (1991), Learning Behavior Networks from Experience. In *Procs. of ECAL-91*, pp. 48–57.

[3] Koza, J.R. (1991), Evolution of Subsumption Using Genetic Programming. In *Procs. of ECAL-91*, pp. 110–119.

[4] Hu, H. and Brady, M. (1996), A Parallel Processing Architecture for Sensor-Based Control of Intelligent Mobile Robots. *Robotics and Autonomous Systems* 17: 235–257

[5] Nakasuka, S., Yairi, T. and Wajima, H. (1996), Autonomous Generation of Reflexion-Based Robot Controller Using Inductive Learning. *Robotics and Autonomous Systems* 17: 287–305

[6] Watanabe, K. and Izumi, K. (1998), Fuzzy Behavior-Based Control (1st Report, A Proposal of Control System Realization). *Trans. of Japan Society of Mechanical Engineers* 64C(620): 1278–1286

[7] Izumi, K. and Watanabe, K. (2000), Fuzzy Behavior-Based Control Trained by Module Learning to Acquire the Adaptive Behaviors of Mobile Robots. *Mathematics and Computers in Simulation* 51(3-4): 233–243

[8] Arkin, R.C. (1998), Behavior-Based Robotics. MIT Press

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Energy Optimal Gait Analysis of Quadruped Robots

Kazuo Kiguchi*, Yukihiro Kusumoto*, Keigo Watanabe*, Kiyotaka Izumi*, Toshio Fukuda**
* Dept. of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University,
1 Honjomachi, Saga 840-8502, Japan
E-mail: kiguchi@me.saga-u.ac.jp
** Center of Cooperative Research in Advanced Science and Tech.
Nagoya University,
1 Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

Abstract

It is important for walking robots such as quadruped robots to decide an efficient gait. Since animals and insects are the basic models of most of walking robots, their walking patterns are good example for the walking robots. In this study, the walking energy consumption of a quadruped robot is analyzed and compared with natural animal gaits.

In this study, genetic algorithms have been applied to obtain the energy optimal gait when the quadruped robot is walking with a certain velocity. In this method, an individual in a population represents a walking pattern of the quadruped robot. The gait (individual) which consumes the least energy is considered as the best gait (individual) in this study. The energy optimal gait is analyzed at several walking velocities, since the amount of walking energy consumption change if the walking velocity of the robot is changed. The results of this study can be applied to decide what kind of gait should be generated for the quadruped robot as its walking velocity changes.

Keywords: Quadruped Robot, Optimal Gait, Energy Analysis, Animal Gait, Genetic Algorithms.

1. Introduction

Recently, many studies have been carried out for walking robots. Walking robots are good at moving in unstructured and rough terrain and avoiding obstacles compared with wheeled robots. It is important for walking robots such as quadruped robots to decide an efficient gait. One of the best ways finding the efficient gait is imitating the animal or insect gaits [1]. Since animals and insects are the basic models of most of walking robots, their walking patterns are good example for the walking robots. On the other hand, it is important for the robots to take into account the walking energy consumption [2]. Many studies on walking energy consumption analysis have been carried out in the field of robotics, cybernetics, biomechanics and zoology up to now. For example, energy optimal analysis has been performed for multilegged robotic vehicles to realize a nearly passive cyclic gait [2] and energy consumption of animal locomotion has been studied by measuring oxygen consumption in the field of zoology [3]-[5].

As an animal changes walking pattern, the amount of its energy consumption changes. It is known that animal's walking patterns change depending on its walking velocity and the load on it [1]. In this study, energy optimal gaits of quadruped robots are analyzed with respect to walking velocity. Since most of quadruped robots are inspired by animals such as horses, dogs, or cats, the optimal gaits of the quadruped robots are supposed to be similar to those of animals.

Each leg of the quadruped robot is modeled as a simple two-link structure. Torque is supposed to be supplied at each joint in this model. Genetic algorithms (GA) [6], one of the best optimization algorithms, have been applied to obtain the energy optimal gait when the quadruped robot is walking with a certain velocity. In this method, an individual in a population represents a walking pattern of the quadruped robot. In the evaluation process of the GA, the robot is supposed to walk a certain distance on a flat floor with the same gait and the same velocity. Body height and step length of the robot and duty ratio are also taken into account as variables. In the evaluation process of the GA, energy (the kinetic and potential energy) consumption of the quadruped robot is calculated. Here, most of energy is consumed for accelerating and decelerating the leg swing. The gait (chromosome in GA) which consumes the least energy is considered as the best gait in this study. The energy optimal gait is analyzed at several walking velocities, since the amount of walking energy consumption change if the walking velocity of the robot is changed. The results of this study can be applied to decide what kind of gait should be generated for the quadruped robot as its walking velocity changes. Finally, the analyzed results are compared with natural animal gaits.

2. Energy Analysis of Quadruped Robot

The model of the quadruped robot used in this study is depicted in Fig. 1. Each leg of the quadruped robot consists of two links. In this section, walking energy consumption of the quadruped robot is analyzed. In this study, the body of the quadruped robot is supposed to move only along x-axis with a certain velocity. The translational motion in the other direction and the rotational motion is contstraint.

The equation of the kinetic energy of the swing leg of the quadruped robot is written as:

Link 1:

$$\begin{aligned} K_1 &= \frac{1}{2}m_1\left(v_1^2+v_0^2\right)+\frac{1}{2}I_1\dot{\theta}_1^2 \\ &= \frac{1}{2}m_1\left(\dot{x}_1^2+\dot{z}_1^2+\dot{x}_0^2\right)+\frac{1}{2}\left(\frac{1}{3}m_1l_1^2\right)\dot{\theta}_1^2 \\ &= \frac{1}{2}m_1\left(\, l_{g1}^2\dot{\theta}_1^2+\dot{x}_0^2\right)+\frac{1}{6}m_1l_1^2\dot{\theta}_1^2 \end{aligned} \qquad (1)$$

Link 2:

$$\begin{aligned} K_2 &= \frac{1}{2}m_2\left(v_2^2+v_0^2\right)+\frac{1}{2}I_2\dot{\theta}_2^2 \\ &= \frac{1}{2}m_2\left(\dot{x}_2^2+\dot{z}_2^2+\dot{x}_0^2\right)+\frac{1}{2}\left(\frac{1}{3}m_2l_2^2\right)\dot{\theta}_2^2 \\ &= \frac{1}{2}m_2\{l_1^2\dot{\theta}_1^2+2l_1l_{g2}\dot{\theta}_1\left(\dot{\theta}_1+\dot{\theta}_2\right)\cos\theta_2 \\ &\quad +l_{g2}^2\left(\dot{\theta}_1+\dot{\theta}_2\right)^2+\dot{x}_0^2\}+\frac{1}{6}m_2l_2^2\dot{\theta}_2^2 \end{aligned} \qquad (2)$$

where m_i is mass of link i, v_i denotes the translational velocity of link i, v_0 denotes the translational velocity of the body, I_i represents the inertia matrix of the link i with respect to its center of mass, and θ_i is the angle of link i.

The equation of the potential energy of the swing leg of the quadruped robot is written as:

Link 1:

$$U_1 = m_1gz_1 = m_1\left(z_0 - l_{g1}\sin\theta_1\right)g \qquad (3)$$

Link 2:

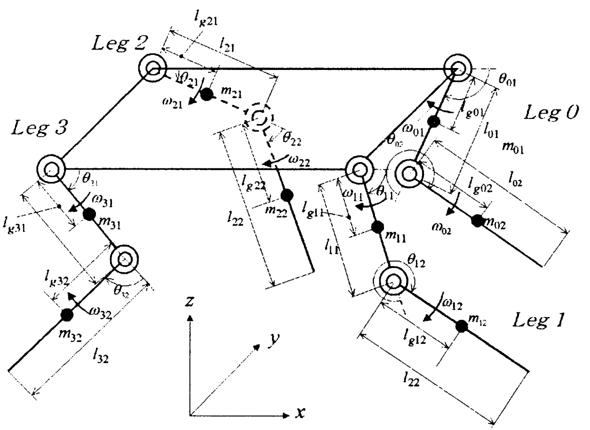


Figure 1 Quadruped robot model

$$U_2 = m_2gz_2 = m_2\left\{z_0 - l_1\sin\theta_1 - l_{g2}\sin\left(\theta_1+\theta_2\right)\right\}g \qquad (4)$$

where z_i is the height of center of mass of the link i.

The equation of the kinetic energy and potential energy of the body of the quadruped robot are written as:

$$\begin{aligned} K_0 &= \frac{1}{2}m_0v_0^2 = \frac{1}{2}m_0\left(\dot{x}_0^2+\dot{z}_0^2\right) \\ &= \frac{1}{2}m_0\left\{l_2^2\left(\dot{\theta}_1+\dot{\theta}_2\right)^2+2l_1l_2\left(\dot{\theta}_1+\dot{\theta}_2\right)\dot{\theta}_1\cos\theta_2+l_1^2\dot{\theta}_1^2\right\} \end{aligned} \qquad (5)$$

$$U_0 = m_0gz_0 = m_0\left\{z_f + l_2\sin\left(\theta_1+\theta_2\right)+l_1\sin\theta_1\right\}g \qquad (6)$$

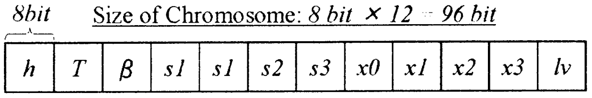
where m_0 is weight of body, v_0 denotes the translational velocity of the body, and z_0 is the height of center of mass of the body.

3. Analysis of Energy Optimal Gait Using GA

In order to find out the energy optimal gait for the quadruped robot, walking energy consumption is analyzed with respect to several walking velocities in this study. The GA has been applied to obtain the energy optimal gait when the quadruped robot is walking with a certain velocity.

The structure of each individual (chromosome) in a population is shown in Fig. 2. The process of the GA is depicted in Fig. 3. This GA process is performed for several walking velocities.

In the evaluation process of the GA, walking energy (the kinetic and potential energy) consumption of the quadruped robot in a certain distance is calculated. Here, most of energy is consumed for accelerating and decelerating the leg swing. The gait (individual) which consumes the least energy is considered as the best gait (individual) in this study. The robot is supposed keep the same gait for a trial in the evaluation process. The optimal body height of the quadruped robot during walking



H: Body height

T: Time of walking cycle

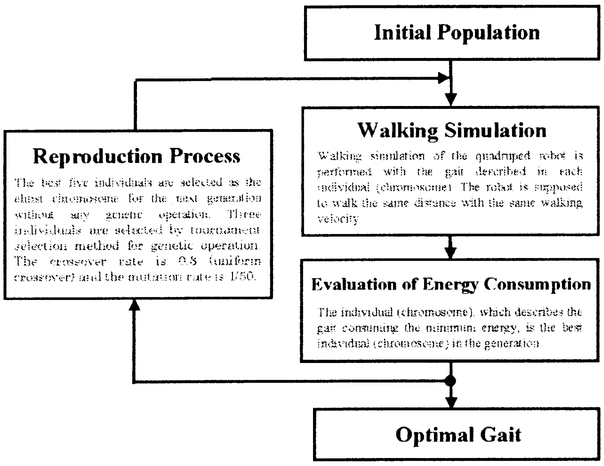
β: Duty ratio

sl: Swing leg starting time of leg *l*

xl: Initial position in x-axis of leg *l*

lv: Maximum height of swing leg

Figure 2 Structure of chromosome



Walking simulation of the quadruped robot is performed with the gait described in each individual (chromosome). The robot is supposed to walk the same distance with the same walking velocity.

The best five individuals are selected as the elitist chromosome for the next generation without any genetic operation. Three individuals are selected by tournament selection method for genetic operation. The crossover rate is 0.8 (uniform crossover) and the mutation rate is 1/50.

The individual (chromosome), which describes the gait consuming the minimum energy, is the best individual (chromosome) in the generation.

Figure 3 Process of GA

is also decided in GA. The energy optimal gait is analyzed at several walking velocities, since the amount of walking energy consumption change if the walking velocity of the robot is changed.

4. Simulation

The size of the quadruped robot used in simulation is defined to be similar to that of large dogs as shown in Fig. 4. There are 500 individuals (chromosome) in a population in this simulation. The best five individuals are supposed to remain in the next generation without any genetic operation. Three individuals are selected by tournament selection method for genetic operation for the next generation. In the genetic operation, the crossover rate is 0.8 (uniform crossover), and the mutation rate is 1/50. The walking distance for each evaluation is set to be 4m.

Figure 5-7 show the generated optimal gaits in the simulation when the walking velocity is 1.5m/s, 1.7m/s, and 1.9m/s, respectively. The generated optimal gait depicted in Fig. 5 is the same as the typical static gait of the quadruped robots and its duty ratio β is close to 0.75.

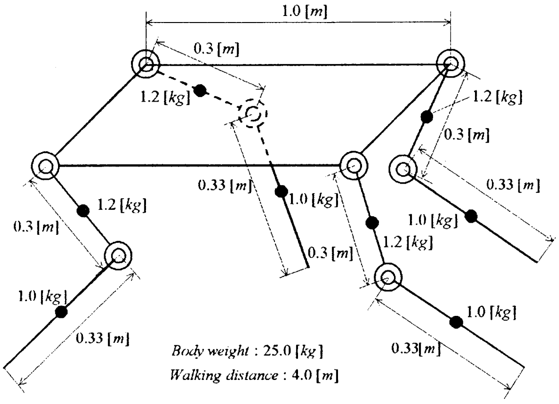


Figure 4 Simulated quadruped robot model

The generated optimal gait shown in Fig. 6 is also the static gait. The generated optimal gait shown in Fig. 7 is very similar to the typical dynamic gait called trot and its duty ratio β is close to 0.5. One can see that the energy optimal gait is changing as the walking velocity is increasing.

5. Discussion

Since the size of the quadruped robot is inspired by the large size dog, the optimal gaits of the quadruped robot are supposed to be similar to those of the dog. It is known that Froude Rate can be used to compare the walking gait between different kinds of animals [4]. The equation of Froude Rate for animal is written as:

$$Froude\ Rate = \frac{v_0^2}{gl} \tag{7}$$

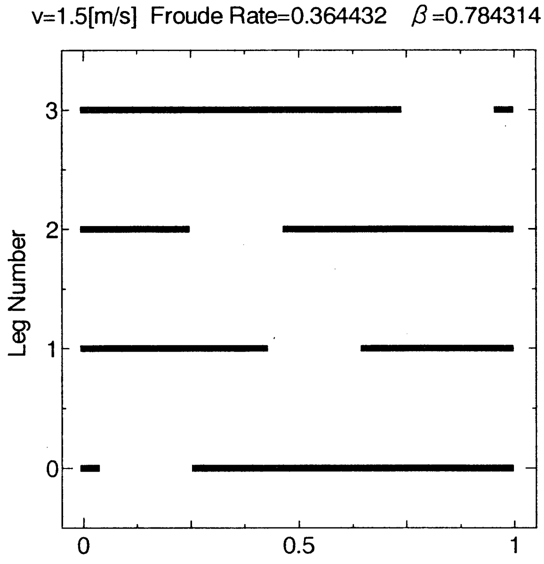


Figure 5 Generated optimal gait (v_0=1.5m/s)

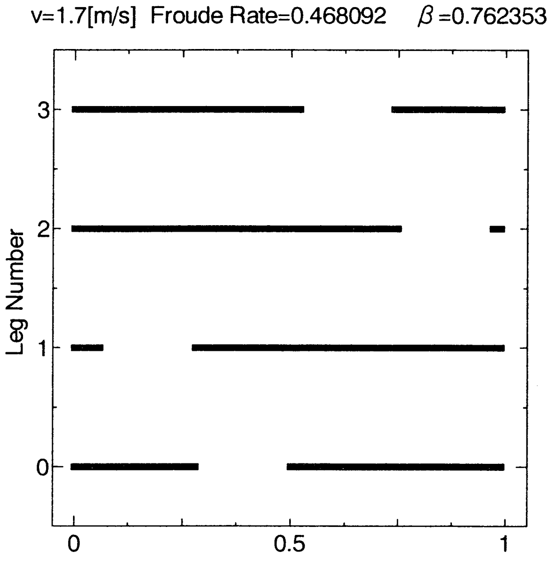


Figure 6 Generated optimal gait (v_0=1.7m/s)

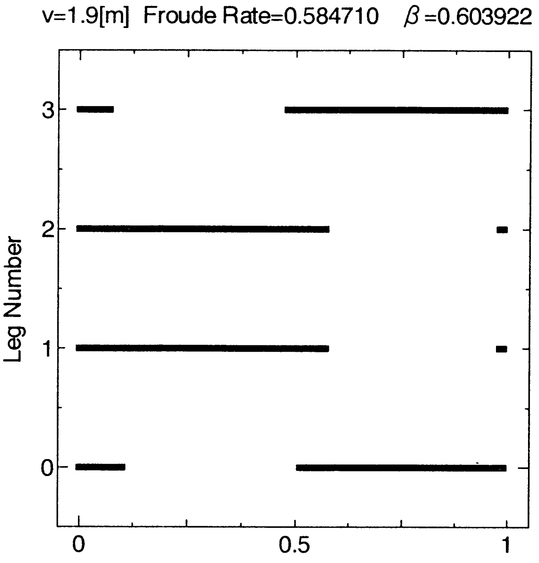


Figure 7 Generated optimal gait (v_0=1.9m/s)

where v_0 represents the walking velocity, g is the gravity acceleration, and l denotes the leg length. The mammals (including dogs) change their gait from static walking gait (crawl) to dynamic walking gait (trot or pace) when the Froude Rate is about 0.5, and from dynamic walking gait to running gait when the Froude Rate is about 2.5 [4]. Therefore, if the generated energy optimal gaits of the quadruped robot are the same as those of animals, the walking gaits of the simulated quadruped robot are supposed to be changed when the Froude Rate of the robot is about 0.5.

The optimal gait generated in simulation is changed when the walking velocity is about 1.8m/s. That means the energy optimal gaits of the simulated quadruped robot are changed when the Froude Rate is about 0.5. Consequently, the quadruped robot can be thought as one of animals as far as the walking gaits are concerned. We can apply the animal gaits for the locomotion of quadruped robots in order to minimize the walking energy consumption.

6. Conclusion

Walking energy consumption of the quadruped robot has been roughly analyzed and compared with that of an animal (dog) under several walking velocities in this study. The simulation results show that the energy optimal gaits of the quadruped robot are almost the same as those of the animals. Since the quadruped robots are considered as one of animals as far as the walking gaits are concerned, we can just apply animal gaits for the locomotion of quadruped robots in order to realize the minimum walking energy consumption.

We would like to study the energy optimal gait transition from dynamic walking to running of the quadruped robot for the future research.

References

[1] K.Akimoto, Y.Makino, M.Yano (1999), Highly Adaptable Optimal Control of an Autonomous Insect Walking Robot in the Real World, Proc. of IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, 1999, pp.VI-942-VI-947

[2] E.Y.Raby, D.E.Orin (1999), Passive Walking with Leg Compliance for Energy Efficient Multilegged Vehicles, Proc. of IEEE International Conference on Robotics and Automation, Detroit, Michigan, 1999, pp.1702-1707

[3] D.F.Hoyt, C.R.Taylor (1981), Gait and the Energetics of Locomotion in Horses, Nature 292(16): 239-240

[4] R.M.Alexander (1992), *Exploring Biomechanics*, W.H.Freeman and Company

[5] R.M.Alexander (1998), *Energy for Animal Life*, Oxford University Press

[6] D.E.Goldberg (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addson-Wesley Publishing Co.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

MemeStorms: Cellular Working Memory and Dynamics of Judgment

Andrzej Buller[1,2], Tomasz Chodakowski[1,3], Łukasz Kaiser[4],
Andrzej Nowak[5] & Katsunori Shimohara[1,6]

[1]ATR International, Information Sciences Division, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
{buller, tomasz, katsu}@isd.atr.co.jp

[2] Starlab N.V., Latour de Freins, Rue Engelandstraat 555, 1180 Brussel, Belgium; buller@starlab.net

[3] Department of Mathematics and Physics, Gdańsk University, ul. Wita Stwosza 57, 80-952 Gdańsk
tch@manta.univ.gda.pl

[4] Faculty of Mathematics, Wrocław University, pl. Grunwaldzki 2/4, Wrocław 50-348, Poland; kaiser@tenet.pl

[5] Institute for Social Studies, Warsaw University, ul. Stawki 5/7, Warszawa 00-183, Poland;
anowak@samba.iss.uw.edu.pl

[6]ATR International, HIP Research Laboratories,2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
katsu@hip.atr.co.jp

Abstract We present results of an experiment aimed to confirm psychological plausibility of a model of human Working Memory (WM) called *MemeStorms*.

There is empirical evidence that subjects' feelings sometimes oscillate between highly positive and highly negative values. Oscillation of judgments occurs more often in the case of ambiguous cues while in the case of clear cues a quick permanent polarization of judgment takes place. There is also evidence that tendency to oscillate is higher for important judgments. We showed that this principle applies also to the *MemeStorms* model in which chunks of information, called *memes*, concerning the object of judgment, are progressively integrated in WM into higher order structures. WM is a cellular spáce where thousands of memes navigate and search for partners suitable to create together a part of an idea. As a result of meme interactions, contradictory ideas appear. When a meme meets a meme carrying an opposite idea, both of them annihilate. Hence, contradictory ideas compete for domination in WM. Since dynamics demonstrated by *MemeStorms* fits to empirical evidence, we argue that the model approximates cognitive mechanisms of human mind.

1 Introduction

Vallacher *et al.* (1994) showed that social judgments may demonstrate intrinsic dynamics i.e. changes without any external stimuli, which put in question traditional views of social judgment. It was also shown that mixed valence produced dynamics, while univalent information produced polarization (Nowak & Vallacher 1998: 99-101). To explain the phenomena, Nowak and Vallacher proposed a theory of dynamic integration, that assumes that the observed oscillations reflect the alternation between two alternative schemata. According to the authors, static integration occurs when univalent information underlies the judgment or when the judgment is not important, while dynamic integration occurs for ambivalent and important judgments (Nowak & Vallacher 1988: 102-115).

Buller (2000) proposes that both the scenario of static and the scenario of dynamic integration are natural consequence of a single, distributed mechanism of information integration in Working Memory (WM). The *MemeStorms* model discussed in this paper assumes that small chunks of information concerning the object of judgment are progressively integrated in WM into higher order structures such that inconsistencies are reduced and the system strives toward coherence. The chunks of information are called *memes*.

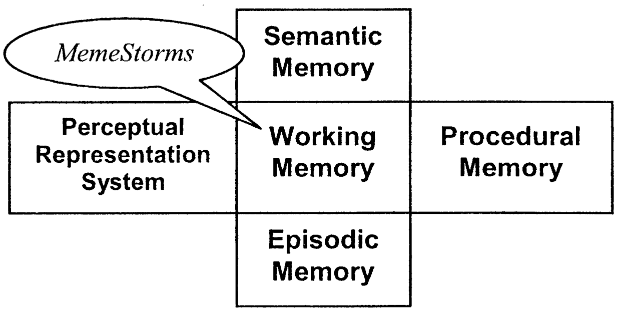


Figure 1. The "4 + 1" memory model (adopted from Buller 1998)

2 *MemeStorms* model

The key concepts of the *MemeStorms* model are: WM and a "society" of memes. WM is a grid of hexagonal processing cells. Memes are pieces of information that move all over the grid (jumping from one cell to another). When two memes meet, they interact. While classic definitions of meme consider it in a social environment, i.e. as a unit of information in a mind whose existence influences events such that more copies of itself get created in other minds (Dawkins 1989; Plotkin 1993; Brodie 1996), the *MemeStorms* concept concentrates on meme interactions inside a single mind. Hence, a meme is understood as an elementary piece of information that can contribute to a more complex meme, while the most complex memes are ideas. A set of appropriately configured memes may constitute an idea of a behavior (*eg* raising one's hand), or an idea of an object (recalled or invented) (Buller & Shimohara 1999), or an idea of a judgment.

Thinking is assumed to be a process in subject's WM, where thousands of memes navigate and search for partners suitable to create together a part of an idea.

WM is a pivotal part of a model of mind, called "4 + 1" showed in Figure 1 (Buller 1988; Buller & de Garis 1988; Buller & Shimohara 2000) developed based on the five-element model of memory system proposed by Endel Tulving (Tulving 1995; Schacter et al. 2000). In the framework of the "4 + 1" model, memes come to WM from supporting memories: PRS (Perceptual Representation System that converts perceived signals onto streams of appropriate memes), Episodic Memory (where ideas are stored chronologically), Semantic Memory (where ideas are organized by categories), and Procedural Memory (that has a direct connection to actuators).

A given meme occurs in WM in hundreds of its copies. Hence, resulting ideas also appear in multiple copies. The idea that is carried by the members of the victorious meme population is equal to a subject's belief or a decision (cf Calvin 1996).

There are four types of meme interaction in the *MemeStorms*: (i) elastic collision, where two memes that meet in WM have nothing in common and then they behave as balls on the snooker table, (ii) crossover, when two memes are related, and exchange some of their informational contents in the way resembling genetic cross-over, (iii) negative cross-over, i.e. a special cross-over that results in a production of a meme containing a negation of a certain assertion, and (iv) annihilation of two contradictory memes.

It may, therefore, happen that, as a result of meme interactions, populations of contradictory ideas appear in WM. Any pair of contradictory memes may meet and annihilate. Hence, WM appears to be a theater in which ideas compete for domination.

WM populated by a "society" of memes is a complex dynamic system in which the *control parameter* is a distribution of basic facts and rules in the stream of incoming memes, while the *order parameter* is a percentage of resulting memes of particular kind in the population of resulting memes. Cue ambiguity and judgment importance are assumed to be functions of the control parameter.

3 An example of judgmental dynamics

Let us assume that a subject looks for an energetic and possibly experienced co-worker. The already interviewed candidate **C** looks "rather energetic", hence, a quite dense stream of memes **G|C**, where **G** represents the feature "energetic", starts flowing into the subject's WM. Because of the "rather", at the same time a weak stream of memes **¬G|C** (where "¬" is the symbol of negation) also starts flowing into the subject's WM. It is hard to say that **C** is an experienced person, hence, a weak stream of memes **X|C** and a denser stream of **¬X|C** flow into the subject's WM, where **X** represents the feature "experienced". At the same time the subject's Semantic Memory releases memes **E|G** and **E|X**, where **E|G** represents the imperative "Employ an energetic person" where **E|X** represents the imperative "Employ an experienced person". Since experience is of lesser importance, the stream of memes of **E|G** is appropriately denser than the stream of copies of **E|X**. Every meme moves through WM, meets other memes, and interact with them. Every encounter of **E|G** and **G|C**, as well as an encounter of **E|X** and **X|C**, results in birth of **E|C** ("Employ **C**!"), while every encounter of **E|G** and **¬G|C**, as well as an encounter of **E|X** and **¬X|C**, results in birth of **¬E|C** ("Don't employ **C**!"). When **E|C** mets **¬E|C**, they annihilate. As a result, after a period of time, WM becomes dominated by the population of memes **E|C**. This means the subject's positive feelings about the idea to employ **C**. Suddenly, the population of memes **E|C** loses to the population of memes **¬E|C**. This means that the subject's feelings turned negative. We argue that this phenomenon may be a natural consequence of dynamic structure of human mind.

4 Experiment

We intended (i) to check whether the simulated subject's tendency to oscillate between highly positive judgments and highly negative judgments is higher for judgments of high importance, and (ii) to confirm, that, unlike in the theory 'static vs. dynamic' integration, both the scenario of static and the scenario of dynamic integration can be explained in the framework of a single mechanism of information integration as in the *MemeStorms* model.

3.1 Assumptions

A set of N=40 individuals has been generated based on the *MemeStorms* model. WM of each of the individuals was a grid of 16×16 cells. Each of the individuals perceived a set of cues that were transformed onto four streams of memes. There were also two streams of memes carrying related rules producing memes contributing to two contradictory judgments. As a result of meme interactions every individual was to express particular judgment calculated as a current percentage of memes "Employ C!" in the population of all newborn memes (order parameter). The judgment was expected to change over time despite constant average densities of incoming streams of memes. 12 kcycles, as the duration of judgment scanning was assumed.

3.2 Experimental variables

The cue ambiguity factor $\mathbf{a}_C$, as well as the **p**-factor, intended to represent judgmental importance, were considered as *independent variables*. $\mathbf{a}_C$ is a real number calculated as a function of densities of four meme streams entering WM. In the testing example there were two pairs of streams of contradictory memes: a stream of $\mathbf{G|C}$, a stream $\neg\mathbf{G|C}$, a stream of $\mathbf{X|C}$, and a stream of $\neg\mathbf{X|C}$. Their densities are denoted d_{GC}, d_{gC}, d_{XC}, and d_{xC}, respectively. It was assumed that $\mathbf{a}_C = 1 - | \delta_{GC} + \delta_{XC} - 1|$, where: $\delta_{GC} = d_{GC} / (d_{GC} + d_{gC})$, $\delta_{XC} = d_{XC} / (d_{XC} + d_{xC})$.
$\mathbf{a}_C = 0$ when either only two non-zero external streams of memes arrive to WM and they are $\mathbf{G|C}$ with $\mathbf{X|C}$, or $\neg\mathbf{G|C}$ with $\neg\mathbf{X|C}$. $\mathbf{a}_C = 0$ when, for example, the number of incoming $\neg\mathbf{G|C}$, equals the number of incoming $\neg\mathbf{G|C}$, and the number of incoming $\mathbf{X|C}$, equals the number of incoming $\neg\mathbf{X|C}$.
p equals the percentage of the memes representing facts ($\mathbf{G|C}$, $\neg\mathbf{G|C}$, $\mathbf{X|C}$, $\neg\mathbf{X|C}$) in the population of all memes incoming to WM (including the rules $\mathbf{E|G}$ and $\mathbf{E|X}$). Hence, the formula: $\mathbf{p} = 100\ (d_{GC} + d_{gC} + d_{XC} + d_{xC}) / (d_{GC} + d_{gC} + d_{XC} + d_{xC} + d_{EG} + d_{EX})$.
As for the *dependent variable*, **s**-factor has been taken. It represents a hesitation level, i.e. the subject's tendency to oscillate between highly positive and highly negative feelings. $\mathbf{s} = 100$ would mean that for a half of the time-cycles in the period of simulation the value of the order parameter was between 0.9 and 1.0, while for the half of the cycles the value was between 0.0 and 0.1. $\mathbf{s} = -100$ would mean that for the entire period of simulation the value of the order parameter was between 0.4 and 0.6. Hence the formula:

$$\mathbf{s} = (1/65) \sum_{i=1}^{10} (u_i \sum_{Ki=1}^{Ni} q(K_i))$$

where:
$u_i = 0.5\ | i - 5.5 | - 1.25$, $q(K_i) = 25.5 - | 25.5 - K_i |$, N_i is a percentage of time-cycles for which $100h$ belonged to the i-th range, where h is the order parameter while the consecutive ranges are: [0, 1), [1, 2), [2, 3), [3, 4), [4, 5), [5, 6], (6, 7], (7, 8], (8, 9], (9, 10].

3.3 Hypothesis

The hypothesis states, that (1) **p**-factor, representing the percentage of memes containing facts in the population of all memes entering WM, can represent importance of a judgment to the judging subject, and (2) like in the case of people, for important judgments, the higher the cue ambiguity, the higher the level of hesitation demonstrated by the *MemeStorms* model, while this rule does not work for non-important judgments.

3.4 Results

In the first step of the experiment, that $\mathbf{a}_C = 1$, for 99 values of **p**-factor (ranged from 1 to 99), for 40 individuals, the value of **s**-factor was taken. The resulting plot average **s** vs. **p** was non-linear. Near $\mathbf{p} = 40$ a dramatic growth of the average/maximum value of **s** occurred. Hence the proposal to treat certain values of **p** as quantities resulting from the level of importance of a given act judgment. Indeed, assuming that a steady stream of memes representing problem-related flows from the Semantic Memory, the density of the stream of memes representing perceived facts may vary depending on the importance of a given judgment.
Assuming that **p**-factor is an acceptable representation of judgment importance, two series of simulations were run for two selected values of **p** (0.25 representing low judgment importance and 0.77 representing high judgment importance). Six selected values of cue ambiguity (0.75, 0.8, 0.05, 0.9, 0.95, 1.0) were taken. For each of the twelve pairs (**p**, a_C), the behaviors of the 40 individuals were simulated and the values of **s** have been taken. The results are shown in Figure 2.

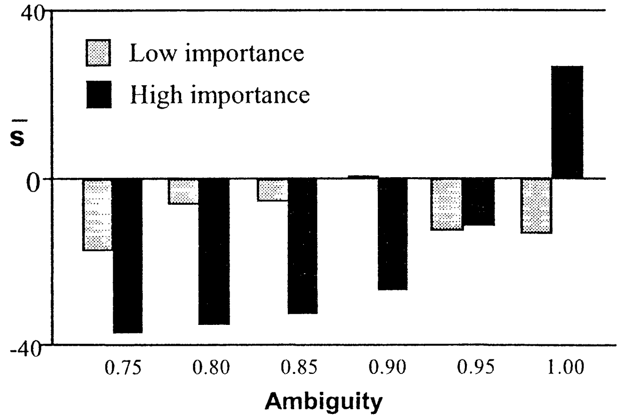


Figure 2. Average values of **s**-factor vs. **ambiguity** of perceived cues. **s** refers to tendency of simulated subjects' feelings to oscillate. Two sets of values of **s**, one for assumed low judgment importance, second for high importance, has been taken for a 16x16-cell Working Memory. In both sets, for six values of ambiguity of perceived cues, 40 simulations were run and average **s** taken. As the figure shows, for important judgments, the bigger ambiguity, the bigger **s**. For less importance judgments, in case of full ambiguity, the simulated subject's feelings may oscillate, but probably never between highly positive and highly negative values. Statistical significance of the results was tested using MANOVA that provided: $F_r = 36.95$ ($p<0.001$), $F_c = 30.01$ ($p<0.001$), $F_{rc} = 41.47$ ($p<0.001$), with $df=479$.

3.5 Discussion

Can **p**-factor represent judgment importance? The plot of **s** (hesitation level) vs. **p** suggests such possibility. For **p** <40, the hesitation level remains low (with small tendency to decrease), while for **p** of the value between 40 and 55 a dramatic growth of the hesitation level takes place. The high level of hesitation remains until **p** ≈ 90. For **p**>90 the hesitation level dramatically decreases.

Based on the above results we took 25 and 77 as the recommended values of **p** to represent low and high importance, respectively. Indeed, as it can be seen in the Figure 1, for assumed high judgment importance (**p** = 77) the subjects' hesitation level (**s**), i.e. the tendency to oscillate between highly positive and highly negative judgments, strongly depends on cue ambiguity, while for assumed low judgment importance (**p** = 25) the tendency is low and not significantly dependent on cue ambiguity. This means that with **p** taken as a representation of judgment importance, simulated subjects strongly tend to oscillate in their judgments when the perceived cues are strongly ambiguous, provided a given judgment is important to a given subject.

The above results are comparable with the psychological evidence presented by Nowak and Vallacher (1998). As for the suggested mental mechanisms, the principles of functioning of the *MemeStorms* model do not fit to the explanation proposed by Nowak and Vallacher (switch between a static integration mode and dynamic integration mode), or to the explanation proposed by Petty and Cacioppo (1986) (the higher importance, the bigger number of mechanisms recruited to information processing). The suggested *MemeStorms*-based explanation admits the possibility that the higher importance, the bigger stream of incoming memes carrying perceived cues, while the stream of incoming memes carrying rules can remain constant regardless the importance of a given judgment. This way, the higher importance, the larger value of **p** (the percentage of memes carrying fact in the stream of incoming memes). Another justification of this proposal is the natural tendency to search for new information in the case of important judgment. In the case of a judgment based on remembered facts, the fact may be replicated in the mental module responsible for an "injection" of memes to WM, which can be associated with the phenomenon of rumination.

A prediction comes from the analysis of the obtained results. Maybe humans, like *MemeStorms* model, lose their tendency to oscillate for extremely high importance of a given judgment (**p**>90). If future experiments with human subjects support this hypothesis, it will mean an additional justification of the *MemeStorms* model.

Conclusion

Computer simulation of meme intractions in the *MemeStorms* model showed that depending on the control parameter, the order parameter can either polarize (for low ambiguity), or oscillate between high positive and highly negative values (for high ambiguity and high importance), or remain close to a medium value (for low importance). Since the results feet to the empirical evidence collected by Vallacher and Nowak during their experiments with human subjects, we argue that the *MemeStorms* model approximates the cognitive mechanisms of human mind.

References

Brodie R (1996) *Virus of the Mind: The New Science of the Meme*, Seattle: Integral Press.

Buller A (1998) *Sztuczny mózg. To już nie fantazje*, Prószyński i S-ka: Warszawa.

Buller A (2000) *Self-Organization of Mind: Theoretical Model, Computer Simulation and Empirical Grounds* (unpublished manuscript), Department of Psychology, Warsaw University.

Buller A, de Garis H (1998) Brain-Building Strategy: Some Remarks and Questions, *Intelligent Information Systems VII, Proceedings of the Workshop held in Malbork, Poland, June 15-19*, 188-193.

Buller A & Shimohara K (1999) Decision Making as a Debate in the Society of Memes in a Neural Working Memory, *The Journal of 3D Forum, 13 (3)*, 77-82.

Buller A & Shimohara K (2000) Does the 'Butterfly Effect' Take Place in Human Working Memory? *The Fifth International Symposium on Artificial Life and Robotics (AROB 5th '00), Jan. 26-28, 2000, Oita, Japan*, 204-207.

Calvin WH (1996) *The Cerebral Code: Thinking a Thought in the Mosaics of the Mind*, Cambridge, Mass: A Bradford Book/The MIT Press.

Dawkins R (1989) *The Selfish Gene, New Edition*, Oxford University Press.

Nowak A & Vallacher RA (1998) *Dynamical Social Psychology*, Guilford Press.

Petty RE & Cacioppo JT (1986). *Communication and persuasion: Central and peripheral routes to attitude change*, New York: Springer Verlag.

Plotkin H (1993) *Darwin Machines and the nature of Knowledge*, Cambridge, Mass: Harvard University Press.

Schacter DL, Wagner AD & Buckner RL (2000) Memory Systems of 1999, In: Tulving E & Craik FIM (2000) *The Oxford Handbook of Memory*, Oxford: Oxford University Press, 627-643.

Tulving E (1995) Organization of Memory: Quo Vadis? In: Gazzaniga MS (Ed.) *The Cognitive Neurosciences*, Cambridge MA: The MIT Press, 839-847.

Vallacher RA, Nowak A & Kaufman J (1994) Intrinsic dynamics of social judgment, *Journal of Personality and Social Psychology, 67*, 20-34.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Dynamic Fuzzy Sets for Cognitive Modeling

Andrzej Buller
ATR International, Information Sciences Division, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
buller@isd.atr.co.jp
and
Starlab N.V., Latour de Freins, Rue Engelandstraat 555, 1180 Brussel, Belgium
buller@starlab.net

Abstract This paper introduces the notion of *Dynamic Fuzzy Set (DFS)*—construct helpful in modeling dynamics of mental processes in biological and artificial brains. Based on DFS one can perform an extraordinary fuzzy inferencing that allows for resulting membership value to change in time even when the processed data remain constant.

Since psychologists empirically confirmed that subject's feelings about a perceived person or social situation change in time and there are cases that subjects' feelings oscillate from a highly positive value to a highly negative value and back, the motivation for building an adequate cognitive model emerged.

The solution proposed in this paper adds dynamics to the classic fuzzy-set-theoretic concept of membership. Membership values representing cues characterizing a perceived object are used as selected elements of the vector $\boldsymbol{a}$—a parameter of the dynamic model $(\boldsymbol{a}_{t+1}, \boldsymbol{x}_{t+1})=F(\boldsymbol{a}_t, \boldsymbol{x}_t)$, where $\boldsymbol{x}$ is a set of state variables characterizing an inferencing engine. Among other elements of $\boldsymbol{a}$ there is a resulting dynamic membership of a perceived object to a fuzzy set representing the resulting judgment.

Preliminary simulation on a certain instance of F show that for some sets of constant membership values constituting the parameter **a**, the resulting judgment tends to a two-state limited cycle attractor, which satisfactorily explains the empirically confirmed dynamics of humans' feelings.

The above approach seems to lead to the philosophical thesis that mind is nothing but relationships that occur between certain notions, and that change over time.

1 Introduction

The concept of fuzzy set, invented in the 60's [1] has proved to reflect the way humans categorize [2]. The development of social cognition caused in the 90's an increasing interest in intrinsic dynamics of human categorization. Experiments confirmed that when judging a perceived person or social situation people sometimes oscillate from highly positive feelings to highly negative feelings even in absence of new data [3]. Similar oscillations were demonstrated by computational models of human working memory that treated mental process as a "debate" in a "society of memes" in a cellular working space [4]. A need to extend the concept of fuzzy set emerged from the research on the models. The proposed kind of fuzzy set, called *Dynamic Fuzzy Set (DFS)* facilitates dealing with elements whose membership changes in time. Below, a draft of a DFS-based model of a mind-brain system has been presented.

2 DFS-based mind-brain model

Let the entity of interest be a space $\mathbf{B}^\circ$ such that $\forall_{a\in \mathbf{B}^\circ}$ $\boldsymbol{a} : \mathbf{T} \rightarrow \mathbf{A}$ where:

i. $\mathbf{T}$ is the space of integers representing *time*;

ii. $\mathbf{A}$ is a space of functions such that $\forall_{\mu\in \mathbf{A}}\, \mu : \mathbf{E}^2 \rightarrow [0, 1]^K$, where K is a positive integer, while

iii. $\mathbf{E} = \{\varnothing, e_1, \neg e_1, e_2, \neg e_2, \ldots\}$ is the space of particular *notions*, where $\varnothing$ denotes *empty set*, while $\neg$ is the operator of *negation*.

Any element of $\mathbf{B}^\circ$ may be considered as a system of relationships between notions. Each of the relationships applies to a pair of notions and expresses itself as a K-element vector of certain real numbers. Elements of the vector can be interpreted as strengths of several sorts of memberships of the second notion in a pair to a fuzzy set represented by the first notion in the pair. Each of the strengths may change over time. Let a fuzzy set in which memberships of its elements change in time be called *DFS* (*Dynamic Fuzzy Set*).

What does in mean when a system consists of a set of related notions and the relationships between the notions change in time? Could not we say that in the system a kind of mental activity takes palace? If we admitted that thinking is nothing but changing relationships between notions, any element of the space $\mathbf{B}^\circ$ could be considered to be a model of a *mind* or even to be a mind itself.

What, in the framework of $\mathbf{B}^\circ$, can be the source of inconstancy of the relationships? In order to not to let the question remain open, let us introduce a *brain* understood as any machine dedicated to a given $\boldsymbol{a}$ that processes every $\boldsymbol{a}_t$ onto $\boldsymbol{a}_{t+1}$. Let us, therefore, introduce the space $\mathbf{B}'$ of minds that use an external device to determine relationships between specific notions for consecutive moments of time. Formally:

$\boldsymbol{a} \in \mathbf{B}' \Leftrightarrow (\boldsymbol{a} \in \mathbf{B}^\circ, \exists_{\langle x,\, F\rangle}\ \boldsymbol{x} = (x_0, x_1, x_2, \ldots, x_N), \boldsymbol{x} : \mathbf{T} \rightarrow \mathbf{M}^N, F : \mathbf{A} \times \mathbf{M}^N \rightarrow \mathbf{A} \times \mathbf{M}^N, (\boldsymbol{a}_{t+1}, \boldsymbol{x}_{t+1}) = F(\boldsymbol{a}_t, \boldsymbol{x}_t))$, where $\mathbf{M} = \mathbf{E}^2$ is the space of *memes*, while N is a positive integer.

The brain is the couple $\langle \boldsymbol{x}, F\rangle$.

In order to test usefulness of the introduced meaning of mind and brain, let us consider an instance of $\boldsymbol{a} \in \mathbf{B}'$ such that:

1. K=2;

2. $\mathbf{E} = \{\varnothing, \mathbf{N}, \mathbf{n}, \mathbf{R}, \mathbf{r}, \mathbf{A}, \mathbf{a}\}$,
where $\mathbf{n} = \neg\mathbf{N}$, $\mathbf{r} = \neg\mathbf{R}$, $\mathbf{a} = \neg\mathbf{A}$;

3. $\forall_{t\in\mathbf{T}}\ \mathbf{m} \notin \{\langle s|m\rangle \in \mathbf{E}^2 \mid m \neq \varnothing \Rightarrow s \notin \{\varnothing, m\}\} \Rightarrow \mu_{s,t}(m) = 0$;

4. $\forall_{t\in\mathbf{T}}\ \mu_{s,t}(\neg m) = 1 - \forall_{t\in\mathbf{T}}\ \mu_{s,t}(\neg m)$;

5. $\forall_{t\in\mathbf{T}}\ x_{0,t} = \langle\varnothing|\varnothing\rangle$,

6. if $u_{i,t} = \langle\varnothing|\varnothing\rangle$ then $x_{i,t} = v_t$, while if $u_{i,t} \neq \langle\varnothing|\varnothing\rangle$ then $x_{i,t} = u_{i,t}$, where

7. $\forall_{\mathbf{m}'\in\mathbf{M}}\ P(v_t = \mathbf{m}') = \mu_1(\mathbf{m}', t)\mu_2(\mathbf{m}', t) / \sum_{\mathbf{m}\in\mathbf{M}} \mu_1(\mathbf{m}, t)\mu_2(\mathbf{m}, t)$, where $\forall_Z\ P(Z)$ is probability of Z;

8. $u_{i,t} : \mathbf{Z}_N \times \mathbf{T} \rightarrow \mathbf{M}$, $\forall$ i,j,k$\in\mathbf{Z}_N$, $u_{i,t} = \psi(x_{j,t}, x_{k,t})$, where $j = L_i(\boldsymbol{x}_t)$, $k = R_i(\boldsymbol{x}_t)$, where $\mathbf{Z}_N = \{\iota \in \mathbf{I} \mid 0 \leq \iota \leq N\}$, where $\mathbf{Z}$ is the space of integers, while

9. $\psi : \mathbf{M}^2 \rightarrow \mathbf{M}$, $\forall_{m\in\mathbf{M}}\ \forall_{\alpha,\beta\in\mathbf{E}\ \mid\ \alpha\neq\beta,\ \neg\alpha\neq\beta}$
$m \in \{ \langle\beta|\alpha\rangle, \langle\neg\alpha|\varnothing\rangle \} \Rightarrow \psi(\langle\alpha|\varnothing\rangle, m) = \langle\varnothing|\varnothing\rangle$;
$m \notin \{ \langle\beta|\alpha\rangle, \langle\neg\alpha|\varnothing\rangle \} \Rightarrow \psi(\langle\alpha|\varnothing\rangle, m) = \langle\alpha|\varnothing\rangle$;
$m \notin \{ \langle\alpha|\varnothing\rangle, \langle\neg\alpha|\varnothing\rangle \} \Rightarrow \psi(\langle\beta|\alpha\rangle, m) = \langle\beta|\alpha\rangle$;
$\psi(\langle\beta|\alpha\rangle, \langle\alpha|\varnothing\rangle) = \langle\beta|\varnothing\rangle$; $\psi(\langle\beta|\alpha\rangle, \langle\neg\alpha|\varnothing\rangle) = \langle\neg\beta|\varnothing\rangle$;

10. $L, R : \mathbf{Z}_N \times \mathbf{M}^N \rightarrow \mathbf{Z}_N$,

$x_{p,t} \neq \langle\varnothing|\varnothing\rangle$	0	0	0	0	1	1	1	1
$x_{q,t} \neq \langle\varnothing|\varnothing\rangle$	0	1	0	1	0	1	0	1
$x_{r,t} \neq \langle\varnothing|\varnothing\rangle$	0	0	1	1	0	0	1	1
$L_{w(\varphi,\lambda,0)}(\boldsymbol{x}_t)$	r	p	r	p	r	p	p	p
$R_{w(\varphi,\lambda,0)}(\boldsymbol{x}_t)$	q	q	p	q	p	q	r	r
$L_{w(\varphi,\lambda,1)}(\boldsymbol{x}_t)$	q	q	q	q	q	q	q	q
$R_{w(\varphi,\lambda,1)}(\boldsymbol{x}_t)$	p	p	p	p	p	p	p	0
$L_{w(\varphi,\lambda,2)}(\boldsymbol{x}_t)$	r	r	r	r	p	r	r	r
$R_{w(\varphi,\lambda,2)}(\boldsymbol{x}_t)$	p	p	q	q	q	q	p	p

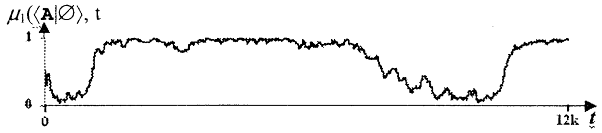
where $p = w(\varphi_p, \lambda_p, 2)$, $q = w(\varphi_q, \lambda_q, 1)$, $r = w(\varphi_r, \lambda_r, 0)$, $\varphi_p = (\varphi + \varphi_{max}) \bmod \varphi_{max} + 1$, $\varphi_q = \varphi$, $\varphi_r = (\varphi + 1) \bmod \varphi_{max} + 1$, $\lambda_q = (\lambda + \lambda_{max}) \bmod \lambda_{max} + 1$, if $\varphi \bmod 2 \neq 0$ then $\lambda_p = \lambda_r = \lambda$, if $\varphi \bmod 2 = 0$ then $\lambda_p = \lambda_r = (\lambda + \lambda_{max}) \bmod \lambda_{max} + 1$, where

11. w is any one-by-one function such that $w : \mathbf{G} \rightarrow \mathbf{Z}_N$, where $\mathbf{G} = \{(\varphi, \lambda, \delta) \in \mathbf{Z} \mid 0 \leq \varphi \leq \varphi_{max}, 0 \leq \lambda \leq \lambda_{max}, 0 \leq \delta \leq 2\} \cup \{e\}$, $(\varphi_{max}, + 1)(\lambda_{max} + 1) = N$, e is any element such that $e \neq \varnothing$;

12. $\forall_{\mathbf{m}\notin\{\langle\mathbf{A}|\varnothing\rangle,\, \langle\mathbf{a}|\varnothing\rangle\}}\ \mu_1(\mathbf{m}, t) = \text{const.}$, while $\mu_1(\langle\mathbf{A}|\varnothing\rangle, t) = N_{\mathbf{A}} / (N_{\mathbf{A}} + N_{\mathbf{a}})$, where for a given notion e, $N_e = C_{e,\,N}$, $C_{e,\,0} = 0$, $C_{e,\,i+1} = C_{e,\,i} + j$, $j = 1$ if $x_{i,\,t} = e$, while $j = 0$ if $x_{i,\,t} \neq e$; where $C_{J,\,i}$ is an auxiliary counter.

13. $\forall_{\mathbf{m}\in\mathbf{M}}\ \mu_2(\mathbf{m}, t) = \text{const.}$

Assuming that **N** and **R** represent 'nicety' and 'richness' of a date proponent, respectively, while $\mathbf{A}|\varnothing$ is a meme suggesting agreement to the date, let us simulate the plot of $\mu_1(\langle\mathbf{A}|\varnothing\rangle, t)$. For given constant values $\mu_1(\langle\mathbf{N}|\varnothing\rangle, t) = .6$, $\mu_1(\langle\mathbf{R}|\varnothing\rangle, t) = .4$, $\mu_1(\langle\mathbf{A}|\mathbf{N}\rangle, t) = \mu_1(\langle\mathbf{A}|\mathbf{R}\rangle, t) = 1.0$, $\mu_2(\langle\mathbf{N}|\varnothing\rangle, t) = \mu_2(\langle\mathbf{R}|\varnothing\rangle, t) = .67$, $\mu_2(\langle\mathbf{A}|\mathbf{N}\rangle, t) = \mu_2(\langle\mathbf{A}|\mathbf{R}\rangle, t) = .2$, F produces:



Conclusions

An essential feature of the resulting plot is its intrinsic dynamics. In the presented case the membership of ∅ to the dynamic fuzzy set **A** tends to a 2-state limit cycle attractor. This result can be treated as a counterpart of hesitation demonstrated by human subjects during evaluation of a perceived object. Hence, the concept of dynamic fuzzy set seems to be a valuable tool for modeling of and theoretical investigations on natural and artificial mind-brain systems. DFS-based modeling also seems to lead to the philosophical thesis that mind is nothing but relationships that occur between certain notions, and that change over time.

References

[1] Zadeh L (1965) Fuzzy Sets, *Information and Control*, *8*, 338-353.

[2] Rosch E (1978) Principles of categorization, In: Rosch E & Lloyd BB (1978) *Cognition and categorization*, Hillsdale, NJ: Erlbaum, 27-48.

[3] Nowak A & Vallacher RA (1998) *Dynamical Social Psychology*, Guilford Press.

[4] Buller A & Shimohara K (2000) Does the 'Butterfly Effect' Take Place in Human Working Memory? *The Fifth International Symposium on Artificial Life and Robotics (AROB 5th '00), January 26-28, 2000, Oita, Japan*, 204-207.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Suitable evolutionary strategies for large scale neural networks

Peter Eggenberger
Emergent Communication Mechanisms Project
Information Sciences Division
ATR International
2-2-2, Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0288, JAPAN
Email: eggen@isd.atr.co.jp

Abstract

Evolving large neural networks with user-defined properties is still an unsolved problem in the field of artificial evolution. The encoding problem (what the genes should represent) and the scaling problem (the number of genes grows with the number of neurons) are not yet satisfactorily solved. Nature solved these problems by a combination of explicit instructions and cooperative processes (these are procedures that generate large-scale effects by sequences of local operations). Therefore the use of mechanisms is proposed which are imitating these biological mechanisms in order to create more powerful artificial evolutionary systems. Many biological mechanisms such as the gene regulation, the patterning of the neural networks or the mechanisms of synaptic plasticity are all based on more or less specific molecular interactions. With evolutionary strategies it is possible to explore such simulated cooperative processes on an evolutionary as well as an ontogenetic level by varying the properties of artificial molecules and their dynamics. Every modeled cell contains a genome, which allows them to communicate and to specify their properties in space and time. By tuning these communication pathways in an usual evolutionary cycle, it is shown that the developmental regulation is evolved and that the system's encoding scheme scales independently of the quantity of the cells. The principles are illustrated by evolving a simple, but large retina.

1 Introduction

In this paper I address three issues in relation to artificial evolution: the genetic encoding of a neural networks, the scaling properties of the encoding scheme and the robustness against change of the evolved results. Methods which have to evolve large neural networks have to solve these problems efficiently and reliably. Standard evolutionary techniques ([1, 2] are used to create a population of an array of artificial genes (bit strings or real-valued numbers). These genes should represent the number of neurons, their properties and their connectivity in an efficient way to allow the evolution of large neural networks. These properties can then be changed by mutation and selection. This process is then repeated and over time better and better solutions of designer defined functions (fitness function) should be found. In order to be able to evolve large neural networks, the length of the genome should be independent of the number of the neurons or synapses. In the literature two broad groups can be distinguished. In the first group [3, 4, 5] direct encoding schemes are used to evolve neural networks, which are simple to implement, but have the drawback that the genome will grow with the number of neurons or the synapses. The other group [6, 7, 8, 9, 10, 11, 12, 13] proposed developmental processes to evolve neural networks. In contrast to these approaches in the proposed model the genes are regulated by continuous regulatory genes which have good evolutionary properties. The efficiency of the biological encoding is illustrated by the following numbers: approximately $1o^5$ genes of the human genome are able to control the development of a nervous system containing 10^10 neurons with 10^15 synapses [14]. Therefore any direct encoding scheme (one gene represents one neuron and/or one synapse) will be the disadvantage of increasing unnecessarily the genome length and therefore also the search time will increase. To overcome this drawback the proposed method implements a continuous genetic regulatory network, which is able to explore cooperative mechanisms. The key idea is that the genes do not encode directly the properties of the neural networks, but encode processes which allow them to produce a neural network with certain properties. As an illustration a retina is evolved which is moved by two pairs of antagonistic motors and has to learn to focus on an incoming stimuli. As the proposed encoding scheme is independent on the number of neurons and synapses, the genome can be used for small as well as large retinas (see figure 3). It is not only possible to encode the structure of a neural network [15], but also mechanisms which change the synaptic weights. The proposed method is not based on a simplification of the encoding scheme, but simple problems have simple genomes, and complex problems will have complex genomes. Furthermore, as the genes encode developmental processes to create neural networks and their connections to evolved sensory and motor systems, I will show that these processes are quite robust against change. For example, if once a single sensory cell is able to find and connect to a neuron, it is easy to evolve a system with many (thousands) of sensors and neurons. Also the positions of the sensors or motors can be changed without decreasing the functions very much. Therefore, during evolution the system can explore different morphological arrangements of the effectors and sensors without disturbing the neural functioning.

2 Method

In earlier work we described how the AES is able to control cell growth, the patterning of neural structures [15] and to imitate the patterning of butterfly wing patterns [16]. In this paper the scope is extended to the evolution of mechanisms which change the synaptic weights. Again the mechanism is based on specific molecular interactions, called ligand-receptor interactions. The synapses are able to express receptors, which are used to change their weights. Substance stemming from other cells can interact with these synaptic receptors and if the impact is high enough the weight of a synapse may change. The impact depends on the affinity between the substance (ligand) and the receptor, their effect (negative or positive) and their concentrations. By varying the receptors and ligands, the AES is able to evolve learning rule without predefining classes of learning rules as in other approaches [4, 5]. No predefined rules such as Hebbian or Antihebbian rules exist in the system, but the AES has to find them by evolving the "right" communications between the cells and the correct interactions. The advantage is that also other learning rules can be evolved which are quite different than those a designer had in mind during the implementation. The principles of how learning rules are evolved is illustrated in figure 1.

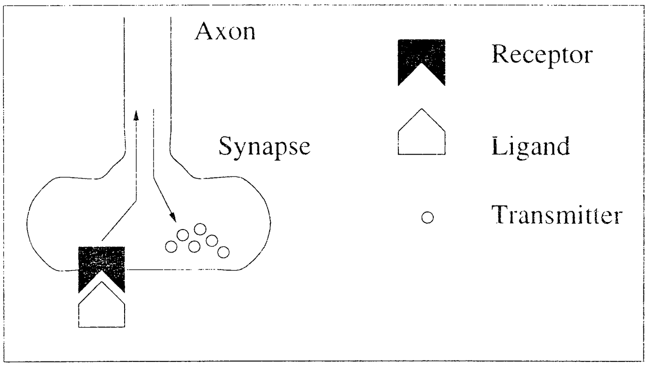


Figure 1: Axons can express receptors on their synapses which control the synthesis of the transmitters. In this way the AES can control the weights of the synapses and can explore different mechanisms how to change the weights.The ligand-receptor interactions have an effect on the gene activity and control the synthesis of transmitters. The concentration of the transmitter represents the weight of a synapse.

3 Results

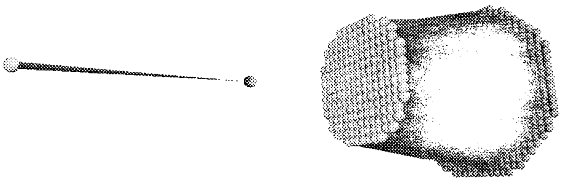


Figure 2: Once a mechanism is evolved to connect a sensory cell to a neuron, the genome length is independent of the number of cells or their connections.The two pictures illustrate this process. The same developmental mechanism is able to connect one receptor or many of them with the same genetic information.

Two different experiments illustrate the above discussed principles. In a first experiment the structure of a retina had to be evolved in which sensory neurons had to connect to interneurons. Every sensory neuron had to have at least one connection to a motor cell. The fitness function simply tested the cells if an activation of the sensor would lead to an activation of a motor cell. Compared to the direct encoding

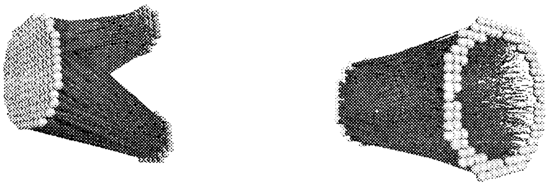


Figure 3: A further advantage of the AES is its robustness against change in locations. As the neurons call the sensory axons by diffusing signals, the neurons as well as the sensors can change their locations without disturbing the function of the whole. Mechanisms of this kind are important as the system can explore its possibilities without double mutations.

schemes several components have to play together before a correct function appears. The sensors have to be differentiated in a way that they emit an axon, the target cells emitted a substance, which gave the sensory axons cues for their searching behavior. Furthermore the axons and the target cells had to establish a synapse with an activating effect on the target cell. There are at least five different things which have to be correct concurrently. Therefore the probability of such interactions should not be to low to get different fitness values. But over time, to avoid disturbing interactions, the probabilities will decrease by increasing the specificity of the molecular interactions.

The second experiment was done with an evolved retina which had to learn to focus on a visual stimuli applied at the periphery of the retina. The task for the AES was to evolve the neural connections and cellular interactions, which were able to teach the retina to focus on a stimuli. The result of the best found solution was dependent on the structure as well as the cellular interactions. The sensory cells emitted axons to the center of the neural cells in such a way that movements in direction of the center activated the neurons, but movements in the opposite direction had no effect. The synapses of the neurons evolved two type of receptors, which were sensitive to the transmitter substances of the neighboring cells and to a value signal emitted by the active cell itself.

The important points to note are that the AES allows to explore the structure of a neural network concurrently with mechanisms of changing synapses and that the structure of the neurons for the sensory cells is important for a correct learning

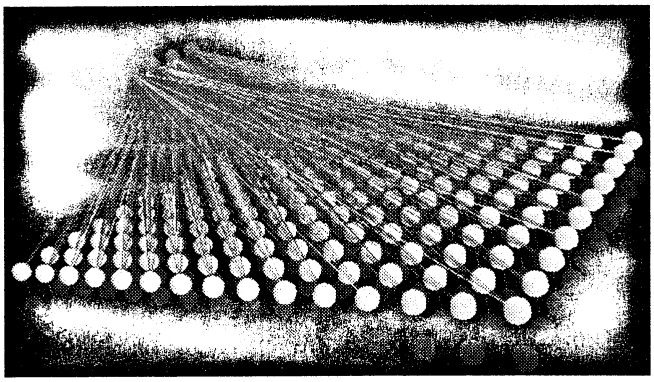


Figure 4: An example of an evolved retina able to focus on a stimuli. If the sensory cells are stimulated in the periphery the retina moves in such a way that the stimuli will be in the center of the retina. The evolved learning mechanism depend on the cellular interactions and on the structure of the neural network.

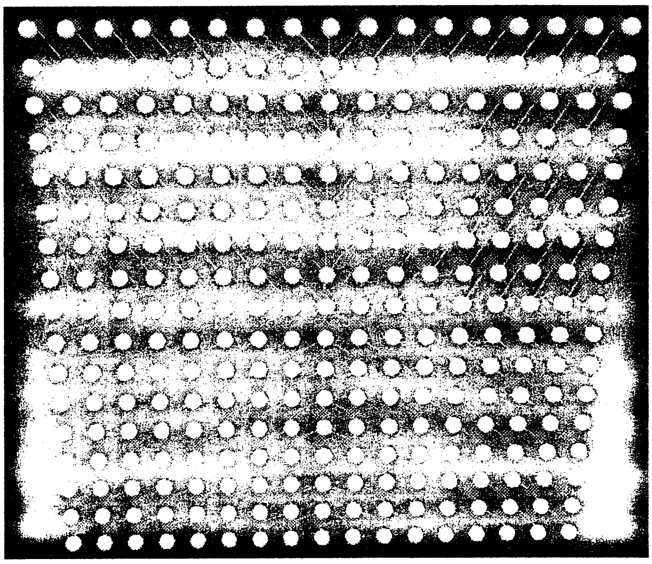


Figure 5: The connections of the sensory cells to the interneurons are shown. The synapses had expressed two receptors, one for a signal indicating the activity of a neuron and a second one indicating the firing of the synapses (analog to the transmitter substance). In this way the simultaneous activities of the synapses as well as the neuron can be detected and used to change the synaptic weight.

4 Summary

The use of cooperative processes such as continuous genetic regulatory networks, the patterning mechanisms as well as mechanisms for the change of synaptic weights have all an advantage for large neural networks, because they can decrease the necessary information decoded in the genome. The genetic search space will decrease and the search time will be less. Although being more complex than any direct encoding scheme the proposed AES shows good scaling behavior, can be easily extended to incorporate adaptive mechanisms by evolving synaptic mechanisms allowing the change of their weights. Furthermore, the rules where and when to apply a developmental rule are all evolved and the designer restrictions are only on the level of which kind of different structural genes (representing effectors, sensor types, pigments for wing patterns etc). Also the interplay between the patterning of a neural structure and the mechanisms changing the neural communication (synaptic as well as cellular) are now in the reach of the proposed AES as shown with the example of the evolved retina. In future work I will apply the proposed methods for the evolution of complete agents in a real-world simulator.

Acknowledgments

I thank Dr. K. Shimohara of having supported this work.

References

[1] Ingo Rechenberg. *Evolutionsstrategie.* frommann-holzboog, 1973.

[2] John H. Holland. *Adaptatin in natural and artificial systems.* Ann Harbor: The University of Michigan Press, 1975.

[3] Inman Harvey. Species adaptation genetic algorithms: The basis for a continuing saga. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 346–354. MIT Press/Bradford Books, Cambridge, MA, 1992.

[4] Dario Floreano and Francesco Mondada. Evolution of plastic neurocontrollers for situated agents. In [17], pages 402–410, 1996.

[5] Dario Floreano and Joseba Urzelai. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13:431–443, 2000.

[6] Richard K. Belew. Interposing an ontogenic model between genetic algorithms and neural networks. In *Advances in Neural Information Processing Systems (NIPS),S.J. Hanson and J.D. Cowan and C.L.Giles,Morgan Kauffman:San Mate,1993*, 1993.

[7] Hiroaki Kitano. Evolution of metabolism for morphogenesis. In Rodney A. Brooks and Pattie Maes, editors, [18], pages 49–58, 1994.

[8] Stefano Nolfi, Dario Floreano, Orazio Miglino, and Francesco Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In P. Meas R. A. Brooks, editor, *Artificial Life IV*. Cambridge, MA: MIT Press, 1994.

[9] Angelo Cangelosi, Domenico Parisi, and Stefano Nolfi. Cell division and migration in a 'genotype' for neural networks. *Network*, 5:497–515, 1994.

[10] Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial Life*, 1:353–372, 1995.

[11] Frank Dellaert and Randall D. Beer. A developmental model for the evolution of complete autonomous agents. In [17], pages 393–401, 1996.

[12] Jari Vaario. Modelling adaptive self-organization. In *Proceedings of the Fourth International Workshop on the Synthesis of Living Systems*. 1994.

[13] Jari Vaario and Katsunori Shimohara. Synthesis of developmental and evolutionary modeling of adaptive autonomous agents. In [19], pages 721–726, 1997.

[14] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. *Essentials of Neural Science and Behavior.* Appleton & Lange, 1995.

[15] Peter Eggenberger. Evolving neural network structures using axonal growth mechanisms. In *Proceedings of The Fifth Int. Symp. on Artificial Life and Robotics (AROB 5th'00)*, pages 208–211, 2000.

[16] Peter Eggenberger and Raja Dravid. An evolutionary approach to pattern formation mechanisms on lepidopteran wings. In *Congress of Evolutionary Computation, Washington*, pages 337–342, 1999.

[17] Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors. *From animals to animats 4: Proceedings of the fourth international conference on simulation of adaptive behavior.* MIT Press, 1996.

[18] Rodney Brooks and Pattie Maes, editors. *Artificial Life IV: Proceedings of the Workshop on Artificial Life*, Cambridge, MA, 1994. MIT Press. Workshop held at the MIT.

[19] Wulfram Gerstner, Alain Germond, Martin Hasler, and Jean-Daniel Nicoud (Eds.), editors. *Seventh International Conference of Artificial Neural Networks (ICANN'97)*, Cambridge, MA, 1997. Springer.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Development of an Artificial Brain Structure for the Behavior Control of the Welfare Liferobot.

○Andrey Loukianov, Masanori Sugisaka
Department of Electrical and Electronic Engineering, Oita University
700 Dannoharu, Oita 870-1192, JAPAN.
aloukian@cc.oita-u.ac.jp msugi@cc.oita-u.ac.jp

Abstract

This paper discusses the development of an artificial brain to control the behavior of an intelligent mobile welfare service robot (called Liferobot). The artificial brain has to provide the robot's control system with adaptability to the environment changes and ability to learn new skills. The structure of the artificial brain is proposed and discussed. The part of liferobot vision system and the demonstration case of low-level behavior learning are presented.

Keywords: *Artificial brain, Intelligent control, Behavior control*

1. Introduction

The problem of the artificial brain design is a new developing research area. During the recent years there has been a growing number of research projects aiming to understand and reproduce the functions of brain in available electronic hardware. To model a living brain using electronic and digital hardware, it is important to understand the biological functions of brain and processes inside it before trying to implement them in hardware and software.

Neural networks were found to be best suited for the modeling of brain functions because of their highly parallel and distributed information processing capabilities [7]. These features of neural networks are fully exploited in case of their hardware realization. Some interesting publications from MIT and ATR can be found concerning this topic [1, 2]. While the hardware implementation of a complex artificial brain is still in the "proof of concept" stage some researchers are looking for a more specific and simple solution of this problem designing primitive artificial brains to control some objects using existing hardware [4].

But the artificial brain itself is not a final goal in the applied fields where it is supposed to control a specific agent and give this agent some degree of intelligence and adaptability. In order to control an agent the artificial brain, like human brain, needs to receive information from the external world through the set of sensing devices. Since the sensing devices are of different kinds it is necessary to develop the methods to convert their output into the stimuli signals for later processing in the artificial brain.

This paper aims to discuss the structure of an artificial brain to be used in the intelligent control system of a mobile service robot for human welfare application. The prototype of such a robot, which we call a Welfare Liferobot, is being developed in Oita University for the purpose of assisting elderly and infirm people at hospitals [5]. The paper will concentrate on laying down the basic principles of this specific artificial brain design and present several methods developed within this framework. In section 3 we provide a description of the liferobot configuration and its functions. In section 4 we propose the structure of the artificial brain for liferobot. In sections 5 and 6 we present a part of liferobot vision system and demonstrate a case of low-level behavior learning.

2. Background

In the area of hardware implementation of artificial brain significant contributions were made by research teams at MIT and ATR. The research team at MIT has proposed Cellular Automata Machine based hardware tool (CAM-8 brain) [1] which can evolve a complex neural network module inside it and run it in real time. The ATR's neuron artificial brain project has succeeded to develop "CBM Brain" [2] which is a programmable logic device capable of growing and evolving thousands of cellular automata network modules with total up to 75 million neurons inside it and running them at rate of 130 billion updates per second. While the projects have achieved good results for modeling of low-level functions of the brain it is still not clear how these brain devices can be used to control the behavior of intelligent agents or learn new knowledge after their structure is evolved. Cho and

Song [3] had attempted to utilize CAM-Brain to control the motion of a Khepera robot.

Sugisaka [4] has reported development of some primitive artificial brains based on combination of an usual computer and a neurocomputer for pattern recognition and motion control of robots. But these brains also are unable to learn on-line because the training of neurocomputer is performed off-line in a supervised manner and because a neurocomputer can offer neural networks of unalterable type and structure. From this viewpoint a flexible software realization of different neural networks types and structures seems to be more promising.

Behavior based approaches to the robot control have proven to be robust in many applications. The paper [6] provides a good review and an extensive discussion on different concepts of robot behavior control. It was noted that behavior based methods usually deal with simple tasks but yield real-time reaction to environmental changes and unexpected events.

3. Liferobot description and functions

The Welfare Liferobot is an intelligent autonomous mobile robot with its own control system on-board and the set of sensors to perceive an environment. The robot is driven by two independent wheels on the sides. The robot control system consists of three networked computers: two PC's and a notebook. The set of sensors includes: two CCD cameras mounted on the rotating head, a microphone for voice recognition, driving wheel encoders and six ultrasonic sensors to detect obstacles. The LCD touch-screen is utilized to provide an interface with a robot user or an operator. The controllable devices of the robot include: two driving motors, six motors for pan-tilt-zoom of the cameras, a head rotation motor and four LEDs to display a status.

The intelligent skills that are expected from the liferobot control system can be described as:

- Ability to navigate in known and unknown environments. This may include a visual orientation or an odometry based motion with obstacle avoidance. Intelligent skills may involve capability of exploration in the unknown environment with a map building, acquisition of knowledge on how to navigate in similar environments, etc.
- Ability to recognize the patterns and objects. Visual recognition can be used to recognize specific patterns that will affect the liferobot behavior. For example, symbols and marks can be used for the navigation purposes, face and body posture recognition can be used to identify and classify a person the robot is dealing with).
- Ability to understand the voice instructions and to communicate by voice. This feature can be used as a high-level control (control goal selection) or as an intelligent feedback to the control system during learning (a new person can be introduced to the robot by voice so it can store his face in the memory to recognize him later).

Apparently, these skills have to be coupled between each other. For example, the voice instructions may be used to resolve a navigation problem. The functions listed above are by no means to be considered complete but they provide a basis for developing the required system architecture.

4. Artificial brain structure

In design of an artificial brain with given above functions it is essential to consider its overall architecture. Based on the ideas from [4, 5] we propose the structure of the liferobot artificial brain, which is shown in Figure 1.

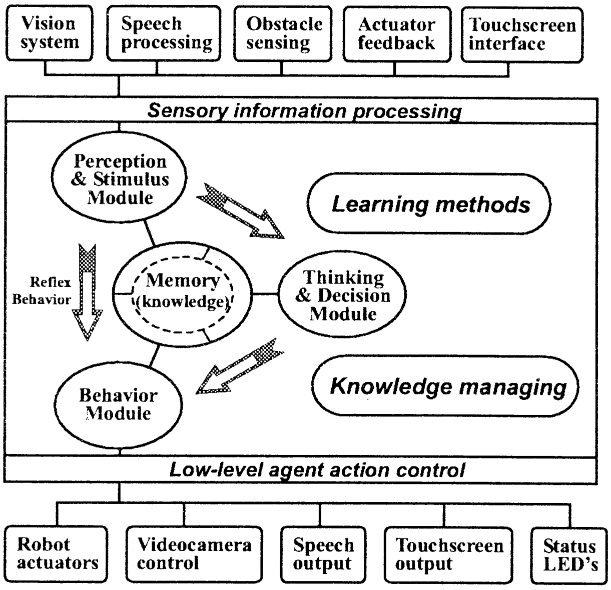


Figure 1. Structure of the artificial brain for the liferobot

The proposed structure is based on the "Sense, Think and Act" paradigm according to which the artificial brain is divided into three components: (a) a sensory system to perceive the world situation and generate corresponding set of the stimuli; (b) a thinking and decision system which develops its internal world representation and has a set of the behavior rules on how to react in specific situations to certain stimuli; (c) a behavior generation system that translates the selected behavior into the sequence of low-level actions. Some behaviors may be generated directly in response to sensory output (reflex behavior).

The processing architecture of the artificial brain is a network of computers with equally distributed computational workload. Since real-time image processing and a speech recognition are computationally intensive tasks the two of three computers will be assigned mostly to these major tasks. The remaining computer will host the thinking-decision and behavior generation modules, a knowledge base and the low-level control procedures. The computation architecture of overall control system will represent the set of the concurrent multithreaded processes communicating over the network.

In the beginning the robot is supposed to be provided with initial knowledge about stable behaviors, learning methods and knowledge managing. With long-term objective to develop a self-learning intelligent control system it is important to review the available methods and approaches which can be used in the components of the artificial brain. Let us briefly discuss this topic.

4.1. Sensory system

Before passing the information from vision and voice sensors to perception module some low-level information processing methods are to be applied. Since these methods are hardcoded and are no subject to change when the system is learning they will greatly affect the system performance and its ability to learn. Obviously, a number of different approaches should be used to provide the system with bigger adaptivity and learning capacity. The high-level perception and recognition methods should be able to learn mostly in unsupervised mode, recognizing similar sensory patterns and memorizing the new ones. The ART neural networks seem to meet this problem condition the best [7].

4.2. Thinking and decision making

Firstly the decision making is going to be realized through the set of fuzzy rules. In the beginning it will form a basic set of liferobot's stable behaviors which could later be extended by reinforcement learning methods. Further development of this module of artificial brain may lead to creation of the behavior-augmented expert system.

4.3. Behavior generation

Once the current behavior is established by decision module its translation into the set of required actions becomes almost a straightforward operation. It may be achieved using an adaptive rule table (actions database) which can be expanded by reinforcement learning. For the purpose of low-level control of robot state variables, for example, tracking tasks, a neural and fuzzy controllers may be applied.

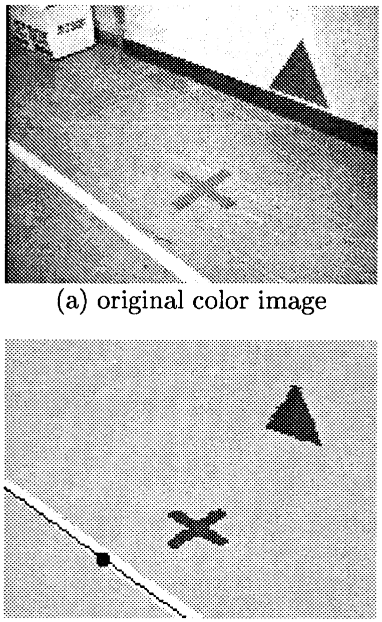
5. Vision system

It is anticipated that the liferobot will receive the largest part of information about the external world through its vision system. For the better results the vision system should have a collection of different image processing methods. An appropriate method is to be chosen automatically according to the situation and the current goal specification. Here we briefly present a part of liferobot vision system, our image processing method for obtaining the information about objects in the surrounding environment.

The visual information is usually represented by large amount of digital image data. The main task of the vision system is to extract desired features from the image and describe them by a small number of attributes. In our proposed method we assume that the world is comprised of objects of different colors. The vision system finds these objects in the camera image, extracts them and describes their shape and properties by the finite number of attributes. To describe the shape of objects the Hu's moment invariants [8] are used. The obtained information may be used for the navigation purposes (color guidelines and reference marks), pattern recognition, etc.

The procedure, which is applied to each image acquired from the camera, can be summarized as follows. The image is compressed and converted to HSV color space. Then the color indexing of image pixels is performed by applying the lookup table. Finally, an object labeling is carried out during which all properties of found objects are computed.

The result of the described steps is a list of extracted objects with the following computed attributes: color, size, location on image, orientation (for elongated objects), first 7 moment invariants. An example of proposed visual system operation is shown in Figure 2. In this figure the output of visual system (b) after processing the original image (a) contains three objects: a line, a triangle and a plus sign. For the line object the center of gravity location and the orientation of line are shown.



(a) original color image

(b) extracted objects

Figure 2. An example of camera image processing by the vision system

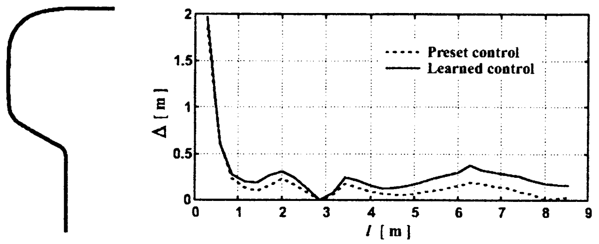
6. An example of behavior learning

The robot is supposed to be provided with an initial set of stable behaviors. The other necessary behaviors have to be learned by the robot in a supervised or unsupervised manner. In this section we present an example of how the supervised learning can be used to generate a new basic behavior. A guideline following behavior was chosen for the learning. It is a low-level behavior which involves a continuous control of the robot state variables.

During the learning procedure the robot is guided manually by an operator along the training paths to collect the training data. In this example the path for a straight line approach were used. Then the learning algorithm processes the gathered data and trains a multi-layer feed-forward neural network, which is used afterwards to control the wheel velocities. The details of the learning algorithm can be found in another our paper submitted to this conference. The experimental results of tracking the 9-meter long "question mark" shaped color guideline with the learned and the preset controllers are shown in Figure 3. The initial large absolute error exists in the graph because the robot starts approaching guideline from distant position.

7. Conclusion

The paper deals with the design of an artificial brain for an intelligent control of the mobile service robot. The architecture of the artificial brain is proposed and discussed. The examples shown illustrate the progress of the system development.



(a) An experimental guideline shape

(b) The absolute tracking error

Figure 3. The results of a guideline following experiment

References

[1] F. Gers, H. de Garis, "Porting a cellular automata based artificial brain to MIT's cellular automata machine 'CAM-8'," *Proc. of the Asia-Pacific Conference on Simulated Evolution and Learning*, Taejeon, Korea, pp. 321–330 (1996).

[2] H. de Garis et al., "Building an artificial brain using an FPGA based CAM-Brain Machine," *Applied Mathematics and Computation*, Vol. 111, pp. 163–192 (2000).

[3] S.-B. Cho, G.-B. Song, "Evolving CAM-Brain to control a mobile robot," *Applied Mathematics and Computation*, Vol. 111, pp. 147–162 (2000).

[4] M. Sugisaka, "Design of an artificial brain for robots," *Artificial Life and Robotics*, Vol. 3, pp. 7–14 (1999).

[5] M. Sugisaka, "Cognitive and behavioral artificial liferobot," *Proc. of the ISIM 2000*, Oct. 3, Korea, pp. 194–198 (1999).

[6] P. Baroni, D. Fogli, "Modelling robot cognitive activity through active mental entities," *Robotics and Autonomous Systems*, Vol. 30, pp. 325–349 (2000).

[7] L. V. Fausett, "Fundamentals of neural networks," *Prentice Hall*, (1994).

[8] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, pp. 179–187 (1962).

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Methodology of Emergent Synthesis

Kanji Ueda
Dept. Mechanical Engr.
Kobe University
Kobe 657-8501, Japan
ueda@mech.kobe-u.ac.jp

Hisashi Tamaki
Dept. Comput. & Systems Engr.
Kobe University
Kobe 657-8501, Japan
tamaki@al.cs.kobe-u.ac.jp

Itsuo Hatono
Information Processing Center
Kobe University
Kobe 657-8501, Japan
hatono@kobe-u.ac.jp

Abstract

The state-of-the-art of the Project "Methodology of Emergent Synthesis" is summarized. In the paper, first, the aim and the framework of the project are shown, where the problems in synthesis of artifactual systems are categorized into three classes, and then, emergent properties to be considered in designing the methodology of emergent synthesis are discussed. Second, the activities of each sub-theme of the researches in this project are briefly introduced.

1 Introduction

Recently, we are facing increased complexity and uncertainty arising from such factors as diversification of culture, individualization of lifestyle and globalization of industrial activities. The increasing complexity and uncertainty bring about practical and theoretical difficulties in all the domains of artifactual activities. The most essential point is how to realize an artifactual system that achieves its purpose in an unpredictable environment. Thus, it is not easy to approach the problems in synthesis only by adopting existing principles.

In this paper, the state-of-the-art of the Project "Methodology of Emergent Synthesis" in the Research for the Future Program supported by JSPS (the Japan Society for the Promotion of Science) is summarized. First, we define the problems in synthesis and classifies the difficulties with synthesis problems. Then, the concept of emergence and its relation to solving the difficulties are clarified. Furthermore, the recent activities of the researches in the project are outlined, and future directions are summarized.

2 Framework of Emergent Synthesis

2.1 Synthesis

Synthesis is clearly related to human activities for creating artificial things, while analysis is related to understanding the things. Analysis is an effective method to clarify the causality of existing systems, and enable one to unfold the structure that realizes the functions of the systems. On the other hand, synthesis is indispensable for creating novel artifacts that satisfy required functions as given by human. The latter type of problem, i.e., from a function to a structure, can be called as an inverse problem, while the former, i.e., from a structure to a function, called a direct problem [1].

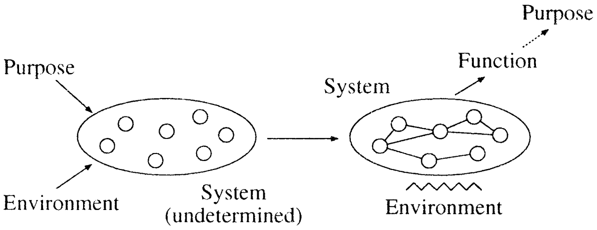


Fig. 1 Problem of synthesis.

Now, consider the solution method of the synthesis problem, i.e., how to determine the structure of a system in order to realize its function to achieve a purpose under the constraints of a certain environment, as shown in Fig. 1 [2]. It is to be noticed, however, that any framework for solving the synthesis problem has to include an analytical part, as indicated in Fig. 2 [2, 3]. It is unlikely that one can find the solution of a synthesis problem by either induction or deduction alone. It is more likely that abduction is needed, too. In abduction, one first makes a proposition of a candidate structure, then one analyzes the structure in order to find the function within the environment. As a consequence, once the function satisfies the specified purpose, one adopts the structure as a solution.

As for the hierarchy of synthesis, one may broadly consider two levels: (a) synthesis of a technique, conventionally by human and (b) synthesis of a system, i.e, a final result, by using the technique, as shown in Fig. 3. The final results correspond to such items as a design solution, a layout of production line and a production schedule, while the technique includes such mechanism as self-organization, evolutionary computation, adaptation and learning.

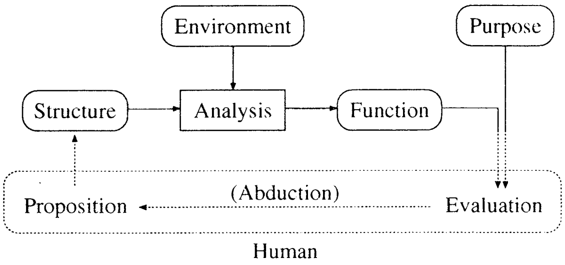


Fig. 2 Framework of synthesis.

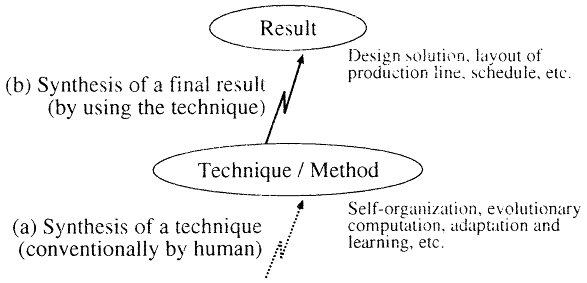

Result
Design solution, layout of production line, schedule, etc.
(b) Synthesis of a final result (by using the technique)
Technique / Method
Self-organization, evolutionary computation, adaptation and learning, etc.
(a) Synthesis of a technique (conventionally by human)

Fig. 3 Hierarchy of synthesis.

2.2 Difficulty in Synthesis

The main concern here is when and whether completeness of the information could be achieved in the description of an environment and in the specification of the purpose of a system. With respect to imcompleteness of the information of the environment and/or the specification, the difficulties in synthesis problem can be categorized into three classes [2, 3].

- Class I — problem with complete description: If the information concerning both the environment and the specification are fully given, then the problem is completely described. However, it is often difficult to finf the optimal solution.
- Class II — problem with incomplete environment description: The specification is complete, but the information on the environment is incomplete. Since the problem is not wholly described now, it is difficult to cope with the dynamic properties of the unknown environment.
- Class III — problem with incomplete specification: Not only the environment description but also the specification are incomplete. The problem solving, therefore, has to start with an ambiguous purpose, and a human interaction becomes significant.

The problem becomes more difficult as the complexity of a system and/or the informational uncertainty increase, which is summarized as shown in Fig. 4.

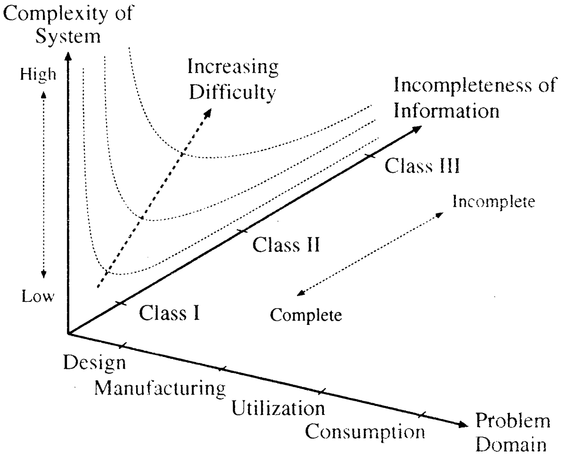


Fig. 4 Classification of synthesis problem and domains.

2.3 Emergent Synthesis

So far, not a few kinds of the definition of emergence have been presented [1]. Here, we use the term "emergence" in the following sense: a global order of a structure expressing a new function is formed through bi-directional dynamic processes, where local interactions among elements reveal a global behavior and the global behavior results in new constraints on the behavior of the elements [1]. According to this definition, the concept of emergence is schematized as shown in Fig. 5. The importance is the formation of a stable global order, which is neither fixed, periodical nor chaotic, but a complex structure. A stable global order gives a new function, and if the function meets the specified purpose, one can adopt it as a solution.

The concepts of the purpose and the environment are defined more definitely in our study, where emergence is investigated in association with the synthesis of artifactual systems — which has to be distinguished from the understanding of emergent phenomena related to existing natural things as in the case of physics and biology.

Furthermore, instead of the traditional, analytic and deterministic approaches based on the top-down problem decomposition such as operational research, artificial intelligence and knowledge-based engineering, the emergence related approaches are being developed with both the bottom-up and the top-down features, which include evolutionary computation, self-organization, behavior-based methods, reinforcement learning, multi-agent systems, etc., and they seems to be feasible for offering efficient, robust and adaptive solutions to the problem of synthesis.

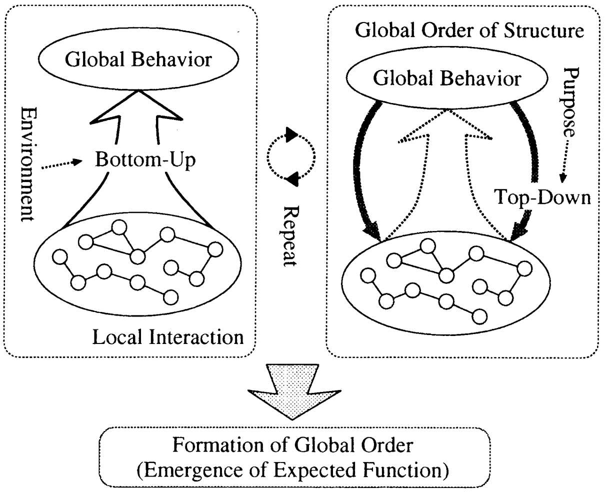


Fig. 5 Concept of emergence.

3 Methodology of Emergent Synthesis

In the project "Methodology of Emergent Synthesis", the difficulties of synthesis problem solving are classified, as described in the above section, and the role of the emergent processes is clarified in each stage of design, manufacturing, utilization and interaction with users. By systematically composing an emergent methodology and verifying its effectiveness in solving the problems of synthesis, the project aims at contributing to establishment of the science of synthesis [4] and to applicability to practical engineering phases. The research members participated from multiple disciplines are about twenty, such as mechanical engineering, information science, mathematics, biology, economics and philosophy.

The project includes the following sub-themes (the details are available in http://www.mi-2.mech.kobe-u.ac.jp/~mirai/index-e.html): emergent synthesis of design systems, emergent synthesis of manufacturing systems, emergent computation for utilizing artifactual systems, and synthesis of relational emergence in artifactual environment.

3.1 Emergent synthesis of design systems

In this sub-theme, the characteristics of engineering systems and products are considered with respect to their structures as hierarchical systems. In synthesizing artifactual systems and/or products, such structures and their attributes should be fixed according to implicit structures. Since design aims at the construction of the whole structure of a system so as to realize the required function, the design process as a whole — from the conceptual design to the detail design — semms to be top-down at a first glance. In order to describe this process of design, however, one has to compose the whole system with accumulating a number of subsystems. Thus, emergent methodologies could play important roles in discovering novel patterns of structures in an efficient way. As for the recent research, see the references [5]–[10].

3.2 Emergent synthesis of manufacturing systems

This sub-theme is carried out by the concept of biological manufacturing system (BMS) [11]. The BMS aims at dealing with non-foreseen changes in the manufacturing environment. It is based on the ideas inspired by biology such as self-organization, evolution, learning and adaptation. For the problems in Class III, especially, an approach of interactive manufacturing has been proposed. In this approach, human beings — designers, manufacturers and customers — and artifacts interact with each other in a virtual space with the aid of self-organization, so that the performance of each participant is improved in an interactive manner. The recent research in this sub-theme includes [12]–[17].

3.3 Emergent computation for utilizing artifactual. systems

In order to construct an emergent approach for the optimization, the following research works have been carried on: a multi-agent support systems for manufacturing scheduling, multi-objective optimization with evolutionary computation, optimization in uncertain environments, and optimization of the utilization of artifactual systems by reinforcement learning. Based on these approaches, the final goal is to develop a framework which may deals with such phases as modeling the problem, designing the solution, and interaction with human being. See the references [18]–[22] for the recent research.

3.4 Synthesis of relational emergence in artifactual environment

Emergence in the relationship between manufacturers, consumers and artifacts is considered, where a network model is adopted with nodes consisting of autonomous agents. These agents interact with each other via products. The interaction results in the products' evolution driven by consumers' tastes, manufacturers' productivity and their feedback. In order to make the consumers' behavior realistic, the model includes advertisement, trends, history and technical evaluation of the products. The manufacturers' behavior is described from two aspects: design and marketing. See the references [23]–[27] for the results as well as the details.

4 Summary

This paper summarizes the state-of-the-art of the Project "Methodology of Emergent Synthesis". Starting with the definition of the problem of synthesis and the classification of the difficulties with synthesis problems, the paper has clarified the concept of emergent synthesis. Then, the activities of the researches in the project have been briefly shown.

In the project, so far, the emergent approaches to the problems of synthesis related to design, manufacturing, utilization and consumption of artifactual systems have been composed, and their effectiveness and potential have been confirmed. Furthermore, based on the concatenation of these approaches, establishing the general and systematic methodology for emergent synthesis and also applying the methodology to real artifactual systems are left for further studies.

Acknowledgements

This research has been pursued partially under the financial support through the project "Methodology for Emergent Synthesis" (96P00702) in Research for the Future Program of the Japan Society for the Promotion of Science.

References

[1] K. Ueda, Analysis and Synthesis, *Proc. of Int. Workshop on Emergent Synthesis*, 7–12 (1999)

[2] K. Ueda, Introduction of the Project "Methodology for Emergent Synthesis", *Proc. of Workshop on the Methodology of Emergent Synthesis*, 1–4 (1998)

[3] K. Ueda and H. Tamaki, Aim of the Project "Methodology for Emergent Synthesis", *JSME Design & Systems Conf.*, 121–124 (1997)

[4] H. Yoshikawa, *Techno-Globe*, Kogyochosakai (1993)

[5] K. Fujita and S. Akagi, Agent-Based Distributed Design System Architecture for Basic Ship Design, *Concurrent Engineering – Research and Applications,* **7** (2), 83–94 (1999)

[6] T. Inamoto, H. Murao, V. Kryssanov, Y. Kurematsu and S. Kitamura, Designing a Pattern Generator for a Walking Robot by Modeling the Human Creative Process, *Prof. of 5th Int. Symp. on Artificial Life and Robotics,* 809-812 (2000)

[7] I. Nagasaka, K. Ueda, A. Markus and T. Taura, Adaptive-Growth-Type Assembly Structure Representation for Assembly Layout Design, *Proc. of 32nd CIRP Int. Seminar on Manufacturing Systems,* 323–332 (1999)

[8] V. Kryssanov, H. Tamaki and K. Ueda, The Mechanism of Creative Reasoning and Emergent Constructions in Conceptual Design, *Proc. of 1999 Int. CIRP Design Seminar,* 239–248 (1999)

[9] V. Kryssanov, H. Tamaki and K. Ueda, Agents for Assessing Requirements in Dynamic Environments to Support Evolutionary Design, *Pro.c of 6th Int. Seminar on Life Cycle Engineering,* 151–160 (1999)

[10] S. Kitamura, Y. Kakuda and H. Tamaki, An Approach to the Emergent Design Theory and Applications, *Artificial Life and Robotics,* **3**, 86–89 (1999)

[11] K. Ueda, *Biological Manufacturing Systems,* Kogyochosakai (1994)

[12] K. Ueda, I. Hatono, N. Fujii, and J. Vaario, Reinforcement Learning Approaches to Biological Manufacturing Systems, *Annals of the CIRP,* **49** (1), in press (2000)

[13] J. Vaario, K. Ueda, An Emergent Modeling Method for Dynamic Scheduling, *J. of Intelligent Manufacturing,* 129–140 (1998)

[14] J. Vaario and K. Ueda, Lineless Production for Highly Flexible and Adaptive manufacturing, *Proc. of CIRP Int. Seminor on Intelligent Computation in manufacturing Engineering,* 391–398 (1998)

[15] J. Kozasa, H. Tamaki, S. Abe and S. Kitamura, An Autonomous Decentralized Model Based Approach to Production Planning, *Proc. of 1999 IEEE Int. Conf. on Systems, Man and Cybernetics,* IV, 424–429 (1999)

[16] K. Ueda, J. Vaario and N. Fujii, Interactive Manufacturing: Human Aspects for Biological Manufacturing Systems, *Annals of the CIRP,* **47** (1), 389–392 (1998)

[17] N. Fujii, I. Hatono and K. Ueda, Self-Organization Process in Interactive Manufacturing Environment, *Proc. of 4th Int. Symp. on Artificial Life and Robotics,* 1, 166–169 (1999)

[18] H. Murao and S. Kitamura, Q-Learning with Adaptive State Space Construction, *Learning Robots* (Lecture Notes in Artificial Intelligence 1545), 13–28, Springer-Verlag (1999)

[19] K. Kawakami, K. Ohkura and K. Ueda, Adaptive Role Development in a Homogeneous Connected Robot Group, *Proc. of 1999 IEEE Int. Conf. on Systems, Man and Cybernetics,* 3, 251–256 (1999)

[20] K. Ohkura, Y. Matsumura and K. Ueda, Robust Evolution Strategies, *Lecture Notes in Computer Science 1585,* 10–17, Springer-Verlag (1999)

[21] M. Svinin, K. Yamada, and K. Ueda, Reinforcement Learning Approach to Acquisition of Stable Gaits for Locomotion Robots, *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics,* 12–15 (1999)

[22] M. Svinin, K. Ueda and M. Kaneko, Analytical Conditions for the Rotational Stability of an Object in Multi-Finger Grasping, *Proc. of IEEE Int. Conf. on Robotics and Automation,* 1, 257–262 (1999)

[23] K. Ueda, J. Vaario, T. Takeshita and I. Hatono, An Emergent Synthetic Approach to Supply Network, *Annals of the CIRP,* **48** (1), 377–380 (1999)

[24] S. H. Oda, K. Iyori, K. Miura and K. Ueda, The Application of Cellular Automata to the Consumer's Theory: Simulating a Duopolistic Market, *Simulated Evolution and Learning* (Lecture Notes in Artificial Intelligence 1585), 454–461, Springer Verlag (1999)

[25] S. H. Oda and K. Miura, The Application of Cellular Automata and Agent Model to Markets with Network Externalities, *Commerce, Complexity and Evolution,* Cambridge University Press, 351–371 (2000)

[26] A. Iwasaki S. H. Oda and K. Ueda, The Emergence and Collapse of the State: A Game Theoretic Analysis with Computer Simulations, *Proc. of Complex Systems 1998,* 282–290 (1998)

[27] A. Iwasaki, S. H. Oda and K. Ueda, Simulating an N-person Multi-stage Game for Making a State, *Simulated Evolution and Learning* (Lecture Notes in Artificial Intelligence 1585), 309–316, Springer Verlag (1999)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A coevolutionary approach to adaptive encoding for genetic algorithms

Hajime Murao†, Akio Yamamoto‡, Hisashi Tamaki†, and Shinzo Kitamura†

†) Dept. of Computer and Systems Eng.,
Faculty of Eng.,
Kobe University,
Kobe 657-8501, JAPAN

‡) Dept. of Computer and Systems Eng.,
Graduate School of Science and Technology,
Kobe University
Kobe 657-8501, JAPAN

Abstract

This paper introduces a coevolutionary genetic algorithm to explore not only the search space defined by a genotype-phenotype mapping over the problem space but also the mapping itself. The proposed algorithm consists of two populations of genes and loci separately. A chromosome of a solution candidate is generated by rearranging a sequence of genes given by an individual in one population using loci infomation given by one in another population. Genetic operations are not directly applied to chromosomes but indirectly through geneteic operations on genes and loci. We firstly discuss about the expectable performance of the proposed method from this point of view in connection with the schema theorem. Then, it is applied to the deceptive problem introduced by Goldberg for which the difficulty against genetic algorithms can be controled by the mapping from a binary string as a chromosome to a set of 3-bit strings as a solution candidate. As a result, it shows fairly good performance beside the conventional genetic algorithm.

1 Introduction

Genetic algorithms are methods to explore the problem space by applying genetic operations onto chromosomes for which solution candidates as phenotypes are encoded. We here called a sub-space of the problem space can be explored by a genotype-phenotype mapping and genetic operations as a search space. How large search space can be explored depends on both of the genotype-phenotype mapping and genetic operations, and therefore they should be carefully determined according to the probelm. However, since there are usually no *a-priori* knowledge of the problem when designing a genetic algorithm beforehand, they can only be determined empirically.

There are reports about so-called fitness landscape in which the influence of genetic operations over the search space are discussed[1, 2]. Efficient parameters of genetic operations are investigated but can be applied for limited problems only. Some adaptic ways to adjust parameters according to the problem have been also proposed[3, 4, 5].

Concerning about the genotype-phenotype mapping, a well-known characteristic of the string type chromosome is the schema theorem, which gives an aspect for the evolutionary process where long, high-order schemata will have a less survival probability. Therefore, it is thought better to place genes related each other at closer loci. To realize it without *a-priori* knowledge of the problem, methods modifying a string of 2-tuples of gene and locus are proposed[6, 7].

There is another interesting work proposing a method to find not a genotype-phenotype mapping directly but better schemata as building-blocks for relaxing the difficulty of finding better genotype-phenotype mapping[8], where a coevolutionary approach is utilized. It is closely related interesting approach to our method.

In this paper, we also utilize a coevolutionary approach to explore not only the search space defined by a genotype-phenotype mapping over the problem space but also the mapping itself. The proposed algorithm consists of two populations of genes and loci separately. A chromosome of a solution candidate is generated by rearranging a sequence of genes given by an individual in one population using loci information given by one in another population. Genetic operations are not directly applied to chromosomes but indirectly make an influence over chromosomes through genes and loci. In the following sections, we firstly discuss about the expectable performance of the proposed method from this point of view in connection with the schema theorem. Then, it is applied to the deceptive problem introduced by Goldberg for which the difficulty against genetic algorithms can be controled by the mapping from a binary string as a chromosome to a set of 3-bit strings as a phenotype. Finally, we conclude.

2 Coevolutionary genetic algorithm for exploring genotype-phenotype mappings

2.1 Seed, rule, chromosome and solution candidate

The proposed method consists of two populations, where one consists of sequences of values corresponding to genes, *i.e.* each value is placed at an appropriate position in a chromosome. Another population consists of sequences of values each of which indicates a position of corresponding gene and we call each sequence as a loci placement. A chromosome of a solution candidate is generated by rearranging a seqeunce of genes by using corresponding loci information. We call a sequence of genes as a **seed** of a chromosome and a loci description as a **rule** of generating a chromosome from a seed.

A seed $\boldsymbol{s}$ can be defined according to *a-priori* knowledge of the problem. For instance, you can define $\boldsymbol{s}$ as $\boldsymbol{s} \in R \times R \cdots R$ if you know the real-coded vector type chromosome is enough to describe and explore any solution candidates. We here assume a binary string chromosome and consequently it is enough to think about a binary string seed $\boldsymbol{s} \in \{0,1\} \times \{0,1\} \cdots \{0,1\} = \{0,1\}^l$ where l is the number of genes in a chromosome. A rule $\boldsymbol{r}$ is defined as an permutation of numbers indicating positions of corresponding bits in a chromosome, *i.e.* a rule $\boldsymbol{r}$ is defined as $\boldsymbol{r} \in L \times L \cdots L = L^l$ where $L = \{1, 2, ..., l\}$.

Let S and R as a set of seeds and rules respectively. We indicate a chromosome generated by applying the j-th rule $\boldsymbol{r}_j$ of R onto the i-th seed $\boldsymbol{s}_i$ of S as $\boldsymbol{x}_{ij} = \boldsymbol{s}_i \circ \boldsymbol{r}_j$. We can define any combination manner of $\boldsymbol{s}_i$ and $\boldsymbol{r}_j$ as $\circ$, and apply *a-priori* knowledge of the problem here. Two simple cases for the seed $\boldsymbol{s}_i = (s_i(1), s_i(2), ..., s_i(l))$ and the rule $\boldsymbol{r}_j = (r_j(1), r_j(2), ..., r_j(l))$ are

$$x_{ij}(k) = s_i(r_j(k)), \tag{1}$$

and

$$x_{ij}(r_j(k)) = s_i(k). \tag{2}$$

In both cases, the definition length of sub-strings of the seed $\boldsymbol{s}_i$ and the rule $\boldsymbol{r}_j$ on what schemata of the chromosome $\boldsymbol{x}_{ij}$ is mapped are related to k. That is, schemata of a chromosome are mapped onto same long sub-strings of a corresponding rule using the combination manner in Eq.1, but the length of sub-strings of a seed are independent to the length of schemata of the chromosome. In this case, the length of sub-strings of a seed can become short as enough. On the other hand, Eq.2 shows the length of sub-strings of a rule is equal to that of a corresponding seed but they are indipendent to the length of schemata of the chromosome. From this point of view, Eq.2 is thought to be superior to Eq.1 but it should be investigated more carefully.

2.2 Evaluation of seed and rule

A fitness value of a chromosome $\boldsymbol{x}_{ij}$ can be calculated by a combinatorial function $f(\cdot)$ of mapping from a chromosome to a solution candidate and mapping from a solution candidate to a fitness value. Some kind of fitness values for rules and seeds can be thought but we here introduce the maximum fitness value of chromosomes generated from a seed as a fitness value of the seed and also the maximum fitness value of chromosomes generated by a rule as a fitness value of the rule as,

$$f_s(s_i) = \max_{\boldsymbol{r}_j \in R} f(\boldsymbol{x}_{ij}) = \max_{\boldsymbol{r}_j \in R} f(\boldsymbol{s}_i \circ \boldsymbol{r}_j) \tag{3}$$

$$f_r(\boldsymbol{r}_j) = \max_{\boldsymbol{s}_i \in S} f(\boldsymbol{x}_{ij}) = \max_{\boldsymbol{s}_i \in S} f(\boldsymbol{s}_i \circ \boldsymbol{r}_j) \tag{4}$$

2.3 Algorithm

The procedure of the proposed algorithm is as follows:

1. **Initialization**: Initialize both sets S and R of seeds and rules by randomly generated strings.

2. **Evaluation of chromosomes**: Generate possible chromosomes by using seeds and rules in the current sets of S and R, and evaluate them.

3. **Evaluation of seeds and rules**: Calculate fitness values of seeds and rules as in Eq.3 and Eq.4.

4. **Genetic operations**: Apply genetic operations onto seeds and rules to generate new sets S' and R' of seeds and rules.

5. We call a procedure from 2 to 4 as a generation. Repeat generations until conditions to terminate are satisfied.

6. Output a solutoin candidate corresponding to a chromosome with the maximum fitness value over generations as a result.

3 Computer simulations

3.1 N×3-bit deceptive problem

The deceptive problem is an optimization problem in which a large and smooth basin for not the global optimum but the local optimum is defined so as to mislead the search to the sub-optimal solution. It is done by violating the static building-block hypothesis, *i.e.* when chromosomes are 2-bit strings, the four potential building-blocks of $\{0*, *0, 1*, *1\}$ are schemata, and the static building-block hypothesis is violated if fitness values are defined as $f(0*) > f(1*)$ or $f(*0) > f(*1)$ if the global optimum is at point 11.

The N×3-bit deceptive problem is formulated as a problem finding a combination of N 3-bit strings $\{\boldsymbol{y}_1, ..., \boldsymbol{y}_N\}$ which maximizes the fitness value given by $\sum_{i=1}^{N} e(\boldsymbol{y}_i)$, where $e(\boldsymbol{y}_i)$ is a fitness value of 3-bit strings $\boldsymbol{y}_i = (y_i(1), y_i(2), y_i(3))$ $(y_i(j) \in \{0, 1\}\ \forall j)$ defined so as to violate the static building-block hypothesis.

3.2 Application of the proposed method

The proposed co-evolutionary method is applied to the 10×3-bit deceptive problems P1 and P2, where the optimal 3-bit string is $\boldsymbol{y}_i = (1, 1, 1)$ $(\forall i = 1, ..., N)$ and is $\boldsymbol{y}_i = (0, 1, 0)$ $(\forall i = 1, ..., N)$ respectively. Fitness values for every 3-bit strings are given by Table 1.

Let the size of the set S be 100, for which the length of a seed $\boldsymbol{s}_i \in S$ is $l = 3 \times N = 30$ and the k-th element of a seed is as $s_i(k) \in \{0, 1\}$. Let the size of the set R be also 100, for which the k-th element of a rule $r_j \in R$ is as $r_j(k) \in \{1, 2, ..., l\}$ where $r_j(a) \neq r_j(b)$ for all $a \neq b$ $(a, b \in \{1, 2, ..., l\})$.

A chromosome is generated according to Eq. (1), and is mapped to a solution candidate according to so-called "loose coding" manner, *i.e.* a solution candidate $\boldsymbol{y}_k$ is defined by a chromosome $\boldsymbol{x}_{ij} = (x_{ij}(1), x_{ij}(2), ..., x_{ij}(l))$ as follows,

$$y_k(m) = x_{ij}(k + (m-1)N) \quad m = 1, 2, 3 \qquad (5)$$

A fitness value of a solution candidate is defined so as to normalized in a range [0, 1] as,

$$\frac{1}{30N} \sum_{i=1}^{N} e(\boldsymbol{y}_i) \qquad (6)$$

We also apply so-called elitest strategy where a seed and a rule corresponding to the best solution candidate in a generation are forced to remain in the sets S and R of the next generation.

Table 1: Fitness values of 3-bit strings

bit-string	e( )		bit-string	e( )	
	P1	P2		P1	P2
(0,0,0)	28	0	(1,0,0)	14	26
(0,0,1)	26	14	(1,0,1)	0	28
(0,1,0)	22	30	(1,1,0)	0	0
(0,1,1)	0	0	(1,1,1)	30	22

3.3 Results

Figure 1 shows time courses of the average of the maximum fitness values over 30 trials. A simple genetic algorithm with 10000 individuals is also applied to the same problems P1 and P2. It is clearly shown that the proposed method outperforms the simple genetic algorithm in both problems.

The proposed method shows quite good performance of converging to the optimal solution after only 200 generations for the problem P2. This means that the coevolutionary search proposed here is working well. At first sight, the result might be strange since the problem P1 appears easier than P2, because once a seed $\boldsymbol{s}$ with $s(i) = 1\ \forall i$ is obtained, any rule can be applied to generate the optimal solution and there becomes no need to search for a rule in P1. However, if coevolutionary search works well and rules can contribute enough, the number of seeds with ten "1"s out of 30 bits, which can be translated to the optimal solution in P2, is ${}_{30}C_{10}$. It is quite larger than ${}_{30}C_{30} = 1$, the number of seeds can be translated to the optimal solution in P1.

We then observe of the proportion of "1"s in a seed given by Eq.7.

$$\text{Proportion of "1"} = \frac{1}{l} \sum_{k=1}^{l} s_i(k) \qquad (7)$$

Figure 2 shows time variance of the distribution of the proportions for seeds during a trial. The solid line shows that of the elitest one. Figure 3 shows time variance of the distribution of the proportions for chromosomes. Since the optimal solution of the problem P2 is $\boldsymbol{y} = (0, 1, 0)$, the proportions should be distributed around $1/3 = 0.333$ after sufficient number of generations. Actually, the proposed method leads the proportions to around 0.333 and the elitest one converges to 0.333 while the simple genetic algorithm mislead to the sub-optimal solution of $\boldsymbol{y} = (1, 0, 1)$ with the proportions around $2/3 = 0.666$.

Because of the limited space of the paper, we cannot

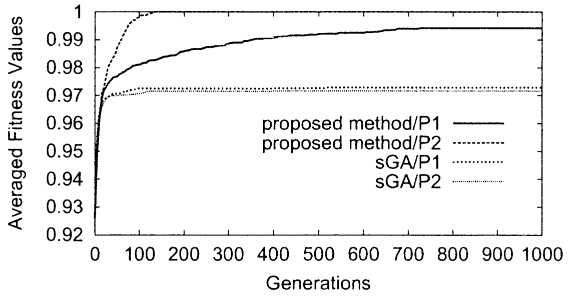


Figure 1: Time courses of the averaged maximum fitness values.

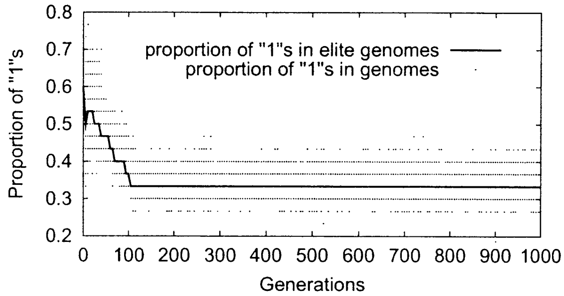


Figure 2: Proportion of "1"s to the bit-length of the chromosome for the proposed method.

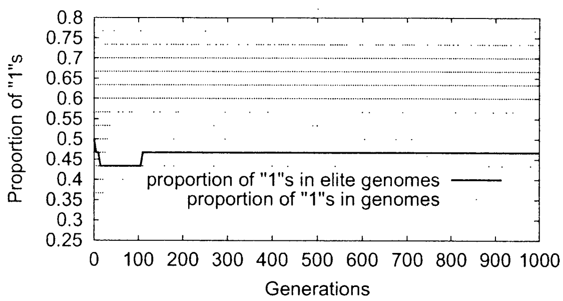


Figure 3: Proportion of "1"s to the bit-length of the chromosome for the simple genetic algorithm.

include results of the observation on rules, but they show that the sub-strings of rules are fragile for genetic operations. It is because we apply so-called "loose coding" as a mapping from a chromosome to a solution candidate in which building-blocks become long, and under the combination manner of Eq.1 we used here, the length of sub-strings of rules are closely related to the definition length of corresponding schemata as I noticed in the section 2.1.

4 Conclusion

In this paper, we proposed a coevolutionary algorithm to evolve not only chromosomes but also structure of chromosomes. It relaxes the difficulty of determining genotype-phenotype mapping before knowing aspects of the problem. The proposed method was applied to the 3-bit deceptive problem and outperformed a simple genetic algorithm. Future work will concentrate more investigation of the behavior of the method in case of using different combination manner described in Eq.2, and application to the practical problems.

Acknowledgements

This research was supported by the 'Methodology of Emergent Synthesis' research project (No. 96P00702) conducted under the program 'Research for the Future' of Japan Society for the Promotion of Science.

References

[1] Kauffman, S., and Levin, S. (1987) Towards a General Theory of Adaptive Walks on Rugged Landscapes, *J. theor. Biol*, Vol.128, pp.11–45

[2] Mathias, K., and Whitley, D. (1992) Genetic operators, the fitness landscape and the traveling salesman problem, *PPSN*, Vol.2, pp.219–228

[3] Bäck, T. (1993) Optimal Mutation Rates in Genetic Search, *Proc. of the Fifth ICGA*, pp.2–8

[4] Maley, C. (1995) The Coevolution of Mutation Rates, *Advances in Artificial Life*, F. Moran, A. Moreno, J.J.Merelo, and P. Chacon eds., pp.234–245

[5] Spears, W.M. (1995) Adaptive Crossover in Evolutionary Algorithms, *Proc. of the 4th Annual Conf. on Evolutionary Computing*, pp.367–384

[6] Holland, J.H. (1992) Adaptation In Natural And Artificial Systems, MIT Press

[7] Goldberg, D.E., Deb, K., and Korb,B. (1991) Don't Worry, Be Messy, *Proc. of ICGA '91*, pp.24–30

[8] Handa, H., Baba, N., and Katai, O. (1997) Genetic Algorithm involving Coevolution Mechanism to Search for Effective Genetic Information, *Proc. of the 1997 IEEE Int. Conf. on Evolutionary Computation*, pp.709–714

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Adaptive Segmentation of the State Space based on Bayesian Discrimination in Reinforcement Learning *

K. Yamada*, K. Ohkura*, M.Svinin** and K. Ueda*

*Mechanical Engineering Department Kobe University Rokko-dai, Nada-ku, Kobe 657-8501, Japan

**Bio-Mimetic Control Research Center, RIKEN, Anagahora, Shimoshidami, Nagoya, 463-0003, Japan

Abstract

This paper proposes reinforcement learning in the continuous state and action spaces. A learning system with new mechanisms of action selection and reproduction is proposed. The system makes use of a Bayesian discrimination method for dynamic segmentation of the continuous state space. As learning progresses, the continuous state space is segmented by Bayes boundaries. The proposed system is tested under simulation for an autonomous mobile robot. It is shown that the learning performance can be improved if clustering of the production rules is employed.

Keyword: reinforcement learning, Bayesian discrimination.

1 Introduction

One of the main problems in reinforcement learning is the construction of a mapping from the state space to the action space for an autonomous agent. The problem becomes complicated when the action space and the sensor space are continuous, as the discretization density directly affects the learning performance. A coarse discrete state space entails "incomplete perception", "hidden state" or "perceptual aliasing". On the other hand, a detailed discrete state space representation may extend the learning time. To overcome this problem, it is necessary to segment the continuous state space adaptively.

It is evident that a learning process, if not properly organized, may take a long time to converge since the agent has to explore a wide state space. To improve the learning performance, basic mechanisms of the learning system should be carefully designed.

There are two approaches to overcome this problem. One way is to apply standard function-approximation techniques such as artificial neural networks to the Q-function. Sutton[1] used CMAC[2] as a method of the function approximation. Lin[3] represented the Q-function by the multi-layer neural networks called *Q-nets*. Morimoto et al. [4] use Gaussian Softmax Basis Functions (GSBF) as the method of the function approximation. This method was applied to the stand-up task for a two-joint, three link robot, and they obtained good results. However, those techniques still have difficulties in designing the structure of the neural networks and the number of the neurons.

The other way is to segment the continuous state space according to an agent's experiences. Asada et al. [5] proposed a method to segment the state space by the hyperellipsoid-shaped regions. However, the segmented state spaces were not re-organized even if the segments were found to be inappropriate for given tasks. Takahashi et al. [6] proposed a method which segments the continuous state space by the Nearest Neighbor method while executing a task. In this method, the segmented state spaces were not fixed, but were reconstructed its shape. Ueno et al. [7] focused on the accuracy and efficiency of the state space division, and used Mahalanobis' distance and Nearest Neighbor method to segment the state space. In these approaches, an agent first obtains sets of the input data, and classify them according to the similarities of the rewards or of the actions. Since in these method, a learning agents is required to classify all the input data at every time step, extra costs caused by unnecessary modification of segments that have nothing to do with the current input.

In this paper, we develop a Bayesian discrimination method to segment the continuous state space into clusters. The clusters are represented as production rules (*IF-THEN* rules). In addition to the Bayes decision theory, the method blends the ideas of reinforcement learning and learning vector quantization in order to adaptively divide and segment the continuous state space into clusters.

2 Basic concept

In this section, we outline a learning procedure, operating on a set of production rules, which is based on the Bayes decision theory. When a new sensory

*This study has been supported, in part, by the Japan Society for the Promotion of Science within the Project RFTF96P00702, "Methodology of Emergent Synthesis."

input is memorised. the control system should generate a new rule that maps the current sensory input to an action to be executed. However, if a similar rule that maps a sensory input close to the current input already exists, it is not necessary to generate a new rule. To be able to classify the similar rules, we utilize Bayesian discrimination method for comparing the sensory input with the states of the memorized rules.

Bayesian discrimination method is one of the well-known methods of pattern classification[8]. In this method, the conditional probability distribution $p(\boldsymbol{x}|C_i)$ of the i-th cluster, $i = 1, 2, \ldots$, and the prior probability $P(C_i)$ are assumed to be known in advance. Given the input data $\boldsymbol{x}$, each cluster estimates the posterior probability $P(C_i|\boldsymbol{x})$ using Bayes' formula, $P(C_i|\boldsymbol{x}) = P(C_i) \cdot p(\boldsymbol{x}|C_i)/p(\boldsymbol{x})$, and the input data $\boldsymbol{x}$ is assigned to the cluster with the maximal posterior probability. Assuming that the probabilistic density function of each cluster is given by Gaussian distribution, the covariance matrix and the mean of each cluster can be estimated step by step.

Bayesian discrimination method estimates the loss of misclassification of the sensory input. In our system, the action rules are associated with clusters, and the clusters are segmented by the Bayes boundaries (Fig.1) Basically, the agent selects one of the rules with the smallest estimated losses and executes a corresponding action. If the action gets no punishment, the selected rules are placed near the sensory input and its utility increases. On the other hand, if the selected rule gets a punishment its utility decreases. In our system, the agent can not only generate and cluster new rules, but also can remove the rules whose fitness is under a certain threshold.

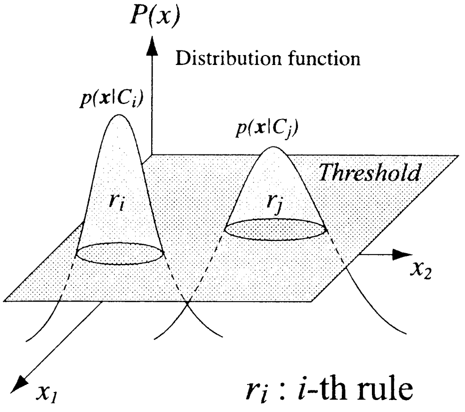


Figure 1: State space which is segmented by Bayes boundaries.

2.1 Structure of the Action Rules

Let n_s be the number of sensors an agent is equipped with, and $\boldsymbol{x} = \{x_1, \ldots, x_{n_s}\}^T$ be the sensory input to the agent. The learning system operates on a set of action rules, R. An individual rule can be associated with a cluster in the state space where the rule's action is appropriate. Formally, a rule $r \in R$ is defined as follows:

$$r := < \boldsymbol{v}, u, \boldsymbol{a}, f, \boldsymbol{\Sigma}, \Phi > \tag{1}$$

where $\boldsymbol{v} = \{v_1, \ldots, v_{n_s}\}^T$ is the state vector associated with the rule r; u is the utility of the rule; $\boldsymbol{a} = \{a_1, \ldots, a_{n_a}\}^T$ is the action corresponding to the rule, n_a is the number of actuators; f is the occurrence (prior probability) of the rule; $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1, \ldots, \sigma_{n_s}\}$ is the covariance matrix of the rule, $\Phi = \{\phi_1, \ldots \phi_{n_l}\}$ is the sample set associated with the rule, n_l is the number of samples.

Each sample $\phi_i = \{x_1, \ldots x_{n_s}\}$ of the set Φ memorizes the sensory input recorded when the rule r was selected for executing its action A. In other words, whenever the rule r is a winner, the current sensory input $\boldsymbol{x}$ is added to the sample set Φ associated with this rule. In this representation $\boldsymbol{v}$ is the mean value of the sample set Φ, and $\boldsymbol{\Sigma}$ is its covariance.

If $\boldsymbol{v}$ matches to a certain degree the current sensory input $\boldsymbol{x}$, the rule r becomes active and can trigger its action $\boldsymbol{a}$. In the beginning rule set R is empty u_0. As learning progresses, the total number of rules in R, n_r, varies by processes of reproduction and extinction.

2.2 Action selection

Action selection is performed using the Bayesian discrimination technique which originated in the field of pattern recognition. The selection procedure is organized as follows. Given the conditional probability distribution of the i-th rule's cluster,

$$p(\boldsymbol{x}|C_i) = \frac{1}{(2\pi)^{\frac{n_s}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \cdot \exp\left\{\frac{-1}{2}(\boldsymbol{x} - \boldsymbol{v}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{v}_i)\right\} \tag{2}$$

and the estimated value of f_i, the risk of misclassification of the sensory input $\boldsymbol{x}$ into other clusters can be defined as

$$g_i = -\log\{f_i \cdot p(\boldsymbol{x}|C_i)\} \tag{3}$$

The rule with the minimal value of g_i is selected as a winner and is denoted as r_w. If the value g_w of the winner risk is larger than a threshold g_{th}, then the winner is rejected, and the agent performers an action selected randomly. Otherwise, the agent performers the action specified by the winner rule.

2.3 Temporal credit assignment

The utilities of the rules are updated every time after the winner executes its action. The mechanism of utility adjustment consists of four parts.

1. *Direct payoff distribution* : The direct payoff P is given to the winner rules only in specific states. There are two types of payoffs: reward ($P > 0$) and punishment ($P < 0$). The payoff is spreading back along the sequence of the winner rules with the discount rate $0 < \gamma < 1$: $u_w^{(k)} \leftarrow +u_w^{(k)} + \gamma^k P, k = 0 \ldots N$, where N is the length of the winners chain. Here, we denote the parent of r_w as $r_w^{(1)}$, the parent of $r_w^{(1)}$ as $r_w^{(2)}$ and so on.

2. *"Bucket brigade" strategy* : The current winner r_w hands over a part of its utility, Δu, back to the previous winner, $r_w^{(1)}$: $u_w^{(1)} \leftarrow u_w^{(1)} + \Delta u$, where $\Delta u = \kappa(u_w - u_w^{(1)})$, $\kappa \in [0,1]$, if $u_w > u_w^{(1)}$ and $\Delta u = 0$, otherwise.

3. *Taxation* : Whenever a winner rule r_w triggers its action, its utility is updated as $u_w \leftarrow (1 - cf)u_w$.

4. *Evaporation* : All the rules reduce their utilities at the evaporation rate $\eta < 1$ when the agent reaches the goal state: $u_w \leftarrow \eta u_w$. The rules decreasing their utility below the threshold u_{min} are removed.

2.4 Modification and Reproduction

The modification and reproduction phase is performed if the action triggered by the winner rule r_w did not result in a situation where the agent is punished.

If the winner rule is rejected (that is if $g_w > g_{th}$, see Section 2.2) a new rule, memorizing the current sensory input and the executed action, is produced. The parameters of the new rule are set as

$$\boldsymbol{v}_c = \boldsymbol{x}, \quad \boldsymbol{\Sigma}_c = \sigma_0^2 \boldsymbol{I}, \quad \boldsymbol{a}_c = \boldsymbol{a}_w, \qquad (4)$$

$$u_c = u_0, \quad f_c = f_0 \qquad (5)$$

where σ_0, u_0 and f_0 are constants, I is the unit matrix.

On the other hand, if the winner rule r_w is not rejected, its parameters are modified as follows. First, the sample set Φ_w is updated, i.e. the current sensory input $\boldsymbol{x}$ is added to Φ_w. Then, the sample mean $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_{n_s}\}^T$ and the sample variance $s^2 = \{s_1^2, \ldots, s_{n_s}^2\}^T$ are estimated from the updated set Φ_w, and the confidence intervals for $\bar{X}$ and s^2 are defined[1].

If all the components of the state vector $\boldsymbol{v}$ and the covariance matrix $\boldsymbol{\Sigma}$ of the winner rule r_w are within the confidence intervals, the parameters of the rule are left unchanged. Otherwise, even if one of the components is out of the corresponding confidence interval, the parameters of the rule r_w are modified as follows:

$$
\begin{aligned}
v_i &\leftarrow v_i + \alpha(\bar{x}_i - v_i) && (6)\\
\sigma_i^2 &\leftarrow \sigma_i^2 + \alpha^2[s_i^2 - \sigma_i^2] && (7)\\
f_w &\leftarrow f_w + \beta(1 - f_w) && (8)
\end{aligned}
$$

where α, β are a constant. For all the other rules, the prior probabilities f_i are modified as follows: $f_i \leftarrow (1 - \beta)f_i$.

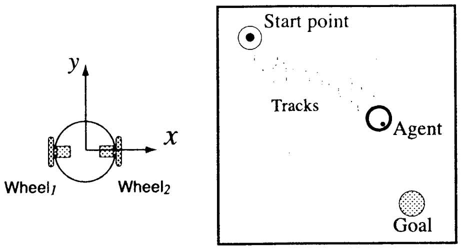


Figure 2: Simulation setting

3 Simulation

Feasibility of the proposed method is tested under the simulation. This simulation shows how the proposed learning system segments the state space in a two-dimensional environment.

3.1 Simulation setting

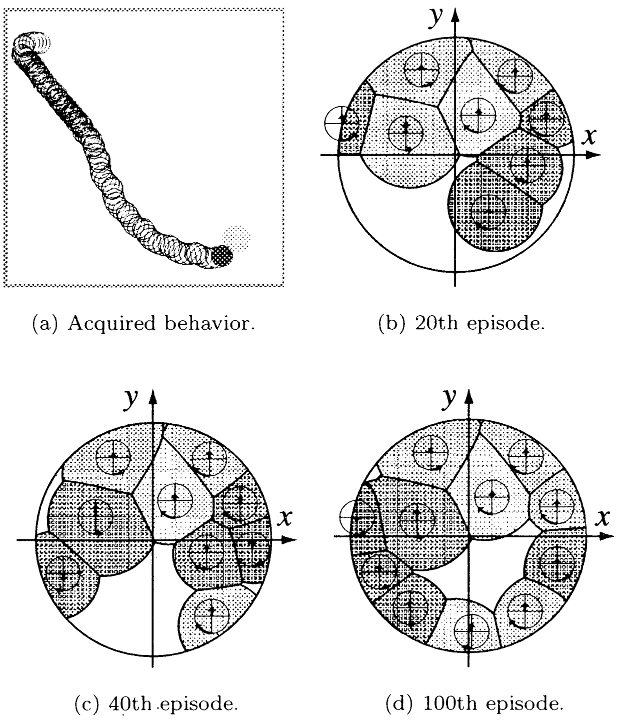
The simulation setup is shown in Fig.2. The sensory input is the real valued coordinates (x, y) of the goal in the coordinate system fixed to the agent. The agent should avoid collision with the wall. The time interval between the starting motion and reaching the goal is called episode. The agent is placed at the starting point, with its heading direction oriented randomly at every episode. The episodes are updated when the agent reaches the goal, or when the number of produced actions reaches 1000. The agent is rewarded only upon reaching the goal and is punished for every collision.

The parameters are set as follows: $P = 20$(reward), $P = -0.05u$(punishment), c_f=0.01, γ=0.8, η=0.98, κ=0.15, u_{min}=9.5, u_0=10, σ_0=0.05, f_0=0.001, α=0.1, β=0.0001

3.2 Simulation result

Fig.3 shows the agent's behavior in episode No. 100 and the segmented state space (Bayes boundaries) in

[1] The mathematical details are omitted due to the space limitation.



(a) Acquired behavior.

(b) 20th episode.

(c) 40th episode.

(d) 100th episode.

Figure 3: Simulation result and segmented state space

episode No. 20, No. 40 and No. 100. The number of collisions and the number of actions (step) are plotted in Fig.4. In the first episode, the state space of the agent is undivided. The agent behavior is random and the number of collisions is large. It takes 20 episodes for the agent to acquire the goal seeking and collision avoidance patterns.

Fig.3(b)-3(d) shows the segmented state space at each episode. The circle indicates the area covered with the rules, and undivided state space is covered by the white area. The arcs and the arrows indicate the rule's action[2]. As learning progresses, the state space is segmented and the shape of each cluster is modified.

4 Conclusions

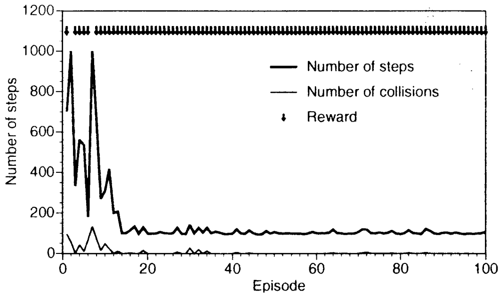


Figure 4: Learning history

A new reinforcement learning technique for adaptive segmentation of the continuous state space was formulated. The technique makes use of Bayesian discrimination method, where the segmented state space is represented by Bayes boundaries. The proposed technique was applied to a navigation task and tested under simulation. The simulation results show that the agent can segment its continuous state space adaptively and, by doing so, reach the goal. Though we omitted a simulation result in the sixteen-dimensional state space, an agent acquired the collision avoidance and goal seeking behavior. Currently, the exploration of the action space is performed randomly in our system. In future, we plan to employ more complex techniques (similar to REINFORCE algorithm [9] or RBF network [4]) to abstract and decompose the action space.

References

[1] R.S. Sutton, Generalization in reinforcement learning: Successful examples using sparse coarse coding, Advances in Neural Information Processing Systems 8, pp.1038-1044, MIT Press, 1996.

[2] J.S. Albus, Brains, Behavior and Robotics, BYTE Books, McGraw-Hill,1981.

[3] Long-Ji Lin. Scaling Up Reinforcement Learning for Robot Control, Proc. of the 10th Inter. Conf. on Machine Learning, pp.182-189, 1993.

[4] J. Morimoto and K. Doya, Hierarchical reinforcement learning for motion learning: learning "stand-up" trajectories", Advanced Robotics, Vol.13, No.3, pp.267-268, 1999.

[5] M. Asada, S. Noda, and K. Hosoda. Action-Based Sensor Space Categorization for Robot Learning, Proc. of IROS '96, pp.1502-1509, 1996.

[6] Y. Takahashi, M. Asada and K. Hosoda, Reasonable Performance in Less Learning Time by Real Roobot Based on Incremental State Space Segmentation, Proc. of IROS'96, pp.1518-1524, 1996.

[7] Atsushi Ueno, Hideaki Takeda, and Toyoaki Nishida. Learning of the way of abstraction in real robots, Proc. of SMC'99 (CD-ROM), Vol.1, pp.II746-II751, 1999.

[8] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley-Interscience, N.Y. 1972.

[9] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning, Vol.8, pp.229-256, 1992.

[2] The agent turns in the direction specified by the arc, and then moves in the direction (resulting after the turn) specified by the arrow.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Study on the Multicriteria Optimization Support by Using Evolutionary Algorighms

Inamoto, T.[†], Kryssanov, V.V.[‡], Tamaki, H.[‡], and Kitamura, S.[‡]

[†] Graduate School of Science and Technology, Kobe University
[‡] Dept. Computer and Systems Engineering, Faculty of Engineering, Kobe University

Rokko-dai, Nada-ku, Kobe 657-8501, Japan
E-mail: inamoto@al.cs.kobe-u.ac.jp

Abstract

In this paper, we propose a method to support human decision making in multicriteria optimization. Acquiring a pareto frontier is a well-established research topic in evolutionary computation. Studies on this topic often set the main objective as to obtain a surface covering pareto optimal solutions for a task. The decision maker has then to select some preferred candidates on the surface. Obviously, however, that it would be just a waste of resources and time to consider undesirable/unfeasible solution candidates on the surface. The method developed through in this study is intended to allow for avoiding such wasting. The method is realized within the framework of evolutionary algorithms, and result of a computational experiments are given.

1 Introduction

Many problems in engineering can be considered as optimization problems aimed at minimizing cost and maximizing benefit. Solving optimization problems is one of the main tasks that engineers have daily to deal with. Optimization problems are ubiquitous in real world, and it is hard to find a universal approach to solving them. The optimization problem can usually be divided into subtopics restricted by some specific circumstances, and each of the subtopics forms the corresponding field of science. Linear programming, one of such subtopics, succeeds because of the linear feature specific to the considered problems. But the world is full of problems, which are too difficult to be solved within only the linear programming paradigm. Multicriteria optimization is one of these difficult problems[1].

While dealing with the problem of multicriteria optimization, it seems logical to consider two issues: 1) identifying a utility function and 2) acquiring pareto optimal solutions. These two issues are, in fact, complementary. The process of identifying a utility function most typically involves perturbing parameters of an existing solution candidate and generating a new solution candidate, exposing the new candidate to a decision maker, and analyzing the result to effectively make use of the reaction of the decision maker. If the non-linearity of the target problem is significant, the efficiency of this process is limited as a small perturbation can lead to a large difference in evaluating.

The issue of acquiring pareto optimal solutions is a difficult problem by itself. Exploring a set of solution candidates, which covers all around the pareto frontier, is the main task in the problem. It is left to the decision maker which candidates should be selected among the pareto optimal solutions. Generally, there must be many solution candidates, and it is expected that reducing the selection time would facilitate the decision making process.

2 Multicriteria Optimization by Using Evolutionary Algorithms

Pareto optimality is a concept that plays a central role in the field of multicriteria optimization. The solution candidates to be used by the decision maker should be of the pareto optimal solutions, irrespective of his or her subjective preferences. So, searching pareto optimal solutions is an essential part of the decision making process.

There are many studies aimed at obtaining pareto optimal solutions by adopting some procedural heuristic mechanisms, because it is hard to do so only with mathematical apparatuses. The rapid growth of computer performance makes it possible to employ more complicated mechanisms, such as population-based methods like genetic algorithms[2][3], in order to search pareto optimal solutions.

Although it may seem preferable to exhibit all the obtained alternatives to the decision maker, the burden of the decision maker is quite heavy if the number of the alternatives is too large. It would then be meaningful to exclude anyhow undesirable candidates and focus on only interesting solution candidates, even if having a smaller probability of being inspired from them.

The utility function of the decision maker has to be clear to allow for choosing among the solution candidates. In reality, however, the utility function is often not clear, and the process of "digging up" the function is required. Studies on identifying the utility function usually focus on generating a new candidate by interactively perturbing the old candidate. If the problem is complex enough, the influence of perturbation addition may be significant and the efficiency of the decision making process may be little. In contrast with it, the approach based on evolutionary algorithms has an advantage of the robustness resulting from the characteristics of population-based search.

3 The Proposed Method

The method described in this section is to lighten the burden of the decision maker. It is proposed to leave out the solution candidates which are strongly undesirable for the decision maker, even if they belong to the pareto frontier. The exclusion of solution candidates is performed by interactively modifying a fitness function in evolutionary algorithms. To do so, the following points are to be clarified: 1) the definition of the evaluation of the fitness function, and 2) the procedure of changing the fitness function.

In our study, the evaluation of a solution candidate is defined as a value, which is inversely proportional to the number of solution candidates that were given a high fitness but were not chosen by the decision maker at the previous stage of the problem solving process. Regarding the second point, it will be assumed that the fitness function is changed following the standard procedures of evolutionary algorithms. We define the fitness function as an average of the phenotypes of individuals, where each individual has a phenotype that can be used as an instance of the fitness function. It is expected that the population comprised of such individuals evolves in accordance with the preferences of the decision maker by interactively evaluating each individual, resulting in a fitness function most "suitable" for the decision maker.

The following consequences are expected by utilizing the proposed method.

- The "best" (for the particular decision maker) fitness function is found by applying genetic operators to the population.

- Solution candidates are searched within the neighborhood of the favorites of the decision maker.

- Then, the above causes that only "good" solution candidates are revealed to the decision maker in few steps.

4 The Application Problem

The problem of designing a pattern generator for a walking robot will be considered in this section. The behavior of the pattern generator is represented by the following differential equation (see [4] for details):

$$\begin{aligned} T_a \frac{dx_i(t)}{dt} + x_i(t) &= -\sum_{j=0, j\neq i}^{n-1} a_{ij} y_i(t) \\ &\quad - b f_i(t) + s_i \\ T_f \frac{df_i(t)}{dt} + f_i(t) &= y_i(t) \qquad (1) \\ y_i(t) &= \begin{cases} x_i(t) & \text{if } x_i(t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \\ i &= (0, 1, \cdots, n-1) \end{aligned}$$

In (1), the variable n signifies the number of neurons which compose the pattern generator. n is set to 4 corresponding to the number of joints of the robot model. The correspondence between the pattern generator and the robot model is shown in Fig.1.

The reasons of selecting the problem for the application of the method are as follows:

1. Designing a pattern generator is more difficult than, for instance, traditionally considered toy problems, and

2. There exists no universal function to evaluate a walking trajectory, so it is usual for the decision maker to define an ad hoc function for every particular design.

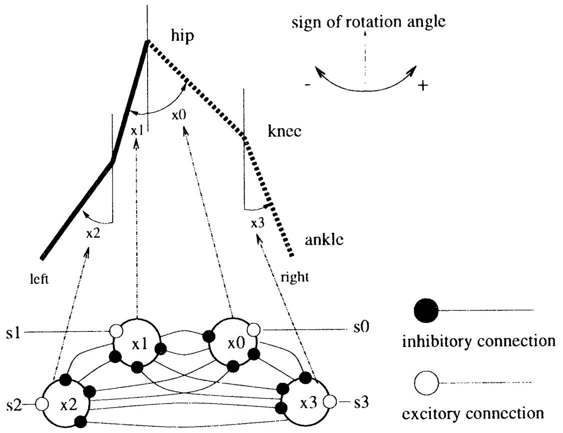


Figure 1: The correspondence between the pattern generator and the robot model.

In this study, the function to evaluate walking trajectories is defined as a composition of five criteria:

$$F = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 + w_5 f_5,$$

where f_1 : the whole walking distance,
f_2 : the penalty against vertical movement,
f_3 : the count of steps,
f_4 : the penalty against unaccustomed joint movement,
f_5 : the maximum stride,
w_i : the weight for $f_i (i=1,2,\cdots,5)$.
(2)

Each f_i in (2) is normalized in the range $[0,1]$ by the minimum value and the maximum value obtained through a run at the algorithm. For the simplicity purpose, some parameters, which can intuitively be defined, are fixed as shown in Tbl.1. Then, the main task in designing a pattern generator for a walking robot is reduced to searching the parameters a_{ij} of the equation (1).

5 Experimental Results

First, let us introduce the following notation:

I_p : an individual which encodes a set of parameters of a pattern generator,

Table 1: The fixed parameters of the pattern generator.

name of the parameter	T_a	T_f	b	s_0, s_3	s_1, s_2
its value	1	12	2.5	0.3	1.0

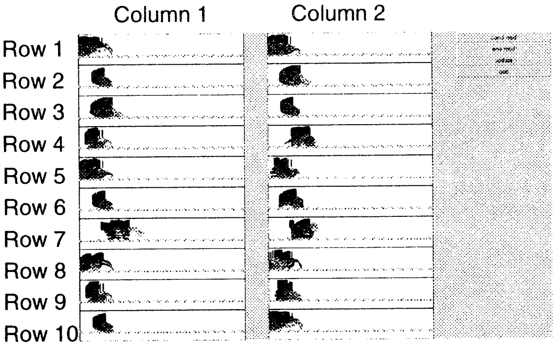


Figure 2. The snapshot of the tool.

I_f : an individual which encodes a set of weights for (2),

P_p : a population consisting of I_ps,

P_f : a population comprised of I_fs.

Fig.2 shows a snapshot of an experimental software tool developed to realize the proposed method. Some walking patterns are displayed in Fig.2 in the matrix form. The number of rows of the matrix represents the number of P_f, and the number of columns has been determined by a decision maker as the number of I_ps to be evaluated for each I_f. The I_ps displayed for one I_f are chosen among P_p in the order of the values which are calculated as the inner product of the phenotype of the I_f and $f_i (i = 1, 2, \cdots, 5)$ of the corresponding walking trajectory.

Fig.3 presents one of the walking trajectories obtained by applying the tool with the parameters shown in Tbl.2 according to the following procedure:

Step 1. Randomly initialize P_p and P_f.

Step 2. Display the walking trajectories of I_ps which have the highest value resulted from the inner product of every I_fs.

Step 3. Evaluate some I_f negatively by clicking the figures corresponding to an undesirable walking trajectories.

Step 4. Apply genetic operators to P_p.

Step 5. Renew the displayed data to be consistent with P_p.

Step 6. Evaluate P_f.

Step 7. Apply genetic operators.

Step 8. Go back to Step 2 if P_f is not sufficiently converged.

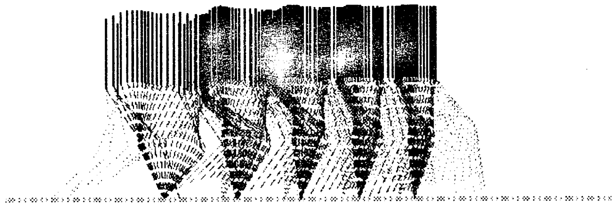


Figure 3. A walking trajectory.

6 Summary

In this paper, we have proposed a method to lighten the burden of a decision maker in multicriteria optimization. The main idea of the method is to exclude the searching of undesirable solution candidates by interactively refining the fitness function in evolutionary algorithms. An experimental tool has been developed to verify the method, and a case study of designing a walking pattern generator for a biped robot has been made. The study results have shown us that:

- It is difficult for the decision maker to evaluate walking trajectories, in which movement is an important factor, from a static figure.
- There is almost no difference between I_fs in fitness.
- I_f exhibiting only mediocre solution candidates gets better evaluation than I_f exhibiting both a preferred candidate and an undesirable candidate.

The advantages of the method proposed in this study are as follows:

- By applying the developed tool, it becomes possible to control the decision making process in the plain and predictable manner.
- The time required for the solution generation is decreased.
- It becomes possible to assess individual preferences of the decision maker by analyzing the transition of P_f.

It is considered that problems, in which the evaluation of each solution candidates largely depends on visual data, could be most suitable for the application of the proposed method.

Table 2: The parameters and its Values of Experiment.

the size of P_f	10
the number of elites of P_f	1
the size of P_p	200
the number of elites of P_p	10
crossover ratio	0.8
mutation ratio	0.002
the number of candidates to be evaluated for each I_f	2
the maximum value of a_{ij} at Eqn.(1)	5
the maximum value of w_i at Eqn.(2)	2
the length of the chromosome of I_p	6
the length of the chromosome of I_f	6

References

[1] Pratyush Sen, Jian-Bo Yang: "Multiple Criteria Decision Support in Engineering Design", Springer, 1998

[2] Carlos M. Fonseca, Peter J. Fleming: "An Overview of Evolutionary Algorithms in Multiobjective Optimization", *Evolutionary Computation*, Vol 3(1), pp.1–16, MIT Press, 1995

[3] Jeffrey Horn: "Multicriteria decision making", *Handbook of Evolutionary Computation*, pp,1–15, IOP Publishing Ltd. and Oxford University Press, 1997

[4] Matsuoka K., "Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators", *Biol. Cybern.*, Vol.56, pp.345–353, 1987

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001.

An Evolutionary Approach to Decentralized Reinforcement Learning for Walking Robots

○S. Ushio*, M. Svinin**, K. Ueda*, and S. Hosoe**

*Mechanical Engineering Department, Kobe University, Rokko-dai, Nada-ku, Kobe, 657-8501, Japan
**Bio-Mimetic Control Research Center, RIKEN, Anagahora, Shimoshidami, Nagoya, 463-0003, Japan

Abstract

An evolutionary approach to decentralized reinforcement learning for an eight-legged walking robot is studied in this paper. The decentralized reinforcement learning system is composed of eight learning subsystems, one for each leg of the robot. A genetic algorithm is used to determine optimal connections between the eight learning systems, and classifier systems are used to control the robot legs for a specific connection structure. The evolutionary approach is tested under computer simulation.

Key Words: decentralized reinforcement learning, walking robot, evolutionary method, connection.

1 Introduction

It is known from biology that movements of individual legs of insects are controlled by independent control systems, and that the individual controllers share information between themselves according to a certain connection structure [1, 2, 3]. It is also known that this connection structure is different for different types of insects.

In this paper we try to understand how such a structure of the leg interconnection can be determined or acquired in an evolutionary manner by learning a specific walking task. In our approach, the individual leg controllers are modeled by reinforcement learning modules, implemented as classifier systems, that are highly adaptive and possess self-organizing properties. Integration of the individual controllers into one scheme is done by specifying how the total sensor space is shared by the classifier systems.

In our previous research [4], the connection structure was specified by the designer. This can be done more or less easily for a relatively small number of legs. However, as the number of legs increases, it becomes more and more difficult to justify the optimality of an intuitively pre-specified connection structure.

Note that the problem under consideration can be viewed in a broader context in which it is required to design a reinforcement learning system with an optimal degree of decentralization or with an optimal reduction of the state space. It is well-known that reinforcement learning systems operating in large state and action spaces require long learning time [5]. In this situation, decentralization of one control system into several autonomous modules can be beneficial. However, it is not clear how to define an optimal sharing of the state space between the subsystems. In our opinion, the sharing scheme can be a crucial factor in the learning performance. The use of evolutionary techniques for finding an optimal sharing scheme is reasonable, as the design problem under consideration cannot be formulated in an analytical form.

This paper is organized as follows. Section 2 gives the description of the robot and outlines basic components of our approach. Section 3 discusses results of the computer experiments. Finally, Section 4 concludes the paper.

2 System Description

2.1 Walking robot

An eight-legged walking robot is used in the computer experiments. The robot is shown in Figure 1. Every leg has two degree of freedom (swinging in the body plane, and lifting up and down). The robot leg is controlled discretely, and it (the leg) can be in one of the following states.

[1]: The leg is swung forward and lifted down.
[2]: The leg is swung backward and lifted down.
[3]: The leg is swung backward and lifted up.
[4]: The leg is swung forward and lifted up

Depending on the states of the legs, the robot can be in stable or unstable configurations. The stability is understood in the static sense, i.e., it is preserved if the gravity center of the body, projected onto the ground, lies within the contact leg polygon.

It is assumed that the robot is equipped with three light sensors located at the front part of the robot body. The reading of the light sensor is within the interval $[0, 150]$. The smaller the reading, the brighter the light signal.

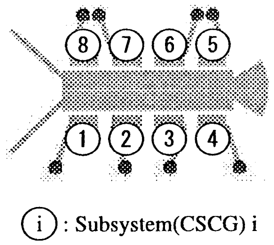


Figure 1: Eight-legged walking robot

2.2 Robot's task

In the computer experiments, we pursue a simple goal: to make the robot able to reach the light source located at a certain distance in front of the robot. The experimental environment is shown in Figure 2.

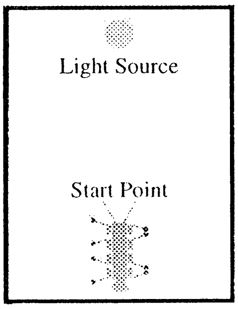


Figure 2: Simulation environment

It is assumed that the robot leg moves in a sequential manner, changing its state as follows

$$\ldots[1] \rightarrow [2] \rightarrow [4] \rightarrow [1] \rightarrow [2] \rightarrow [4] \rightarrow [1] \ldots \quad (1)$$

For example, in the [1] → [2] transition the leg is swung backward while keeping the contact with the ground, which drives the robot body forward. In the [2] → [4] transition the leg is lifted up and swung forward. In the [4] → [1] transition the leg is simply lifted down.

While the trajectory of the leg states is pre-specified in the above sequence, the timing between the changes of the states is not fixed. It should be regulated in such a way so that to preserve the robot stability.

2.3 Reinforcement learning system

A reinforcement learning system called CSCG (Continuous Space Classifier Generator, see [4] for the details) is used in controlling the robot. It is assumed that each leg of the robot is controlled by its own CSCG, as shown in Figure 1. The CSCGs can share information between one another, forming a certain connection structure.

Depending on the connection structure, a CSCG is inputted current values of the light sensors, the state of its own leg, and the states of the legs connected to this CSCG. The output of the CSCG (that is the next state of the leg, associated with the CSCG) is defined as follows. The CSCG outputs 1 if the state of the leg should be kept same as the current one, and 2 if the state should be changed to the next one in the sequence (1). For example, if the current state of the leg is [4] and the CSCG outputs 2, then the next state should be [1].

The robot gets a positive global reward when it reaches the light source[1]. In addition, the robot may get a positive local reward whenever it goes forward to the goal. The robot is punished whenever it falls down or deviates from the goal.

The time interval between the starting motion and stopping the robot is called an episode. The episodes are updated when the robot reaches the light source, or when the number of produced actions (steps) reaches 300. The learning performance of the robot is measured by the number of goals reached in a fixed period of time.

2.4 Evolutionary method

To define an optimal connection structure (the one with the highest learning performance), we employ a simple genetic algorithm (GA) as described in [6]. The genetic string consists of 64 bits, concatenating 8 bits substrings of all the CSCGs. For a given CSCG the i-th bit is 0 if the information about the state of the i-th CSCG is not available for the CSCG under consideration. Otherwise, the i-th bit is set to 1.

The fitness function is defined as follows:

$$Fitness = P + const \qquad (2)$$

where P is the learning performance given by the number of successful episodes (when the robot reaches the goal) in the series of 100 trials.

3 Simulation results

The results of the computer experiments are shown in Table 1. The connection structure, defined by the GA, is not symmetric. In a symmetric connection structure the i-th and j-th CSCGs share the information on the states between each other. In an asymmetric connection structure, the i-th CSCG can get the information on the state of the j-th CSCG but the j-th CSCG not necessarily gets the information on the state of the i-th CSCG.

[1] Strictly speaking, rewards and punishment are given not to the robot but to certain classifiers of the CSCGs.

Table 1: Asymmetric connection

Input State	Leg1	Leg2	Leg3	Leg4	Leg5	Leg6	Leg7	Leg8
Subsystem1	1	1	1	0	1	0	0	0
Subsystem2	0	1	0	0	0	0	0	0
Subsystem3	1	0	1	0	1	0	0	0
Subsystem4	0	0	1	1	0	0	0	0
Subsystem5	0	0	0	0	1	0	1	0
Subsystem6	0	1	0	0	1	1	0	0
Subsystem7	0	0	0	0	0	0	1	1
Subsystem8	0	1	0	0	0	0	1	1

Table 2: Symmetric connection

Input State	Leg1	Leg2	Leg3	Leg4	Leg5	Leg6	Leg7	Leg8
Subsystem1	1	1	0	0	0	0	0	1
Subsystem2	1	1	1	0	0	0	0	0
Subsystem3	0	1	1	1	0	0	0	0
Subsystem4	0	0	1	1	1	0	0	0
Subsystem5	0	0	0	1	1	1	0	0
Subsystem6	0	0	0	0	1	1	1	0
Subsystem7	0	0	0	0	0	1	1	1
Subsystem8	1	0	0	0	0	0	1	1

To confirm that the structure evolved by the GA is suboptimal, we measured the learning performance for the structures of the following six types.

Type 1: centralized control system, with one CSCG controlling all the legs.

Type 2: decentralized control system with 8 CSCGs not connected between themselves.

Type 3: control system with 8 CSCGs, where each CSCG is connected to all others.

Type 4: control system for which the connection structure is defined randomly.

Type 5: control system with the symmetric connection structure given by Table 2.

Type 6: control system with the asymmetric connection structure obtained by the GA (Table 1).

For each type of connections we performed 100 computer experiments (one experiments consists of 100 episode) under different random seeds. The results of measurements are summarized in Figure 3 which plots the average (over 100 experiments) value of the learning performance. As can be seen, the asymmetric connection structure evolved by the GA on average scores the highest learning performance.

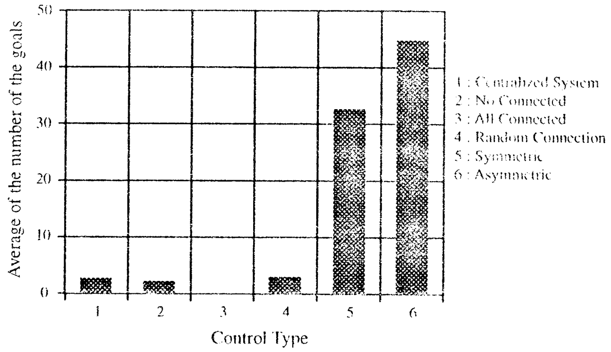
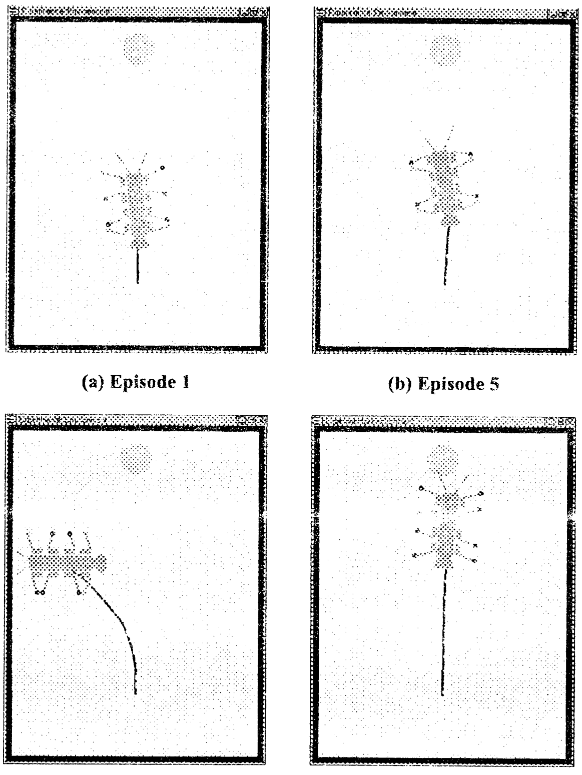


Figure 3: Average learning performance

An example of the learning process with the leg connection structure of type 6 is shown in Figure 4, Figure 5, and Figure 6. The behavior acquisition process is animated (episodes 1, 5, 29 and 41) in Figure 4. The number of local rewards, the number of punishments, and the number of steps[2] necessary to reach the light source, are plotted in Figure 5. The action history of the legs in episode 1, 5 and 41 are shown in Figure 6.


(a) Episode 1

(b) Episode 5

(c) Episode 29

(d) Episode 41

Figure 4: Acquired behavior of the robot

[2] This number strongly correlates with the global rewards. When it was less than 300 the goal was reached and the global reward was given.

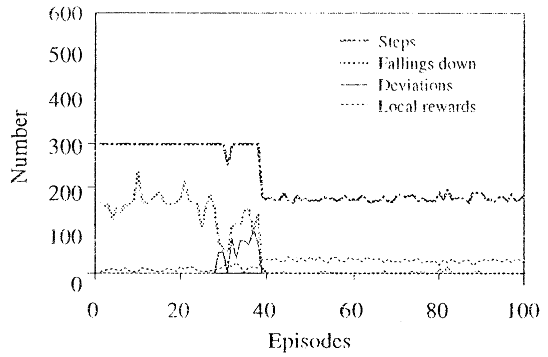


Figure 5: Learning history

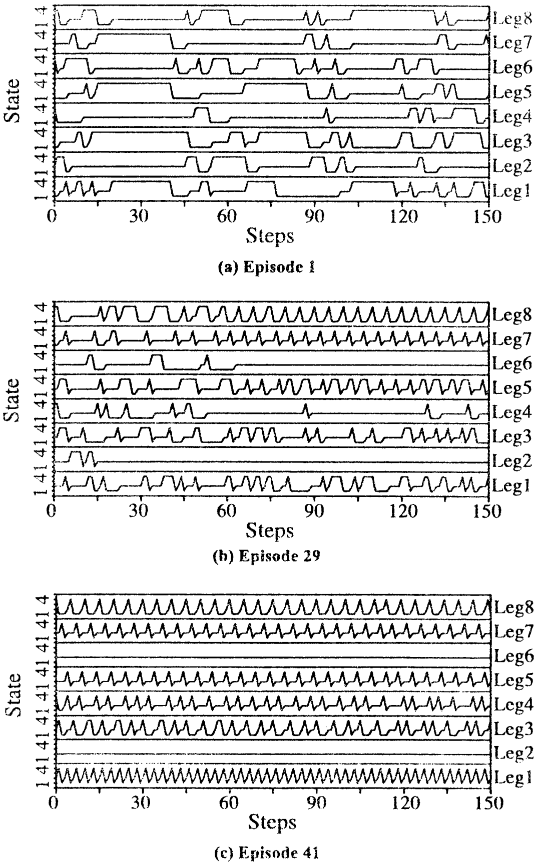
In the first episode, the leg controllers (the CSCGs) have no prior knowledge of environment. The CSCGs output random actions and, as a result, the number of unstable configurations is high. In the 29th episode, the 2nd, 6th, 7th and 8th CSCGs begin to generate specific action patterns. Note that the 2nd CSCG is completely independent as its state is defined only by the state of the corresponding 2nd leg. On the other hand, the state of the 6th CSCG includes the states of the 2nd and 5th legs, and the state of the 8th CSCG does the states of the 2nd and 7th legs. The 7th CSCG gets information on the state of the 8th leg. For this specific connection structure, from the begining of the 41st episode, all the CSCGs begin to generate specific action patterns and the robot reaches the light source in approximately 180 steps

4 Conclusion

An evolutionary approach to designing a reinforcement learning control system for an eight-legged walking robot is proposed in this paper. The controllers of the legs have autonomous reinforcement learning modules implemented as classifier systems. The modules can be connected to one another, forming a certain connection structure. A genetic algorithm is used to determine a suboptimal structure of the connection between the modules for the learning performance. The proposed approach is tested under simulation. The simulation results show sub-optimality of the structure obtained by the GA. In future, we plan to test the evolutionary approach on the real eight-legged walking robot OCT-1b.

Acknowledgements

The first and the third authors acknowledge the support of the Japan Society for the Promotion of Science within the Project JSPS-RFTF96P00702, "Methodology of Emergent Synthesis."



(a) Episode 1

(b) Episode 29

(c) Episode 41

Figure 6: Leg action history for the asymmetric connection structure.

References

[1] E. von Holst, Über relative Koordination bei Arthropoden. *Pflügers Archiv* **246**, 1943, pp. 847–865.

[2] G. Wendler, Laufen und Stehen der Stabheuschrecke: Sinnesborsten in den Beingelenken als Glieder von Regelkreisen. *Z. Vergl. Physiol.* **48**, 1964, pp. 198–250.

[3] H. Cruse, T. Kindermann, M. Schumm, J. Dean and J Schmitz, Walnet—a biologically inspired network to control six-legged walking. *Neural Networks* **11**, 1998, pp. 1435–1447.

[4] M. Svinin, K. Yamada, and K. Ueda, Reinforcement Learning Approach to Acquisition of Stable Gaits for Locomotion Robots, *Proc. IEEE International Conference on Systems, Man & Cybernetics*, 1999, Vol. VI, pp. 936–941.

[5] R.S. Sutton, and A.G. Barto, *Reinforcement Learning : An Introduction*, MIT Press, Cambridge, Massachusetts, A Bradford Book, 1998.

[6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Emergence of Supply Chains by Producers' Selection of Suppliers

Daisuke Nakanishi[1], Itsuo Hatono[2], Kanji Ueda[3]

[1]Graduate School of Science and Technology, Kobe University,
[2]Information Processing Center, Kobe University
[3]Department of Mechanical Engineering, Kobe University
Rokkodai, Nada, Kobe 657-8501, Japan

Abstract

This paper describes an approach to modeling supply chains, which consist of agents, such as material suppliers, parts suppliers, assembly makers, dealers, consumers, and so on. The agents have strategies to deal with different objectives. The agent's behavior is based on the strategies and depends on the agent's internal states and interactions between other agents. We apply this approach to simple market simulations. The simulation shows that emergence of supply chains can be observed, and it suggests that this approach can be applied in decision making support systems for supply chain management.

Key Words: Emergence of Supply Chains, Utility Theory, Decision Making, Simulation of Supply Chains

1 Introduction

Supply chain management has become one of the most important question of virtual enterprises, which consist of many assembly makers, material suppliers, parts suppliers, and dealers, to gain profits. On the market, we can find many software packages developed for supply chain management. The most part of these software packages deals with the "fixed supply chains." Although, to organize efficient supply chains is important to cope with difficulties, such as the rapid change of economic environment, and the diversified consumers' needs. There are few approaches to support organization of supply chains [1].

The aim of this paper is to propose an approach to cope with the difficulties mentioned above. In our approach, supply chains are created as the result of local interactions among agents. And supply chains affect the agents as constraints. By using the model, we can simulate the markets, and can observe "emergence of supply chains." Furthermore, we suggest the model can be used for decision support tools to organize and analyze supply chains.

2 Modeling of Supply Chains

In this paper, we assume that supply chains are created as the result of local interactions among agents, such as material suppliers, parts suppliers, assembly makers, dealers, consumers and so on. The agents have strategies to deal with different objectives. The strategies are described by using utility functions. The agent's behavior is based on the strategies and depends on the agent's internal states and interactions between other agents.

2.1 Emergence of Supply Chains

We assume that *emergence* in this paper is used as the phenomenon of the appearance of the tendentious order through continuous interactive processes: local interactions between agents reveal a global behavior, and the global behavior affects the behavior of the agents as constraints. Therefore, we can interpret the process forming supply chains as emergence of supply chains.

2.2 Agents

An agent $A_i \in Agent$ is defined by

$$A_i = \langle State_{A_i}, Act_i, state_{A_i}, state_{F_j A_i}, \vartheta_i, \varpi_i \rangle, \quad (1)$$

where $Agent$ is a set of agents. $State_{A_i}$ is a set of states of the ith agent A_i, and $state_{A_i} \in State_{A_i}$ is a state of A_i. $state_{F_j A_i}$ is a state of the jth field F_j from the viewpoint of A_i, and Act_i is a set of actions of A_i. ϑ_i is a transfer function, and ϖ_i is a decision making function [2].

$$state_{A_i} = \langle Char_i, Prop_i, \textstyle\bigcup_k I_k \rangle, \quad (2)$$

where $Char_i$ is a set of characteristics of A_i, $Prop_i$ is a set of properties, which are factors to decide their strategies. I_k is the kth intermediate which is in A_i. The transfer function ϑ_i is as follows:

$$\vartheta_i(state_{A_i}, state_{F_j A_i}, act_i) = (state'_{A_i}, state'_{F_j A_i}), \quad (3)$$

where $state_{A_i}$ is the current state of A_i, and $state_{F_j A_i}$ is the current state of F_j from the viewpoint of A_i. $state'_{A_i}$ is the future state of A_i, and $state'_{F_j A_i}$ is the future state of the jth field F_j from the viewpoint of A_i.

In decision making, each agent selects the action act_i^* if

$$\vartheta_i(state_{A_i}, state_{F_j A_i}, act_i^*) = (state^*_{A_i}, state'_{F_j A_i}), \quad (4)$$

where $state^*_{A_i}$ is the best state for A_i in a set of future states of A_i.

The decision making function ϖ_i can be defined as follows:

$$\varpi_i(state_{A_i}, state_{F_j A_i}) = act_i^*. \quad (5)$$

2.3 Intermediates

Intermediates are transfered among agents. For example, the intermediates represent money, products, and information. An intermediate I_k is defined by

$$I_k = \langle Type_{I_k}, Char_k \rangle, \quad (6)$$

where $Type_{I_k}$ is a set of types of the kth intermediate I_k, and $Char_k$ is a set of characteristics of I_k.

2.4 Fields

A field $F_j \in Field$ is given by

$$F_j = \langle \bigcup_i A_i, \bigcup_k I_k, \bigcup_k Type_{I_k}, State_{F_j}, \varphi_j \rangle, \quad (7)$$

where $Field$ is a set of fields, A_i is the ith agent which is in the jth field F_j, and I_k is an intermediate which is in F_j. $Type_{I_k}$ is a set of types of the kth intermediate I_k. $State_{F_j}$ is a set of states of F_j. φ_j is a mapping function defined as follows:

$$\varphi_j : \prod_i (State_{A_i} \times State_{F_j A_i}) \to State_{F_j}, \quad (8)$$

where $State_{A_i}$ is a set of states of A_i, and $State_{F_j A_i}$ is a set of states of F_j from the viewpoint of A_i.

2.5 Utility Function

Utility is defined by a subjective value and expresses the quantitative preference order among agents. Utility is also used here as a standardized scale to compare different factors in decision making. A utility function u is defined by the following equation:

$$u(x_1, x_2, \ldots, x_l) = f(u_1(x_1), u_2(x_2), \ldots, u_l(x_l)), \quad (9)$$

where x_i is a factor, f is a scalar function, and u_i is a utility function against x_i [3].

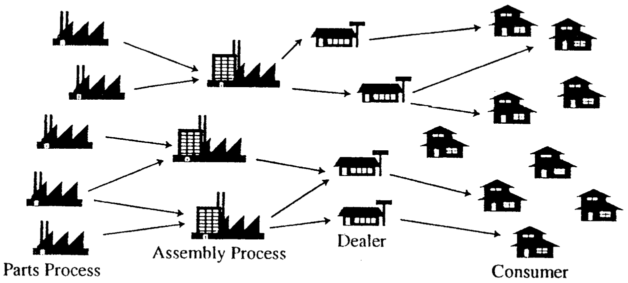


Figure 1: Supply Chains

3 Modeling Examples

In this paper, we focus on supply chains and their effects against strategies at first (Fig. 1). We simulate simple markets which include individual consumers, dealers, assembly makers and parts suppliers as agents. This section describes product models and a behavior of producers and consumers in the simple markets.

3.1 Product Models

We assume that the model of the jth product which consists of n parts is described with n objectively measured characteristics. A set of characteristics $Char_j$ is as follow:

$$Char_j = \{z_{ij} \mid i = 1, 2, \ldots, n\}, \quad (10)$$

where z_{ij} is the value of the ith characteristic.

3.2 Behavior of Agents

Agents maximize their utility, which is estimated from several factors in decision making to contract with other agents for co-operation [4]. A decision making function is defined as

$$\varpi = \max U, \quad (11)$$

where U is a utility function of an agent. The utility function

$$U = u(x_1, x_2, \ldots, x_l) \quad (12)$$

is written as

$$U = \sum_i w_i u_i(x_i) = \sum_i w_i u(x_i), \qquad (13)$$

where w_i is a weight for the ith factor x_i. Therefore, a utility function U_i is described using the ith factor x_i as follow:

$$U_i = u_i(x_i) = u(x_i). \qquad (14)$$

A utility function u is defined as

$$u(x_i) = \begin{cases} 1 - \exp(-x_i/\lambda_i) & (du_i(x_i)/dx_i \geq 0) \\ \exp\left(-(x_i/\lambda_i)^2\right) & (du_i(x_i)/dx_i < 0) \end{cases} \qquad (15)$$

where x_i is a factor of decision making, and λ_i is constant.

3.3 Behavior of Producers and Consumers

Agents contain producer and consumer ones. Dealers, assembly makers and parts suppliers are described as producer agents, On the other hand, individuals and enterprises are described as consumer agents. This is because individuals buy a product by a decision, and enterprises buy some products by a decision.

Producers usually use the strategy "maximization of profits" in decision making. However, in general, each producer must take into account other factors, such as risk of delay, reliability, and so on, in addition to profits. Since it is difficult to convert the factors into money, we assume that each producer determines its behavior based on a utility function. The utility function has three attributes: profits, the reliability of the agent, and the desired due date.

The utility function U_1 for profits, U_2 for the due date and U_3 for the reliability are as follows:

$$U_1 = u(\sum_K \Phi(X_K, z) - C(x, z) - \sum_k C_k(x_k, z)), \qquad (16)$$

$$U_2 = u\left(\max D_k(x_k, z)\right), \qquad (17)$$

$$U_3 = u(\sum_k r_k), \qquad (18)$$

where x is the number of products, x_k is the number of products to order to the kth co-operator ($k = 1, 2, \ldots, n$), X_K is the number of products ordered form Kth co-operator, $\Phi(X_K, z)$ is an offer function, $C(x, z)$ is a cost function, and $C_k(x_k, z)$ is an offer function of the kth co-operator. r_k is the reliability of the kth co-operator and it is written as the history of the number of products which are ordered to the kth co-operator, and $D_k(x_k, z)$ is products' due date of the ith co-operator against x_k.

Utility of consumers is defined as the value of characteristics of products and the available money [5, 6]. Then an utility function U_1 and U_2 are as follows:

$$U_1 = u(Y(z, x)), \qquad (19)$$

$$U_2 = u(o - \Theta(z, x, o)), \qquad (20)$$

where $Y(z, x)$ is a consumption activity, o is an income of a consumer, and $\Theta(z, x, o)$ is a bid function of a consumer.

4 Simulation Results

In the simulated simple market, there are 45 consumers, 15 dealers, 5 assembly makers and 14 suppliers as agents on the fields. Each product consists of 7 parts. A supplier makes only one type of parts. There are two suppliers for one type. An assembly maker also makes one type of products. Assembly makers put emphasis on profits or the due date. The 1st and the 5th assembly makers put emphasis on profits. The rest put emphasis on the due date. All the assembly makers make products with the same characteristics.

We assume that the 5th assembly maker will withdraw form the market at 500 time steps in the simulation.

4.1 Emergence of Supply Chains

Through simulation, we observe emergence of supply chains. For example, the supply chain including the 1st assembly maker is also changed after the withdraw of the 5th assembly maker. This result suggests the supply chain can adapt itself to the environmental change.

4.2 Analysis of Market Shares

Before the withdraw of the 5th assembly maker, the 1st and 5th assembly maker have relatively larger shares than the others (Fig. 2). After 500 time steps, the 1st assembly maker becomes to have a top share. The reason is that it is the only assembly maker which puts emphasis on profits. On the other hand, assembly makers which put emphasis on the due date are rarely affected. The simulation results suggest that the difference of producers' strategies strongly affects the market share of products.

4.3 Analysis of the Process of Contracting

In interactions among the 1st assembly maker, the 4th supplier and the 11th supplier, the contracts between the 1st assembly maker and the 11th supplier increase after the environmental change (Table 1). This means that the 1st assembly maker gets a higher utility to contract with the 11th supplier than with the

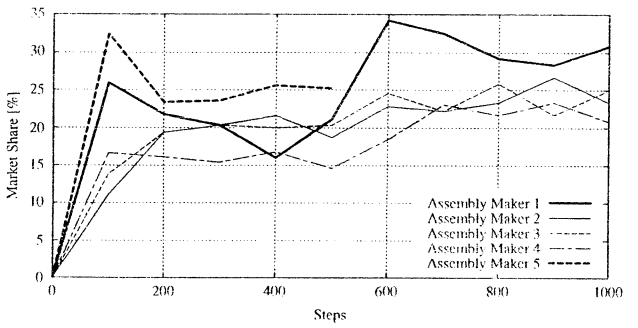


Figure 2: Market shares

Table 1: Tendencies of contracts

Contract Connection		Value	
		Before	After
Assembly maker 1	Supplier 4	58	13
	Supplier 11	72	170
Assembly maker 5	Supplier 4	41	0
	Supplier 11	112	0

4th supplier before the 5th assembly maker withdraws. Furthermore, the 11th supplier is often contracted by the 5th assembly maker before the change. After the change, the 1st assembly maker increases contracts with the 11th supplier because of the absence of competition. This means that the 1st assembly maker adjusts itself to a dynamic environment change and its condition, and the 1st assembly maker gets a higher utility of contracts after the absence of competition.

5 Conclusion

In this paper, we described a new approach: supply chains are created as the result of local interactions among agents and the behavior of supply chains strongly affects the behavior of the agents as constraints.

Furthermore, we developed simple market simulations to evaluate the suggested approach. We can observe emergence of supply chains in the simulation. The analysis of market shares suggests that the deference of producers' strategies affects the market share of products. The analysis of the emergent process shows that each agent and supply chain can adapt itself to environmental changes. These considerations indicate that it is possible apply the suggested approach to decision making support systems for supply chain management.

Acknowledgments

This study has been supported in part by the "Methodology of Emergent Synthesis" Project (JSPS-RFTF96P00702) in Research for the Future Program of the Japan Society for the Promotion of Science and BMS Group in NGMS project in IMS program.

References

[1] K. Ueda, J. Vaario, T. Takeshita, I. Hatono, "An Emergent Synthetic Approach to Supply Network," *Annals of the CIRP*, Vol. 48, No. 1, 1999, pp. 377–380.

[2] T. Sueyoshi and M. Tokoro. "Dynamic modeling of agents for coordination," *Decentralized A.I. 2*, Elsevier Science, 1991, pp. 161–176.

[3] R. L. Keeney, H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley and Sons, 1976.

[4] R. G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, Vol. C-29, No. 12, 1980, pp. 1104–1113.

[5] K. Lancaster, *Modern Consumer Theory*, Edward Elgar, 1991.

[6] S. Rosen, "Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition," *Journal of Political Economy*, Vol. 82, No. 1, 1974, pp. 34–55.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Chaotic Information Maximization for Blind Source Separation

○ Wenwei Yu, Hiroshi Yokoi, Yukinori Kakazu
Complex Engineering Department, Hokkaido University, Japan
{yu, yokoi, kakazu}@complex.eng.hokudai.ac.jp

ABSTRACT

Blind Source Separation (BSS) is a process that uses mixtures of several signal sources only, to extract the original signals. Some learning methods that can perform BSS were suggested. However, the adaptation rules of the parameter has the problem that, the learning tends to be trapped by local minimums caused by periodicity of signal sources, multi-path transfer, etc. So that, we introduce Chaotic Steepest Decent (CSD) Model, a chaotic method, into the BSS network architecture. Taking the mutual information as the Objective Function, we first derive the adaptation rules for each element of the parameter matrix. Then we analyze the dynamics of the chaotic BSS system, and suggest a cross-correlation based monitoring algorithm for the chaotic BSS system. At last, we verify the validation of the system by results of experiments on audio recordings.

Keywords: *information-maximization, BBS, learning, local minimum, chaotic, search*

INTRODUCTION

Blind Source Separation (BSS) is a process that uses mixtures of several signal sources only, to extract the original signals. BSS has a large number of potential applications in communications, speech and medical signal processing, radar or sonar array signal processing, etc.

Existing algorithms for BSS can be roughly divided into two categories, one is that utilizes some kinds of higher order statistics implicitly through some nonlinear functions[3], and another is that does it explicitly. In 1995, Bell et al[2] presented a self-organization network architecture capable of performing blind separation, based on the information maximization principle. Later, Torkkola[4] extended the architecture into more general cases where the sources may have been delayed with respect to each other. The problem is formulated as to maximize the mutual information that the output Y of a neural network processor contains about its input X, by adapting the elements of a weight matrix W to suitable values.

However, just like the other neural network learning algorithms, the adaptation rules of the weight matrix W and delay D matrix have the problem that, the learning tends to be trapped by local minimums caused by periodicity of signal sources, multi-path transfer, etc. This is not trivial, since at least, in most of the potential application areas, such as speech signal processing and biomedical signal processing, signal sources are usually periodic. The problem is aggravated by the self-organization nature of the learning algorithms.

Apparently, it is necessary to introduce mechanisms that are capable of avoiding the local minimum problem. Three classes of methods can be considered. The first one includes statistical methods, such as Simulated Annealing, which supply perturbation to search processes explicitly. The second one includes some global search methods, like GA and ES, which generate a diversity of search patterns by maintaining a large number of gene groups[5]. The third one includes chaotic methods, in which the deterministic chaos is embedded into systems to create diverse dynamics[5].

Considering that BSS is a process rather than only a result, we expect to investigate the internal dynamics of the BSS network architecture. So that, we introduce Chaotic Steepest Decent (CSD) Model, a chaotic method, into the BSS network architecture.

In CSD model, first proposed by Jun Tani[1], a nonlinear resistance which depends on the derivative of state vector and varies periodically is introduced into the energy (Objective Function) landscape of a system, so that, the state of the system tends to travel from one energy basin to another.

Taking the mutual information as the Objective Function, we first derive the adaptation rules for each element of the delay matrix. Then we analyze the dynamics of the chaotic BSS system, and suggest a cross-correlation based monitoring algorithm for the chaotic BSS system. At last, we verify the validation of the system by results of experiments on both artificial data and audio recordings.

METHODS

1. Information-Maximization

For the completeness of the description, we will first describe the Information-Maximization, using the formulation introduced by Torkkola[4]. Assuming two sources s_1 and s_2 and their two delayed mixtures x_1, x_2, we have the expression:

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t-D_{12}) \\ x_2(t) &= a_{22}s_2(t) + a_{21}s_1(t-D_{21}) \end{aligned} \quad (1)$$

After the Torkkola [4] and Bell et al[2], the separation computation can be shown by expression (2), where, u_i are the outputs before the squashing, and w_{0i} are the bias weights.

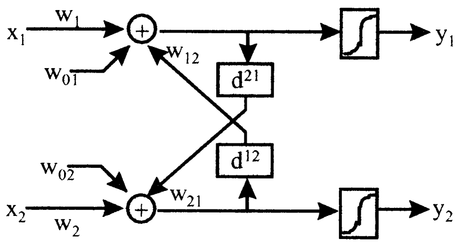


Figure 1. The separation network

$$\begin{aligned} u_1(t) &= w_1x_1(t) + w_{12}(t-d_{12}) + w_{01} \\ u_2(t) &= w_2x_2(t) + w_{21}(t-d_{21}) + w_{02} \end{aligned} \quad (2)$$

Then, figure 1 shows the mixing and separating network under the formulation, where,

$$y_1(t) = g(u_1(t)) \qquad y_2(t) = g(u_2(t)) \quad (3)$$

and we use $g(u) = 1/(1+e^{-u})$ as our squashing function in this paper, however, it can be replaced by other similar functions.

As was shown in the Bell et al[2], the mutual information between outputs y_1 and y_2 can be minimized by maximizing the entropy at the outputs, which equal to maximizing $E[ln(J)]$, which is presented by (4)

$$|J| = \frac{\partial y_1}{\partial x_1}\frac{\partial y_2}{\partial x_2} - \frac{\partial y_1}{\partial x_2}\frac{\partial y_2}{\partial x_1} = y_1'y_2'D$$

$$\ln|J| = \ln(y_1') + \ln(y_2') + \ln(D) \quad (4)$$

where, $D = \frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1} \qquad y_1' = \frac{\partial y_1}{\partial u_1}$.

Taking d_{ij} as examples, we get,

$$\begin{aligned} \nabla J(d_{12}) &\propto -(1-2y_1)w_{12}\,\dot{u}_2(t-d_{12}) \\ \nabla J(d_{21}) &\propto -(1-2y_2)w_{21}\,\dot{u}_1(t-d_{21}) \end{aligned} \quad (5)$$

2. Chaotic Information-Maximization

Instead of using the steepest decent, CSD employs a time development equation:

$$m\ddot{d}_{ij} + f_1(\dot{d}_{ij}, \omega t) = -\varepsilon\nabla J(d_{ij}) + f_2(d_{ij}) \quad (6)$$

where, function f_1, shown in (7), represents a nonlinear damper whose gain oscillates periodically. ω is the angular velocity by which the gain of the nonlinear damper changes. d_0, d_1 are the coefficients for the linear damper; d_2 is the constant gain for the nonlinear damper. f_2 is the function forcing the d_{ij} to itinerate in its range.

$$f_1(\dot{d}_{ij}, \omega t) = [d_0 \sin(\omega t) + d_1]\dot{d}_{ij} + d_2\,\dot{d}_{ij}\,\dot{d}_{ij}\,\mathrm{sgn}(\dot{d}_{ij}) \quad (7)$$

$$f_2(d_{ij}) = \begin{cases} 0 & low\_limit \le d_{ij} \le high\_limit \\ -wed & d_{ij} \ge high\_limit \\ wed & d_{ij} \le low\_limit \end{cases} \quad (8)$$

3. Monitoring for Chaotic Information Maximization

We expect that the information maximization can benefit from the chaotic itinerary and search the whole space without being trapped by local minimums. However, as for BSS problems, we need not only explore the problem space completely, but also to obtain an accurate solution effectively. CSD doesn't supply such kinds of mechanisms, but its chaotic exploration can be restrained to some extents by adjusting the coefficient d_2[1]. We introduce a monitoring algorithm for chaotic search, which is illustrated in figure 2.

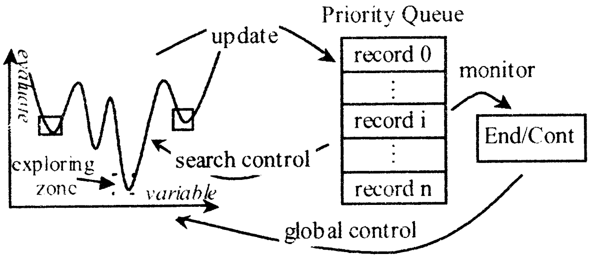


Figure 2. Architecture for Monitoring

The idea is to keep a record of experienced local minimums on the evaluation surface. This local record, as well its neighborhoods form one cluster, and will be explored minutely to find better solutions.

A cross-correlation is employed to evaluate each solution. By the continuity of the cross-correlation, the neighborhoods of one minimum ought to have a relatively good evaluation. 3 main functions are described in table 1.

Table 1. Rules for Monitoring Functions

Function	Condition	Execution
Update	$E_{solution} < E_{queue}$	Register solution *iFlagRegister*=T
	Solution ∈ *Cluster*^{i} and $E_{solution} < E_{min}{}^i$ or $E_{avg}{}^i \downarrow$	Update *Cluster*^{i}
Search control	*Cluster*^{i} == *ON*	*par* = *par*^{i}
	$E_{min}{}^i \downarrow$or $E_{avg}{}^i \downarrow$	*par*^{i} = *par*^{i}+ Δ , *iFlagImproved*=T
	$It_{\text{cluster}}{}^i$ > StepTh1	*par* = *par_high*
Global control	*IflagRegister* == F and *IFlagImproved* == F	$It_{unchanged}$++
	$It_{unchange}$ > StepTh2	End search

$E_{solution}$, E_{queu}:evalution of solution and priority queue
Cluster^i: *i*th cluster in priority queue
$E_{min}{}^i$, $E_{avg}{}^i$:minimal and average evaluation of ith cluster in queue
par, *par*^{i} : current and *i*th cluster's CSD parameter d_2
$It_{\text{cluster}}{}^i$, : consecutive iterations of *i*th cluster
$It_{unchange}$: consecutive unchanged iterations

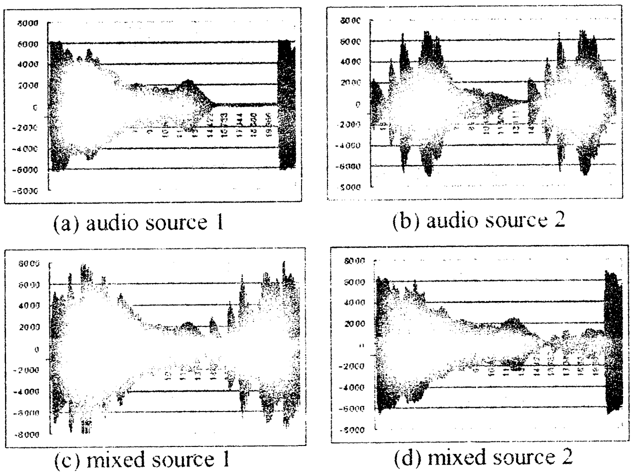
EXPERIMENTS and RESULTS

The time development equation (6) is calculated by a 4th order Runge-Kutta method. Two audio sources, which are presented in figure 3(a), (b) were mixed according to

$$x_1(t) = s_1(t) + 0.2s_2(t-10)$$
$$x_2(t) = s_2(t) + 0.8s_1(t-20) \qquad (9)$$

figure 3(c), (d) show the waveforms of mixed sources.

Figure 4 reveals the local minimum problem caused by mere adaptation. Different initial values cause the convergence to different solutions. That is, the adaptation tends to be trapped by local minimums.



(a) audio source 1 (b) audio source 2

(c) mixed source 1 (d) mixed source 2

Figure 3. Original audio sources and mixed sources

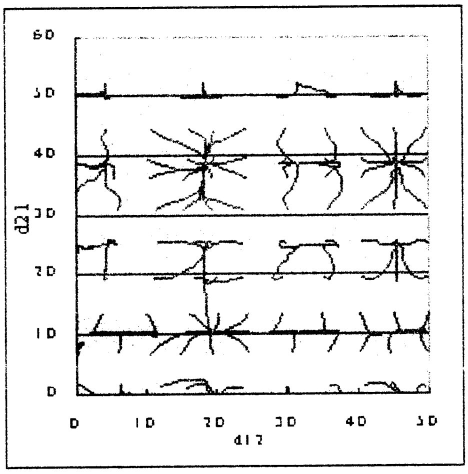


Figure 4. Local Minimums in d_{12} and d_{21} plane

In the case of chaotic adaptation, the nonlinear resistance prevents exploration from being trapped by local minimums. Figure 5 shows the Poincare plotting of 5/21π phase for d_{12}, The parameter d_2 is set to 0.5, which corresponds to a wide-range exploration. As can be seen from the figure, not only the neighborhoods of d_{12}=10 were explored, but also other possible solutions are explored.

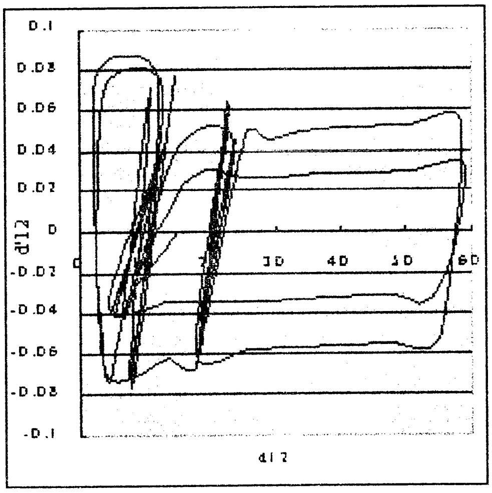


Figure 5. 5/21π phase Poincare plotting for d_{12}

In order to obtain the solution from the CSD, while not disturbing the chaotic itinerary, we suggested a monitoring algorithm. Figure 6 shows the experiment results. In (a), clusters remained in the priority queue are presented by both their minimum evaluation values and average values. Several solutions are even better than real answer, (10, 20), this agrees with the multi-optimum nature of the problem. In (b), explored times and control parameters of the clusters were depicted. Part of the clusters are minutely and frequently explored, while some others are not. Those hardly explored clusters certainly have the low confidential, compared with those well explored. (c) supplies another viewpoint, which plots the evaluation and control parameter of each iteration. The bigger d_2 results in minute explorations, which are usually followed by better solution.

CONCLUSIONS

In this research, in order to solve the local minimum problem, we proposed a chaotic information maximization, with a monitoring algorithms. The validity was verified by experiment results.

As our future works, comparisons with other stochastic method and monte-carlo methods should be done. Moreover, the experiment with real world source should also be considered.

References

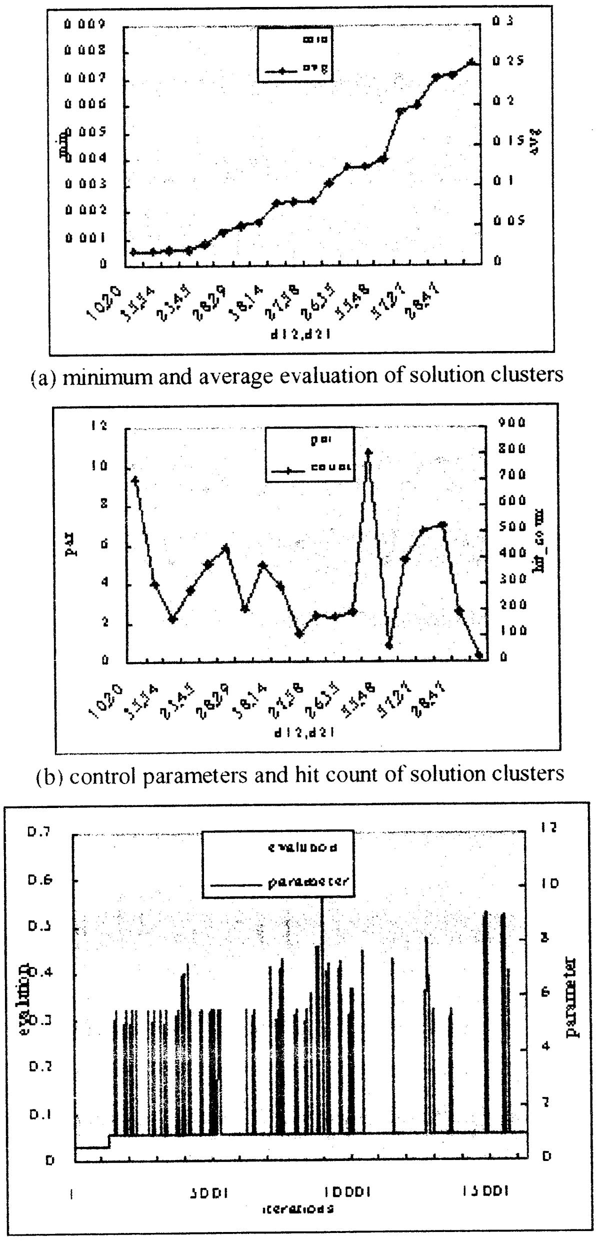
[1] J. Tani (1991), Proposal of Chaotic Steepest Descent Method for Neural Networks and Analysis of Their Dynamics, Trans. IEICE, Vol. J74-A, No. 8, pp. 1208-1215

[2] A. Bell and T. Sejnowski (1995), An information maximization approach to blind separation and blind deconvolution. Neural Computation, 7(6): pp. 1004-1034

[3] P. Common, C. Jutten, and J. Herault (1991), Blind separation of sources, part II: Problem Statement. Signal Processing, 24(1): pp. 11-20

[4] Torkkola, Kari (1996), Blind Separation of Convolved Sources Based on Information Maximization. Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, pp. 423-432

[5] S. Nara, P. Davis and H. Totsuji (1993): Memory Search using Complex Dynamics in a Recurrent Neural Network Model, Neural Networks, vol 6, pp. 963



(a) minimum and average evaluation of solution clusters

(b) control parameters and hit count of solution clusters

(c) iteration-parameter-evaluation relation

Figure 6. results of CSD monitoring

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

An Experimental Platform for Analyzing Interaction Between Human, Machine and Environment

○ Koji Morikawa* Natsuki Oka** Sameer Agarwal***

*Advanced Technology Research Laboratories, Matsushita Electric Industrial Co., Ltd.
3-4, Hikaridai, Seika, Souraku, Kyoto, 619-0237, Japan
morikawa@crl.mei.co.jp

** Information & Network Research Laboratory, Matsushita Research Institute Tokyo, Inc.
3-10-1 Higashimita, Tama-ku, Kawasaki 214-8501, Japan
oka@mrit.mei.co.jp

*** Department of Computer Science and Engineering, University of California, San Diego
La Jolla, California 92093-0114, USA
sagarwal@cs.ucsd.edu

Abstract

In cognitive science, developmental study becomes one of the essential areas to understand the human cognitive ability. Generally, there is a difficulty in collecting interaction data that is suitable for analysis. At first, we began by designing an experimental platform for collecting interaction data among a developing system (or a young child), its environment, and a caregiver. The collected interaction data is used for both interaction analysis and computer simulations of cognitive development. The platform is implemented as simple video games which both developing system and caregiver play with and has the following characteristics: (1) Sufficient platform for full deliberate research, and (2) Flexible experimental settings. We implement the platform in multi-modules, and each module can be executed in different computers connected with TCP/IP.

In this paper, we will report the outline of the designed experimental platform and discuss how to analyze and utilize the interaction data collected by our platform for cognitive developmental research.

Keywords: interaction analysis, cognitive development, platform, video game

1. Introduction

We have started research into building systems that develop like infants through interaction with their environment and caregivers. Although this is a long-range goal which relates to the essence of intelligence, we expect that we can build an easy-to-use human interface that adapts to its environment and users, and contributes to resolve the so-called digital divide as midterm production.

Recently some interesting ideas concerning development of cognition are proposed. For example, starting small [1], science of imitation[2], autopoetic system[3], prediction and planning[4], and so on. In addition, computational models of the cognitive ability are also proposed. Omori and Shimotomai [5] proposed a neural network model PATON in which an attention system controls the overall behavior of PATON. Roy [6] presented a system inspired by infant language learning, which learns from untranscribed speech and images. Sugita and Tani [7] presented a connectionist model developed for the linguistic communication between a robot and human.

These ideas are the keys for realizing the developing system. We especially consider the basic framework of cognitive development as the following (Figure 1): basically the system learns from rewards and the efficiency of learning is improved by imitation that we call *imitation-based reinforcement learning* (Figure 2). For example, the system learns how to act in the environment and how to communicate with caregivers from experience such as the system got *reward1* when the environment was in *state1*, *caregiver1* gave *utterance1* and the system took *action1* and gave

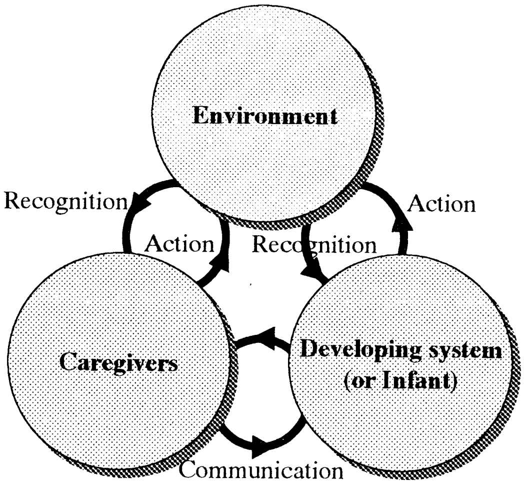


Figure 1. Interaction among a developing system (or an infant), its environment, and its caregivers.

uttereance2. In this framework the meaning of utterances reside in experience of this kind.

We believe that interaction among three constituents, that is, a developing system (or an infant), its environment, and its caregivers, is essential to development of cognition (Figure 1). Both the system and the caregivers can act in their environment, and can recognize the environment. The developing system and the caregivers communicate by language, gesture, and other signals.

For promoting this research, it is important to have an experimental platform, which consists of a minimum structure and is suitable for collecting data easily. At first, we tried to design an experimental platform for analyzing interaction between human, machine, and environment.

In this paper, we will report the outline of the designed experimental platform and discuss how to analyze and utilize the interaction data collected by our platform for cognitive developmental research.

2. An Experimental Platform for Cognitive Development

In this chapter, we will explain an experimental platform for collecting interaction data among a developing system (or a young child), its environment, and a caregiver (a user). The collected interaction data is used for both interaction analysis and computer simulations of cognitive development. The platform is implemented as a simple video game that both developing system and caregiver play with.

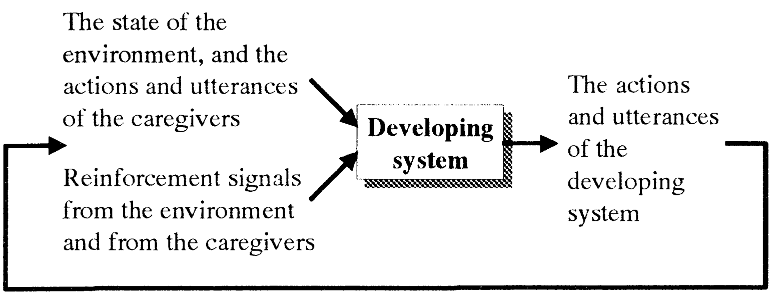

The state of the environment, and the actions and utterances of the caregivers
Reinforcement signals from the environment and from the caregivers
Developing system
The actions and utterances of the developing system

Figure 2. The basic framework of the developing system: imitation-based reinforcement learning. The system learns how to act in the environment and how to communicate with the caregivers by reinforcement learning whose candidates for actions are generated by imitation.

2.1 Task Configuration

We have designed an experimental platform for computer simulations of cognitive development (Figure 3 and Figure 4).

Figure 3 shows the configuration of the simple video game (we call this Video Game I). There are a ball, an infant's racket and a caregiver's racket. The ball moves straightforward and reflects on the rackets and walls. Each racket is controlled by the infant or the caregiver. When the ball hits the racket, a reward is given to the controller of the racket. Developing system (or infant) can control own racket, communicate with the caregiver, and watching the action of the caregiver. With this environment, a caregiver can teach how to

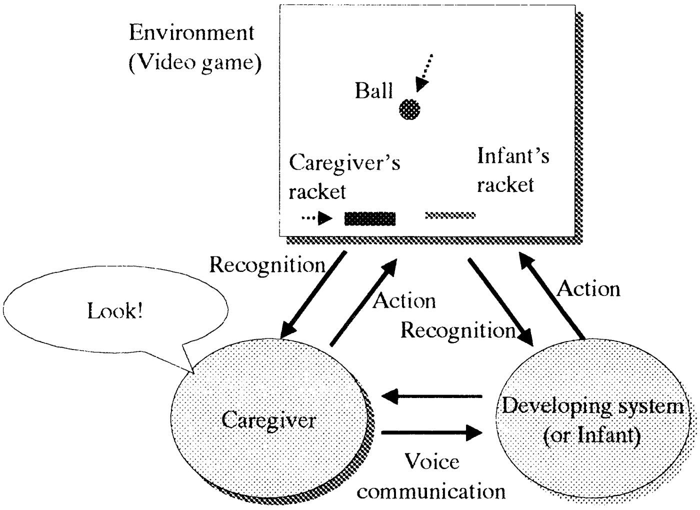


Figure 3. An experimental platform (Video Game I)

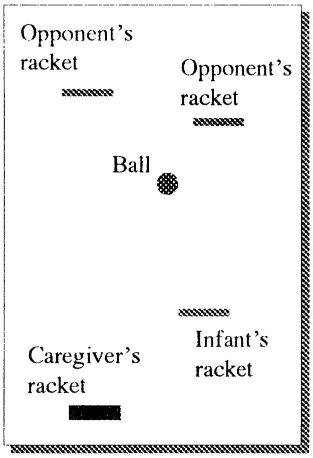


Figure 4. Another example of virtual environment. (Video Game II)

move by showing or by talking. Or a caregiver and infant play just for fun. We can analyze the interaction between parent and infant. The system can record all of the information produced by the environment, caregiver and developing system.

Figure 4 shows the extended platform of Video Game I (we call this Video Game II). The number of a ball is not changed, but the number of the players in the field is changed to four, there are two opponents players. In this setting, the relation between a caregiver and an infant is changed to win the game. Compare with Video Game I, the necessary knowledge for controlling the racket is increased remarkably. By preparing two games, we can collect different kind of data. The collected data can be used for verifying both the development algorithm of the system and the cognitive development of a infant.

2.2 Characteristics of the Platform

The characteristics of the platform are:

(1) Inclusion of the entire interaction among the three constituents shown in Figure 1.

The developing system learns both how to act in the environment and how to communicate with the caregiver, and hence it is necessary to include all the interaction among the three constituents.

The setting shown in Figure 1 also supplies the training data for multiple strategies of learning that include reinforcement learning, imitation learning, and learning by communication [8].

(2) Use of simple virtual environment, especially a video game.

The use of virtual environment makes the platform simple enough to execute with a personal computer. This simplicity and inexpensiveness are important because if we can use a large number of experimental platforms, we can gather a large amount of training data, which will be the key to success in complicated learning, that is, success in the simulation of development.

Simple virtual environment will also be useful in restricting the caregiver's vocabulary appropriately, and in making the recognition of environment practicable by the current technology.

Another merit of virtual environment is that we can freely design its complexity according to the stage of development. For example, we plan to use a two-player Video Game I as the first environment for the developing system, then a more complex four-player Video Game II as the second environment.

Both the caregiver and the developing system can move their own racket, and can see the positions of the rackets and the ball. This compatibility of the input/output signals is useful for imitation learning, and helps the caregiver to give appropriate advises. The developing system receives either the raw image of the game screen or preprocessed signals such as the position and the speed of the ball.

The use of a video game as environment is preferable because if it is fun to participate in the experiment, we can gather a large amount of interaction data, and because we can measure the degree of success by the score of the game.

If we use virtual environment, the developing system does not have any physical body. However, it interacts with the environment and the caregiver, and will learn how to act in the environment and communicate with the caregiver.

(3) Use of voice as the channels of communication between the developing system and the caregivers.

The developing system receives both the segmental feature and the prosodic feature. The former is a chain of speech sound, and the latter is the pitch, the power, and the length of sound. We believe that paralinguistic information, such as the intention of utterances or the control of dialogues, which is propagated through prosodic feature is essential for language acquisition.

The response of the developing system to a caregiver's utterance indicates that it is listening, and also indicates the level of understanding. Consequently, the response facilitates the utterances of the caregiver at the comprehension level of the system.

(4) Replaceability of the developing system with an infant.

2.3 System Design

This platform is flexibly designed. For collecting interaction data from human parent and child, the platform should be simple and portable in a single computer. For simulation with developing system, it should be extendable to multi computers for complicated calculation. We implement the platform in multi-modules, and each module can be executed in different computers.

For implementation to the PC(s), we introduced server-client model (Figure 5). The communication server (CS) takes charge of exchanging information between clients. The CS handles multiple channels. Each

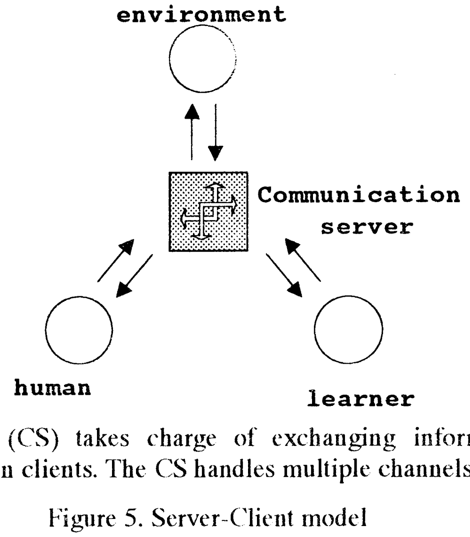


Figure 5. Server-Client model

channel has the list of clients that can share the information. With the CS, this simulation environment is very flexible in the number of clients and the number of PCs. The communication between clients and server is done by TCP/IP protocol.

In this Video Game I, there are one communication server and three clients. The clients are (a) game-server client which is responsible for doing all the computations related to the game, (b) A GUI client which we refer to as the Visual Client, and (c) a client for autonomous players which we call the Learning Client. We can execute each client on difference PCs, or one PC.

3. Collectable Data and its Analysis

Using this environment, we can simultaneously collect data shown below.
- Environmental data: the position of a ball, caregiver's racket and learning system's racket, current score, reward from the environment, etc.
- Racket control signal: control signal from the caregiver and the learning system.
- Voice data: current version records caregiver's voice directly. We can analyze the voice data after game ends. In future, we have a plan to combine the speech recognition and speech synthesis program. After that, we can analyze dialogue in real time.

The use of voice communication, enables the caregiver to give rich and timely information to the developing system, which includes:
(a) Evaluation such as "Good shot.",
(b) Alerting or pointing out the focus of attention such as "Look!",
(c) Instruction such as "Stop.", and more.

With collected data, we have a plan to do several experiments using the same environment. We plan to begin with collecting interaction data between a caregiver and a human infant, and analyze the data to find 1) what kind of modalities (or intentions) of utterances are included, and 2) by what clues the modalities can be estimated. We also plan to analyze interaction data between a caregiver and a learning system to verify and develop a learning algorithm.

4. Conclusions

We have designed an experimental platform for computer simulations of cognitive development. The platform was carefully designed to be sufficient for full deliberate research of cognitive development, however at the same time it is simple enough to execute with a personal computer.

According to our view of cognitive development, imitation-based reinforcement learning, the developing system is confronted with the extremely huge search space of learning. With the experimental platform we made, we can deal with the search space problem by gathering a large number of training data, and by limiting the huge space by a combination of several ideas.

In future, we have a plan to do several experiments using the same environment. We plan to begin with collecting interaction data between a caregiver and a human infant, and analyze the data to find 1) what kind of modalities (or intentions) of utterances are included, and 2) by what clues the modalities can be estimated. We also plan to analyze interaction data between a caregiver and a learning system to verify and develop the learning algorithm.

References

[1] Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A, Parisi, D., and Plunkett, K, *Rethinking Innateness: A Connectionist Perspective on Development,* The MIT Press, 1996.
[2] Kuniyoshi, Y., "The Science of Imitation: Towards Physically and Socially Grounded Intelligence", *RWC Technical Report (TR-94001),* Real World Computing Partnership, Tsukuba, Ibaraki, Japan, 1994.
[3] Maturana, H. R. and Varela, F. J., *Autopoiesis and Cognition: The Realization of the Living,* D. Reidel Publishing, 1980.
[4] Oka, N., "Conscious Level Processing by Dynamic Combination of Neural Network Modules", *Proceedings of the 2nd International Conference on Cognitive Science,* 1999, pp 195-200.
[5] Omori, T. and Shimotomai, T., "Brain Modeling of Word Acquisition Process Acceleration by Syntax Key", *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society,* 7(3), 2000, pp 223-235 (in Japanese).
[6] Roy, D. K., *Learning Words from Sights and Sounds: A Computational Model,* PhD Thesis, Massachusetts Institute of Technology, 1999.
[7] Sugita, Y. and Tani, J., "A Connectionist Model Which Unifies the Behavioral and the Linguistic Processes: Results from Robot Learning Experiments", *Sony CSL Tech. Report: SCSL-TR-00-001,* 2000.
[8] Oka, N. and Morikawa, K., "*Toward a System that develops through interaction with its environment and caregivers*", International Conference on Artificial Intelligence in Science and Technology, 2000 (to be printed)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Criticality of Cooperative Society

○Masao KUBO,Hiroshi SATOH,Akira NAMATAME
Dept. of Computer Science, National Defense Academy, Yokosuka Japan, 239-8686
masaok@nda.ac.jp, hsato@nda.ac.jp, nama@nda.ac.jp

Abstract Recently, there has been growing interest in the interactions among persons. It is needed to consider how a man behaves in the society where the number of people with whom he has to interact is varying. In this paper, we model the society with agents who play iterated prisoner's dilemma game and show that an appropriate interaction size is needed to emerge cooperative behaviors through computer simulations.

keywords prisoner's dilemma

Introduction

This paper examines interplay among human. Especially, we focus on relationship between the size of community and its expansibility and stability. It seems to be good to interact other persons as many as possible. However, there may be an appropriate size of interaction in some type of society. The purpose of this paper is to investigate the dynamics of the society through computer simulations.

We assume that a person behaves rationally and his behavior is subject to payoff that he will get. The iterated prisoner's dilemma is the model that is used to study these situations. In two person iterated prisoner's dilemma (2IPD) game, two players may select cooperation(C) and defect(D). Table 1 shows the payoff matrix of 2IPD. This is a non zero-sum game and players do not know when they will finish the iterations of the game.

The other's strategy Own's strategy	S1 (Cooperate)	S2 (Defect)
S1 (Cooperate)	R R	S T
S2 (Defect)	T S	P P

Table 1 Payoff matrix of the 2IPD (T >R >P > S, 2R>T + S)

If this game is played only once, "D" is the optimal behavior. However, if the game is played many times, "D" is not the optimum any more. There are many researches in various environments [1] [2]. Ordinarily, a person interacts with many persons. There is an approach that uses N-players IPD (NIPD) for interactions among many person [3]. Table 2 shows the payoff matrix of NIPD. NIPD deals with the game (for example, congress) in which N players interact all at once, then the payoff is subject to the number of agents who choose "C". Yao et. al. state that the cooperation rate is inversely proportional to the number of the players [3].

No.of Cooperators	*0*	*1*	・・・*X*	・・・*N*
S1 (Cooperate)	C_0	C_1	・・・C_x	・・・C_N
S2 (Defect)	D_0	D_1	・・・D_x	・・・D_N

Table 2 (1)Dx>Cx for 0≦x≦N,(2) Dx+1>Dx and Cx+1>Cx for 0≦x<N(3)Cx>(Dx+Cx-1)/2 for0<x≦N

In contrast, we extend 2IPD for interactions among many person and adopt the payoff matrix given by Table 1 in this paper, because we are interested in different situation in which players interact individually (one to one interactions).

In addition to using the payoff matrix on Table 1, we restrict the interactions with localization; agents play the game only with their neighborhood. The size of neighborhood *f* is very important, since the spreading speed of information is depend on the size. If *f* is small, two agents who are widely separated are almost independent, on the other hand, if *f* is large, the effects of an agent's behavior is far and wide. Although every person has its unique neighborhood size, we assume all agents have the same *f* value because we have to seek the appropriate size of *f* as first step.

The speed of producing a new strategy is also important. Agents change their strategies based on the behaviors of other agents as human does. We adopt distributed genetic algorithm way to produce new strategies[5].

The remainder of this paper is presented as follows. In section 2, we define players and society; representation of strategies, mechanism of changing behaviors and measuring cooperation rate. Section 3 shows some empirical results and discuss them. Computer simulations are done in several size of *f* (2, 4, 8, 16, 24, 40, 80). The results show a dramatic change of dynamics at some *f* value. This is different from other existing research results.

The Society

In this section, we firstly define the strategies of players. The strategies are represented as if--then rules. Secondly, payoff matrix, measurement of cooperation level and neighborhood structure are described.

Encoding Strategy for the NIPD Game

The players of IPD are defined as [4]. They can select cooperate (C) or defect (D) per a bout and repeat this game over again. The players choose their next move taking into account the series of previous games.

Let us consider a(t) as the action of i th agent at time t; $a_i(t) \in \{0$ (Cooperate); 1 (Defect) $\}$, then the strategy is represented as a rule set. For the NIPD game remembering *n* previous step, there are 2^{2n} combinations of possible histories. Therefore, at least 2^{2n} bits are needed to represent a strategy. Besides, agents need more 2n bits for an early stage of iterations in which the size of their history is too short to use their rule. Thus, we represent the strategy as 2^{2n+n} bits. Let us consider s_{ik}, where $k \in \{1,..., 2^{2n+n}\}$, as the strategy of i th agent.

In this paper, an agent interacts with their neighborhoods and each pair of agents plays the games 20 times. The size of history is two (n = 2). For example, if the history of the game is ((C, D), (C, C)), then agent i choose "C" according to the condition,0100.

Payoff Matrix

We model the N-player's interactions as 2IPD with N-players, so the profit of an agent is sum of each 2IPD games. The payoff matrix of 2IPD is shown by Table 1. Prisoner's dilemma is subject to following two conditions; (1) T > R > P > S, (2) 2R > T + S. In our experiments, (R; S; T; P) = (3; 0; 5; 1).

Let us denote the profit of an agent i who play games with agent j at time t as p(ai(t),aj(t)). The payoff of τ times iteration with same agents is

$$\sum_{k=t-\tau}^{t} p(a_i(k), a_j(k)),$$

and the sum of the payoff which is gathered from the interactions with *f* neighborhoods is

$$\sum_{j}^{f}\sum_{k}^{t} p(a_i(k), a_j(k))$$

Next, we define the cooperation rate of population. This is the ratio of cooperate actions to all actions.

Assume that agent i has played the game b times at moment t, the average cooperation rate $r_i(t)$ is

$$r_i(t) = \frac{(b - \Sigma_{k-1}^{h} a_i(t-k))}{b}$$

In this paper, we set $b = f\tau$. The cooperation rate of entire population is written as follows,

$$R(t) = \frac{\Sigma_i^N r_i(t)}{N}$$

Localization and Player's Definition

We now define the neighborhood structure. There are a number of research that deal with neighborhood structure in IPD [2] [3]. For example, ring, Moore, Gaylord and so on. In this paper, agents are settled at the each grid of a plane. Neighborhood structure is defined considering the symmetry because the speed of spreading information should be uniform. Figure 1 shows some neighborhood structures up to *f*=16.

For larger size, neighborhood structures are given by a quadratic terrian,*d*. In this paper, *f*=24, 48, 80 using *d*=2,3,4, respectively.

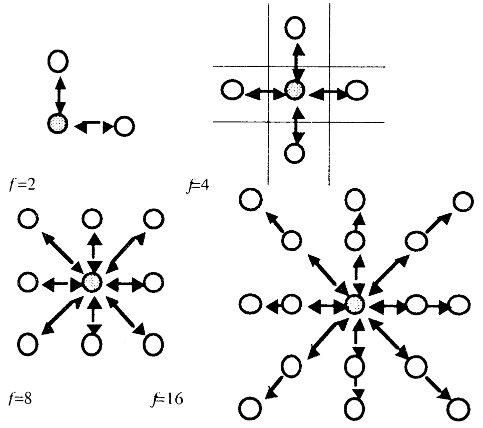


Figure 1

Learning Strategies of the Players

Learning ability is one of the most important issues for agents who wish to adapt dynamically to changing society. A person learns behavior of his friends who go about with well. Also, he take care not to make same mistakes which his friends make. We refer these examples as one of the learning by imitation. Uno proposed two learning strategies inspired from the imitation[6]. He named his strategies Complete Mimicry strategy (= copy) and Mutant strategy (= crossover) respectively. Both procedures were examined but the following experiments used only the mutant strategy only. Therefore, the procedure of the mutant strategy for player *i* is described

Strategy Decision Part

C=0 D=1

	t=1,t=2		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
positio n	1	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
stik	C	C	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D

Fig.3 Example of Player i(TFT)

as follows.

(Step1) Select a player *p* who scores highest profit p(t) than other its competitors.

(Step2) Apply one-point crossover operation to s_{ik} and s_{pk}. Set one of the new offsprings to its new string candidate.

(Step3) Apply mutation operation to the candidate s_{ik} Each s_{ik} is examined whether the operation is applied or not. Note that the probability per bit is called as mutation rate. If the operation is executed, the value is flipped.

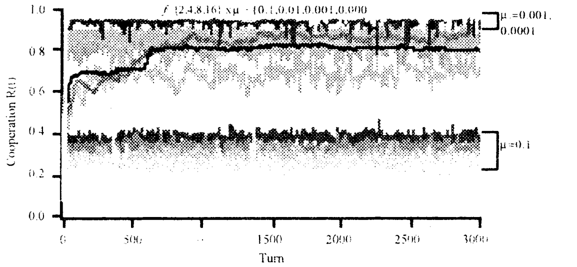
(Step4) p(t) = 0.0 r_i(t) = 0.0

In the following experiments, the mutation rate is one of 0.1, 0.01, 0.001 and 0.0001.

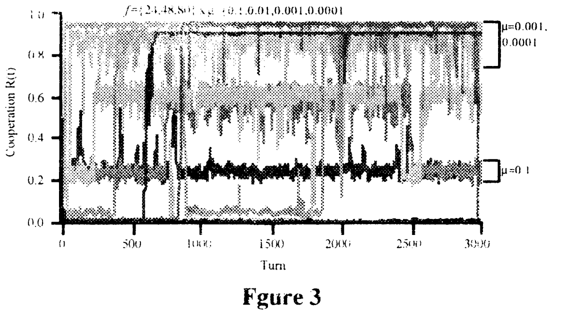
Simulation Results

We set the grid width to 20 and the number of the players to 400. The parameter *f* and the mutation rate of all players were set equally. Their initial strategies were defined randomly and each run was executed as the following sequence that we call a turn. Firstly every player competes with its neighborhoods. Secondly, all players update their strategies. This sequence, a turn, was repeated 3,000 times. The 48 societies were examined (*f*={2, 4, 8, 16, 24, 48, 80} x mutation rate μ = {0.1, 0.01, 0.001, 0.0001}).

Figure 2, 3 how the results. The y-axes show the turn, the x-axes indicate the cooperation rate, R(t). Figure 2 illustrates progress of R(t) of small *f* societies. Figure 3 and 4 present the societies with *f* = {24, 48, 80}. These graphs suggest the following remarks.



Fgure 2



Fgure 3

The society which mutation rate is high will not reach high cooperative state. In figure 2, several graphs exist from 0.2 to 0.4. It means that it was difficult for a society which *f* is smaller than 24 and its mutation rate was high, for example, μ = 0.1, to shift to cooperative state (over 0.8). Among this group, the more the *f* enlarged, the higher cooperation it achieved.

The societies with mutation rate 0.001 and 0.0001 tended to keep high cooperative state. Figure 2 illustrates the progress of the cooperative societies with *f* = {2, 4, 8, 16}. They scored higher than the last group. Especially, among the group, the 2 societies achieved higher than the others did. The one is the society with *f* = 24, μ = 0.01. The *f* and mutation rate of the other is 80, 0.001, respectively. However, the figure manifests that these were not stable consistently and their cooperation rates were fluctuated from 0.92 to 0.8. The fluctuation of the society with *f* = 24 is small as same as the societies with *f* = {8, 16}. On the other hand, magnitude of the fluctuation of the society with *f* = 80 was larger than the others shown by figure 4, in spite that this is one of the 2 highest cooperative societies. It expressly provides that the society needed much time until it reached cooperative state and the state was broken down. On the other runs which we do not show because the space limitations, this kind of breakdown phenomena was observed frequently. Also, the cost was sometimes larger, sometimes smaller.

Therefore, we conceived that it needed to compare the number of the breakdown and the cooperation rate for classification of societies. Hence, we plotted 2-tuple, variance and average of R(t). The x-axes and y-axes of figure 5, 6, and 7 mean the average of R(t), the variance of R(t), respectively. Note that there is a difference in meaning among these graphs. In figure 5 and 6, points that have same *f* were connected. For example, a line was drawn between the point (*f* = 4 and μ =0.01) and the point (*f* = 4 and μ = 0.001). On the other hand, in figure 7, points which has same μ value were connected. Moreover, take care that the ranges of y-axes between figure 5 and 6 are different. Figure 6 is a macrograph of figure 5 and it ranges within smaller interval. Therefore, in figure 6, the graphs of societies with *f* that is over 16 were vanished. Using by this analysis, the following results are highlighted.

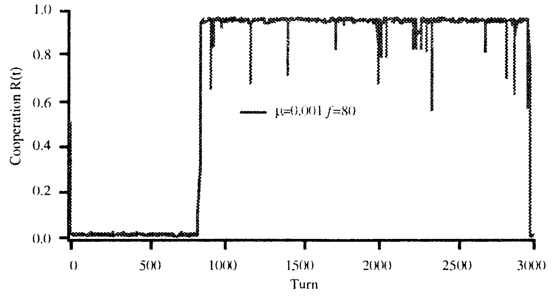


Figure 4

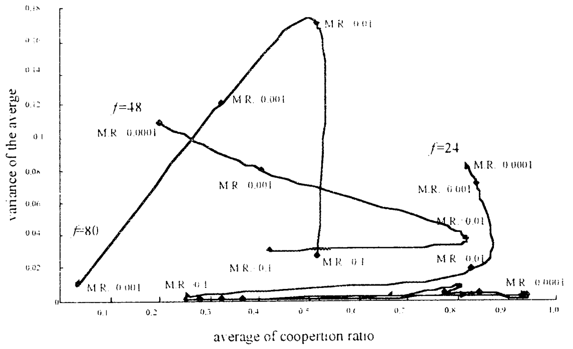


Figure 5

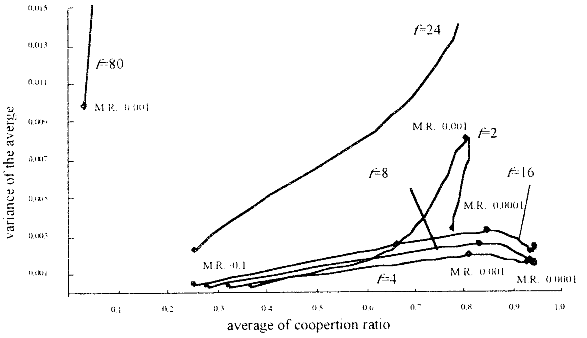


Figure 6

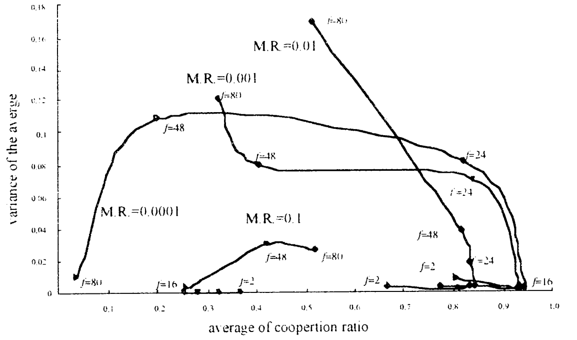


Figure 7

When$f$ is very small (e.g. $f = 2$), the cooperation rate is not improved.

If the$f$ is gradually increased ($f = 4, 8$), the cooperation rate is improved. Within this range of$f$, the smaller the mutation rate is, the smaller the variation is.

If the f enlarges over the level described at second remark, the variation will be high suddenly ($f = 16, 24$). In figure 5, the variation at$f = 16$ and $\mu = 0.0001$ is larger than that of $\mu = 0.001$. Also, the variation of $f = 24$ in figure 8 is suddenly enlarged at $\mu = 0.0001$.

Mutation rates that could produce cooperative society before can not work, if f is larger than the level denoted by the last remark. The cooperative rate is quickly converged to 0.0. Of course, if the mutation rate is too small and$f$is huge, the initial state of society makes big influence to its state of future.

These experiments suggest that there is a critical neighborhood size$f$ and critical mutation rate μ to emerge cooperative society. According to the traditional approach, the larger$f$ is, the more difficult a society reaches a cooperative state.

Conclusions

To study about the interaction among persons, we considered how a man behaves in the society where the number of people with whom he has to interact is varying. In this paper, we model the society with agents who play 2-player iterated prisoner's dilemma game with his n-players around him. From several computer simulations, we showed that an appropriate interaction size and an appropriate learning speed are needed to emerge cooperative behaviors.

References

[1] Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)

[2] Epstein, J. and Axtell, R.: Growing Artificial Societies: Social Science from the Bottom Up. Brookings institutions (1996)

[3] Yao, X. and Darwen, P. "The experimental study of N-player iterated prisoner's dilemma," Informatica, Vol.18, (1994) pp.435-450

[4] Lindgren, K.: Evolutionary phenomena in Simulating simple Dynamics. Artificial Life II (1991) 295-311

[5]Hansaryi, J. and Selten, R. (1988) "A Game Theory of Equilibrium Selection in Games," MIT Press.

[6] Uno, K. and Namatame, A.: An Evolution Design of the Networks of Mutual Reliability. Proceedings of Congress on Evolutionary Computation (1999) 1717-1723

Real Time Gaze Control of Active Head-eye System without Calibration

Do-yoon Kim*, Jung Rae Ryoo, Hyun-keun Park and Myung Jin Chung**
KAIST EECS
**nice@cheonji.kaist.ac.kr and **mjchung@ee.kaist.ac.kr*

Abstract - This paper describes the real time gaze control using position-based visual servoing. To solve the calibration problem in the position-based approach, we construct the end point closed-loop system using active head-eye system. The main control objective is to make a gaze point to track the target and the image feature of the target should be located at each image center. As cameras are not calibrated, the reconstructed positions are not correct. As the reconstructed position error is used with Jacobian matrix of kinematic relation, though, the system stability is locally guaranteed. The proposed approach is successfully applied to the tracking task of the moving target in some simulation and real experiments. And the processing speed is satisfied with real time.

Key words *– Active Head-eye system, Position-based Visual Servoing, Adaptive Look-up table, Gaze Control*

1. Introduction

Using sensor information to control robots has become a very popular field of robotics research, since it promises to lead to the design of intelligent autonomous robots. In contrast to their preprogrammed industrial counterparts, intelligent autonomous robots must be able to deal with unexpected events such as obstacles or misplaced objects. Consequently the importance of sensor that obtains the state of the external world is more emphasized. Nowadays, vision is the most commonly used sensor in robotics because CCD cameras offer rich information than any other sensors.

In the field of visually controlled robots, two strategies have been proposed [1]. In image-based systems the distance from current pose to final pose is defined in image space. The great advantage of the image-based systems is that the accuracy can be made relatively insensitive to calibration errors. On the other hands position-based systems try to determine the object's pose from the visual input as accurately as possible and then move the robot appropriately. This approach lends itself very well to integration in manufacturing. Unfortunately, its accuracy heavily depends on the accuracy of the sensor/robot calibration and of the sensor itself because the object's pose is described in Cartesian space. The defect of the position-based systems is overcome by endpoint closed-loop systems (ECL) that observes both the target position and the current position [1].

In this paper we construct ECL system using an active head-eye system with stereo camera controlled to gaze a target with position-based visual servo algorithm. There are lots of advantages that can be achieved from introducing the active vision system. First, as the target object is located at the center of each image, disparity error can be minimized and extraction of 3D information becomes easier. Second, image distortion from camera lens is also suppressed. Third, it is much easier to recognize an object because it remains at the same location on each image plane. And forth, ECL systems can easily be implemented by active head-eye system because it can continuously track the moving target position.

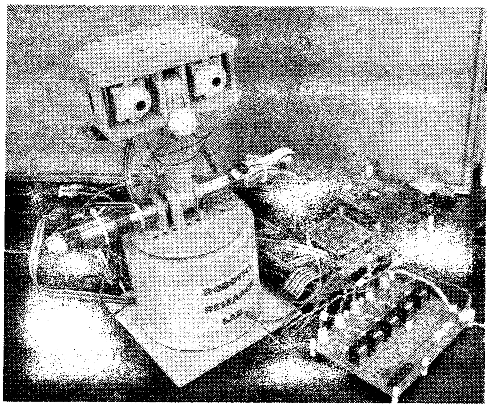


Figure 1. The implemented Active Head-eye System

For the purpose of utilizing these advantages, an active head-eye system is made like Fig. 1. Each camera can rotate independently by vergence control and the head can rotate in the vertical(tilt) and horizontal(pan) direction. In addition, the head can move back and forth. The entire structure is a redundant joint mechanism when the gaze is controlled to be located at a 3D point.

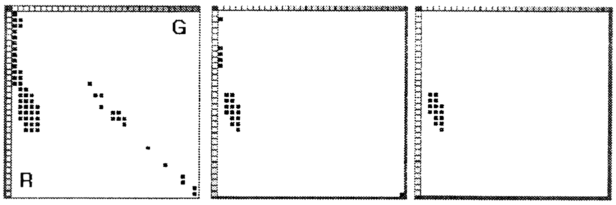
This paper is organized as follows: In section 2, in order to achieve the position of target independent illumination condition, we proposed a adaptive look-up table approach having two advantages of both speed as well as robustness in RGB space. In section 3, the 3D reconstruction method in our active head-eye system is introduced. In section 4, based on the reconstructed position, the position-based visual servoing of the active head-eye system controlled to gaze the estimated target position is shown. In section 5 simulations and experiments show the feasibility of the proposed method. Finally we conclude in section 6.

2. Extraction of Feature Point using Color

The color feature is broadly used in object tracking because the color is one of the intrinsic properties characterizing the object. The most natural way to represent the colors is expression by the (Red, Green, Blue) triplet. However, the condition of illumination or the shape of object affects the color. This dependence about various conditions causes many problems when conventional color threshold in RGB space is executed. This approach does not represent intrinsic colors in the best way, since the chromatic properties of the spectrum are not separated form the non-chromatic ones, such as brightness in colors. A number of representations have been developed to differentiate the intensity within colors. Normalized RGB and HIS color models are good examples. But if we use these models we must also put up with decreasing the acquisition speed of the information to the position of target, since these are obtained by a complex computation. On the other hand, the Look-up Table method that represents the color distribution of the object with the table in RGB space can not only separate object color and background color but also it is fast because only simple comparison is used without complex computation [3]. However, although LUT method has many advantages, the alternation of color distribution as the movement of object remains as problem. To solve the problem, we use an adaptive Look-up table method. The adaptive LUT method changes the configuration of LUT depending on color distribution of object at each sampling time.

The algorithm of adaptive look-up table is as follows:

1. Manually set look-up table at initialize stage.
2. Separate object and Background through LUT
3. Calculating minimum rectangular size within the object in image space
4. The bigger color values than the predefined value are remained in RGB space and others are eliminated.
5. Grouping as two classes. One is the object color group and the other is the background color group.
6. Find the group near previous LUT that is the object color group in two groups.
7. Change the LUT with the selected group.
8. Repeat from step 2 to 7 whenever image is updated.



(a) Step 2 (b) Step 5 (c) Step 6

Figure 2. The process of updating LUT is depicted. (a) shows all pixels through LUT in step 2. (b) is pixels bigger than 0.25% and grouping with two classes. (c) is selection of near object color between two groups

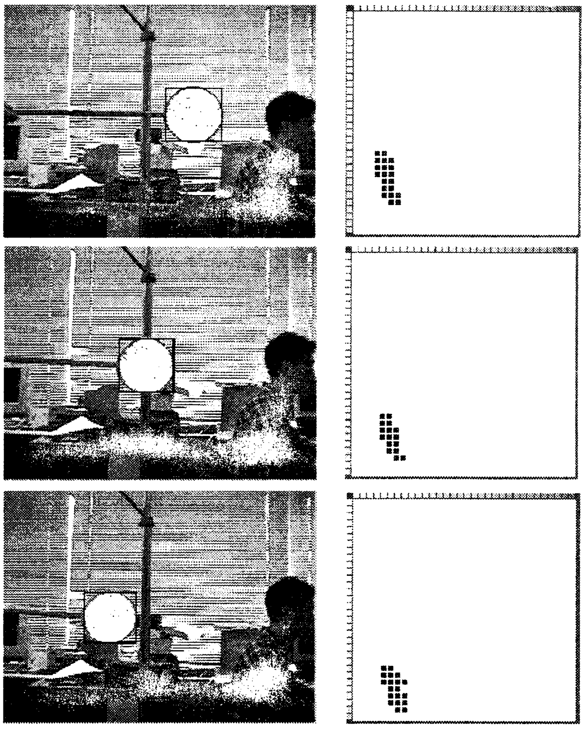


Figure 3. The left side is test image and the right side is the final updated LUT at each step. LUT is change the configuration according to the position of target.

In fig. 2, the changes of LUT according to the flow of algorithm are depicted. Fig. 3 shows that the proposed adaptive LUT method works well in vary illumination condition. The target used in experiment is the circular color patch and the pixels finding with LUT are displayed by white color. Although the left side is brighter than the right side, the change of the color distribution according to move the object is well represented by the Adaptive LUT method.

Using the adaptive LUT, we can robustly find the feature in the image rather than the previous LUT method.

3. 3D Reconstruction from Image Feature

A schematic diagram of the active head-eye system controlled to gaze a target is depicted in fig 4. In order to estimate the 3D position of target and gaze point from stereo image, a pin-hole camera model is defined with nominal camera parameters. Camera models with internal and external parameters merged into one matrix are written as follows:

$$U_l = M_l(\theta)\cdot X \quad \text{and} \quad U_r = M_r(\theta)\cdot X \text{ , (1)}$$

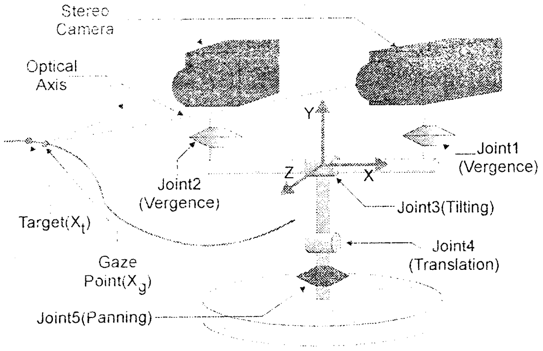


Figure 4. Coordination of Active Head-eye System

respectively, where $U_l = [u_l, v_l, 1]^T$ and $U_r = [u_r, v_r, 1]^T$ are image coordinates, subscript l and r represent each left and right image. $X = [x, y, z, 1]^T$ is defined in 3D coordinates, $M_l(\theta)$ and $M_r(\theta)$ are camera projection matrix, and θ is the joint angle of the active head-eye system. We assume that the stereo camera is not calibrated and only nominal values in the specification of camera are known. Eq. (1) is reformed as follows:

$$U = H(\theta) \cdot X \quad , \tag{2}$$

where $H(\theta)$ is 4×3 transformation matrix form world coordinate to stereo image coordinate and $U = [U_l, U_r]^T$. As Eq. (2) is an overconstrained problem, pseudo-inverse of $H(\theta)$ is utilized to get X from U. Therefore, 3D reconstructed position is estimated as follows,

$$X = H^+(\theta) \cdot U, \tag{3}$$

where $H^+(\theta)$ is a pseudo-inverse of $H(\theta)$. When a point in 3D space is projected to stereo camera, image coordinate of each camera is uniquely determined. On the other hand, not all the image coordinate pairs are reconstructed into 3D coordinates because of the epipolar geometry of stereo vision [7], whereby there are only three independent variables among four elements of the column vector U. Because we use pseudo-inverse, when there is no 3D point corresponding to a given image coordinate pair, eq. (3) returns the best point that minimizes the error [5].

4. Position-based Visual Servo Control

Based on the reconstructed position, the active head-eye system is controlled to gaze the estimated target position. The overall control structure is presented in Fig 4. As the active head-eye system has 5 DOF and gazing orientation is out of consideration, it is categorized as a

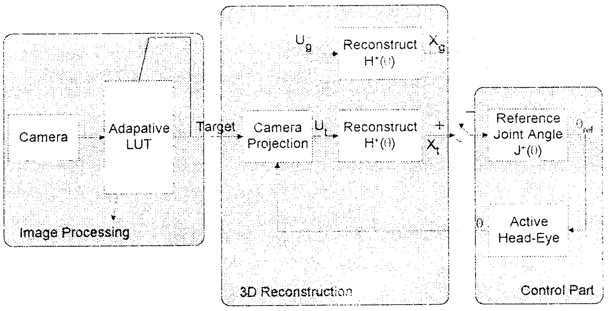


Figure 5. The structure of whole system

redundant joint mechanism and Jacobian is a solution to resolve the redundancy [5]. From fig. 4, the joint angle θ is represented in terms of the position error, such that

$$\dot{\theta} = K_p J^+(\theta) \cdot (X_t - X_{gaze}) \tag{4}$$

where $J^+(\theta)$ is the pseudo-inverse of Jacobian matrix, X_t is a reconstructed target position, X_{gaze} is a gaze position, and K_p is an appropriate gain. Therefore, if the overall stability of the system is ensured and the gaze point enters a steady state, *i.e.* $\dot{\theta} = 0$, X_{gaze} is ensured to be equal to X_t. According to the Fig. 3, the current gaze point X_{gaze} and the estimated target point X_t are generated from the reconstruction routine as follows,

$$X_{gaze} = H^+(\theta) \cdot U_{gaze} \tag{5}$$

and

$$X_t = H^+(\theta) \cdot U_t \quad , \tag{6}$$

respectively, where U_{gaze} is image center coordinate pair and U_t is target image coordinate pair.

5. Experimental Setup and Results

5.1 Simulation

To show that the position-based gaze control is converge though the calibration error, simulation was performed with the active head-eye system model. To affect the incorrect calibration parameters, the reconstruction routine is programmed with different values from those of camera projection routine. The nominal internal parameters and different parameters that intentionally included error are shown in Table 1. The simulation result is depicted in Fig. 6 and Fig 7. As there is the calibration error in 3D reconstruction part, the trajectory from initial position to target position is not optimum, which is straight line. However, the gaze point finally converges to target through a lot of steps because the gaze controller goes toward error minimization at each step.

	Image Center	CCD Size[mm^2]	Focal Length	Angle Error
Nominal	(240,240)	4.3×3.8	12mm	0°
Simulation in fig 6	(235, 245)	2.0×1.8	10mm	±5°

Table 1. Simulation Parameters

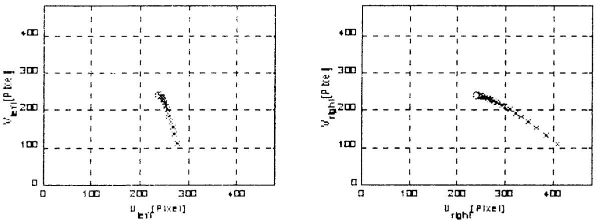


Figure 6. Simulation result with camera parameter's error

5.2 Applying to Real Active Head-Eye System

An target moves in 3D space in a circular manner about 1.5m in front of the system and the gaze of the active head-eye system is controlled to track the target. An experimental result is shown in Fig. 8. As shown in the figure, the gaze tracks the target well. A numerical mean-squared error of the result is 13.5 cm. The execution time table is represented in Fig 9. The required time for the gaze control is 14ms at Pentium III-450Mhz. Generally the required time at image processing is very large. However the proposed adaptive LUT robustly and fast extract the image feature against illumination condition. Allowing that the speed of NTSC signal is 30Hz, the proposed method is expected to work as real time.

6. Conclusion

In this paper, we have presented an active head-eye control scheme based on position-based visual servo control without camera calibration. First of all, in order to control real time we adopt LUT. The fast and robust extraction of image feature is important to real time tracking. The proposed adaptive LUT is effective to measure the position of image feature at vary illumination condition.

And then we controlled gaze position using position-based. Notwithstanding the reconstructed positions of gaze point and target are corrupted by incorrect calibration parameters, the projected target point is located at the center of image plane. If the overall stability is ensured, the steady state error of the position is diminished negligibly for the effect of integrator introduced in control loop. In addition, the target feature can be controlled to be located at an arbitrary point, which should be a feasible coordinate pair of image.

The control considering orientation of object remains as the further work.

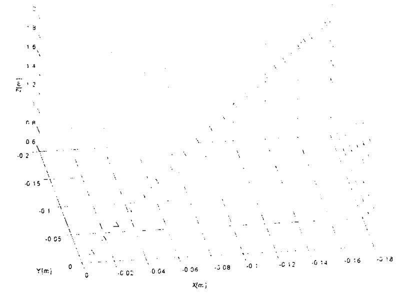


Figure 7. The trajectory of gaze point in Cartesian Coordinate

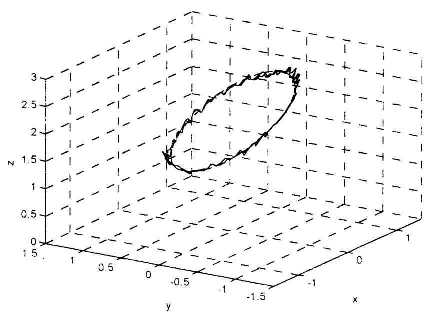


Figure 8. 3D view of experimental result (dashed: target, solid: gaze point)

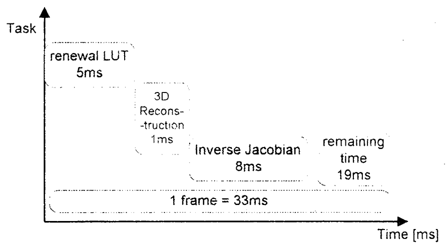


Figure 9. Time schedule of gaze control

References

[1] S. Hutchinson, G. D. Harger and P. I. Corke (1996) A Tutorial on Visual Servo Control, *IEEE Trans. On Robotics and Automation,* Vol.12, No. 5. pp.651-670

[2] O. Faugeras (1993), Three-Dimensional Computer Vision, *MIT Press*

[3] D.Y Kim, H. K Park, and M. J. Chung (1999), "Robust and Fast Color-Detection using a Look-Up Talbe," *Proc. Of Int. Symposium on Artificial Life and Robots*, Vol. 2, pp.650-653

[4] J. R. Ryoo, D.Y. Kim and M. J. Chung (2000), An On-line Trajectory Generation for Control of Active Head-Eye System, *Proceedings of ASCC*

[5] T. Yoshikawa (1990), Foundation of Robotics Analysis and Control, Cambridge, MIT Press

[6] T. F. Chan and R. V. Dubey (1995), A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators", *IEEE Trans. On Robot and Automation*, Vol. 11, No. 2, pp.286-292

[7] G. Xu and Z. Zhang (1996), Epipolar Geometry in Stereo, *Motion and Object Recognition*, Kluwer Academic

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Fast and Stable Learning in Direct-Vision-Based Reinforcement Learning

Katsunari Shibata*, Masanori Sugisaka* & Koji Ito**

**Dept. of Electrical & Electronics Engineering, Oita University, 700 Dannoharu, Oita 870-1192, Japan.
***Dept. of Computational Intelligence & Systems Science, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan. shibata@cc.oita-u.ac.jp

Abstract

Direct-Vision-Based Reinforcement Learning has been proposed not only for the motion planning but for the learning of the whole process from sensors to motors in robots, including recognition, attention and so on. In this learning, raw visual sensory signals are put into a layered neural network directly, and the network is trained by the training signals generated based on reinforcement learning. On the other hand, it has been pointed out that the combination of neural network and TD-type reinforcement learning sometimes leads to instability of learning.

In this paper, it is shown that each visual sensory cell makes a role of localization of our continuous 3-dimensional space and it helps the learning to be fast and stable. Further by processing the localized input signals in the layered neural network, a global representation is reconstructed adaptively in the hidden layer through learning as shown in the previous papers.

1 Introduction

Reinforcement learning has been focused recently for its autonomous and adaptive learning ability. However, in general, only the mapping from the state space to the action space is formed by reinforcement learning. The state space is generated from sensory signals through some pre-processings, and is usually designed by a designer. Some of the authors have proposed Direct-Vision-Based Reinforcement Leaning [1]. In this learning, raw visual sensory signals are put into a neural network directly without any pre-processings and the outputs of the network are utilized as the motion commands. By this, the continuous and adaptive state space is formed in the hidden layer through learning. Furthermore, the reinforcement learning is extended to the total learning for the whole process from sensors to motors, including recognition, memory, sensory integration, and so on[2] [3].

Layered neural networks have been already utilized widely in TD(Temporal Difference)-type reinforcement learning [4][5][6]. However, Boyan et al. has been pointed out that the combination of TD-type reinforcement learning and layered neural network is not robust and may produce an entirely wrong policy[7]. The problem occurs when the discontinuous motion or value function is required such as "2-D puddle world" and "car-on-the-hill" task in [7]. The reason might be that the sigmoid function in the neural network is a smooth and monotonically incremental function, in other words, a semi-linear function and has weak non-linearity.

For this problem, it has been shown that localization of the continuous input space such as CMAC, k-nearest-neightbor and RBF (Radial Basis Function) is effective [8][9][10]. However, some of the authors have pointed out that since they are close to the table-look-up approach and have no hidden units, they cannot reconstruct the adaptive continuous state space from localized signals[11]. This means that whenever a robot with RBF or CMAC learns a new task, it cannot utilize the knowledge obtained from the learning of the previous tasks and has to learn from scratch. Therefore it is not suitable as a brain of intelligent robots.

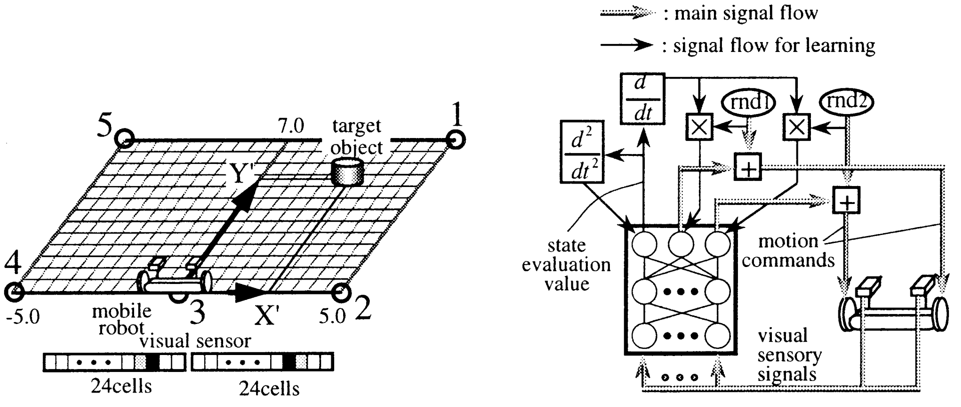
Some of the authors have proposed Gauss-Sigmoid neural network[11]. In this network, continuous input signals are put into a RBF network at first, and then the outputs of the RBF network are put into a sigmoid-based neural network. This network has the both advantages of the RBF-based network and the sigmoid-based neural network. In this paper, it is shown that each visual sensory cell makes a role of localization of our 3-dimensional space like the RBF unit in Gauss-Sigmoid neural network, and it helps the learning to be fast and stable.

2 Localization and Visual Sensor

2.1 Localization

Localization means to represent the spatial information using multiple signals each of which mainly shows the information of a local area in the whole continuous space. All of table-look-up, RBF, and CMAC are the localization-based function approximators.

By using local signals, the learning effects only at the local area, and strong non-linear functions from the continuous input space can be approximated easily as the sum of the weighted local signals. So as to realize better approximation, the continuous space has to be localized more finely and many localized signals are required. However, there is a dilemma that the



(a) simulation environment

(b) signal flow

Figure 1: (a)Environment of the basic task. A mobile robot with two wheels has two visual sensors and it can obtain a reward when it arrives at the target. (b)Signal flow of the learning. Visual sensory signals are put into a layered neural network directly, and the neural network is trained by the reinforcement learning based on Temporal Smoothing Learning.

generalization ability becomes less and less when the number of localized signals more. That is the same as table-look-up approach.

2.2 Visual Sensor and Receptive Field

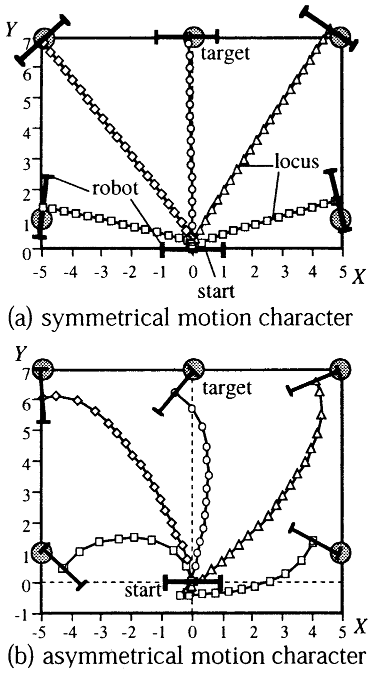
A visual sensor usually consists of many visual cells. Each visual cell has a local receptive field, and catches only a part of the whole visual field. It can be said that the visual sensor makes a role of localization of the continuous 3-dimensional space where we live.

So it can be thought that when the visual sensory signals are put into a neural network, the learning is faster than when the continuous spatial signals, such as object location in the sensor, is put into the neural network. Further, in order to generate the continuous spatial signals from the localized signals, pre-processing is necessary. The pre-processing is usually designed by a designer, and it is difficult to modify the pre-processing flexibly through the learning. It has been already shown that the localized visual signals can be integrated in the hidden layer of the neural network very adaptively through reinforcement learning according to the given task and the motion character of the robot[2].

3 Comparison of Learning Speed and Stability

3.1 Task

Here let's consider the task that a mobile robot with two wheels and two visual sensors obtains an target object as shown in Fig.1(a). Each visual sensor has 24 visual cells arranged in a row. The receptive field of each sensory cell spread out radially, and the sensor has a total of 180 degree of visual field. The robot obtains a reward only when it reaches the target object. Concretely, when the center of the target goes through the robot, the state evaluation output is trained to be 0.4. When the robot misses the target, i.e., the target disappears out of the visual field, it is trained to be -0.4 as a penalty. The diameter of the target is 1.0 and length of the robot is 2.0. Figure 1 (b) shows the signal flow of this simulation. Before learning, the input-hidden connection weights are small random numbers, and all the hidden-output connection weights are 0.0. Accordingly all the outputs of the neural network are



(a) symmetrical motion character

(b) asymmetrical motion character

Figure 2: The robot loci after learning.

0.0 before learning. The number of outputs for motion signals is two, and the robot rotates its wheels according to the outputs. In the early stage, since the robot moves only by random numbers, the target is located within the range that is close to the robot. According to the progress of the learning, the range of the target location becomes wider gradually until $-5 \leq x \leq 5, 0 \leq y \leq 7$. After the robot reaches or misses the target, the target is located at another place chosen randomly. One trial indicates a sequence from the initial state to the target state. Fig. 2(a) shows the robot loci after learning. It is seen that the robot rotates until it catches the target around the center of the visual sensor and then it moves forward.

3.2 Integration of Visual Sensory Signals

Here, at first, it is introduced that the neural network has an ability to represent the spatial information in the hidden layer by integrating the localized visual signals[1]. When the 5-layered sand-glass neural network with two hidden units in the middle layer as shown in Fig. 3 is utilized in the reinforcement learning, the representation of the target location in the middle layer is as shown in Fig. 4(b). It can be noticed that the two neurons represents the target location that is the global and continuous information. The representation is not uniform, and is magnified around the boundary area whether the robot gets the target or miss the target and whether the robot has to change its motion from rotation to forward movement.

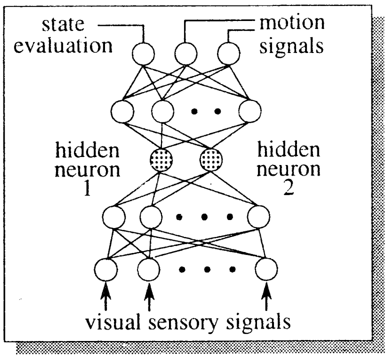
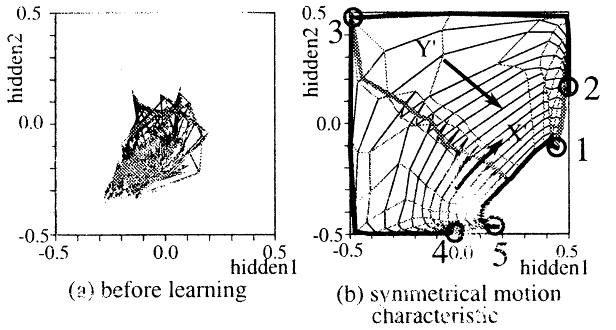


Figure 3: Five-layered sand-glass neural network. It is also used to examine the representation of hidden neurons in the next chapter.



(a) before learning

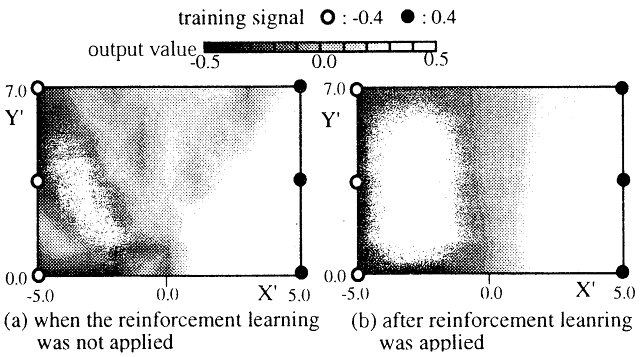
(b) symmetrical motion characteristic

Figure 4: Representation of the target location in hidden neurons' space. Circles in (b) are typical target locations which correspond to the circles in Fig 1 (a).

Next, the hidden representation was also examined for the case of the regular 3-layered neural network. After the reinforcement learning, one output neuron was added to the neural network with 0.0 connection weight and the output is trained only at six locations by the training signals as black and white circles in Fig. 5. Then the output distribution for all the target locations is as shown in Fig. 5(b). Fig. 5(a) shows the distribution when the reinforcement learning was not applied. The output distribution in the case of no reinforcement learning is spread radially. The distribution after reinforcement learning is divided into two regions, left and right. From this result, the hidden representation becomes to represent the global information that is required in the learning.

3.3 Comparison

Here the comparison of the learning performance is made between two input forms, i.e., visual sensory signals input and two dimensional relative location input from the robot to the target. Here three combinations of the input forms and network structures are employed as (1)visual sensory inputs & three-layered network, (2)visual sensory inputs & five-layered sand-glass network, (3)relative location inputs & three-layered neural network. The sand-glass network is the same as the previous subsection that is shown in Fig.3. The number of the hidden neurons are 20 for three-layered network, and 20-2-20 for the sand-glass network. The sand-glass network is employed to map the input information into two dimensional space that is the same as the relative distance inputs. The given task is the target reaching task as mentioned before, but the more difficult task is employed by introducing asymmetry of the wheel motion as follows. The



(a) when the reinforcement learning was not applied

(b) after reinforcement leanring was applied

Figure 5: Comparison of the output distribution after the supervised learning in which the six target locations are used as training data s et.

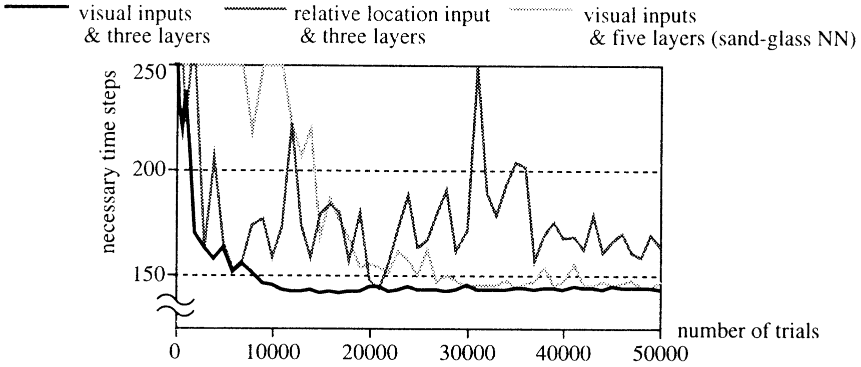


Figure 6: Comparison of the learning performance with respect to the form of the input.

right wheel is rotated according to three times as the corresponding motion signal, while the left wheel is rotated according to the motion signal itself. If the robot rotates both wheels in the same way, the motion direction gradually changes to the left. The robot loci after learning in the case of regular 3-layered neural network are shown in Fig. 2(b).

Figure 6 shows the learning curves for the three cases. The vertical axis shows the average number of steps until reaching the target state over the various target locations. Each trial is stopped at 250 time step even if the robot cannot reach the target. It is known that the learning is the fastest in the combination of visual sensory signals and 3-layered neural network. On the other hand, in the case of the relative location input, the learning is slightly slow and very unstable. This result is similar to that of Boyan et al.[7]. The learning in the case of sand-glass network is very slow, but more stable than the relative location input case.

4 Conclusion

It was shown that the visual sensor makes a role of localizing the spatial information. The difference of the learning speed and stability with respect to the input form was examined between the global spatial signals and the visual sensory signals. Then it was known that the learning is fast and stable in the case of visual sensory signal inputs.

Acknowledgments

A part of this research was supported by the Scientific Research Foundation of the Ministry of Education, Science, Sports and Culture of Japan (#10450165) and by Research for the Future Program by The Japan Society for the Promotion of Science (JSPS-RFTF96I00105).

References

[1] Shibata, K. & Okabe, Y. (1997) "Reinforcement Learning When the Visual Signals are Directly Given as Inputs", *Proc. of ICNN'97*, **3**, pp.1716-1720

[2] Shibata, K., Ito, K. & Okabe, Y. (1998) "Direct-Vision-Based Reinforcement Learning in "Going to an Target"Task with an Obstacle and with a Variety of Target Sizes," *Proc. of NEURAP'98*, pp. 95–102

[3] Shibata, K., Sugisaka, M. & Ito, K. (2000) "Hand Reaching Movement Acquired through Reinforcement Learning" , *Proc. of 2000 KACC*, 90rd (CD-ROM)

[4] Anderson, C. W. (1989) "Learning to Control an Inverted Pendulum Using Neural Networks", *IEEE Control System Magazine*, **9**, 31-37

[5] Williams, R. J. (1992) "Simple Statistical Gradient-Following Algorithm for Connectionist Reinforcement Learning", *Machine Learning*, **8**, 229-256

[6] Tesauro, G., (1992) "Practical Issues in Temporal Difference Learning", *Machine Learning*, **8**, 257-277.

[7] Boyan, J. A. & Moore, A. W. (1995) "Generalization in Reinforcement Learning: safely approximating the value function", in *Advances in Neural Information Processing Systems* (Vol.7),

[8] Sutton, R. S. (1996) "Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding", *Advances in Neural Information Processing Systems*, **8**, pp. 1038–1044

[9] Gordon, G. J.(1995) "Stable Function Approximation in Dynamic Programming", *Proc. of ICML*, pp. 261–268

[10] Sutton, R.S. & Barto, A.G. (1998) "Reinforcement Learning", The MIT Press

[11] Shibata, K., Maehara, S., Sugisaka, M. & Ito, K. (2001) "Gauss-Sigmoid Neural Network", *Proc. of 13th SICE Sympo. on Decentralized Autonomous Systems* (to appear in Japanese)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Evolving Neurofuzzy Systems for System Identification

Yuehui Chen
Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami, Kumamoto 860-8555
chen@actrl-05.cs.kumamoto-u.ac.jp

Shigeyasu Kawaji
Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami, Kumamoto 860-8555
kawaji@cs.kumamoto-u.ac.jp

Abstract

In this paper, based on the data structure of the tree, two kinds of neurofuzzy models of nonlinear systems and its learning algorithmare is proposed. Simulaion results for the identification of nonlinear systems show the feasibility and effectiveness of the proposed method.

Key words:

Neurofuzzy model, PIPE, Random search algorithm, System identification

1 Introduction

Recently, a number of papers have been published in the fields of neurofuzzy models and their learning algorithms [1]-[5].

The problems for designing a neurofuzzy system are that how to determine the architecture of the system and what learning algorithm can be effectively used for the training of the neurofuzzy system. In detail, these problems are refered to the determination of the overall structure of the neurofuzzy model, the position and shape of the fuzzy mumbership functions, the number of fuzzy rules, and the determination of dividing method of input space. As it is known that the curse-of-dimensionality is an un-solved problem in the fields of fuzzy and neurofuzzy systems.

In this paper, based on the data structure of the tree, two kinds of neurofuzzy models of the nonlinear systems and their learning algorithm are proposed. The one is called additive neurofuzzy model in which a multivariable nonlinear system is approximated by a number of weighted sum of the single variable fuzzy system. The other is a direct neurofuzzy model in which multivariable nonlinear system is approximated by a number of fuzzy rules that calculated like in TSK fuzzy model. In dealing with this, a flexible scheme is proposed for evolving the neurofuzzy models in which the architecture of neurofuzzy model and the parameters used in neurofuzzy model are automatically evolved or optimized by probabilistic incremental program evolution algorithm (PIPE) [6][7] and random search algorithm [8], simultaneously.

2 Additive neurofuzzy model

The tree structural representation of additive neurofuzzy model is shown in Fig.1, in which the contents of box below the input x_i denote the data structure of input x_i. Along with the creation of the tree, these parameters are created randomly.

The output of node x_i in the tree is calculated as a single fuzzy TSK model.

The output of overall additive neurofuzzy system is calculated as

$$y = \sum_{i=0}^{N-1} \omega_i * \frac{\sum_{j=1}^{N_j} \mu_{A_j}(x_i)(a_j + b_j * x_i)}{\sum_{j=1}^{N_j} \mu_{A_j}(x_i)} \qquad (1)$$

In this case, the optimization of additive neurofuzzy model to fitting a given data set is that select the appropriate number of fuzzy sets for each input variable, determine the shape and position of each fuzzy mumbership functions and find the appropriate weights of neurofuzzy model.

3 Direct neurofuzzy model

The proposed tree structural representation of direct neurofuzzy model is shown in Fig.2, in which the j-th node of the tree denotes the j-th rule which can be represented as

$$\text{if } x_0 \text{ is } A_{j0} \text{ and } x_1 \text{ is } A_{j1} \ldots \text{ and } x_{n-1} \text{ is } A_{j(n-1)}$$
$$\text{then } y_j = b_{j0}x_0 + b_{j1}x_1 + \ldots + b_{j(n-1)}x_{n-1} + b_{jn}$$
$$(\text{j=0, 1, ... N-1})$$

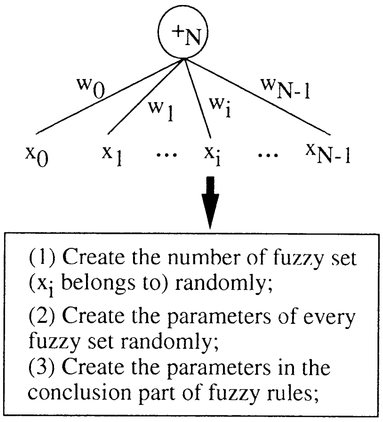

+N
w_0
w_1
w_i
w_{N-1}
x_0 x_1 ... x_i ... x_{N-1}
(1) Create the number of fuzzy set (x_i belongs to) randomly;
(2) Create the parameters of every fuzzy set randomly;
(3) Create the parameters in the conclusion part of fuzzy rules;

Fig. 1: Tree Structural representation of additive neurofuzzy model

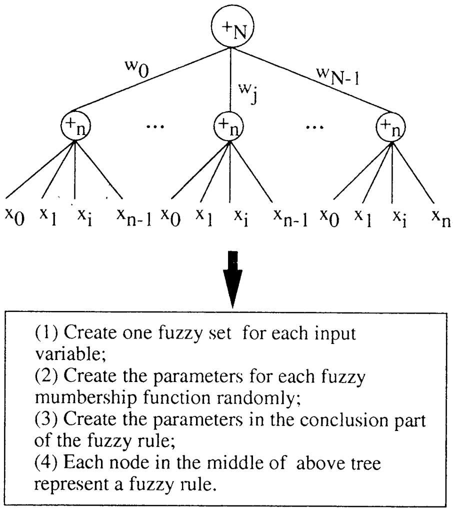

+N
w_0
w_j
w_{N-1}
+n ... +n ... +n
x_0 x_1 x_i x_{n-1} x_0 x_1 x_i x_{n-1} x_0 x_1 x_i x_{n-1}
(1) Create one fuzzy set for each input variable;
(2) Create the parameters for each fuzzy mumbership function randomly;
(3) Create the parameters in the conclusion part of the fuzzy rule;
(4) Each node in the middle of above tree represent a fuzzy rule.

Fig. 2: Tree Structural representation of direct neurofuzzy model

In the tree (Fig.2), the output of node j is calculted according to j-th rule as follows

$$\prod_{i=0}^{n-1} \mu_{A_{ji}}(x_i) * y_j \qquad (2)$$

Therefore, the output of direct neurofuzzy model is calculated as

$$y = \sum_{j=0}^{N-1} \omega_j * (\prod_{i=0}^{n-1} \mu_{A_{ji}}(x_i) * y_j) \qquad (3)$$

In this case, the objective of direct neurofuzzy model optimization is directly learning and selecting a number of fuzzy rules for fitting a given data set.

The used fuzzy membership function in both of additive and direct neurofuzzy models is second order B-spline function. B-spline can be defined in a recursive way,

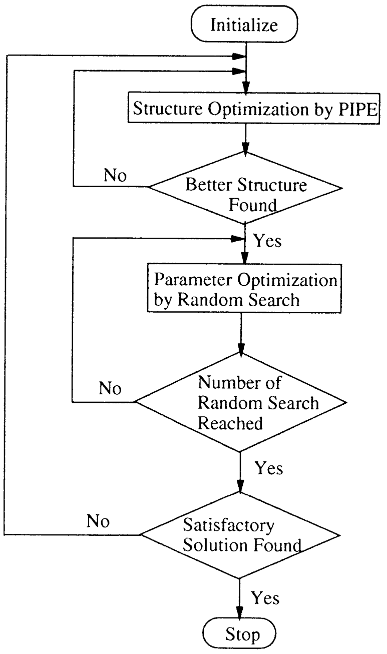

Initialize
Structure Optimization by PIPE
No
Better Structure Found
Yes
Parameter Optimization by Random Search
No
Number of Random Search Reached
Yes
No
Satisfactory Solution Found
Yes
Stop

Fig. 3: The flow chat of the random search algorithm

$$B_0(t) = \begin{cases} 1, & t \in [-\frac{1}{2}, \frac{1}{2}] \\ 0, & otherwise \end{cases} \qquad (4)$$

$$B_m(t) = (B_{m-1} * B_0)(t) \qquad (5)$$

The translation and dilation of second order B-spline used in this experiment is given as

$$f_{a,b}(t) = B_2(\frac{t-b}{a})$$

$$= \begin{cases} \frac{9}{8} + \frac{3}{2}(\frac{t-b}{a}) + \frac{1}{2}(\frac{t-b}{a})^2, & t \in [-\frac{3}{2}a + b, -\frac{1}{2}a + b) \\ \frac{3}{4} - (\frac{t-b}{a})^2, & t \in [-\frac{1}{2}a + b, \frac{1}{2}a + b) \\ \frac{9}{8} - \frac{3}{2}(\frac{t-b}{a}) + \frac{1}{2}(\frac{t-b}{a})^2, & t \in [\frac{1}{2}a + b, \frac{3}{2}a + b] \\ 0, & otherwise \end{cases} \qquad (6)$$

4 Design of the neurofuzzy models

Based on the above representation of additive and direct neurofuzzy models, a hybrid learning scheme (see Fig.3) is proposed, in which the architectures of additive and direct neurofuzzy models are evolved by PIPE algorithm [6][7] and the parameters of the models are optimized by random search algorithm [8].

Design algorithm of neurofuzzy models

1) Create the initial population including the sparse tree and the corresponding parameters used in the additive and direct neurofuzzy model.
2) Structure optimization by PIPE sub-program, in which the fitness function is calculated by

$$Fit(i) = \sum_{j=1}^{P} |y_1^j - y_2^j| \tag{7}$$

where P is the total number of samples, y_1^i and y_2^i are the actual and model output of i-th sample, $Fit(i)$ denotes the fitness value of i-th individual.
3) If the better structure found, then go to step 4, otherwise go to step 2. The creterion concerning with better structure found is distinguished as follows: if the fitness value of the best progam is smaller than the fitness value of the elitist program, or the fitness values of two programs are equal but the nodes of the former is lower than the later, then we say that the better structure is found. Where the best and elitist program are the best program at current generation and the one found so far, respectively.
4) Parameter optimization by random search subprogram. In this step, the parameter vector $W(k)$ of best program (tree) is taken out from the population and optimized by random search algorithm in order to decrease the fitness value of best program.
5) If the maximum number of random search is reached, then go to step 6; otherwise go to step 4.
6) If satisfied solution is found, then stop; otherwise go to step 2.

5 Experiments

The plant to be identified is given by the following equation [9]

$$y(k+1) = \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2)-1)+u(k)}{1+y^2(k-1)+y^2(k-2)} \tag{8}$$

The input and output of system are $x(k) = [u(k), u(k-1), y(k), y(k-1), y(k-2)]$ and $y(k+1)$, respectively.

The training samples and the test data set are generated by using input signal of eqn.(7) and eqn.(8), respectively.

$$u(k) = \begin{cases} sin\left(\frac{2\pi k}{250}\right), & if\,(k < 500) \\ 0.8sin\left(\frac{2\pi k}{250}\right) + 0.2sin\left(\frac{2\pi k}{25}\right), & \\ & if\,(k >= 500) \end{cases} \tag{9}$$

Table 1: Parameters used in PIPE Algorithm

Population Size PS	10
Elitist Learning Probability P_{el}	0.01
Learning Rate lr	0.01
Fitness Constant ε	0.000001
Overall Mutation Probability P_M	0.4
Mutation Rate mr	0.4
Prune Threshold T_P	0.999999

$$\begin{aligned} u(k) = \; & 0.3sin(k\pi/25) + 0.1sin(k\pi/32) \\ & +0.1sin(k\pi/10) \end{aligned} \tag{10}$$

The used parameters in PIPE are shown in Table 1. The initial values of parameters in random search are listed as: $\beta_0 = 0.1$, $\beta_1 = 1000$, $\alpha = 0.995$, $\phi_0 = 0.1$, $\phi_{min} = 0.001$, $I_{sf0} = 10$, $P_{sf0} = 0.3$, $\triangle I_{sf1} = 0.02$, $\triangle I_{sf2} = 0.1$, $I_{sfmax} = 100$, $K_{er} = 1.001$, $c_i = 1.01$, and $c_d = 0.995$. The basic random search step $\alpha = 2000$.

Experiment 1: The evolutionary induction of additive neuro-fuzzy models The solution was returned as the best individual found in the run after the 10110 search steps with fitness 4.446611. The average error of each sample is 0.004446611.

The numbers of fuzzy sets for each input variable of $u(k), u(k-1), y(k), y(k-1), y(k-2)$ are 3, 5, 3, 3, 7 in the evolved neurofuzzy model, respectively. The comparision of actual and evolved outputs of the plant for are shown in Fig.4 and Fig.5, respectively.

Experiment 2: The evolutionary induction of direct neuro-fuzzy models

The solution was returned as the best individual found in the run after the 27737 search steps with fitness 2.986719. The average error of each sample is 0.002986719 (the training sample is same with thoese used in experiment 1).

The evolved direct neurofuzzy model only has 11 fuzzy rules. The comparision of actual and evolved outputs of the plant for test data set are shown in Fig.6 and Fig.7, respectively.

From above simulation results, it can be seen that the proposed methods works very well for generating the proper additive and direct neurofuzzy models.

6 Conclusion

Based on a new calculation structure of type constrained sparse tree, additive and direct neurofuzzy

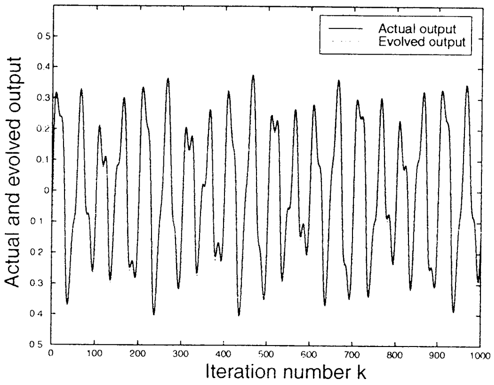


Fig. 4: The actual and evolved outputs of the plant for the test data set (experiment 1)

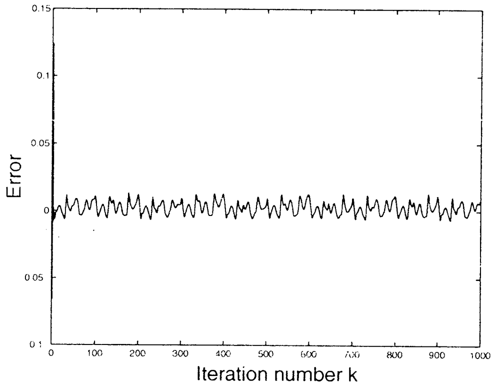


Fig. 5: The test error (experiment 1)

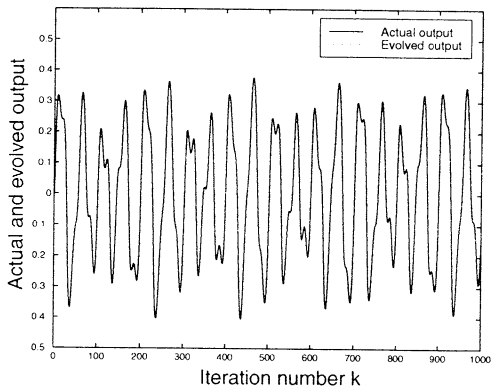


Fig. 6: The actual and evolved outputs of the plant for the test data set (experiment 2)

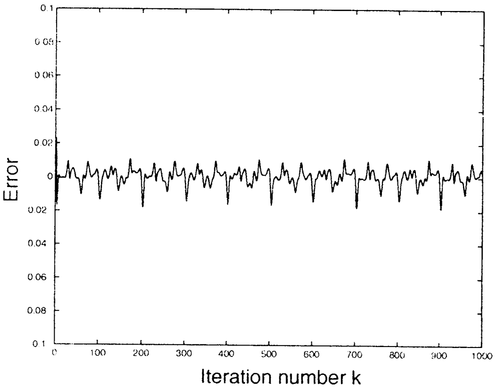


Fig. 7: The test error (experiment 2)

models are proposed in this paper. The advantage of the proposed method is that the distribution of fuzzy sets and the portion of fuzzy rules are evolved without any a prior. Simulation results shown that the evolved additive and direct neurofuzzy model are effective for the identification of nonlinear system.

References

[1] J.-S. R. Jang, "ANFIS, Adaptive-network-based fuzzy inference systems", *IEEE Trans. on System, Man and Cybernetics*, Vol.23, No.3, pp.665-685, 1992.

[2] S. Barada etal., "Generating Optimal Adaptive Fuzzy-Neural Models of Dynamical Systems with Applications to Control", *IEEE Trans. on SMC*, Vol.28, No.3, pp.371-390, 1998.

[3] M. Russo, "FuGeNeSys-A Fuzzy Genetic Neural System for the Fuzzy Modeling", *IEEE Trans. on SMC*, Vol.6, No.3, pp.373-388, 1998.

[4] N. Kasabov, "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems", *Fuzzy sets and Systems*, Vol.82, No.2, pp.2-20, 1996.

[5] N. Kasabov et al., "FuNN/2- A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition", *Information Science - Applications*, Vol.101, No.3-4, pp.155-175, 1997.

[6] Y. Chen and S. Kawaji, "Identification and Control of Nonlinear System using PIPE Algorithm", *Proc. of Workshop on Soft Computing in Industry'99 (IWSCI'99)*, Muroran, pp.60-65, 1999.

[7] Y. Chen and S. Kawaji, "Evolutionary Control of Discrte-Time Nonlinear System using PIPE Algorithm", *Proc. of IEEE Int. Conference on Systems,Man and Cybernetics*, Vol.4, pp.1078-1083, Tokyo, 1999.

[8] J. Hu, et. al., "RasID - Random Search for Neural Network Training", *Int. J. of Advanced Computational Intelligence*, Vol.2, No.2, 134-141, (1998).

[9] K. S. Narendra et al., "Identification and Control of Dynamic System using Neural Networks", *IEEE Trans. on Neural Networks*, Vol.1, No. 2, Jan. pp.4-27, 1990.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Labeling Q-Learning In Hidden State Environments

○Hae Yeon Lee*, Hiroyuki Kamaya**, Kenich Abe*

*Dept. Electrical and Communication Engineering, Graduate School of Engineering, Tohoku Univ.
Aoba05, 980-8579, Sendai, Japan
(Tel : +81-22-217-7075 ; Fax : +81-22-263-9290 ;
E-mail : yeon@abe.ecei.tohoku.ac.jp, abe@abe.ecei.tohoku.ac.jp)
**Dept. Electrical Engineering, Hachinohe National College of Technology
Tamonoki, 139-1192, Hachinohe, Japan
(Tel : +81-178-27-7283; Fax : +81-178-27-9379 ; Tel ; E-mail : kamaya-e@hachinohe-c.ac.jp)

Abstract

Recently, *Reinforcement Learning*(RL) methods have been used for learning problems in environments with hidden states. However, conventional RL-methods have been limited to the *Markov Decision Process* problems. To overcome the hidden states, several algorithms were proposed [1], [2], [3]. But these need extreme amount of memory of past sequences.

The aim of this paper is to extend our previous algorithm for hidden states environments, called *Labeling Q-learning*(*LQ-learning*)[4][5][6], which reinforces incomplete observed perception by labeling. Namely, in the LQ-learning, the agent has the perception structure constructed with pairs of observations and labels. By these pairs, the agent can distinguish states more exactly, which look same but really different. Labeling is executed by labeling functions. Numerous labeling functions can be considered, but in this paper, we will introduce several kinds of labeling functions based only on 1 immediate past sequential observation and the current observation.

We will introduce the basic idea of LQ-learning briefly, apply it to maze problems with hidden state, and show its availability of LQ-learning by simulations.

Keywords;
Reinforcement Learning, Labeling Q-learning, Hidden States Environment, Agent, Grid-World

1. Introduction

Sequential decision problems in which agent's observations provide the agent with the complete state of its environment can be formulated as *Markov Decision Processes*(MDP). For MDP problems, a number of successful plans and RL methods have been developed.

However, in many areas, e.g., mobile robotics, multi-agents or distributed control environments, etc., the environment's perceptions at best give the agent incomplete information about the state of environment. Such interactions between the agent and environments suffer from hidden states. Therefore, recent studies on the hidden states environments have concentrated on overcoming these hidden states by using less memory to estimate states.

Thus, this emphasis on state estimation has come about because it has been widely noted that the presence of hidden states renders popular and successful RL methods for MDP, such as Q-learning by Watkins[7], TD(λ) by Sutton[8], and Sarsa(λ) by Dayan[9], useless on hidden states environments. Next, we introduce the basic algorithm of RL below.

In the RL, the environment is defined by a finite set of state *S*, and the agent has a finite set of actions *A*. In the RL process, the agent observes its environment's state $s \in S$ time-discretely, and at each time step, determines an action $a \in A$ based on the observation and receives a scalar reward *r*. Thus, the object of RL is to construct an optimal action policy that maximizes the expected discounted reward.

$$V(s_t) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{1}$$

Here, r_t is the reward at time step *t* and $0 \leq \gamma \leq 1$ is discount factor that makes immediate reward more available than reward more distant in time.

An action selection at each step is based on Q-values, $Q(s,a)$, which is relative to goodness of actions. The Q-value, $Q(s,a)$, is the total discounted reward that the agent would receive when it starts at a state *s*, performs an action *a,* and behaves optimally thereafter.

In most RL methods, the Q-values are estimated by the *Temporal Difference*(TD) method, see Sutton[10]. A standard form of the algorithms to estimate the Q-values with heuristic traces is expressed as follows;

$$Q(s,a) \leftarrow Q(s,a) + \left[r_t + \gamma Q(s_{t+1}, a_t) - Q(s_t, a_t)\right] e(s,a) \quad (2)$$

where α is learning rate, γ is discount factor, and $e(s,a)$ is *eligibility traces*, see Loch and Singh,[11]. The eligibility traces emphasize more recent and frequent events more strongly. But in our paper, we use a modified form of the eligibility traces, called *replacing traces*, see Singh and Sutton[12].

$$\begin{aligned} &e(s_t, a_t) \leftarrow 1 \\ &e(s,a) \leftarrow \gamma \lambda e(s,a) \quad \text{for} \quad s \neq s_t, \text{ or } a \neq a_t \end{aligned} \quad (3)$$

2. Labeling Q-learning

In the environment with hidden states, the agent is not allowed to observe the state directly, but the agent gets an observation containing information about the state. Let $o_t \in O$ be the observation at step t. Then, in our LQ-learning process, an integer $\theta_t \in \Theta$ is attached to the observation o_t based on previous actions and observations, i.e., past history, where $\Theta = \{0, 1, 2, \cdots, M-1\}$. This defines a new perception of the environment, $\tilde{o}_t = (o_t, \theta_t) \in O \times \Theta (= O)$. For these new perceptions, Q-values $Q(\tilde{o}, a)$ are updated by conventional Q-learning algorithm.

To save memories, a label θ_t, attached to observation o_t, is used to successive observations as many as possible. When the agent gets an observation with a noticeable feature, a kind of landmark, then new label different from the label just before should be attached to the observation.

Let Λ_t be the history at step t, i.e., a sequence of previous actions and observations with a fixed length and the present observation. Then we first define a logical variable "*CHANGE*" on the history space. *CHANGE* gives the condition under which new label is attached to the present observation. By introducing some random variables, we are able to define a probabilistic *CHANGE*. In our previous study, we introduced a probabilistic *CHANGE*(see [6]). Here, we will introduce two types of deterministic *CHANGE*s.

In addition, we use a rule of attaching labels to observations. We call this rule as Labeling Function(LF), say h, on the history space. Thus our labeling process is stated in general as follows.

$$\theta(o_t) = h(\Lambda_t) \quad \text{if } CHANGE(\Lambda_t) \text{ is True.} \quad (4)$$

$$\theta(o_t) = \theta(o_{t-1}) \quad \text{if } CHANGE(\Lambda_t) \text{ is False.} \quad (5)$$

Here, however, we consider a rather simple case. That is, we consider the pair of present observation o_t and preceding observation o_{t-1} as the history.

3. Environments and Tasks

3.1 Grid-World

In this paper, we use Grid-Worlds as test problems of hidden states Environments. Because the Grid-World is not restricted in arrangement of obstacles, starting points, and goal points, then environments with various complexities may be realized. Fig.1 is an example used in our simulations.

In Fig.1, gray cells mean obstacles. Each state transition is a movement from one white cell to a next white cell. And in Fig.1, "S" means the starting point, and "G" the goal. The ultimate object of the agent is to reach the goal in as few steps as possible. The dotted line in each figure shows the shortest path. It is ideal for the agent to learn the shortest path.

At each cell or step, the agent can only observe which of the adjacent 4 cells are obstacles. The values of the north, east, south and west gray cells next to the agent are 1, 2, 4, and 8, respectively, and the agent's observation is the sum of values of gray cells surrounding the agent. Thus, there are 16 possible observations. Moreover, Q-table of the last success trial is memorized. And this is used at the starting of trial after the failure.

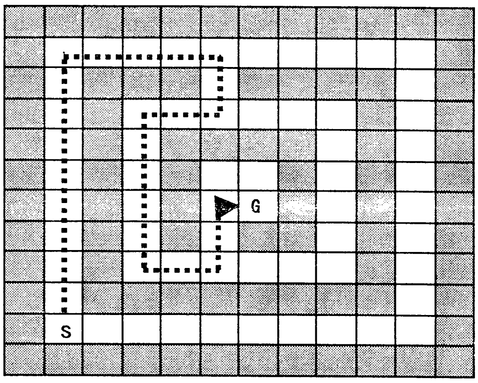


Fig.1 Experimented Maze: The optimal path from S to G, is shown by dotted line, it takes 27.

3.2 Task Representation

The agent's task is to discover a path leading from the starting position to the goal in the maze. The agent is assumed to have the lifetime, $T_{\max}$, separated into "trials". Each trial consists of at most $S_{\max}$ steps. Each trial ends either if the agent arrives at the goal, or cannot find the goal in within $S_{\max}$ steps. In each trial, only when the agent arrives at the goal, it receives a reward r, 10 in our tests. Otherwise the reward is zero. In our simulations, $T_{\max}$ is fixed at 50,000 and $S_{\max}$ at 100.

3.3 Action Selection

Though numerous number of action-selection rules can be considered, we use *ε -greedy* rule, which means that most of time the agent choose an action that has maximal estimated action value, but with probability ε they instead select an action at random

In addition, we try to make the convergence of Q-values more stable. Thus the ε is decreased in proportion to learning trials, because the randomly selected action would disturb the convergence of Q-values in the latter part of the learning. At a trial T, ε is given as follows:

$$\varepsilon_T = (1 - \frac{T}{T_{\max}})\varepsilon_{\max} \qquad (6)$$

Here, ε_T is the epsilon at trial T, and $\varepsilon_{\max}$ is the maximum value of epsilon fixed at 0.1 in our experiment, e.g., if $T_{\max}$ =10,000, the ε is 0.01 at the 1000th trial.

4. Experiments & Results

4.1 Experiments

In this paper, we test three type of labeling methods. The first one, which we call here LQ-1, has probabilistic CHANGE, and it was introduced in our previous paper[5].

The second method, called LQ-2, is rather simple structure. It has no stochastic variables, and labeling is done deterministically at all landmark-like observations.

The third one, called LQ-3, is deterministic also, but it has a variable, called *count*. This variable is introduced for saving the number of labels. In our previous methods [4], [5], [6], the agents changed their labels at each noticeable steps, though probabilistically, but in LQ-3, the label can not be changed except when the variable *count* reaches to a positive constant, say *max-count*, which is a parameter in the algorithm. More details are explained below.

LQ-1

In the LQ-l, the *CHANGE* is defined as follows.

$$CHANGE = (o_t \neq o_{t-1}) \wedge (\chi(o_t) = 1) \qquad (7)$$

where χ is a probabilistic variable, and the details are seen in [6]. And we use the LF shown below.

$$\theta(o_t) \leftarrow h_2(o_t, o_{t-1}) = (o_t - o_{t-1}) \bmod M \qquad (8)$$

Here, M is maximum number of labels.

LQ-2

In the LQ-2, labeling is deterministic. The *CHANGE* is defined as follows.

$$CHANGE = (o_t \neq o_{t-1}) \qquad (9)$$

And the LF is more simple than LQ-1.

$$\theta(o_t) \leftarrow \theta(o_t) + 1 \qquad (10)$$

LQ-3

As mentioned above, in the LQ-3, the agent has a integer variable, *count*, and the labeling is done under the condition of *CHANGE* shown below.

$$CHANGE = (o_t \neq o_{t-1}) \wedge (count = max\_count) \qquad (11)$$

The LF is the same with above LQ-2. In our experiment, *max_count*=3. The variable count is updated as follows.

```
if       $CHANGE = TRUE$
             $count \leftarrow 0$    (reset)
else if  $(o_t \neq o_{t-1}) \wedge (count < max\_count)$        (12)
             $count \leftarrow count + 1$
else
             $count \leftarrow count$
```

3.2 Results

Simulation results of above three LQs are summarized in Table 1. All entries of Table 1 are averages of 30 experiments. The 2nd column lists average steps(path lengths), and 3rd column lists goal hit percentages, at the final trial. We can see that all methods find near-optimal path. LQ-3 performs better than other two methods.

Labeling methods	Ave. Steps	(%) Goal
LQ-1	33	90
LQ-2	38	100
LQ-3	31	100

Table.1 Results of LQ-learning

Fig.2 shows average steps against trials.

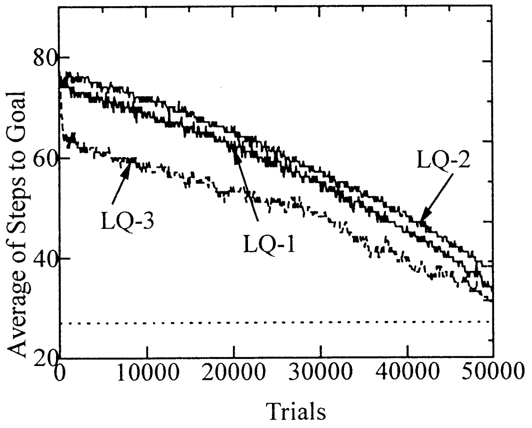


Fig.2. The following parameters were used for all experiments: $\lambda = 0.9$, $\gamma = 0.9$, and $\alpha = 0.1$. In LQ-1, the probability p_0 is initialized with 0.95 at each experiments.

5.Conclusion

In this paper, we presented several extended versions of Labeling Q-learning, and applied these to a test problem of mazes, a simple environment with hidden states. And results of experiments demonstrated LQ-learning's ability to learn the paths of the maze though they are near-optimal.

The LQ-learning, however, has several limitations. In all experiments, we used LQ-learning with only 5 labels (i.e., M=5). We tested several experiments with different numbers of labels, and found that the LQ-learning at least with 5 labels worked well in the environment like Fig.1. We should exploit how to decide the appropriate number of labels by learning. This is one of our future works.

Further, we should develop rather efficient labeling functions also. We should develop a learning agent, which can learn the LFs by itself. This is another future possibility of our approach.

References

[1] Holland. J. H. Adaptation in Neural and Artificial Systems, Univ. Michigan Press, Ann Arbor, MI, 1995.

2] Lin, L. and Mitchell, T. M. Reinforcement learning with hidden states. in Proceeding of the Second International Conference on Simulation of Adaptive Behavior, 271-280, 1992.

[3] Wiering. M. and Schmidhuber. J. HQ-Learning, Adaptive Behavior 6:2, 1997.

[4]Haeyeon Lee, Hiroyuki Kamaya, Kenichi Abe, Labeling Q-Learning For Non-Markovian Environments. IEEE International Conference on System, Man, and Cybernetics (SMC'99), V-487-491, 1999

[5] Haeyeon Lee, Hiroyuki Kamaya, Kenichi Abe, Labeling Q-Learning for Partially Observable Markov Decision Process Environments, International Symposium on ARTIFICIAL LIFE AND ROBOTICS, Proc. of The Fifth Int. Symp. On Artificial Life and Robotics(AROB 5th'00) , 484-487, 2000

[6]Haeyeon Lee, Hiroyuki Kamaya, Kenichi Abe, Labeling Q-Learning for Maze Problems with Partially Observable States, Korean Automatic Control Conference (KACC2000), 489-492, 2000

[7] Watkins, C. J. C. H. and Dayan, P. Q-learning. Machine Learning, 8, 279-292, 1992.

[8] Sutton, R. S. and Barto, A. G. Reinforcement Learning : An Introduction, Cambridge, MA, MIT Press, 1988.

[9]Dayan, P. D. The convergence of TD() for general λ , Machine Learning , 8, 341-362, 1992.

[10] Sutton, R. S. Learning to predict by the methods of temporal differences, Machine Learning, 3, 9-44, 1998.

[11] Loch, J. and Singh, S. Using Eligibility Traces to Find the Best Memoryless Policy in Partially Observable Markov Decision Processes. ICML-98, 1998.

[12] Singh, S. P. and Sutton, R. S. Reinforcement Learning with Replacing Eligibility Traces. Machine Learning, 22, 123-158, 1996.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

The Voice Sound Source Separation based on the Grouping in the Frequency Domain

○K.Ninagawa, T.Umeyama, Y.Sagawa, N.Sugie
Department of Electrical and Electronic Engineering
Graduate School of Science and Technology, Meijo University
1-501 Shiogamaguchi, Tempaku-ku, Nagoya, 468-8502, Japan
c3002022@meijo-u.ac.jp

Abstract

We propose a new scheme of sound source separation. The mixed speech sound is received with one microphone. The sound signal is transformed to a sound spectrogram. Then the high power components or formants are extracted. Exploiting the concurrency of onsets of those formants belonging to one voice, and other constraints, formants are separated into groups corresponding to each voice. Then applying a set of band-pass filters corresponding to grouped formants to the mixed voices, we can extract each voice. The experiments yielded promising results.

Key words: Sound source separation, cocktail party effect, sound spectrogram

1 Introduction

Various sounds exist around us. We can hear only one voice among them, and understand the speech. The human ability of sound source separation is known as a cocktail party effect.

It is reported that one can understand better when one listens to binaurally than monaurally. Furthermore it seems obvious that some sort of semantic processing plays certain positive roles as well.

Suggested by the above biological findings, there is proposed a scheme for sound source separation. It receives the mixed sounds with two or three microphones, and then it separates two sound sources. The scheme consists of the following two stages: first the direction of each sound with respect to the microphones is determined by examining the intermicrophone temporal differences of the onsets of mixed sounds; then it is possible to eliminate one of the two sounds using the determined temporal difference, resulting in separation[1].

Independent of the proposal of the above scheme, another scheme, blind separation or independent component analysis, has been intensively under study[2]. In these schemes, n microphones are assumed to separate n sounds (in most cases, n=2).

In this paper we propose a new scheme for sound separation. The scheme is new in that
(1) only one microphone is required to separate two mixed voice and that
(2) the processing is based on processing applied to sound spectrograms.

2 The Voice Sound Source Separation

The processing of voice sound source separation consists of two stages shown in Figure 1.
Stage 1: Extraction of formants and the grouping.
Stage 2: Extraction of separated voice.

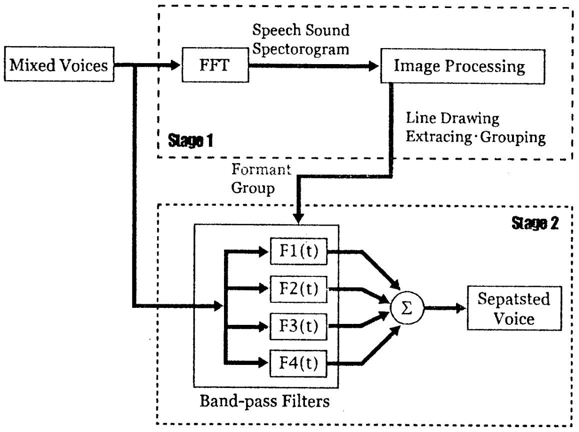


Figure 1: The sound source separation processing Figure

2.1 Grouping in the Frequency Domain

Two mixed voices are received by one microphone. Then the sound spectrogram is obtained. The sound spectrogram shows the time variation of frequency spectrum as a grayscale image. By processing the sound spectrogram, formants of two mixed voices can be extracted. Then we separate formants to two groups corresponding to two voices.

2.1.1 Line Drawing Extraction

By extracting formants, the sound spectrogram is transformed to a line drawing as shown in Figure 2.

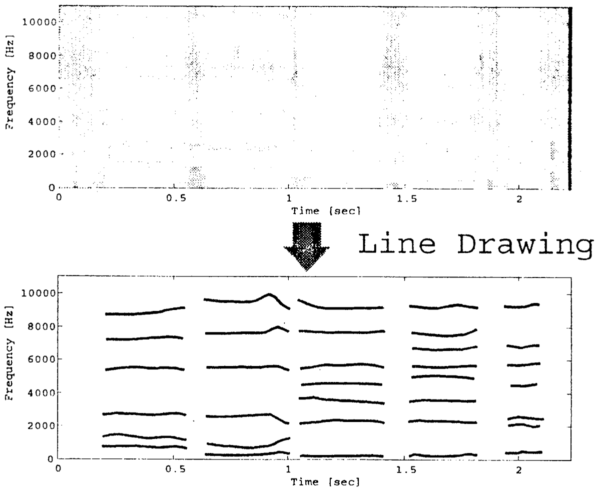


Figure 2: The line drawing obtained from the sound spectrogram.

2.1.2 Grouping

The line drawing version of the sound spectrogram consists of several formants, some of which are relate to one voice and others to the other voice. Exploiting the concurrency of transient events such as onsets(Figure 3), the relationship in the frequency domain among those formants constituting vowels, and so on, we can separate formants to two groups corresponding to two voices.

2.2 The Extraction of Separated Voice

When we separate the formants from the mixed two voices to two groups, the voice separation can be done as follows. Let two voices be Voice A and Voice B. Referring to Figure 1, Fi(t) denotes the frequency of the i-th formant of one voice, at time t. These formants constitute the main components of the voice. Thus by applying a set of band-pass filters, whose central frequencies at t are Fi(t)s, to the mixed voices, we can separate the mixed voices to Voice A and Voice B.

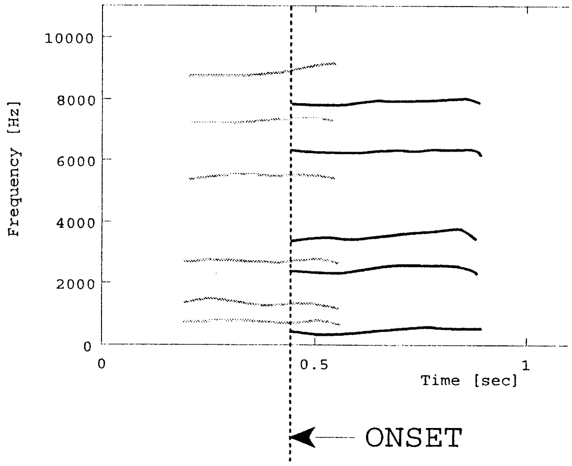


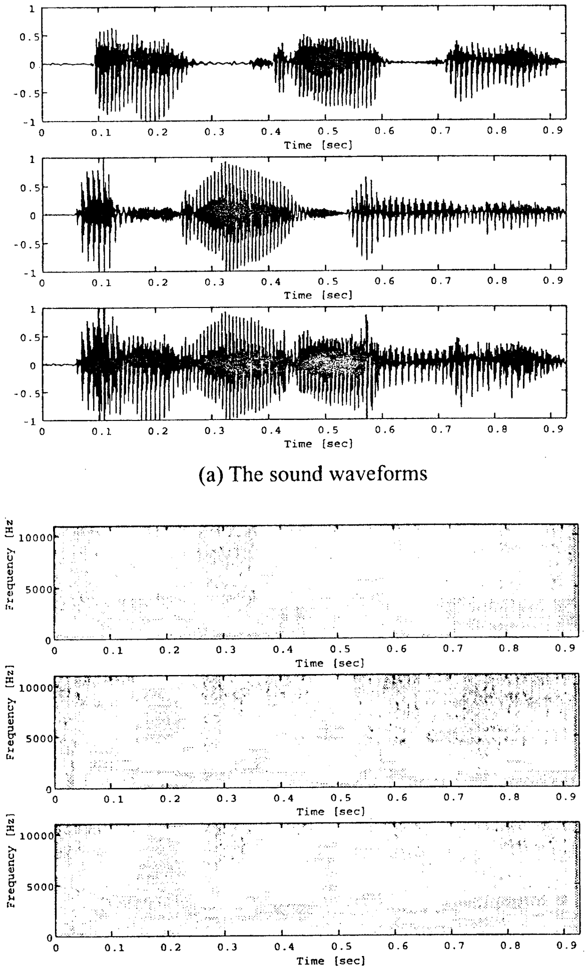
Figure 3: Separation to two groups.

3 Experiments

3.1 Methods

At the moment, the processing at the first stage is still incomplete. Therefore we extracted F'i(t)s of each voice from the corresponding unmixed voice, Voice A or Voice B. The Fi(t)s from Voice A and Voice B are extracted using the maximum entropy method. We chose those frequency components with higher power. We used four frequency components and their second harmonics. Thus F1(t),...,F8(t) were obtained. We found that they contained the formants of the vowels within the voice.

As Voice A, we used "sports news in Japanese", while we used "weather forecast in Japanese" as Voice B. The mixed voices were synthesized by mixing Voice A and Voice B. In Figure 4(a), the top-, the middle-, and the bottom-waveforms represent Voice A, Voice B, and the mixed Voices, respectively. In Figure 4(b), the corresponding sound spectrograms are shown.



(a) The sound waveforms

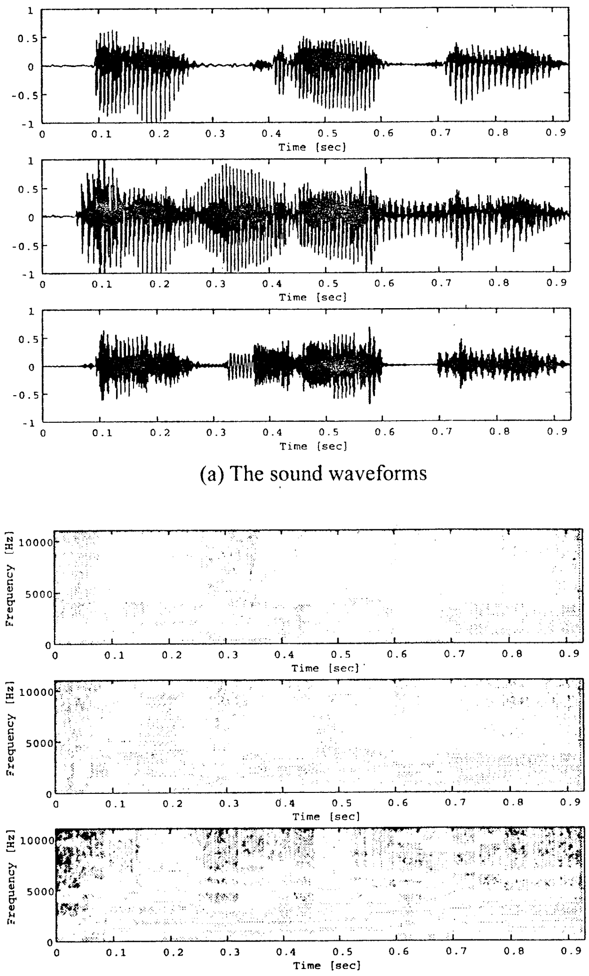
(b) The sound spectrograms

Figure 4: The voices used by the experiment.

3.2 Results of Sound Source Separation

Exploiting the scheme shown in Figure 1, we separated Voice A from the mixed voices. In Figure 5(a) are shown three waveforms. The top shows pure Voice A itself. The middle is the mixed voices. The bottom is the separated Voice A from the mixed voices. The bottom waveform is reasonably similar to the top waveform, the ideal one. When we listened to the bottom waveform, we could recognize the speech. In Figure 5(b) are shown the sound spectrograms corresponding to Figure 5(a).

In Figure 6 are shown the results of extraction of Voice B. As a whole, the listening tests of the extracted Voice A and Voice B suggest that the vowel portions of the voice can be well recognized, but the consonant portions are difficult to be recognized.



(a) The sound waveforms

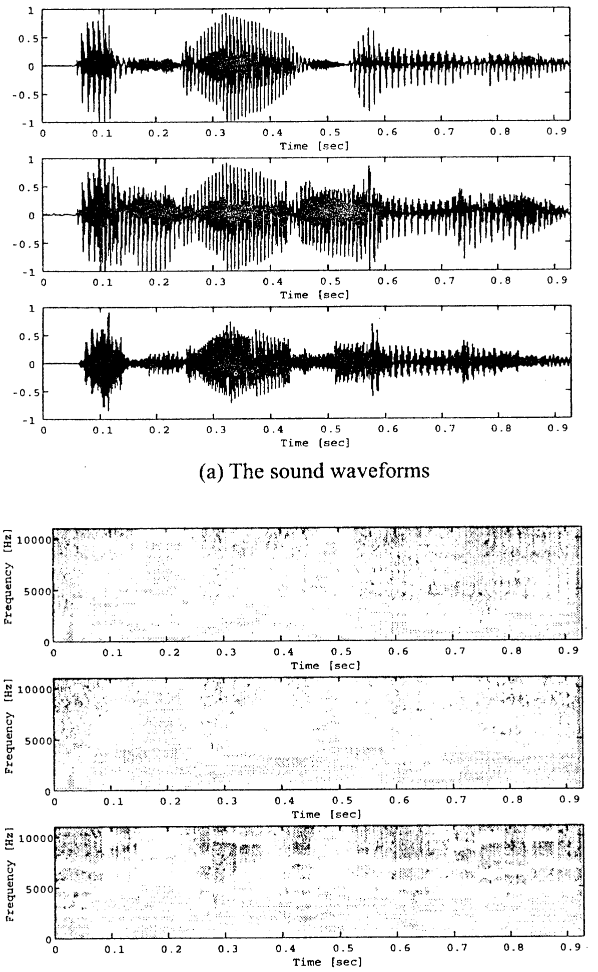
(b) The sound spectrograms

Figure 5: The result of the sound source separation for Voice A.

4 Conclusions

We proposed a scheme of mixed voice separation in the frequency domain. The scheme consists of two stages. The first stage is concerned with the separation in the frequency domain, while the second with the extraction of each constituent voice from the mixed voices.

At the moment, the first stage is only in completely implemented, while the second stage is completed.



(a) The sound waveforms

(b) The sound spectrograms

Figure 6: The result of the sound source separation for Voice B.

We conducted experiments, where two voices are Japanese sports news and Japanese weather forecast. We could extract each voice from the mixed voices. The quality of the voices was reasonable in the vowel portions, but poor in the consonant portions.

The future issues are
(1) completion of stage 1,
(2) improvement of voice quality of separated voice.

References

[1] J.Huang, N.Ohnishi, N.Sugie, "A biomimetic system for localization and separation of multiple sound sources", *IEEE Trans. Instrumentation and Measurement*, 44, 3, pp.733-738, June 1995.

[2] C.Jutten, J.Herault, "Blind separation of sources, Part 1:An adaptive algorithm based on a neuromimetic architecture", *Signal Processing*, 24,1,pp.1-10, 1991.

[3] M.Kawamoto, K.Matsumoto, N.Ohnishi, "A method of blind separation for convolved non-stationary signals", *Neurcomputing*, 22, pp.157-171, July 1998.

[4] N.Sugie, "Extracting Order from the Chaotic World -Mechanism of Extended Perceptual Grouping-", *Journal of Signal Processing*, Vol.4, No.5, pp.343-348, September 2000.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Social Evolution in Imperfect World

Hiroshi SATOH, Kimitaka UNO, Masao KUBO, Akira NAMATAME
Department of Computer Sciencc
National Defense Academy of Japan
1-10-20 Hashirimizu, Yokosuka, Kanagawa, 239-8686 Japan
E-mail: {hsato, kimi, masaok, nama}@nda.ac.jp

Abstract

Agent based simulation is a powerful tool for the research of complex systems such as human society. The important key word of this kind of research is "bounded rationality" of the agents. From the standpoint, we adopt following two hypotheses as the basis of our analysis: (1) each agent interacts only with his neighbors, and (2) each agent behaves by mimicking its neighbors. How the society gropes its way towards equilibrium in this imperfect world? The purpose of this research is to investigate the role of mimicry in social evolution. In this paper, we examined the growth of conventions in the society where individuals rely on hearsay to determine their action. Specific conditions as to which conventions are most likely to emerge from an amorphous state are clarified with computer simulations.

1 Introduction

The social evolution is the complex process made by interactions between individuals in the society. Game theory is commonly used when analyzing social evolution and there are two styles: one is the standard game theory and the other is the evolutionary game theory. The interpretation of the standard game theory is that the game is played exactly once between fully rational individuals. Evolutionary game theory, instead, assumes that the game is repeated many times by evolvable individuals chosen from large populations randomly. Much literature exists that investigates connections between the long run aggregate behavior and concepts of equilibrium solutions of the games [1].

The evolutionary game theory with bounded rationality [3] and local interaction must be used to construct an appropriate model for analysis of social evolution in real world, because the ability and the activity of each individual in society are not sufficient. So, the following two concepts form the basis of our analysis: (1) each agent interacts only with his neighbors, (2) each agent myopically behaves by imitating the behavior of his neighbors. We formalize these ideas in a model with a finite population of agents in which agents are repeatedly matched within a period to play a game.

One key concept for the analysis of the evolutionary game is the evolutionary stable strategy [2]. The predominant strategy is said to be evolutionary stable if there exists a barrier which prevents other strategies from invading. It is known that aggregate behavior converges to a Nash equilibrium that also satisfies the condition of the evolutionary stability [2]. The evolutionary stability can explain whether a strategy is robust to evolutionary pressures, but can not explain how a population arrives at such a strategy.

Another key concept for analysis of the evolutionary game is the replicator dynamics. This is a simultaneous differential equations of the number of agents who adopt some strategy. The description is an accurate, but we can only describe the strictly limited situation such as random matching with homogeneous agents.

In general, an evolutionary process has two basic elements: a mutation and a selection. The evolutionary process based on a biological mutation mechanism is very slow. However, social evolution may be fast because there exists a mechanism of mimicry, the transmission process of superior strategies from one head to another by imitation.

The criterion of evolutionary equilibrium highlights the role of mutations. The replicator dynamics, on the other hand, highlight the role of selection. Agent based simulation is a new methodology for studying an evolutionary games in the large that highlights the concepts of both mutation and selection. The purpose of this paper is to clarify the role of mimicry in social evolution.

2 The Model of Social Evolution

2.1 Local Matching

In order to describe the interactions among agents, we adopt local matching model. In this model, agents interact much more with their neighbors than with those who are far away [4]. They adapt other agents' successful strategies as guides for their own choices. Hence, their success depends in large part on how well they do in their interactions with their neighbors. Neighbors can serve another function: A neighbor can provide a role model. If the neighbor is doing well, the behavior of the neighbor can be imitated. In this way, successful strategies can spread throughout a population from neighbor to neighbor [5]. The position of each agent remains fixed in his locations, but their strategies can spread.

2.2 Learning Strategies

We assume that agents are less sophisticated and that they do not know how to calculate best replies. Agents are not so rational or knowledgeable as to correctly guess or anticipate the other agent's strategies.

An important aspect of social evolution is the learning strategy adapted by each individual. In the previous literature, agents are viewed as being genetically coded with a strategy and selection pressure favors agents that are fitter, i.e., whose strategy yields a higher payoff against the population. We consider two types of learning strategies: complete mimicry and partial mimicry. The latter is implemented as crossover in a genetic algorithm's words, we then call it crossover strategy. Each agent interacts with the agents on all eight adjacent squares and imitates the strategy of any better performing one. In each generation, each agent attains a success score measured by its average performance with its eight neighbors. Then if an agent has one or more neighbors who are more successful, the agent copy the strategy of the most successful neighbors in complete mimicry model and crossover his strategy with the strategy of the most successful neighbor in partial mimicry model.

2.3 Interactions Formulated as Games

Possible combinations of interactions can be formulated games as shown in Table 1. We classify those interactions into several types based on the payoff structures that describe interactions, and they are given the following special names in game theory. Later, we explain about a description of parameters (a, b, k) in detail; (1) The dilemma game (Table 1(a)), (2) The coordination game (Table 1(b)), (3) The mixed-motivation game (Table 1(c)).

Table 1: Local interactions formulated as games

(a) Dilemma game ($\beta \geq 1$, $\alpha \leq 0$)

	The other's strategy	
Own strategy	S_1 (Cooperate)	S_2 (Defect)
S_1 (Cooperate)	1 1	β α
S_2 (Defect)	α β	0 0

(b) Coordination game ($0 < k \leq 1$)

	The other's strategy	
Own strategy	S_1 (Cooperate)	S_2 (Defect)
S_1 (Cooperate)	k k	0 0
S_2 (Defect)	0 0	1 1

(c) Mixed-motivation game ($0 < k \leq 1$)

	The other's strategy	
Own strategy	S_1 (Cooperate)	S_2 (Defect)
S_1 (Cooperate)	0 0	k 1
S_2 (Defect)	1 k	0 0

3 The Implementation

3.1 Agents

Each agent's decision rules are represented as binary strings. A binary string of each agent consists of 22 positions (Table 2). Each position p_j, $j \in [1, 22]$, is represented as follows. The first and second position encodes the action that the agent takes at iteration $t = 1$ and $t = 2$. A position p_j, $j \in [3, 6]$, encodes the history of mutual hands (cooperate or defect) that the agent took at iteration $t-1$ and $t-2$ with his neighbor (opponent). A position p_j, $j \in [7, 22]$, encodes the rule of action that agent i takes at iteration $t > 2$, corresponding to the history (p_j, $j \in [3, 6]$). We consider agents' actions as follows: C = 0 and D = 1, and line up the actions of him and his opponent at $t - 2$ and $t - 1$. First position of rule (p_7) represent the action for (0000) = (CCCC), and second position (p_8) represent for (0001), . . . , and the last position represent for (1111).

Table 2: Example of genes of agent i (TFT)

	$t = 1, 2$		memory				strategy decision part (0 = C, 1 = D)															
p_j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
i	0	0					0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

3.2 Algorithm

The complete mimicry and partial mimicry algorithms are implemented like [8]. Details are as follows:

(Step1) An agent i, $i \in [1 \dots N]$, interacts with his neighbors at all eight adjacent squares, and the game is repeated for T iterations.

(Step2) An agent i chooses his action. His initial action, at $t = 1$ and $t = 2$, is determined by decoding bits in the position p_1 and p_2. For $t > 2$, $t \in [1 \dots T]$, an agent i chooses his action at t by using the information about his memories (position p_j, $j \in [3, 6]$) and strategy decision part (position p_j, $j \in [7, 22]$).

(Step3) The payoffs during T iterations of the games are accumulated.

(Step4) After T iterations, an agent i compares his payoff with his neighbors

(Step5) An agent i updates his binary strings using the genetic operator: [for complete mimicry] copy from, [for partial mimicry] crossover with, the neighbor who acquires the highest payoff for partial mimicry and mutation.

(Step6) if generation is *lastgen* then exit, else goto Steps 1.

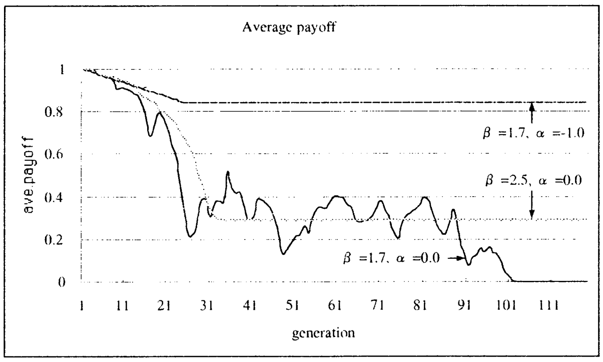
4 Simulation Results

Simulations are done in three types of games with both mimicry strategy and partial mimicry strategy (= crossover strategy). Agents are set in an area of 20×20 ($N = 400$ agents), with wrap-around. Parameters are as follows: $lastgen = 500$, $T = 10$, $mutate = 0.001$.

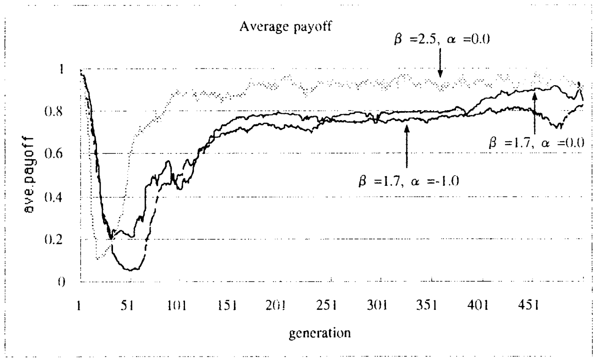
4.1 Social Dilemma Game

One agent located in a center who adopts All_D strategy and his strategy gives an influence to his neighboring agents. Fig. 1(a) represents the average payoff of each agent in mimicry case. As a result of this simulation, the population rapidly evolves but it is very unstable and the stability of social evolution depends on the parameters α and β.

On the other hand, The payoff of each agent with the partial mimicry case (= crossover strategy) are shown Fig. 1(b). For any combination of parameters (β, α), each agent can acquire higher payoff. We can also conclude that the social evolution with the crossover strategy is very stable. The evolutionary process with the crossover strategy doesn't depend on a combination of parameters (β, α). However, many generations are required to reach this desirable state compared with the former model.



(a)



(b)

Figure 1: Average payoff in Dilemma Game: (a) mimicry strategy, (b) crossover strategy

4.2 Social Coordination Game

In this game, we set up random strategy as the initial state of all agents in order to investigate whether an agent that has various strategies in the initial state cooperates mutually or not. The parameter $k = 1.0$ in Table 1(b). The result of simulation is shown in Fig. 2. In both cases of mimicry and crossover strategies, each agent is able to acquire a high payoff.

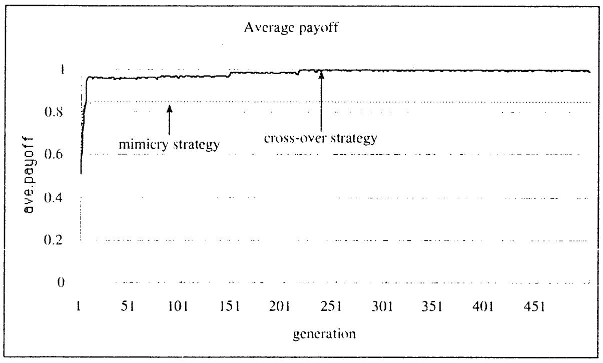


Figure 2: Average payoff in Coordination Game

4.3 Social Mixed-motivation Game

In this game, significant differences were observed between the mimicry strategy and the crossover strategy. Fig. 3 shows the case in which $k = 1.0$ in Table 1(c). we also investigate another k value. As a result, in the evolution with the mimicry strategy, there are many agents who persist with either strategy S_1 or S_2, and then, they can not acquire the payoff at high level. On the other hand, the evolution with the crossover strategy, whatever value we set for parameter k, each agent can acquire a higher payoff. Consequently, in the mixed-motivation game, we can say that evolution with the crossover strategy leads to a more efficient society than with the mimicry strategy.

5 Conclusion

The role of mimicry on social evolution is investigated by analysis of the interactions in a finite population of agents in which agents are repeatedly matched with their neighbors to play games. The hypotheses employed here is the limited ability of agents to receive, decide, and act upon information they get in the course of interactions. We analyzed social learning and showed how the society as a whole learns even when the individuals composing it do not. Especially, we examined how conventions evolve in a society from an amorphous state where there is no established custom and individuals rely on hearsay to determine what to do. With simulations, we find specific conditions as to which conventions are most likely to emerge.

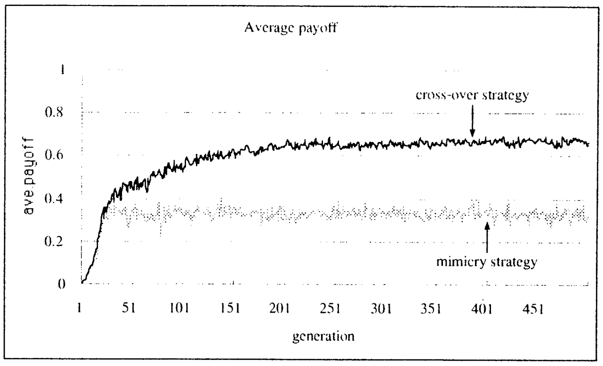


Figure 3: Average payoff in social mixed-motivation game

References

[1] Weibull J., *Evolutionary Game Theory*, The MIT Press, 1996.

[2] Smith J. M., *Evolution and the Theory of Games*, Cambridge University Press, 1982.

[3] Rubinstein A., *Modeling Bounded Rationality*, The MIT Press, 1998.

[4] Axelrod R., *The Complexity of Cooperation*, Princeton Univ. Press, 1997.

[5] Lloyd A. L., "Computing Bouts of the Prisoner's Dilemma," *Scientific American*, June, 1995.

[6] Yao X. and Darwen P., "The experimental study of N-player iterated prisoner's dilemma," *Informatica*, Vol.18:pp.435-450, 1994.

[7] Uno K. and Namatame A., "Evolutionary Behaviors Emerged through Strategic Interactions in the Large," *Proc. of the Genetic and Evolutionary Computation Conference*, pp.1414-1421, 1999.

[8] Lindgren, K., "Evolutionary phenomena in Simulating simple Dynamics," *Artificial Life II*, pp. 295-311, 1991

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

APPLICATION OF UNCERTAIN VARIABLES AND LOGICS TO COMPLEX INTELLIGENT SYSTEMS

Z. Bubnicki

Institute of Control and Systems Engineering,
Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, POLAND
Phone: +48 71 320 33 28, email: bubnicki@ists.pwr.wroc.pl

Keywords: uncertain variables, uncertain systems, intelligent systems, knowledge-based systems

Abstract

A class of complex uncertain systems with unknown parameters in the knowledge representation is considered. The unknown parameters are assumed to be values of uncertain variables. The decision problems for two-level system, for a complex of operations and for the system with a learning process are shortly described.

1. Introduction

In [1,2] the application of uncertain variables to decision making in a class of knowledge-based uncertain systems has been described. The purpose of this paper is to show how this approach may be applied to a class of complex knowledge-based uncertain systems and how it may be combined with a learning process described in [3,4,5,6].

2. Uncertain logics and variables

Consider a universal set Ω, $\omega \in \Omega$, a vector space X, a function $g\colon \Omega \to X$, and a crisp property $P(x)$ in the set X. The property P and the function g generate the crisp property $\Psi(\omega,P)$ in Ω : "For the value $\bar{x} = g(\omega) \overset{\ddot{A}}{=} \bar{x}(\omega)$ assigned to ω the property P is satisfied", i.e. $\Psi(\omega, P) = P[\bar{x}(\omega)]$. Let us introduce now the property $G(\bar{x}, x) =$ "$\bar{x} \cong x$" for $\bar{x}, x \in X$, which means: "$\bar{x}$ is approximately equal to x". For the fixed ω, $G[\bar{x}(\omega), x] \overset{\ddot{A}}{=} G_\omega(x)$ is a soft property in X. The properties $P(x)$ and $G_\omega(x)$ generate the soft property $\overline{\Psi}(\omega,P)$ in Ω : "the approximate value of $\bar{x}(\omega)$ satisfies P", i.e.

$$\overline{\Psi}(\omega,P) = G_\omega(x) \wedge P(x) = [\bar{x}(\omega) \cong x] \wedge P(x) \qquad (1)$$

where x is a free variable. The property $\overline{\Psi}$ may be denoted by

$$\overline{\Psi}(\omega,P) = \text{"}\bar{x}(\omega) \mathrel{\tilde{\in}} D_x\text{"}$$

where $D_x = \{x \in X\colon P(x)\}$, and "$\bar{x} \mathrel{\tilde{\in}} D_x$" means: "the approximate value of $\bar{x}$ belongs to D_x" or "$\bar{x}$ approximately belongs to D_x". Denote by $h_\omega(x)$ the logic value of $G_\omega(x)$:

$$w[G_\omega(x)] \overset{\ddot{A}}{=} h_\omega(x), \bigwedge_{x \in X} (h_\omega(x) \geq 0), \max_X h_\omega(x) = 1 .$$

Definition 1 (uncertain logic): The uncertain logic is defined by a universal set Ω, a metric space X, crisp properties (predicates) $P(x)$, the properties $G_\omega(x)$ and the corresponding functions h for $\omega \in \Omega$. In this logic we consider soft properties (1) generated by P and G_ω. The logic value of $\overline{\Psi}$ is defined in the following way

$$v[\overline{\Psi}(\omega, P)] = \begin{cases} \max\limits_{x \in D_x} h_\omega(x) & \text{for } D_x \neq \varnothing \\ 0 & \text{for } D_x = \varnothing \end{cases}$$

and is called a degree of certainty or *certainty index*. The operations for the certainty indexes are defined as follows:

$$v[\neg \overline{\Psi}(\omega,P)] = 1 - v[\overline{\Psi}(\omega,P)],$$

$$v[\Psi_1(\omega, P_1) \vee \Psi_2(\omega,P_2)] = \\ = \max\{v[\Psi_1(\omega,P_1)], v[\Psi_2(\omega, P_2)]\},$$

$$v[\Psi_1(\omega,P_1) \wedge \Psi_2(\omega,P_2)] = \min\{v[\Psi_1(\omega,P_1)], v[\Psi_2(\omega,P_2)]\}$$

where Ψ_1 is $\overline{\Psi}$ or $\neg\overline{\Psi}$, and Ψ_2 is $\overline{\Psi}$ or $\neg\overline{\Psi}$. □

Definition 2 (*C*-uncertain logic): The first part is the same as in Definition 1. The certainty index of $\overline{\Psi}$ and the operations for the certainty indexes are defined as follows:

$$v_c[\overline{\Psi}(\omega,P)] = \frac{v_p[\overline{\Psi}(\omega,P)] + v_n[\overline{\Psi}(\omega,P)]}{2} =$$

$$= \frac{1}{2}[\max_{x \in D_x} h_\omega(x) + 1 - \max_{x \in \overline{D}_x} h_\omega(x)],$$

$$v_c[\neg\overline{\Psi}(\omega,P)] = v_c[\overline{\Psi}(\omega,\neg P)],$$

$$v_c[\overline{\Psi}(\omega,P_1)\vee\overline{\Psi}(\omega,P_2)] = v_c[\overline{\Psi}(\omega,P_1\vee P_2)],$$

$$v_c[\overline{\Psi}(\omega,P_1)\wedge\overline{\Psi}(\omega,P_2)] = v_c[\overline{\Psi}(\omega,P_1\wedge P_2)]. \qquad \square$$

Definition 3 (uncertain variable): The uncertain variable $\bar{x}$ is defined by the set of values *X*, the function $h(x) = v(\bar{x} \cong x)$ (i.e. the certainty index that $\bar{x} \cong x$, given by an expert) and the following definitions:

$$v(\bar{x}\,\tilde{\in}\,D_x) = \begin{cases} \max\limits_{x\in D_x} h(x) & \text{for } D_x \neq \varnothing \\ 0 & \text{for } D_x = \varnothing \end{cases},$$

$$v(\bar{x}\,\tilde{\notin}\,D_x) = 1 - v(\bar{x}\,\tilde{\in}\,D_x),$$

$$v(\bar{x}\,\tilde{\in}\,D_1 \vee \bar{x}\,\tilde{\in}\,D_2) = \max\{v(\bar{x}\,\tilde{\in}\,D_1), v(\bar{x}\,\tilde{\in}\,D_2)\},$$

$$v(\bar{x}\,\tilde{\in}\,D_1 \wedge \bar{x}\,\tilde{\in}\,D_2) =$$

$$= \begin{cases} \min\{v(\bar{x}\,\tilde{\in}\,D_1), v(\bar{x}\,\tilde{\in}\,D_2)\} & \text{for } D_1\cap D_2 \neq \varnothing \\ 0 & \text{for } D_1\cap D_2 = \varnothing \end{cases}.$$

The function $h(x)$ will be called a *certainty distribution*.

□

Definition 4 (*C*-uncertain variable): *C*-uncertain variable $\bar{x}$ is defined by the set of values *X*, the function $h(x) = v(\bar{x} \cong x)$ given by an expert, and the following definitions:

$$v_c(\bar{x}\,\tilde{\in}\,D_x) = \frac{1}{2}[\max_{x\in D_x} h(x) + 1 - \max_{x\in \overline{D}_x} h(x)], \qquad (2)$$

$$v_c(\bar{x}\,\tilde{\notin}\,D_x) = 1 - v_c(\bar{x}\,\tilde{\in}\,D_x),$$

$$v_c(\bar{x}\,\tilde{\in}\,D_1 \vee \bar{x}\,\tilde{\in}\,D_2) = v_c(\bar{x}\,\tilde{\in}\,D_1\cup D_2),$$

$$v_c(\bar{x}\,\tilde{\in}\,D_1 \wedge \bar{x}\,\tilde{\in}\,D_2) = v_c(\bar{x}\,\tilde{\in}\,D_1\cap D_2) \qquad \square$$

The definitions of uncertain variable and *C*-uncertain variable are based on uncertain logic and *C*-uncertain logic, respectively.

Consider now a static system with input vector $x\in X$ and output vector $y\in Y$, described by a relation $R(x,y,c) \subset X\times Y\times C$ where $c\in C$ is an unknown vector parameter which is assumed to be a value of an uncertain variable $\bar{c}$ with $h_c(c)$ given by an expert.

Decision problem: for the given $R(x,y,c)$, $h_c(c)$ and D_y (where $y\in D_y$ is a desirable output property) find $\hat{x}$ maximizing $v(\bar{y}\,\tilde{\in}\,D_y)$. For the fixed *x*

$$v(\bar{y}\,\tilde{\in}\,D_y) \overset{Ä}{=} v(x) = v[\bar{c}\,\tilde{\in}\,D_c(x)] = \max_{c\in D_c(x)} h_c(c) \quad (3)$$

where $D_c(x) = \{c\in C:\ (x,y,c)\in R \rightarrow y\in D_y\}$ and $\hat{x} = \arg\max v(x)$. When $\bar{c}$ is considered as *C*-uncertain variable it is necessary to find v (3) and

$$v(\bar{y}\,\tilde{\in}\,\overline{D}_y) = v[\bar{c}\,\tilde{\in}\,\overline{D}_c(x)] = \max_{c\in\overline{D}_c(x)} h_c(c) .$$

Then, according to (2) with y in the place of c,

$$v_c(\bar{y}\,\tilde{\in}\,D_y) \overset{Ä}{=} v_c(x) = \frac{1}{2}[v(\bar{y}\,\tilde{\in}\,D_y) + 1 - v(\bar{y}\,\tilde{\in}\,\overline{D}_y)] \quad (4)$$

and $\hat{x}_c = \arg\max v_c(x)$. Let us note that in both cases (functional and relational) the solution $\hat{x}$ (or $\hat{x}_c$) may not exist or may be not unique (as in the deterministic system without *c*).

Example 1: Let x, y, $c\in R^1$, the relation R is given by inequality $cx \le y \le 2cx$, $D_y = [y_1, y_2]$, $y_1 > 0, y_2 > 2y_1$. Then $D_x(c) = [\frac{y_1}{c}, \frac{y_2}{2c}]$, $D_c(x) = [\frac{y_1}{x}, \frac{y_2}{2x}]$. Assume that c is a value of an uncertain variable $\bar{c}$ with triangular distribution $h_c(c)$ determined by $(0, \frac{1}{2}, 1)$. It is easy to note that $\hat{x}$ is any value from [$2y_1$, y_2] and $v(\hat{x}) = 1$. Using (4) we obtain

$$v_c(x) = \begin{cases} y_2(2x)^{-1} & \text{when} & x \ge y_1 + 0.5y_2 \\ 1 - y_1x^{-1} & \text{when} & y_1 \le x \le y_1 + 0.5y_2 \\ 0 & \text{when} & x \le y_1 . \end{cases}$$

It is easy to see that $\hat{x}_c = y_1 + 0.5y_2$ and $v_c(\hat{x}_c) = \frac{y_2}{2y_1 + y_2}$. E.g. for y_1=2, y_2=12 the results are the following: $\hat{x}\in[4, 12]$ and v = 1, $\hat{x}_c$ = 8 and v_c = 0.75. □

3. Knowledge representation and decision making problem for complex system

Let us consider a distributed system with a structure presented in Fig.1 where $u_j\in U_j$, $x_j\in X_j$ ($j\in\overline{1,k}$), $y\in Y$ are real number vectors. The lower level subsystems are described by relations

$$R_j(u_j, x_j; c_j) \subseteq U_j\times X_j , \quad j\in\overline{1,k}$$

where $c_j\in C_j$ is a parameter (real number vector), and upper level subsystem is described by a relation

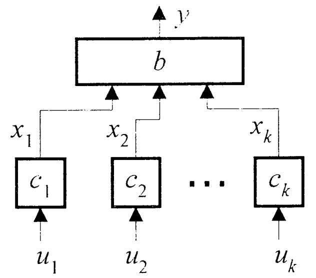


Fig.1

$$R(x_1, ..., x_k, y; b) \subseteq X_1 \times ... \times X_k \times Y$$

where $b \in B$ is a parameter (real number vector). The set of relations forms the knowledge representation of our system. We can formulate the following **decision problem**: For the given knowledge representation and the set $D_y \subset Y$ find the largest set $D_u \subset U$ such that the implication $\overline{u} \in D_u \Rightarrow y \in D_y$ is satisfied. In this notation $\overline{u} = (u_1, ..., u_k)$ is the input of the system as a whole, $U = U_1 \times U_2 \times ... \times U_k$ and $y \in D_y$ is a user's requirement. The decision problem for the system as a whole may be decomposed according to the decomposition of the knowledge representation in the distributed system.

1. The decision problem for the upper level: Given D_y, find the sets $D_{xj} \subset X_j$ such that the implication

$$(x_1 \in D_{x1}) \wedge ... \wedge (x_k \in D_{xk}) \Rightarrow y \in D_y$$

is satisfied.

2. The decision problems for the lower level: Given D_{xj}, find the largest set D_{uj} such that the implication $u_j \in D_{uj} \Rightarrow x_j \in D_{xj}$ is satisfied. In general, the solutions for the system as a whole and via the decomposition are not equivalent.

Assume now that the unknown vector parameters b and c_j are values of uncertain variables $\overline{b}$ and $\overline{c}_j$, respectively, described by $h_b(b)$ and $h_{cj}(c)$ given by an expert. The decision problem for the system as a whole is the same as in the previous section and consists in finding $\hat{\overline{u}}$ maximizing $v(\overline{y} \tilde{\in} D_y)$ for the given output. The direct solution of this problem may be very complicated and it may be reasonable to apply the decomposition.

1. **Upper level problem**: for the given $R(x_1, ..., x_k, y; b)$, $h_b(b)$ and D_y find $\hat{x}$ maximizing $v(\overline{y} \tilde{\in} D_y)$ where $x = (x_1, ..., x_k)$.

2. **Lower level problem**: for the given $R_j(u_j, x_j; c_j)$, $h_{cj}(c_j)$ and D_{xj} find $\hat{u}_j$ maximizing $v(\overline{x}_j \tilde{\in} D_{xj})$; $j \in \overline{1, k}$. The sets D_{xj} are such that

$$D_{x1} \times D_{x2} \times ... \times D_{xk} \subseteq D_x \qquad (5)$$

where D_x is a set of solutions $\hat{x}$ of the upper level problem.

We may assume the different sets D_{xj} satisfying (5) and obtain the different sets $\overline{D}_u$ of $\hat{\overline{u}}$ as the results of the lower level problem. In general, the results obtained for the system as a whole and via the decomposition may be different, i.e. $\overline{D}_u \subseteq D_u$ where D_u is the set of $\hat{\overline{u}}$ obtained for the system as a whole, $\overline{D}_u = D_{u1} \times ... \times D_{uk}$ and D_{uj} are the results of the lower level problem. It is easy to show that the direct approach and the approach via decomposition are equivalent (i.e. $\overline{D}_u = D_u$) if $D_{x1} \times ... \times D_{xk} = D_x$.

Example 2: Consider a simple production system with two subsystems on the lower level and onedimensional positive variables y, x_1, x_2, u_1, u_2: y is the amount of a final product, x_1, x_2 are the amounts of products and u_1, u_2 are the amounts of raw materials in the lower subsystems. The upper level is described by a relation $bx_1 \le y \le bx_2$, and relations for the lower level are reduced to functions: $x_1 = c_1 u_1$, $x_2 = c_2 u_2$. The certainty distributions for b, c_1 and c_2 are equal to 1 for b^*, c_1^* and c_2^*, respcctively. The requirement is determined by $D_y = [y_1, y_2]$. Then, for the upper level we obtain $D_b(x_1)$ and $D_b(x_2)$ in the form: $b \ge y_1 x_1^{-1}$ and $b \le y_2 x_2^{-1}$. According to the solution of the decision problem in sec.2

$$\hat{x}_1 = \arg\max_{x_1}[\max_{D_b(x_1)} h_b(b)], \quad \hat{x}_2 = \arg\max_{x_2}[\max_{D_b(x_2)} h_b(b)]$$

and finally the set of $\hat{x}_1$ i.e. D_{x1} and the set of $\hat{x}_2$, i.e. D_{x2} are as follows: $x_1 \ge y_1 b^{*-1}$ and $x_2 \le y_2 b^{*-1}$. Consequently, it is easy to obtain the results for the lower level: $u_1 \ge y_1 (c_1^* b^*)^{-1}$ and $u_2 \le y_2 (c_2^* b^*)^{-1}$. If these conditions are satisfied then the requirement concerning the amount of the product is satisfied with $v = 1$.

4. Complex of operations

The approach presented in the previous sections may be applied to a special case: a complex of parallel operations executed by a group of executors (operators or robots in a production system or computers for computational operations). The operations are described by relations $T_j \le c_j u_j$ where $T_j = x_j$ is the execution time and u_j denotes the size of a task (e.g. the amount of the raw material). The requirement is $y \le \alpha$ where $y = T = \max T_j$ is the execution time of the whole complex. The **decision problem** is as follows: for the given $h_{cj}(c_j)$ and α, find the task distribution

$\hat{u} = (\hat{u}_1,...,\hat{u}_k)$ maximizing $v[(\bar{y}\tilde{\in}[0,\alpha]) = v(\bar{y} \tilde{\leq} \alpha)$, subject to constraint $u_1 + ... + u_k = U$ where U is the size of the task to be distributed. It is easy to see that

$$v(\bar{y} \tilde{\leq} \alpha) = v(\bar{x}_1 \tilde{\leq} \alpha) \wedge ... \wedge v(\bar{x}_k \tilde{\leq} \alpha) = \min v_j(u_j) \overset{\text{Ä}}{=} v^*$$

where

$$v_j(u_j) = v(\bar{x}_j \tilde{\leq} \alpha) = \max_{c_j \in D_{cj}(u_j)} h_{cj}(c_j)$$

and $D_{cj}(u_j)$ is described by $c_j \leq \alpha u_j^{-1}$.

Example 3: Let $h_{cj}(c_j)$ have a triangular form (Fig.2). Then it is easy to obtain: $v_j(u_j) = 1$ for $u_j \leq 2\alpha(a_j + b_j)^{-1}$, $v_j(u_j) = 0$ for $u_j \geq \alpha a_j^{-1}$ and $v_j(u_j) = -A_j u_j + B_j$ otherwise, where $A_j = a_j(a_j + b_j)\alpha^{-1}(b_j - a_j)^{-1}$, $B_j = A_j \alpha a_j^{-1}$. For $k = 2$ decision $\hat{u}_1$ is found by solving the equation $v_1(u_1) = v_2(U - u_1)$. The result is as follows: (i) For $U \geq \alpha(a_1^{-1} + a_2^{-1})$ $v^* = 0$ for any u_1 . (ii) For $U \leq 2\alpha[(a_1 + b_1)^{-1} + (a_2 + b_2)^{-1}]$ the set of $\hat{u}_1$ is $[U - 2\alpha(a_2 + b_2)^{-1}, 2\alpha(a_1 + b_1)^{-1}]$ and $v^* = 1$. (iii) Otherwise, $\hat{u}_1 = (B_1 - B_2 + A_2 U)(A_1 + A_2)$ and $v^* = B_1 - A_1 \hat{u}_1$. E.g. if $U = 2$, $\alpha = 2$, $a_1 = 1$, $b_1 = 3$, $a_2 = 2$, $b_2 = 4$ then $\hat{u}_1 = 1.25$, $\hat{u}_2 = 0.75$ and for this distribution the requirement $T \leq \alpha$ is approximately satisfied with $v^* = 0.75$.

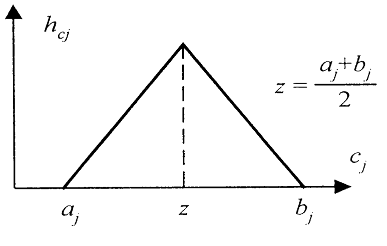


Fig.2

5. Learning process and conclusions

In [4,7,8] the learning process consisting in *step by step* knowledge updating has been described for the uncertain system under consideration. At each step the set of possible values of the unknown parameters b and c may be determined and used for the determination of the current decisions. It may be useful to combine the information obtained in the learning process with the information given by an expert in the form of certainty distributions. If at n-th step D_{bn} and D_{cn} denote the sets of the possible values of b and c, then $h_b(b;n)$ and $h_c(c;n)$ proposed at this step should take into account the forms of D_{bn} , D_{cn} and should be equal to zero outside these sets. Having $h_b(b;n)$ and $h_c(c;n)$ one can determine the current decisions $\hat{u}_n$. The approach presented here for two-level system may be extended to more complicated structures and to the logical knowledge representations [5]. Another extension may be developed for resources distribution in the complex of operations for a general formulation with so called soft variables, including uncertain variables and fuzzy numbers [9].

Acknowledgements

This work was supported by the Polish State Committee for Scientific Research under the grant no. 8 T11C 012 16.

References

[1] Bubnicki Z (2000), Uncertain variables in the computer aided analysis of uncertain systems. In: F. Pichler, R. Moreno-Diaz and P. Kopacek (eds), *Computer Aided Systems Theory. Lecture Notes in Computer Science.* Springer Verlag, Berlin, pp. 528-542

[2] Bubnicki Z (2000), Uncertain variables and their applications for a class of uncertain systems, *Int. J. of Systems Science* (in press)

[3] Bubnicki Z (2000), Learning algorithms in a class of knowledge-based systems. *Artificial Life and Robotics*, 4: 22-26

[4] Bubnicki Z (2000), Learning processes in a class of knowledge-based systems, *Kybernetes*, 29: 1016-1028

[5] Bubnicki Z (1998), Learning processes and logic-algebraic method in knowledge-based control systems. In: S. G. Tzafestas and G. Schmidt (eds) *Progress in System and Robot Analysis and Control Design. Lecture Notes in Control and Information Sciences.* Springer Verlag, London, pp. 183-194

[6] Bubnicki Z (2000), The application of learning algorithms and uncertain variables in the knowledge-based pattern recognition. Proc. V AROB Symp., Oita, Japan, pp. I-1 – I-5

[7] Bubnicki Z (2000), Knowledge validation and updating in a class of uncertain distributed knowledge systems. In: Zhongzhi Shi, Boi Faltings, Mark Musen (eds.), Proc. of 16th IFIP World Computer Congress. Intelligent Information Processing, Beijing, China, pp. 516-523

[8] Bubnicki Z (2000), Learning control system for a class of production operations with parametric uncertainties. In: Peter G. Groumpos, Antonios P. Tzes (eds.), Proc. IFAC Symposium on Manufacturing, Modeling, Management and Control, Patras, Rio, Greece, pp. 228-233.

[9] Bubnicki Z (2001), A unified approach to descriptive and prescriptive concepts in uncertain decision systems. In: Y. Lin (ed), *J. of Applied Systems Studies* (in press)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Simulation of Aerodynamics of Micro Air Vehicles TH380 Airfoils

Huaiyu Wu, Zhaoying Zhou, Shenshu Xiong, Xiaohao Wang, Guiqiu Bao, Sha Li, Zhuanxin Li
Department of Precision Instruments & Mechanology
Tsinghua University, Beijing 100084, P. R. China
E-mail: wuhy@post.pim.tsinghua.edu.cn

Abstract

A full-scale prototype of Micro Air Vehicle TH380 with spans of 380 mm at Tsinghua University is addressed. Computations of aerodynamic performances at Low Reynolds Numbers (Re<1 ×10^5) for a variety of airfoils are presented. An optimum airfoil used for Micro Air Vehicle is selected by comparison between the aerodynamic characteristics. The results of flight-testing of TH380 show that the computation of the typical characteristics at Low Reynolds Numbers is reliability.

Key words: *Micro Air Vehicle; MAV; Low Reynolds Numbers; Aerodynamic characteristics; Flight-testing.*

1. Introduction

Since America scientists proposed the concept of Micro Air Vehicle (MAV) at the future military meeting held by Defense Advanced Research Projects Agency in 1992, the study and development of Micro Air Vehicle has been carried out all over the world. For the time being, many countries included America, Israel, Germany, Australia, India, Japan and so on have established a variety of especial organizations and provided with a large mount of funds for the MAVs projects and relative topics[1].

This paper presents the full-scale prototype of Micro Air Vehicle TH380 at Tsinghua University in China. Computation and analysis of aerodynamic characteristics of several airfoils at Low Reynolds Numbers are given. After comparison between the performances is finished, an optimum airfoil for TH380 is obtained. Flight-testing of TH380 with the optimized airfoil shows that the computation and analysis are practicable for aerodynamic performances at Low Reynolds Numbers.

2. A Full Scale Prototype of Micro Air Vehicle TH380

2.1 Specifications

Basic specifications for the full-scale prototype include the spans of 380 millimeters or less, the speed of 10 meters per second, the payload of 30 grams or less and the flying time of 5-10 minutes.

Because of the limits of the maximum-sized airfoil, the wing payload per square centimeter on MAV would be much larger than that of the conventional wing. Based on the current situation, a triangle airfoil prototype MAV with no level trail wing is selected. The triangle airfoil can enhance the useful lift area and provide with the desirable lift within the finite power. However, the stability and the controllability of the MAV are more important than any conventional fixed wing air vehicles.

2.3 Basic Structure

After computations and comparisons of aerodynamic characteristics of several model airfoils are finished, an optimum airfoil for the triangle MAV TH380 with maximum wingspan of less than 380 millimeters is obtained, as shown in Fig. 1.

The MAV TH380 consists of body structure, battery-driven electric motor, propeller, speed governor, receiver, gyro, servos, cells, and antenna. The body structure is made of foamed plastics by especial handicraft. The two control flaps ailerons are hinged with the body trailing edge. Synchronous deflections serve as elevator and asynchronous deflections serve as rudder. The landing gear is not considered in order to reduce the general weight, and then the MAV is taking off by way of throwing.

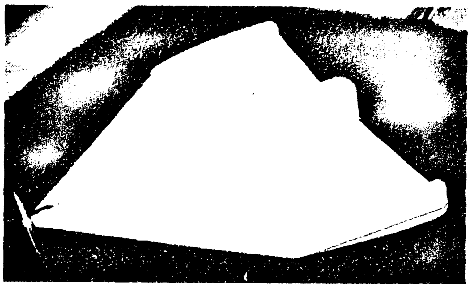


Fig. 1 A full scale prototype of Micro Air Vehicle TH380

3. Computation of Aerodynamic Characteristics

Airfoil aerodynamic characteristics comprise lift and drag. Theory researches show that aerodynamics F can be described as following[2, 3]

$$F = f(\alpha, shape, \rho, l, V_\infty, \mu, a)$$

where α is angle of attack, *shape* is related to the body structure, ρ is the density of air, l is a characteristic length of body, V_∞ is the free-stream velocity, μ is the coefficient of viscosity, a is the velocity of sound.

For low-speed flat-plate airfoil, the lift is a function of V_∞^2, the friction drag of laminar flow is a function of $V_\infty^{2/3}$, and friction drag of turbulent flow is a function of $V_\infty^{9/5}$. For the convenience of computations and analysis, both lift and drag are uniformly defined as a function of V_∞^2. For low-speed airfoil $(Re < 1\times10^5)$, lift coefficient, drag coefficient and moment coefficient can be written

$$C_y = f(\alpha, shape, Re)$$
$$C_x = f(\alpha, shape, Re)$$
$$C_{M_z} = f(\alpha, shape, Re)$$

where Re is Reynolds Number. *Shape* is related to wing area S, span length l, mean geometric chord c_m, aspect ratio λ, tapered ratio η, sweep back angle χ.

3.1 Zero-Lift Drag Coefficients C_{x0}

Drag coefficient when lift is zero is defined as Zero-lift drag coefficient C_{x0}. The zero-lift drag characteristics of MAV TH380 are shown in Fig. 2. As it can be seen from the figure, ① The thinner the airfoil thickness, the less the C_{x0}; The greater the airfoil camber, the greater the C_{x0}; This is because the changes of the thickness and the camber would modify the mean camber line and the boundary layer thickness; ② The greater the Reynolds numbers, the greater C_{x0}; The C_{x0} varies quickly when the Reynolds number is less than 5×10^4. Note that the change of C_{x0} would have direct influence over the stability of the MAV. Furthermore, it is obvious that zero-lift drag characteristics associated with the thickness of 10%c and the camber of 5%c on the condition of Reynolds number below 5×10^5 are moderately superior to others.

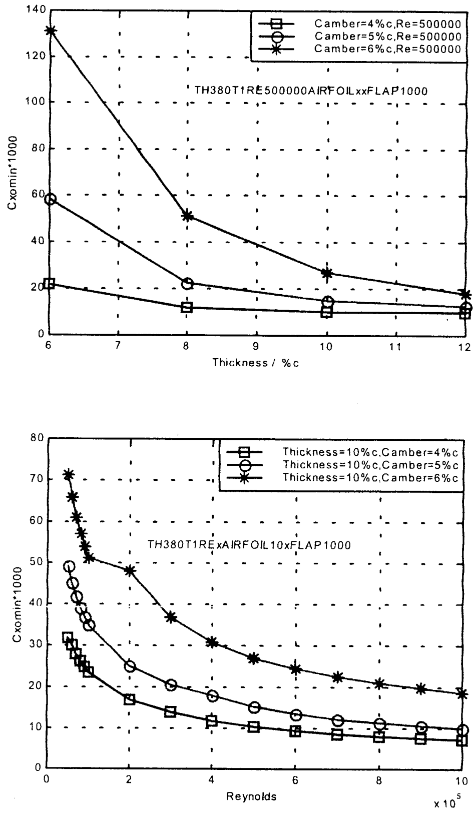


Fig. 2 Comparison of Zero-lift drag characteristics

a) Influence of thickness on C_{x0}

b) Influence of Reynolds on C_{x0}

3.2 Maximum Lift Coefficients $C_{y\max}$

Lift coefficient on the condition of stall speed angle α_s is defined as maximum lift coefficient $C_{y\max}$. The influence of thickness, mean chord line and Reynolds numbers on $C_{y\max}$ is demonstrated in Fig. 3. As it can be observed from the plots, ① $C_{y\max}$ is maximum when thickness is in the range of 10%c—12%c; ② An increment in camber would enhance $C_{y\max}$. This is because there are both adverse pressure gradient $(\partial p/\partial x > 0)$ and favorable pressure gradient $(\partial p/\partial x < 0)$ on the wing surfaces due to camber airfoil, and then there is no tendency of separation at a certain angle of attack; ③ The changes of $C_{y\max}$ in the range of Reynolds numbers of 5×10^4—5×10^5 vary sharply. The plots show that lift characteristics associated with the thickness of 10%c and the camber of 5%c is optimum.

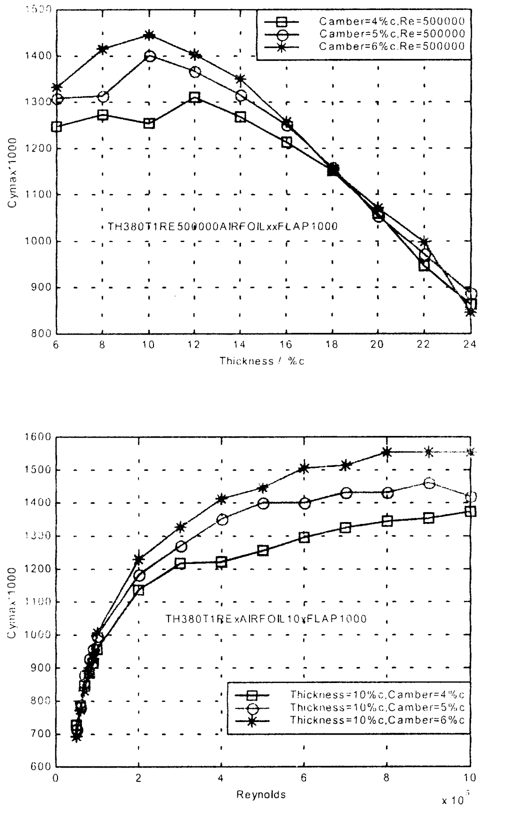


Fig. 3 Comparison of maximum lift characteristics

a) Influence of thickness on $C_{y\max}$

b) Influence of Reynolds on $C_{y\max}$

3.3 Maximum Lift Coefficients with Aileron $C_{y\max}$

According to thin airfoil theory on flaperon aerodynamics[4], lift coefficient could be estimated by

$$C_y = 2\pi\alpha + 2(\pi - \theta_h + \sin\theta_h)\delta_f$$

where δ_f is flap deflection in radian, θ_h is the location of flap axis in radian.

The aerodynamics for MAV TH380 with flaps is shown in Fig. 4. The curves reveal that ① the maximum lift $C_{y\max}$ would enhance as flap deflection increases; ② Under the condition of same flap deflection, the greater the Reynolds numbers, the greater the maximum lift $C_{y\max}$; ③ Under the condition of same flap deflection, the changes of flap length have a little influence on the maximum lift $C_{y\max}$. As it can be seen from the figure, the maximum lift $C_{y\max}$ with the thickness of 10%c and the camber of 5%c is appropriate.

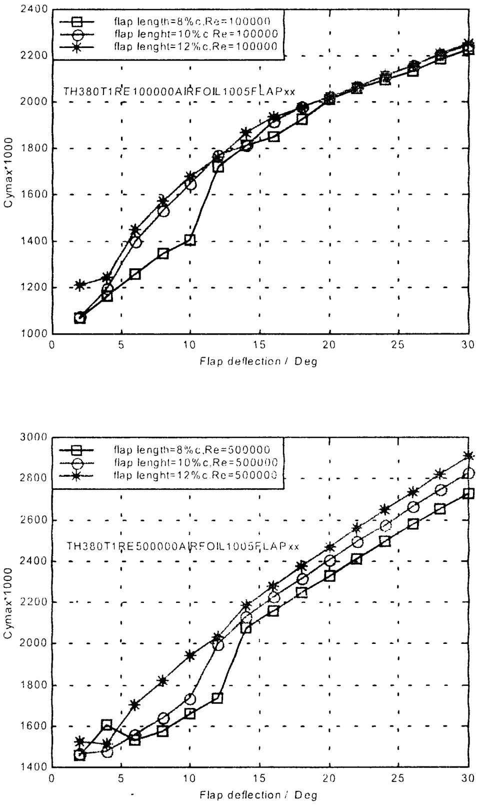


Fig. 4 Influence of flap deflection on $C_{y\max}$

a) Re=1×10^5 b) Re=5×10^5

3.4 Lift Drag Ratio $k = C_y / C_x$

Lift drag ratio is one important performance of aerodynamics. Owing to $k = C_y / C_x$, all factors which affect lift coefficient C_y and drag coefficient C_x could certainly have an influence on lift drag ratio *k*. The curves of lift drag ratio vs. angle of attack are presented in Fig.5. It can be seen that ① The thinner the airfoil thickness, the greater the C_y and the smaller the C_x , and then the greater the lift drag ratio *k*; ② The greater the Reynolds number, the greater the C_y and the smaller the C_x , and then the greater the lift drag ratio *k*; ③ On the condition of same angle of attack, C_x increases quickly in spite of slight increment in C_y , and then lift drag ratio *k* hardly varies. Furthermore, the roughness of airfoil surfaces is another important factor to have a certain influence on the lift drag ratio *k*.

Through comparisons and analyses of aerodynamics for MAV TH380, the thickness 10%c and the camber 5%c are superior to others and are practicable.

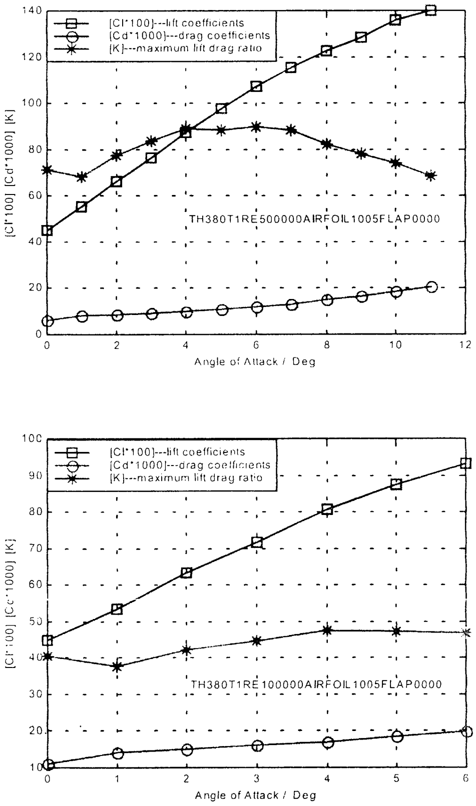


Fig.5 Comparison of lift-drag ratio

4. Flight Testing

Maiden flight-testing of MAV TH380 with triangle airfoil was finished in April 28, 2000 at Beijing in China, as illustrated in Fig. 6. The records of the flight with no payloads ratio remote control show that the speed of 8-10 meters per second, the height of 30-50 meters, the time of about 5 minutes and the distance of 3000 meters are gone through.

The MAV TH380 has the span of 380 millimeters and the weight of 154 grams which include power system (battery-driven electric motor, propeller and nickel cells) 88 grams in 57 percent of general weight, control system (transmitter, speed governor, gyros and servos) 29.8 grams in 19.3 percent of general weight, and body structure 36.2 grams in 23.5 percent of general weight. The MAV TH380 looks like small speck in the sky, which approaches in virtual silence. In flight, it also takes great advantages of its small size and low sound over other conventional aircrafts. One cannot hear any sounds of battery-driven electric motor while being 40-50 meters away from the MAV.

In flight-testing, the MAV TH380 without payloads was thrown into the sky by hand, and then propelled by battery-driven electric motor controlled by radio remote. One mode of synchronous-controlled double servos would be employed while uprising or descent. Another mode of asynchronous-controlled double servos would be used while heading or navigation. In flight, it is very desirable for MAV TH380 to perform level flight, swerving and hovering.

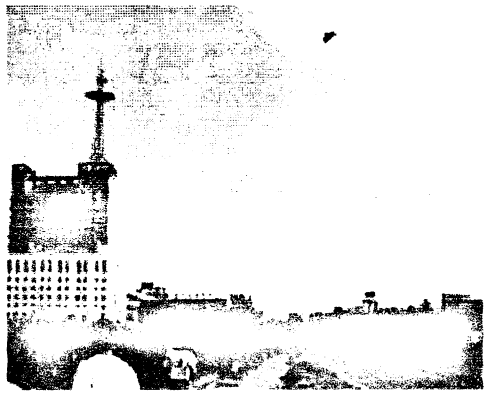


Fig. 6 Flight-testing grounds for MAV TH380

5. Conclusions

(1) Computation and analysis of aerodynamics play an important role in the design of airfoils for Micro Air Vehicle at Low Reynolds Numbers. The paper presents the comparison of the different performances for several airfoils and gives an optimum airfoil for MAV TH380. Furthermore, flight-testing results show that this airfoil is practicable for small-scale airplane.

(2) Besides the airfoil, the other key technologies of Micro Air Vehicles at Low Reynolds Numbers include microminiaturization of apparatus on MAV, higher efficient energy transfer, intelligent flying control and so on, which are being developed all over the world.

References

[1] Zhou Z, Wu H, Xiong S, *et al.*(2000), The research actualities and developing trends of Micro Aerial Vehicles. Journal of Test and Measurement Technology, 2000.A, 14: 724-728

[2] Wu H, Zhou H, Xiong S, *et al.*(2000), The actualities of Micro Air Vehicles and its key technologies. Journal of Wuhan University of Science and Technology, 2000,23(2): 174-178

[3] Barnes WM (1979), Aerodynamics, aeronautics, and flight mechanics. John Wiley & Sons, Inc.

[4] Thomas JM (1989), Low Reynolds number aerodynamics. Springer-Verlag Berlin, Heidelberg.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Study on the Optimal Design of a Mobile Robot based upon Mobility and Recurrency

Tae-Seok Jin
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea
jints@hyowon.pusan.ac.kr

Jang-Myung Lee
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea
jmlee@hyowon.pusan.ac.kr

Abstract

In this paper, We define the Mobility and Recurrency and acquire desired Mobility and Recurrency which result from the dynamic space change between two wheels of a mobile robot assunming that the robot has the fixed wheel radius. When a robot has a fixed interval between two wheels, it cannot avoid a sudden obstacle because of the constraint of Mobility and Recurrency.

The focus of this paper is on the instant Recurrency but high and the stable Mobility. That is, by changing the space between two wheels in simulation, the mobile robot could get the high Recurrency instantly and high Mobility with the stability.

Supposed that the Mobility and Recurrency which are defined in this paper are applied to the design of a mobile robot, We suggest a theoretical basis on the optimal design of the mobile robot. The experimental data support the validity of the aforementioned Mobility and Recurrency

Keywords.: Mobile robot, Mobility, Obstacle avoidance, Recurrency

1. Introduction

Recently, About an intelligent mobile robot, there are many investigations that are referred how to perceive environments, trajectory planing, collision avoidance and mapping technologies [8]. But, in the case of the technologies that are referred how to perceive environments, avoid a collision, it's a good performance on simulation. Because it's usually excluding the sensitivity of sensor. But according to the sensors sensitivity there are many differences between theory and real system. On this research, it is not simply to avoid collision considering just the structural design of mobile robot. We focused on the structural design of mobile robot that can compensate the processing speed to perceive obstacle, mobile robot's speed, and time delay during high speed driving. we made the mobile robot can move on good direction during driving curve, through optimal design of structure of mobile robot, considering interval between two wheels, and it's size. We will check the validity of result and apply to mobility and driving of mobile robot through simulation.

2. Structural characteristics of mobile robot

2.1 Construction of mobile robot

Fig.1. shows the construction of the mobile robot to make in this research. Existing mobile robot has a fixed interval between two wheels and so it can drive by changing wheel's speed but this mobile robot has a flexible axis, prismatic axis between two wheels, and so it can rapidly cope with obstacles by changing the space between two wheels, and has a good performance on the curve driving with a fewer revolution of the wheels.

2.2 Definition of Mobility

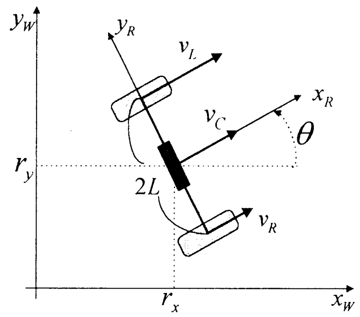


Fig. 1 Modelling and coordinates of a mobile robot for kinematic analysis.

In the Fig. 1, R is the radius of wheel, L is the length between wheel and center, V_C is the velocity of the weight center of mobile robot, w is the angular velocity of two wheels and

$$\dot{x} = \frac{R}{2}(v_R + v_L)\cos\theta \qquad (1)$$

$$\dot{y} = \frac{R}{2}(v_R + v_L)\sin\theta \qquad (2)$$

$$\dot{\theta} = \frac{R}{2L}(v_R - v_L) \qquad (3)$$

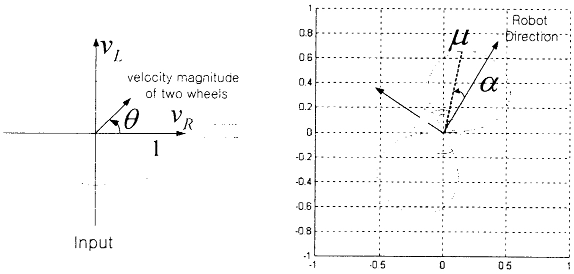
Also if p is present position, $\dot{p}$ is velocity of mobile robot, let us say Jacobian matrix is expressed as formula (4)

$$\dot{p} = J(q)\dot{q} \qquad (4)$$

In formula (4), $\dot{p} = [\, \dot{x} \;\; \dot{y} \;\; \dot{\theta} \,]^T$, $\dot{q} = [\, \dot{q}_r \;\; \dot{q}_l \,]^T$, Jacobian Matrix $J_{3\times2}$ is defined as formula (5).

$$J_{3\times2} = \begin{bmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta \\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta \\ -\frac{R}{2L} & \frac{R}{2L} \end{bmatrix} \qquad (5)$$

The velocity and angular as Fig. 3, for the input velocity vector that satisfied with unit circle of $v^2{}_L + v^2{}_R = 1$ using above formula (1),(2). Fig.3 shows Mobility curve, that relate output to every input direction, with graph.



(a) Unit input (b) Mobility Ellipsoid

Fig. 2 Mapping relationship between unit input and mobility curve.

In order to define Mobility, you can use the following formula using formula (1), (2), (3).

$$w = \frac{R}{2L}(v_R - v_L) \qquad (6)$$

$$v_C = \frac{R}{2}(v_R + v_L) \qquad (7)$$

For unit input, $v^2{}_L + v^2{}_R = 1$ the relation between v and w is defined as following formula (9)

$$\left(\frac{v + wL}{R}\right)^2 + \left(\frac{v - wL}{R}\right)^2 = 1 \qquad (8)$$

$$v^2 + w^2L^2 = \frac{R^2}{2} \qquad (9)$$

From formula (9), Mobility Measure (μ) that is defined as $\|v\|$ = 1 is followed.

$$\mu = \sqrt{\frac{R^2}{2} - w^2L^2} \qquad (10)$$

Therefore, you plot μ for $0 \le \mu \le \pi/2/\sec$, t curve forms are expressed as Fig. 2 (b)
Here, $w = \Delta\theta$ so you can express as following formulas

$$\mu = \sqrt{\frac{R^2}{2} - \Delta\theta^2L^2} \qquad (11)$$

3. Mobility Curve

If Mobility Measure (μ) that is defined from formula (9), is plotted for $0 \le \mu \le \pi/2$, there are two ellipses, that forms is said to curve.
In Fig. 2, Mobility curve shows mobility of mobile robot for unit input velocity vector. The larger, ellipsoid is the faster driving speed in curve of mobile robot become and so Mobility curve show the capability of curve driving. Following Figs show curve that express the capability of curve driving under two conditions that mobile robot which has one radius of wheel has L=1 and L=2 between center and wheel.

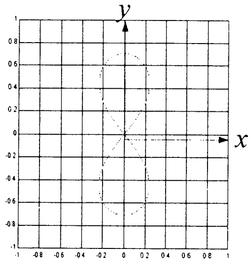


Fig. 3 Mobility Ellipsoid for L=1

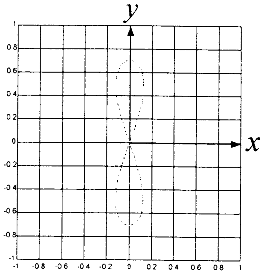


Fig. 4 Mobility Ellipsoid for L=2

Here, the length between center and wheel on Fig. 3 is shorter than that in Fig. 4, The shorter, the length become, the larger, the area of curve become and the better, the capability of curve driving get but the length L has a limit. if you know wheel's radius in the robot design, you can get the relation between the length L and the radius of wheel.

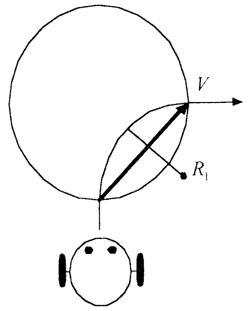


Fig. 5 Radius of curvature in $L < \frac{\sqrt{2}R}{\pi}$

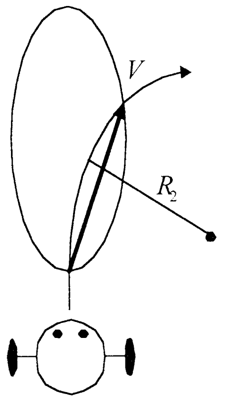


Fig. 6 Radius of curvature in $L > \frac{\sqrt{2}R}{\pi}$

When mobile robot that has same revolution speed of two wheels and progress with α angles between front and moving direction, the routes are different each other according to the length L between wheel and the center of robot. That is shown in Fig. 5, Fig. 6. The route radius of robot in Fig. 5 is R_1, and that in Fig. 6 is R_2. When the length between two wheels is short, route radius is short and so it can avoid a sudden obstacle with a fewer revolution of wheels on the same speed.

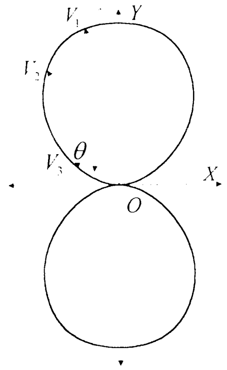


Fig. 7 Mobility curve For $L > \frac{\sqrt{2}R}{\pi}$.

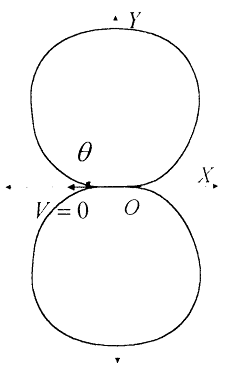


Fig. 8 Mobility curve for $L = \frac{\sqrt{2}R}{\pi}$

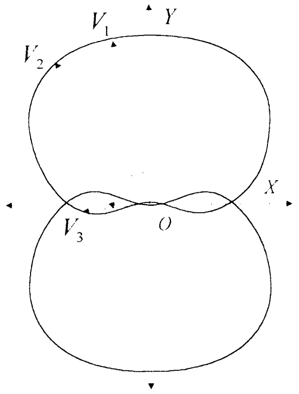


Fig. 9. Mobility curve for $L < \frac{\sqrt{2}R}{\pi}$

4. Optimal design

From Fig. 7, We can get the relation between the radius of wheel, R and the length L, using geometry characteristics of Mobility curve. From Fig. 8, in the case of $\theta \cong \pi/2$, that is to say two wheels of mobile robot revolve opposite direction each other, and so it can't progress, just revolve at the same position. At that time velocity vector $V = 0$ in the $-X$ direction using formula (12), you can get the relation between the radius and the length L of wheels.

Using $v^2 + w^2L^2 = \frac{R^2}{2}$ at $v = 0, w = \frac{\pi}{2}$, you get the following relation.

$$L = \frac{\sqrt{2}R}{\pi} \qquad (12)$$

Fig. 7 shows curve for $L > \sqrt{2}R/\pi$, and mobility of mobile robot become smaller for the same speed. Fig. 8 shows curve for $L = \sqrt{2}R/\pi$ that is optimal ratio at that time.

5. Simulation using Mobility Curve

Using above characteristic, Because the ratio of left and right revolution speed was chosen 1:4 voluntary, The result of simulations is shown in Fig. 10, Fig. 11. Fig. 11 shows the result of simulation at $L/R = \sqrt{2}/\pi$ on the base of Fig. 11, Fig. 12.

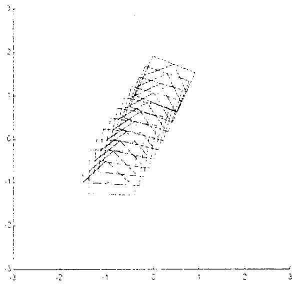


Fig. 10 Simulation of mobility curve in $\frac{L}{R} > \frac{\sqrt{2}}{\pi}$.

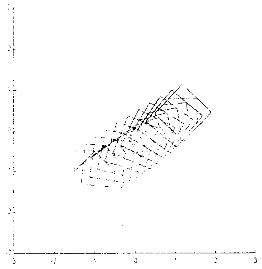


Fig. 11 Simulation of mobility curve in $\frac{L}{R} = \frac{\sqrt{2}}{\pi}$.

Fig 12 appears characteristics of mobile robot to escape form an obstacle. Here, path B represents progressing trajectory to cause an obstacle collision for changing velocity both a tire of left and right to the same moving velocity. However, it find that path A get accomplished to escape from an obstacle for changing velocity both left and right like path B naturally, as $L/R = \sqrt{2}/\pi$. to the interval of two tire when covering curve

In fig 13, it is able to find that it is possible to avoid collision and to reduce a turning radius with changing the interval of two tire for the general progressing path in regard to path A and B.

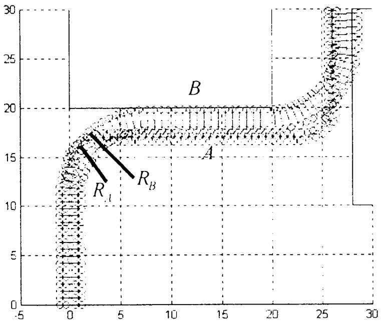


Fig. 12 Path characteristics of Mobile robot based upon L

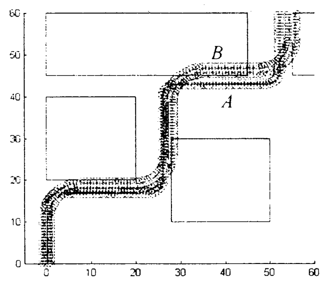


Fig. 13 Obstacle avoidance and path characteristics of Mobile robot based upon L

6. Conclusion

In this paper, considering the relation between the radius of wheels and the length of two wheels that has a prismatic axis. one velocity vector norm of two wheels was mapped to Jacobian of mobile robot. We make a drawing of velocity vector locus, that was defined by Mobility curve. It is possible to analyze capability of curve driving with the geometry model of Mobility curve. Then, If robot find out sudden obstacle during navigation, it compensate real time data process structurally. By changing the space between two wheels, it make radius small during curve driving. Therefore it can avoid a collision rapidly. Simulation shows that the mobility of mobile robot is determined by Mobility curve and mobile robot like to move in the direction of big curve. It is necessary to develop algorithm to control driving of mobile robot, using Mobility curve.

References

[1] Tsuneo Yoshikawa, "*Manipulability of Robotic Mechanisms*." The International Journal of Robotics Research. vol. 4, No.2, pp. 3-9, 1985.

[2] Stephen L. Chiu, "*Task Compatibility of Manipulator Postures*," The International Journal of Robotics Research, vol. 7, No. 5, pp. 13-21, 1988.

[3] Mark W. Spong, "Robot Dynamics and Control," John Wiley & Sons, 1989.

[4] J. C. Alexander and J. H. Maddocks, "On the kinematics of wheeled mobile robots," *Int. J. Robot. Res.*, vol. 8, no. 5, pp. 15-27, 1989; reprinted in *Autonomous Robot Vehicles*. I. J. Cox & G. T. Wilfong, Eds. New York: Springer-Verlag, 1990, pp. 5-24.

[5] J. C. Alexander, J. H. Maddocks and B. A. Michalowski, "Shortest Distance Paths for Wheeled Mobile Robots," *IEEE Trans. Robotics and Automation*, vol. 14, NO. 5, October 1998.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Autonomous Robot Navigation and Dynamical System Tasks Architecture

Maki K. Habib
GMD-Japan Research Laboratory
AIM Building 8F, KokuraKita-Ku, Kitakyushu-City, JAPAN
maki.habib@gmd.gr.jp

Abstract:

Several approaches have addressed the issue of robot control architecture and tried to evaluate its relation to robot's performance in dynamic and complex environment. Different properties of real environment and robot's task impact robot's controller and therefore the choice of control architecture. Understanding issues related to robot's behavior and their interrelation such as, representation, specification, behavior content and time scale for action, interaction, and coordination are essential too. This paper discusses the main approaches in the field of robot control architecture with reference to autonomous robot navigation. It focuses on the challenges, needs, fundamental issues along with the requirements that enable a mobile robot to function autonomously, intelligently, and reliably through unstructured and dynamic real environments.

Keywords:

Autonomous navigation, Robot control architecture, Deliberative planning, Reactive systems, Hybrid control, Behavior based systems, Learning, Robot behavior.

1. INTRODUCTION

Intelligent and autonomous robots that can work in physical world and carry out useful and successful tasks are essential due to their immediate applicability in a variety of fields such as space missions, operations in hazardous environments, underwater, scientific exploration, disaster areas, and other services. Research on mobile robots focuses on the realisation of flexible control architecture that enable the robot to move safely, function autonomously, intelligently, purposefully, and robustly through complex and dynamic real environments that have not specifically prepared for them at design time.

The ability for an autonomous mobile robot to interact, navigate and adapt intelligently in uncertain and dynamic environments represents the core functionality any autonomous mobile robot should be endowed with. At the same time it poses a series of issues that need to be addressed. Real environments contain elements of variability that limit the use of prior information. The robot must be able to operate and act under conditions of imprecision, uncertainty and incomplete information about the environment in a timely manner. Imperfect sensors and the inherent difficulty of the perceptual interpretation process affect robot interaction. Furthermore, the execution of control commands is not completely reliable while the dynamics of real environments is complex and unpredictable. To cope with these difficulties, robot s controller must be able to respond to unforeseen events as soon as they are perceived [2-5,10-12].

Autonomy requires systems that are not only capable of controlling robot's motion in response to sensor inputs by sensing the situation, but that are also capable to react to unexpected events and accordingly change course if necessary by efficiently deciding the next action to be performed while considering multiple conflicting objectives simultaneously. Moreover, it should be able to overcome errors in perceptions and actions. Several approaches have been developed to address these important issues at various levels in mobile robot's control architecture.

Autonomous mobile robots have to be adaptive to different kinds and complex of environments and have to deal with complex task. Hence, control architecture becomes very important. Different properties of real environment and robot's task impact robot's controller and therefore the choice of control architecture. Task requirements can constrain the architecture choice.

Understanding issues related to robot's behavior and their interrelation such as, representation, specification, behavior content and time scale for action, interaction, and coordination are essential to have reference criteria for selecting a control architecture and allowing to compare and evaluate different architectures relative to specific robotic designs, tasks, etc. Such criteria may include ability to execute parallel behaviors simultaneously, runtime flexibility for adaptation and learning, time scale in relation to real time reasoning and actions, action selection mechanism, modularity and the way to use abstraction, robustness and fault tolerance, programming environment and development tools, etc.

This paper discusses the main control architecture approaches in the field of robot in terms of strength, drawbacks and the relation between them while focusing on the critical issues that need to be solved. Behavior based systems are introduced as potential approach that present flexible and reliable solution to robot navigation in regard to dynamical system task architecture.

2. ROBOT CONTROL ARCHITECTURE

Different architectures have been proposed and investigated. These architectures range from symbolic artificial intelligence planners to the increasingly popular behavior based approaches. These include deliberative planning that featured by sequential capabilities of sense, plan, and act and longer time scale. Reactive system that maps perception of robot's world into actions and responds to real-time requirements of the environment. Hybrid control architecture combines layered organization with behavior based decomposition that enable to plan slowly and react quickly, i.e., combining two time-scales in an effective way. Finally, behavior-based systems that attempt to bring the different time-scales closer together by distributing slower computation over concurrent behavior modules.

2.1 Deliberative Planning Control Systems

Deliberative planning is a prerequisite for several capabilities that are often instantiated in an intelligent robot to possess. In almost all cases under such architecture, tasks are expressed as goals to be achieved and a robot must then mix up a series of coordinated actions designed to achieve these goals. Planning based architectures typically involve three generic sequential functional modules, sensing, planning and acting. Planning approach is hierarchical, its information processing functionality decomposed into categories, and based on symbolic planner. Deliberative architecture has to create elaborate models of the environment that may be acquired off-line or be built on-line and reason about how to accomplish their goals. Coarse model may be provided a priori and the model may then by elaborated on-line at the location where interaction is requested.

Most of the early work concerning navigation of mobile robots uses pure symbolic representations of robot s environment to plan and perform navigational tasks. The representation must contain all information needed for planning. Different solutions have been proposed to solve robot s navigational task, such as complex mathematical models about the robot and the environment, statistical methods for extracting information from sensor readings, different kind of maps that reflect robot s environment, logic based reasoning about the map, search algorithms for use on the maps, etc. [29, 30]. The most powerful aspect of these systems is they lend themselves to theoretical analysis and verification of properties such as plan correctness, optimality, etc. Problem solving relies heavily on planning, as most approaches to problem solving consist of incremental movements toward a solution.

Planers assume complete and accurate knowledge of the environment and the robot s sensing and actuating capabilities to generate and execute a correct plan. It assumes the environment does not change during the execution of a plan in a way that affects the plan. It relied on a central world model that required to continual update on the bases of sensory information to generate actions in the real world. Due to vulnerability of this control architecture to the noise and the perturbation that are always present in the real world and the expansion of the search space, a real robot is going to face major difficulties. Limitations of this approach quickly became obvious in which planning control systems is slow with unbounded time constraint and encourage open loop plan execution which is dangerous and have higher probability for reaching unanticipated situations and fail to behave properly in the face of uncertainty. This leads to a conclusion that the resolution of any planning activity is limited by the resolution of the knowledge available at the planning time. Also, it is important to note that no amount of domain knowledge will be sufficient if a robot is expected to learn to cope with novel situations not covered in its domain knowledge. They also require that the designer of the system know beforehand what properties the environment has since the robot s behavior is, to a large extent at least explicitly programmed or modeled.

Problems of this approach can be pointed out as representation and response problems, which can be considered as a temporary ones and solvable with the advance of hardware technologies [32]. But, other researchers such as Brooks do not share this view. His view in rejecting the planning approach related to the principle of intelligence and the information-processing paradigm [16].

An intelligent robot should update its plan when it learns new information that helps it accomplish its current goal more quickly, and it should be ready to respond or to re-plan when the plan fails. Learning and planning have a reciprocating relationship wherein planning creates a new method for carrying out a task, which can then be learned for future use by the planner.

2.2 Reactive Control Systems

Architectures that are designed for a dynamic environment must be able to respond quickly to changes in the environment. This requires the architecture to interrupt its behavior and act accordingly.

In response to the limitations and problems associated with the planning based approach, attempts were made to abandon the planning approach and start to focus on developing new control approach that is capable of interacting with a dynamic environment and better meet the demand of real time performance [16-20]. Accordingly, reactive based robot control architecture was introduced to demonstrate navigation capabilities that are quicker and better than those of planning systems. As the name already suggests, reactive controllers response to unexpected events in robot s environment through coupling of perception and action. The design problem initially is to provide a robot with behaviors that enable it to do its job and ensure its immediate safety as other additional skills can be added gradually. This approach reflects the incremental nature of its design process at each layer of behavior before integrated with the previous one. Purely reactive systems do not use any internal representations of the environment, while they have multiple task modules that react directly to sensory information through collections of rules that map specific situations to specific actions by combining sensors input with state information to produce motor control outputs and state changes.

Under reactive system the perceptual world is divided into mutually exclusive or unique situations, then only one of those situations can be triggered by any sensory input at any one time, and only one action will be activated as a result. Accordingly, reactive systems are built as multiple independent tasks that operate in parallel. Each behavior is responsible for a particular task. A behavior processes its own sensory information and issues its own motor commands. In order to co-ordinate the final motor commands, each behavior can disable the motor commands of the other behaviors that are in conflict with itself based on fixed priorities. Intelligent behavior is achieved through coordination of a set of purposive perception-action units, called behaviors.

This approach aimed at obtaining the desired action without any use of plans. While plans are not represented explicitly in such systems, they are in some sense implicitly designed in to the system through the pre-established interactions between behaviors, i.e., the designer has the responsibility to wire in the required mappings between conditions and actions. Thus, global system behavior is thus required to merge from the interaction between layers. At this level the designer needs to predict these interactions and at each level ensure that the new behavior will integrate properly with the whole system

A particular form of reactive control using a hierarchy of rules or layers is Brooks s subsumption architecture [16]. The resulting control is distributed where each behavior is concurrently active with no notion of a centralised planner. In subsumption architecture, rules are assigned priorities, with high priority rules capable of overriding low-priority rules that would otherwise be enabled. This allows the programmer to develop several levels of competence for the robot. The lower priority rules associated with lower levels of competence, define simple behaviors, while the higher priority rules, associated with higher levels of competence, define more complex behaviors

Reactive control approach has created new set of issues that need to be addressed, such as

- Co-ordinating behaviors into complex system and efficient consideration of conflicts and competitions among multiple behaviors. This is because, reactive systems consists of a number of distributed sensor-motor processes, thus mechanism should be devised to deal with scheduling, management, co-ordination of and communication between these modules so that coherent behavior can be achieved [6,32].
- Specification and design of behaviors, i.e., how to select a set of appropriate behaviors for a given task.
- Pure reactive systems are not able to learn and adapt their behaviors to become skilful in their action.
- Choosing the most appropriate action or to coordinate the behaviors to produce a rational next action.
- Behaviors based approaches relying on the subsumption architecture are not truly adaptive and their behaviors are determined at design time and selection between them are determined by rigid schemes [31].

It is manageable to build and demonstrate methods at the survival layer of reactive systems while higher level layers require much more information and they are correspondingly. much harder to demonstrate. The complexity of behavior that can be achieved using solely reactive actions is still a debated issue.

Afford modular development and provide for incremental growth. Also, no formal guarantees can be made about goal achievement or global optimality of actions.

2.3 Hybrid Control Systems

Planning systems can generate efficient plans and can integrate world knowledge while they lack the ability to efficiently handle changes in the environment unpredictable situations and system complexity. Reactive control systems by sensing repeatedly and acting accordingly provide real time robust performance in dynamic environments but, they cannot guarantee efficient goal achievement, and their ability to demonstrate more sophisticated behaviors seemed limited. Due to the limitations associated with planning and reactive approaches, researchers have their focus on exploiting the benefits of each approach and develop hybrid architecture that integrate the planning and the reactive components to complement the other by having the best of both and compensates for the other s deficiencies. The goal is to combine closed-loop and open-loop execution that implies combining the different time-scales and representations of both. Planning systems usually use abstract symbolic representation of physical objects whereas reactive based systems usually operate on raw sensory information with limited processing. How to close the gap between planning and reactive system components are thus an important issue, which has received a considerable amount of attention in the literature [4, 21-25]. Planning and reactivity are not mutually exclusive characteristics of robot s control architecture. But, they serve complementary purposes while they are implemented in different ways. Interaction between the components of planning system and that of reactive system is an important architecture issue and should be done asynchronously. This allows the fast reactive components to react in real-time whereas the slow planning components can run at slower mode without affecting the system fast response to run time contingencies. In this way the reactive components are able to attend to events as they unfold and trigger appropriate responses in a timely manner, since reactive system components have little or no processing and introduce a tight sensor-to-actuator coupling

Some of the work in hybrid systems deals with integrating planning and reactive based system components. Others, do not exactly integrate the two components, rather they devise new approaches which combine the best of the planning and reactive behavior-based approaches. Another direction considers one of the approaches, either planning or reactive, and tries to push its limits towards the other.

Hybrid method is the hierarchical approach to the navigation problem. It is typically consists of three layers that include standard reactive, planner and a third layer that combine them together, and it is therefore often called three-layer architecture. According to this approach the task is solved at different levels of abstraction, in effect creating several subtasks at the different levels. The logical architecture of most hybrid systems is layered according to decreasing levels of deliberation from top to bottom. The level of abstraction also decreases from top to bottom with highly level planning components operating on symbolic description and the low-level reactive behaviors operating at the signal level. The intermediate layer tries to compensate for the limitations of both the planner and the reactive system by reconcile their different time-scales and different representations while trying to reach to a proper way to resolve contradictory commands between the other layers. Also, the intermediate layer may cache and look to avoid re-planning and reusing previous plans

In Hybrid architectures, rule-based systems, fuzzy logic, neural network, or combination of them typically takes care of low-level tasks. Task that can be achieved at this level often include, obstacle avoidance, wall and corridor following, place recognition, steering signal generation from abstract tasks, e.g., turn. Many of the computationally expensive parts of the system can be done efficiently using neural network. High level tasks of such systems are still symbolic. They symbolic part of the system concerns the abstract task of navigational decision making which they are more suitable for than sensor reading interpretation and other low level tasks.

3. Behavior based Control systems

An intelligent and autonomous robot must be capable to deal with the unexpected and adapt its behavior to meet the new situation. To make rational decisions through proper selection of actions a robot needs to have access to the state of the world and to its own experiences. The critical question is how to make such experience available to the robot? Whether to have such experience explicit in terms of centralized representation or implicit representation with world states distributed through the related behaviors [30-35]. Intelligent systems should exhibit emergence property that is not designed into any of its individual sub-components. This property is important to the behavior based paradigm that ensure the kind of robustness that traditional based systems fail to exhibit when face the richness in the real world that can not be captured by a priori categorization.

Behavior based architectures is gaining increasing potential as new control paradigm that supports autonomous navigation where it addresses the fundamental Artificial Intelligent (AI) issues of sensing, thinking, and acting in real-time and presents a successful approach to solving situated AI problems.

One of the main challenges of autonomous robotics is to devise a theoretical framework for analysis, design and synthesis of systems that have specified properties, performance and behavior. In the behavior based system synthesis, a complex control task is divided into a set of simpler control problems that are implemented by behaviors, i.e. behavior based system uses behaviors as the underlying module of the system that implies the decomposition of robot control problem. Behavior based consist of a collection of rules, taking inputs from sensors or another behaviors in the system, and sending outputs to the actuators. Based on the activation conditions that depend on the state of world and task, the inputs determine the activation level of a behavior [30]. These are often post-conditions of other existing behavior that allow the description of complex temporal sequence. Behavior based systems are not limited in the ways that reactive systems are. As a result, it has key properties expressed in the ability to react in real-time, to express behaviors with various representations, and with its flexible structure it can accommodate reactivness and representation without the need for intermediate layer. Behaviors are feedback controllers that are generally executed concurrently and can achieve and/or maintain the goals of the system individually or as group, thus achieving the task. They are typically higher-level than actions. In building networks of behaviors, behaviors have the ability to take inputs from other behaviors and send outputs to other behaviors. Combining primitive behaviors into one system to enable the creation of complex behaviors in which the complex behavior emerges. Unexpected emergent behavior may be undesirable and also the emergent functionality may carry with it compound problems that could parallel the difficulties in developing the system.

Behaviors typically produce a single-valued output that is viewed as best to perform an action, and accordingly, there is a need to decide what action is the most appropriate. But, the generated actions by different behaviors may be contradictory and hence a mean to resolve conflict situations is needed. If each behavior generates a multi-valued output a more moderate composition scheme could be constructed. Multivalued outputs seem to make an appropriate basis for communication among the behaviors, and to reduce the amount of loss of information. In this regard, the semantics of the behavior outputs or the action selection process is not very clear. A related problem to the action selection is the reliability of the decisions the robot makes

Current behaviors lack the abstract representation that would allow them to be employed at high level and hence they have been not used so much for complex problems involving sequence of behaviors. The key challenge is in how representation in any form of world model can be effectively distributed over the behavior structure. The representation must be able to act on a time-scale that is close if not the same as the real-time parts of the system. Similarly, the representation needs to use the same underlying behavior structure as the rest of the system. Not every part of a behavior-based system needs to be involved with representational computation, i.e. behaviors can be separated into representation based and primitive ones. Most of currently available behavior based system can not have the capability to generate behaviors automatically and they have been typically constructed by hand for each task. Also, they lack the generality that requires system redesign from one task to another even if the underlying behavior remain unchanged [29,30,32].

Most of the approaches have centralized control. Distributed approaches tend to be more robust to failures of single system components and enable graceful degradation. Behavior based systems enable modular system development and this support the trends to choose a decentralized control scheme, but distributed systems add the overhead of coordination, synchronization and communication between system components, which must be overcome by powerful software engineering tools.

Important issues are still facing behavior-based systems and need to have proper solution, such as, how to design simple behaviors that guarantee robust operation in spite of the limited knowledge available at design time, how to coordinate the activity of several and possibly competing behaviors in order to perform a complex task, etc.

4. CONCLUSIONS

A control architecture that enables a robot to function autonomously, purposefully, and reliably through unstructured and dynamic world represents a fundamental milestone towards intelligently behaving robots. The paper has shown that deliberative planning and reactive control is equally important for robot navigation. Each complements the other and compensates for the other s deficiencies, and there is a need for synergetic integration between them. The challenge of hybrid systems is how to achieve the right compromise between the two. Behavior-based systems show promising solution that are flexible, robust to noise/uncertainty and failures, and can deal with the unexpected and adapt its behavior to meet real time requirements. But, there are many issues waiting for efficient solutions such as, behavior generation, scalability of the developed approach when the environment and task become complex, multi-valued actions and action selection, behaviors structure and the distribution of representation about the environment, behaviors and benefiting from experience, distributed approach in behavior control, competing behaviors and resolving conflicts, emergent behaviors and avoiding the undesirable, how complex behavior can emerge from primitive one, the generality of the approach to suit different task, etc.

REFERENCES

[1] Wang L, Tsai WH (1991), Collision Avoidance by a Modified Least-Mean-Square-Error Classification Scheme for Indoor Autonomous Land Vehicle Navigation, Journal of Robotic Systems, Vol. 8, No. 5, pp. 677-698.

[2] Gat E (1995), Towards Principled Experimental Study of Autonomous Mobile Robot, In proceedings of the International Symposium of Experimental Robotics, Stanford CA.

[3] Nehmzow U, Matsui T, Asoh H (1998), Virtual Coordinates: Perception-Based Localisation and Spatial Reasoning in Mobile Robots, Proceeding of the International Conference on Intelligent Autonomous Systems 5, Sapporo/Japan.

[4] Yen J, Peluger N (1995), A Fuzzy Logic Based Extension to Payton and Rosenblat s Command Fusion Method for Mobile Robot Navigation, IEEE Trans. on Systems, Man and Cybernetics, Vol. 25, No. 6, pp. 971-978.

[5] Skarmeta AG, Barber HM (1997), Fuzzy Logic Based Intelligent Agents for Reactive Navigation in Autonomous Systems, Fifth International Conference on Fuzzy Theory and Technology (FT&T'97), Durham, USA, pp. 168-171.

[6] Skarmeta AG, Barber HM, Alonso MS, (1999), Learning Behavior Fusion in Autonomous Mobile Robots", IX ESTYLF'99, Mallorca, SPAIN.

[7] Frank M (1990), Strategies for Intelligent Agent Exploration of Complex Environments.

[http://medg.lcs.mit.edu/mpf/ papers/Frank/Frank-90a.html]

[8] Wilkins DE, Myers KL, Lowrance JD, Wesley LP (1994), Planning and Reacting in Uncertain and Dynamic Environments, Journal of Experimental and Theoretical AI, Vol. 6, pp. 197-227.

[9] Saffioti A (1997), Handling Uncertainty in Control of Autonomous Robots , Technical Report TR/IRIDIA/97-8, Univ. Libre de Bruxelles.

[10] Cattoni R (1993), Behaviours as Bridges Between Symbolic Reasoning and Reactivity , In Proc. of the Italian Workshop on Planning IPW'93, Italy.

[11] Van-Dam J (1998), Environment Modeling for Mobile Robots: Neural Learning for Sensor Fusion, Ph. D. Thesis, University of Amsterdam.

[12] Lacroix S, Flury S, Haddad H etal (1998), Reactive Navigation in Outdoor Environments, International Journal of Robotics Research.

[13] Moravec HP (1982), The Stanford Cart and the CMU Rover , Autonomous Mobile Robots, pp. 407-419.

[14] Kosaka A, Kak AC (1992), Fast Vision Guided Mobile Robot Navigation Using Model Based Reasoning and Prediction of Uncertainties, CVGIP: Image Understanding, Vol. 56, No. 3, pp. 271-329.

[15] Pirjanian P, Christensen HI (1995), Modelling and Planning for Sensor Based Intelligent Robot Systems, World Scientific, Chapter Hierarchical Control for Navigation Using Heterogeneous Models.

[16] Brooks RA (1986), A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 14-23.

[17] Arkin RC (1987), Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior , IEEE ICRA 87, pp. 264-271.

[18] Maes P (1989), How to Do the Right Thing, Technical Report NE43-836, AI-Laboratory, MIT.

[19] Rosenblatt JK, Thorpe CE (1995), Combining multiple goals in Behavior Based Architecture , Proc. of IROS 95.

[20] Mataric M (1997), Behavior-Based Control: Examples from Navigation, Learning and Group Behavior , Journal of Experimental and Theoretical Artificial Intelligence, Vol. 9, No. 2-3, pp. 323-336.

[21] Payton DW, Rosenblatt JK, Keirsey DM (1990), Plan Guided Reaction , IEEE Trans. on Systems, Man, and Cybernetics, Vol. 20, No. 6, pp. 1370-1382.

[22] Arkin RC (1989), Towards the Unification of Navigational Planning and Reactive Control, AAAI 90, 1989.

[23] Gat E (1992), Integrating Planning and Reacting in a Hetrogeneous Asynchronous Architecture for Controlling Real-World Robots , AAAI 92, pp. 809-815.

[24] Saffiotti A. (1993), Some Notes on the Integration of Planning and Reactivity in Autonomous Mobile Robots, AAAI 93, pp. 122-126.

[25] Pirjanian P, H. Blaasvear H, Christensen HI (1994), AMOR: An Autonomous Mobile Robot Navigation System , Proc. IEEE Conf. on Systems, Man and Cybernetics, oI. 3, pp. 2266-2271.

[26] Pirjanian P (1997), An Overview of System Architecture for Action Selection in Mobile Robotics.

[27] Makenzie DC, Arkin RC, Cameron JM (1997), Specification and Execution of Multiagent Missions, Autonomous Robots, Vol. 4, No. 1.

[28] Owen C, Nehmzow U (1998), Map Interpretation in Dynamic Environments, Proc. of the 5th International Workshop on Advanced Motion Control.

[29] Astrom E (1996), A Conecctionist Architecture for High Level Robot Navigation, M. Sc. Dissertation, University of Skovde, Sweden.

[30] Nicolescu M, Mataric M (2000), Extending Behavior-Based Systems Capabilities Using An Abstract Behavior Representation" in the Working Notes of the AAAI Fall Symposium on Parallel Cognition, North Falmouth, MA, Nov 3-5.

[31] Rylatt RM, Czarnecki CA, Routen TW (1995), A Perspective on the Future of Behavior Based Robotics", 10th Biennial Conference on Artificial Intelligence and Simulated Behavior, England.

[32] Rylatt RM, Czarnecki CA, Routen TW (1998), Connectionist Learning in Behavior Based Mobile Robots: A Survey, The Journal of Artificial Intelligence Review, Vol. 12, pp. 445-468.

[33] Maes P (1990), Situated Agents Can Have Goals, Journal of Robotics and Autonomous Systems, Vol. 6, No. 3, pp. 49-70.

[34] Hertzberg J, Jeager H, Morignot P, Zimmer UR (1998), A Framework for Plan Execution in Behavior-Based robots", ISIC/CIRA/ISAS'98, Gaitherburg, USA.

[35] Maes P (1995), Modeling adaptive Autonomous Agents, Artificial Life: An Overview, MIT Press, pp. 135-162.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Supervised Learning Technique for a Mobile Robot Controller in a Visual Line Tracking Task

○Andrey Loukianov, Masanori Sugisaka
Department of Electrical and Electronic Engineering, Oita University
700 Dannoharu, Oita 870-1192, JAPAN.
aloukian@cc.oita-u.ac.jp msugi@cc.oita-u.ac.jp

Abstract

The paper concentrates on the development of learning methods for the intelligent self-learning control system of a mobile service robot. Within this framework an operator assisted learning method for the robot controller in a line tracking task using color CCD camera is proposed. First we present an image processing method which is used for path feature extraction. The multi-layer feed-forward neural network is utilized to control the velocities of the robot wheels. The methods for obtaining of the learning data and training the neural network are described. Experimental results are presented at the end of the paper.

Keywords: *Neural network, Learning, Vision guidance, Image processing*

1. Introduction

Traditional robot control methods are based on well stated mathematical models and algorithms. Though a number of control methods are able to adapt to changing parameters of the system, conventional control systems become unusable when mathematical model of a plant and structure of an environment are unknown. In this case the intelligent control systems with strong learning abilities are needed.

Neural networks have been proven to be important part of the self-learning control systems [2]. It has been shown that the neural networks are very robust and efficient when used not only for fast computations, but also for the tasks related to human brain activities [3] (recognition, associative memory, classification, etc.). Therefore it is possible to implement on the basis of neural networks a brain-like control system that can learn its control law on-line automatically or with the help of a human operator.

The aim of this paper is to present a method of supervised learning for the visual controller of an autonomous robot. The task of visual line tracking is considered. A color line on the floor is used to mark a desired path of the robot. Implemented control scheme utilizes an image processing algorithm to extract path properties from a video image. This information is passed to the neural network which controls wheel motors of the mobile robot. Neural network training data is obtained on the first stage of system operation by manually guiding the robot with the help of a human operator.

2. Background

Various techniques exist that enable mobile robots to navigate in their environments using vision. For the navigation it is crucial to define a path which mobile robot has to follow. The cheap and robust way to define the path is to mark it with color lines on the floor or to set the reference points along it. This reduces the visual navigation problem to a visual line tracking task and allows to implement visual servoing in a closed loop control system [1, 4]. In such systems the control is deduced by using predefined nonlinear control law with visual feedback.

There have been a number of extensive surveys published on the subject of supervised learning of neural networks [6]. It was proved that any nonlinear function can be approximated with arbitrary precision by properly trained multi-layer feed-forward neural network. On this basis neural and neural-fuzzy controllers were implemented for visual tracking tasks [4, 5]. In [5] a neural-fuzzy controller with on-line learning capabilities has been proposed. It has been shown that this controller can improve its tracking skills after several learning test runs, but still it requires some initial preset tracking algorithms to be set before the first run.

The research presented in this paper aims to develop an intelligent self-learning control system for a vision guided mobile robot. Hence the objective is to propose a feasible method which can be used to teach the mobile robot control system to track a line without any a priori knowledge or initially preset algorithms.

3. Mobile robot configuration

The hardware configuration of intelligent mobile service robot is shown in Figure 1. Two CCD cameras with tilt, pan and zoom capabilities are mounted on the robot head to provide a stereo vision. The robot is driven by two DC motors which are connected to two driving wheels on left and right sides of robot. Six ultrasonic sensors are used to detect the obstacles.

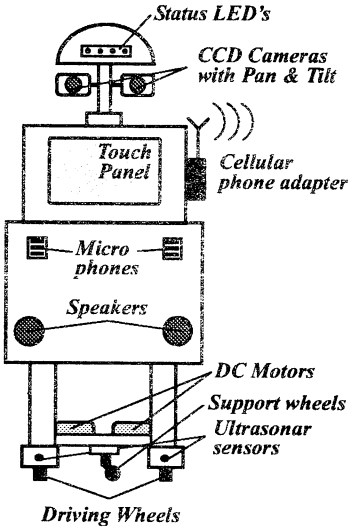


Figure 1. Hardware configuration

The robot is controlled by two computers, located within the robot body and connected by Ethernet network. Zoom, pan and tilt values of the cameras are set from PC serial port using the SONY VISCA protocol. The video images are acquired via standard video capture cards. The pulse wheel encoders are used to determine the relative motion and the angular speed of driving wheels.

The robot nonholonomic kinematics equations can be written as

$$\dot{\boldsymbol{q}}(t) = \pi D\boldsymbol{A}(\boldsymbol{q}(t))\,\boldsymbol{u}(t), \tag{1}$$

where $\boldsymbol{q} = [x\ y\ \phi]^T$ is robot's coordinate and orientation vector in the global reference frame, $\boldsymbol{u} = [\omega_L\ \omega_R]^T$ is angular speed vector of left and right driving wheels, $\boldsymbol{A}(\boldsymbol{q})$ is transformation matrix and D is the diameter of the wheels.

The position of the mobile robot at time moment t can be calculated by numerically integrating Eq. (1) with the known initial position $\boldsymbol{q}(t_0) = \boldsymbol{q}_0$. This implies that the precision of wheel encoders is very crucial in the navigation task, since the errors are propagated and accumulated during the integration.

4. Video image processing

In our task we use the color line on the floor to mark a desired robot path. The objective of image processing is to separate the path line image from background on a video frame and to convert the image information to a small number of parameters. These parameters should define position of the mobile robot relative to the path line as well as direction of the line. Also the image processing algorithm should satisfy sampling time requirements to process every video frame received from the camera with rate of 30 fps.

A video frame received from the camera capture card has size of 640 × 480 pixels and is in RGB color format. To increase the computation speed the size of the image is reduced to 160 × 120 pixels. The image of the line is extracted from resulted source image as a bi-tonal bitmap by applying the following procedure:

1. the source frame is converted from RGB to HSV color format;
2. the HSV image is converted to bi-tonal bitmap by applying line color lookup table with (here we assume that color of the line is distinct in the environment);
3. a median filter is applied to resulting image to remove the noise.

The result of performing of the described steps is a bi-tonal image of the line. Though, even after applying the median filter the image may contain some patches of noise. Remaining patches of noise are efficiently removed by object labeling algorithm which extracts connected areas of the pixels as objects. All objects with area $A < A_{min}$ are discarded, where A_{min} is a minimal object area threshold.

Next problem is how to convert the image information to a small number of parameters. Conventional calculation of the object's center of gravity [4] is not sufficient here because of its ambiguity (see example in Figure 2). To overcome this problem we propose three new parameters that can completely describe position and orientation of the path line in video frame coordinate system.

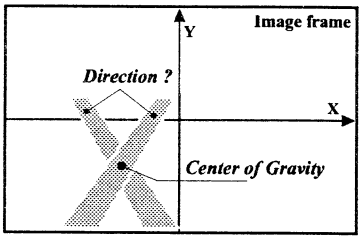


Figure 2. Line direction ambiguity

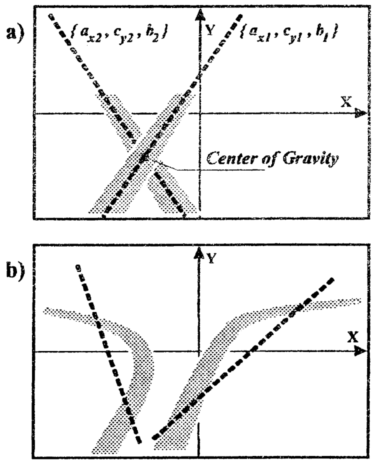


Figure 3. (a) ambiguity resolution; (b) indication of path general direction

The proposed parameters are the coefficients of line equation defined as

$$a_x x + c_y y + b = 0, \qquad (2)$$

which represents a line with best fit to pixels of the path line image. Figure 3 shows how these parameters can resolve ambiguity problem and indicate general path direction. The coefficients a_x, c_y and b are calculated using a least squares method.

It can be easily proved that the line in Eq. (2) passes through the center of gravity point. Thus in our method we have selected three line image parameters as output of image processing. These parameters are:

- pixel coordinates of the center of gravity, x_c and y_c;
- the line direction parameter l_d, which is obtained from Eq. (2) as

$$l_d = \begin{cases} -\frac{c_y}{a_x}, & |c_y| \leq 5|a_x|, \\ 5, & |c_y| > 5|a_x| \text{ and } c_y a_x \geq 0, \\ -5, & |c_y| > 5|a_x| \text{ and } c_y a_x < 0, \end{cases} \qquad (3)$$

that will yield the value in range $[-5,\ 5]$. Geometrically, parameter l_d relates to the angle between least squares fit line (Eq. 2) and Y axis of video frame coordinate system measured clockwise;

The origin of video frame coordinate system is chosen to in the center of the frame.

5. Learning technique for mobile robot controller

From Section 4 let us define the output variable vector of the image processing system as

$$\boldsymbol{y}(t) = [x_c\ y_c\ l_d]^T, \qquad (4)$$

and from Eq. (1) the input variable vector as $\boldsymbol{u}(t)$. If we define the error vector $\boldsymbol{e}(t) = \boldsymbol{y}_d - \boldsymbol{y}(t)$, where $\boldsymbol{y}_d$ is the desired output vector, then the general control law can be written in form

$$\boldsymbol{u}(\boldsymbol{e}, t) = F_u(\boldsymbol{e}(t), \dot{\boldsymbol{e}}(t)), \qquad (5)$$

where F_u is usually a nonlinear function.

In our self-learning control system both the control law F_u and the desired value of output vector $\boldsymbol{y}_d$ are unknown at initial stage. To deal with this problem we suggest using an intelligent controller with multi-layer feed-forward neural network inside which can learn mapping $\mathcal{F}_u : \mathcal{Y} \rightarrow \mathcal{U}$, where $\mathcal{Y}$ and $\mathcal{U}$ are vector spaces of $\boldsymbol{y}$ and $\boldsymbol{u}$, respectively.

In the first phase of mobile robot operation a controller learning procedure is to be carried out. During this phase the sets of input $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ and target $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$ vectors for the neural network training are collected by manually guiding the mobile robot.

5.1. Training data

The training data set has to represent a general visual servoing control law (Eq. 5). It is obtained during training runs along the line with a manual control. Training paths should include left and right turns, deflection compensation, line approaching from an arbitrary position. During training runs of robot it is important to keep the line in the camera's field of view.

After input and output vector sets are gathered, it is necessary to reduce their size for faster training of neural network. First step is to remove duplicates of $\{\boldsymbol{y}_i, \boldsymbol{u}_i\}$ pairs in the recorded sets, which are present due to the small sampling time of controller. Then a simple clustering algorithm is applied. It classifies into the same cluster all learning pairs that satisfy the following condition: $\rho(\boldsymbol{y}_i, \boldsymbol{y}_j) < \rho_{max}$, $(i \neq j$ and $i, j = 1, \ldots, n)$, where $\rho(\cdot)$ is Euclidean distance measure function and ρ_{max} is maximum distance allowed between vectors in the cluster. Let N be the resulting number of clusters. In each cluster a single representation pair $\{\boldsymbol{y}_i, \boldsymbol{u}_i\}$, $i = 1, \ldots, N$ is computed as mean vector.

5.2. Neural network training

A neural network with 3 neurons in input layer, 2 hidden layers (each with 30 neurons) and 2 neurons in output layer is used. Hidden layers and output layer use logsig and tansig activation functions, respectively. Training data is normalized in the range of $[-1,\ 1]$. An experimental training was performed using standard backpropagation, variable learning rate, resilient backpropagation and Levenberg-Marquardt methods. Levenberg-Marquardt method used much less number of iterations to converge than other methods, but its computational time was the longest because of a large size of the training data. The resilient backpropagation method had best timing results and was chosen for the real-time application.

6. Experiment

An experiment which involved training of controller and experimental path tracking using trained controller was conducted. The training of controller was performed on several left and right line approach and compensation paths (Fig. 4). Resulted training data set contained $N = 302$ patterns. Network training time was 8 minutes. Experimental results are presented in Figure 5. In this figure legend "Learn." denotes the trajectory of robot with the learning controller. Legend "Preset" denotes the trajectory of robot guided by controller with preset control law $\boldsymbol{u}(e) = [\omega \ \ \omega]^T + [\Delta\omega \ \ -\Delta\omega]^T$, where $\omega = K_f(y_c - \tilde{y}) + K_{fl}\, l_d \, \mathrm{sgn}(l_d b)$, and $\Delta\omega = K_t(x_c - \tilde{x}) + K_{tl}\, l_d \, \mathrm{sgn}(l_d b)$ (where K_f, K_{fl}, K_t, K_{tl}, $\tilde{y}$, $\tilde{x}$ = const). It can be seen that the difference between tracking errors of learning and preset controllers is small and that learning controller can perform robustly on the paths it had not been trained on. Significant amount of tracking error of both controllers is the result of 0.56 m camera's blind space in front of mobile robot.

7. Conclusion

In this paper we have presented a supervised learning technique for a neural mobile robot controller in a visual line tracking task. This technique can be used in intelligent self-learning control system to acquire line tracking skills without any prior knowledge on how to do it. While guided by an operator the robot obtains its learning data with the help of our suggested image processing method. Experimental results show that after neural controller is trained, the robot is able to follow an arbitrary path. However it should be noted that the quality and stability of learning noticeably depend on selection of training paths and training procedure. This problem has to be addressed in further research.

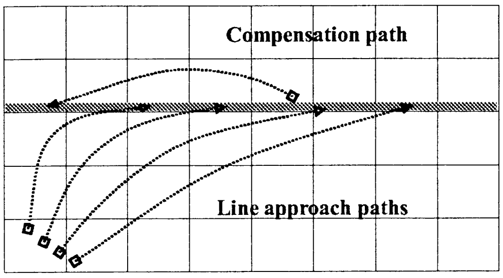


Figure 4. Example of right approach and compensation paths

References

[1] V. Cadenat, R. Swain, P. Souères, M. Devy, "A controller to perform a visually guided tracking task in a cluttered environment," *Proc. of the 1999 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, pp. 775–780 (1999).

[2] B. Jerbic, K. Grolinger, B. Vranješ, "Autonomous agent based on reinforcement learning and adaptive shadowed network," *Arificial Intelligence in Eng.*, Vol. 13, pp. 141–157 (1999).

[3] M. Hattori, M. Hagiwara, "Associative memory for intelligent control," *Mathematics and Computers in Simulation*, Vol. 51, pp. 349–374 (2000).

[4] M. Sugisaka, X. Wang, J.-J. Lee, "Intelligent control with new image processing strategy for a mobile vehicle," *Artif. Life Robotics*, Vol. 2, pp. 113–118 (1998).

[5] X. Wang, M. Sugisaka, "Feasible on-line learning neural and fuzzy intelligent control for a mobile vehicle ," *Proc. of the 1999 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, pp. 769–774 (1999).

[6] D. H. Wolpert (Ed.), "The Mathematics of Generalization: The Proc. of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning," *Perseus Press*, (1994).

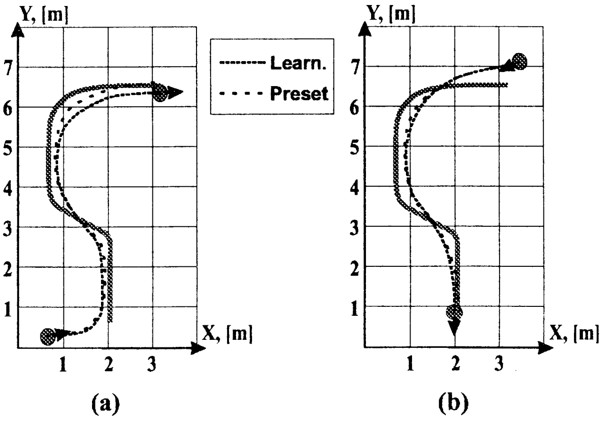


Figure 5. Experimental path tracking: (a) forward; (b) backward.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Resolution of the Puzzle of the Posi-nega Switch Mechanism in the Globally Coupled Map Lattice

T. Shimada and S. Tsukada
Department of Physics, Meiji University, Higashi-Mita 1-1-1, Kawasaki, Kanagawa 214-8571, Japan

Abstract

We revisit the globally coupled map lattice (GCML). We use there is a family of universal 'curves of balance' between two conflicting tendencies in the model – the randomness in each map and the coherence due to an averaging interaction – and we locate the posi-nega switch region in the parameter space of the model. We clarify the mechanism of a basic posi-nega switch in the two-cluster regime, which guarantees no mixing of maps across their mean field in the chaotic transient process. We stress that a special attention must be paid for rounding-off errors; otherwise, there fatally occurs an artifitial numerical degeneracy of maps in a cluster due to a highly negative Lyapunov exponent of the clustor attractor.

1. Introduction

The globally coupled map lattice (GCM L) in the simplest form – the homogeneous GCML – is a system of identical N logistic maps coupled via their mean field. It is a representative of the intelligence activity supposed to come from the synchronization among the neurons in the neural network. The GCML evolution equation can be written down in a one-line equation;

$$x_i(t+1) = (1-\varepsilon)f(x_i(t)) + \frac{\varepsilon}{N}\sum_{j=1}^{N} f(x_j(t)), \quad (1)$$

where t is a discrete time and $f(x) = 1 - ax^2$. Yet, GCML is known to have surprisingly rich clustering phases in the space of the nonlinearity parameter a and the coupling ε. In particular in the two-cluster phase, which occurs in the band region in the (a,ε) plane

$$\varepsilon = 0.08 - 0.16 \text{ at } a = 1.4 \text{ and}$$
$$\varepsilon = 0.24 - 0.33 \text{ at } a = 2.0, \quad (2)$$

it exhibits an interesting posi-nega switch between two clusters of synchronizing maps. The switch found by Kaneko[1] uses an ability of the population ratio θ ($\theta \equiv N^{>}/N$ with $N^{>}$ the number of maps in the majority cluster) as an effective nonlinearity parameter. Before the switch, the maps divide themselves into two clusters oscillating oppositely in phase. The cluster with a higher (lower) average value of maps at even t is called as the positive (negative) cluster. By transporting maps successively from the minority cluster to the majority one, the attractor undergoes successive periodicity doubling. When the unbalance in populations reaches a certain threshold, the system goes into a grand chaotic transient motion. Soon the system comes back to the periodic opposite phase motion. Maps, which were in the positive cluster before the transient steps, are now either all in the positive cluster or all in the negative cluster. The latter case, the half chance, is the posi-nega switch. It is an intriguing phenomenon in that maps behave as if they keep a 'memory' of their former clusters even though they pass through the chaotic transient steps where many channels could open[1].

Already a decade has passed since the discovery, and an application to a more sophisticated case has been also considered[2]. However, the puzzle of the memory has not been elucidated so far. Recently one of the authors (T.S.) has found that the dynamics of the element logistic map – in particular its periodic windows – produces its foliation in the GCML[3]. In this new light, we address ourselves to the old puzzle. We show that there is a natural mechanism that guarantees no mixing of maps between the clusters. New observations to the two-cluster regime are also presented.

2. Cluster Formation

In order to solve the puzzle it is necessary to analyze the cluster attractor dynamics first. The equation (1) is an iteration of two-step process – the mapping and the interaction. In the first, the nonlinearity generally magnifies the variance among the maps. In the second, maps are pulled to the mean field at a fixed rate $1-\varepsilon$. The larger a implies the stronger defocusing of maps, while the larger ε the stronger focusing. In the two-cluster band (2), the conflict is solved by the opposite phase periodic motion of maps in two clusters.

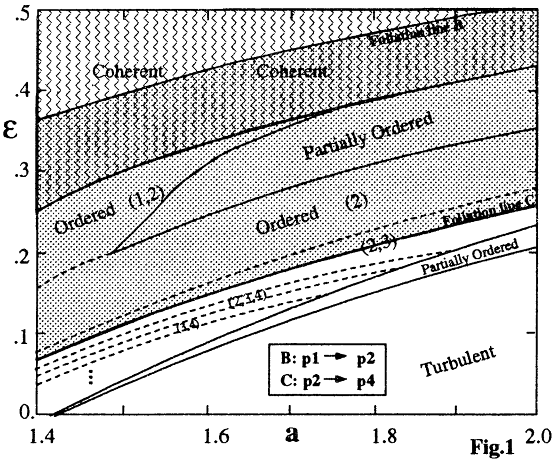


Fig.1

In Fig. 1 we present our prediction (see ref[3]) for the foliation of the period-two region of a single logistic map on the phase diagram. Line B is the foliation of the first bifurcation point, and line C the second. We find that prediction nicely covers the ordered two-cluster regime. It also covers the adjacent part of the coherent phase indicating that the GCML passes through a transient coherent motion with an excess of inputs.

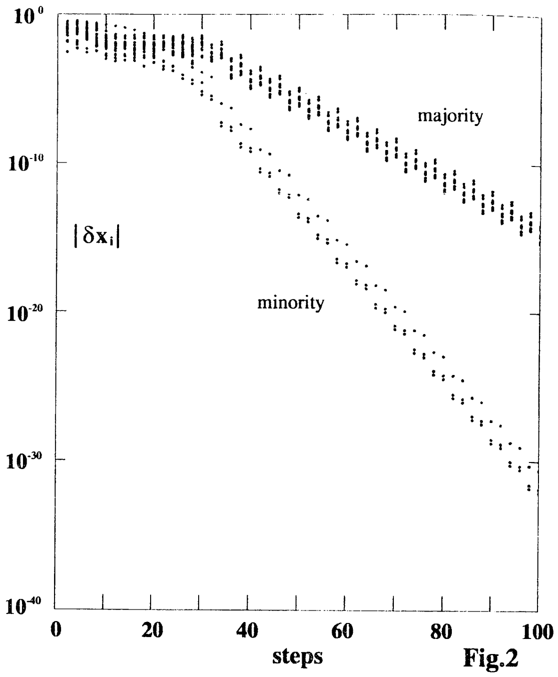


Fig.2

In Fig. 2 we show that the rapid decay of the variations of maps $\delta x_i^\pm \equiv x_i^\pm - x_{av}{}^\pm$ in each cluster. $N = 50, a = 1.98, \varepsilon = 0.3$, $\theta = 0.6$. In general, the maps from a random start fall into the two-cluster attractor in the first ≈ 20 steps and the size of the minority cluster reaches 10^{-40} at ≈ 50 steps.

This is a serious warning for the simulation. The maps become unresolved in the usual double precision computation before the system is compelled to the threshold by inputs. With such an artificial degeneracy, the maps will never split again – the memory becomes almost trivially preserved. To avoid such a triviality, we adopt arbitrary precision calculation throughout this note.

Let us analyze the linear stability matrix of GCML following ref[3]. For the cluster attractor configuration there occurs a high degeneracy of the eigenvalues. For the two-cluster attractor in periodicity p, there are firstly two eigenvalues,

$$\lambda^{(\pm)} = -2a(1-\varepsilon)\left(\prod_{k=1}^{p} x_{av,k}^{\pm}\right)^{1/p}, \qquad (3)$$

each with $N^\pm - 1$ – fold degeneracy. These describe the stability of maps in each cluster. There are also two other eigenvalues $\lambda_\pm$ responsible for the stability of the cluster orbits. The observed decay rates of the GCML clusters are plotted in Fig. 3a as a

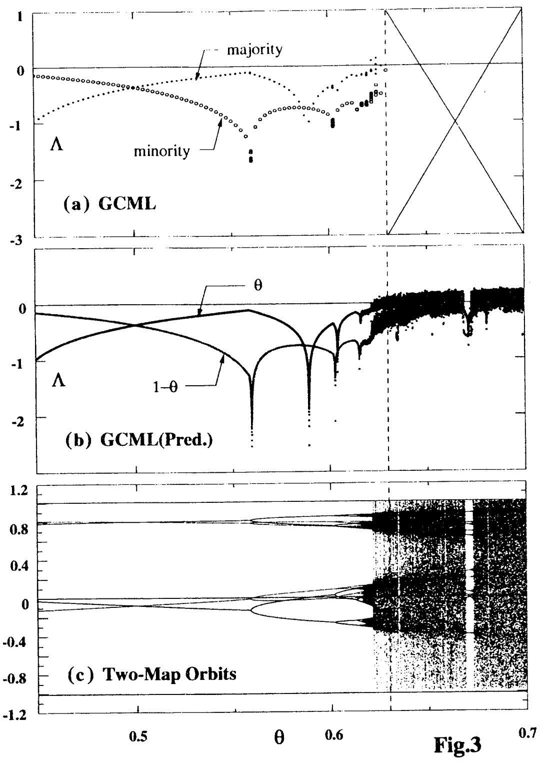


Fig.3

function of θ. These are in an excellent agreement with $\Lambda^{(\pm)} = \log|\lambda^{(\pm)}|$ in Fig. 3b from the matrix-coupled two-map model[1, 4] describing the motion of $x_{av}^{\pm}(t)$ – the 'center of mass' of clusters. *The cluster decay rates are solely controlled by the reduced dynamics of the coupled clusters.*

Fig. 3c exhibits the cluster orbits as determined by the matrix model. We find that the 'orbits successively bifurcate when the cluster decay rates become extreme. On the threshold (the dotted line), the decay almost ceases. This is the start of the grand chaotic motion of the maps. With an excess pulse beyond the threshold, the maps go into the transient steps. The two clusters stop the opposite phase motion and come into a quasi-coherent motion. As there is no stable attractor beyond the threshold, as seen in Fig. 3a, they must soon come back to the original opposite phase motion. Then, the periodic cluster orbits are solely encoded by three parameters a, ε and θ. Therefore, if any mixing between the clusters has not occurred, *they are destined to follow the same orbits before the transient steps.* If the transient steps are even, all maps should come back to the original clusters; if odd, they should be completely swapped between the positive and the negative cluster – this is the posi-nega switch. (This point is hard to see in the original report[1], where only the orbits at even steps are shown.) Thus the real puzzle boils down to a single question. *What is the mechanism that keeps the population unbalance between two clusters unchanged before and after the chaotic steps?* This question is solved below.

3. Transient Process and the No Mixing Mechanism

During the transient time, the clusters move around almost together. Thus, it appears that the maps can easily mix between the clusters. However, there is actually a simple mechanism, which protects them from mixing. The GCML maps evolve under an iteration of a two-step process. From time t to $t+1$, the maps change their value as $x_i \to x_i' \to x_i''$, where the first arrow denotes the mapping, $x_i' \equiv f(x_i)$ and the second the interaction $x_i'' \equiv x_i(t+1)$ given by eq. (1). Their mean field changes accordingly as $h \to h' \to h''$, and the variances of maps (the relative coordinates with respect to the mean field) $\delta_i \to \delta_i' \to \delta_i''$. Let us consider the first step. Since all maps are close to the mean field, we obtain $x_i' = f(h+\delta_i) = f(h) + df/dx|_h\ \delta_i + O(\delta_i^2)$. Averaging this using $\sum_{i=1}^{N} \delta_i = 0$ we find $h' = f(h) + O(\max(\delta_i^2))$. Therefore we find that the new variance is given by

$$\delta_i' \equiv x_i' - h' = df/dx|_h\ \delta_i + O(\max(\delta_i^2)). \quad (4)$$

The second step is a uniform contraction and the interaction does not affect the mean field, that is, $h'' = h'$, as can be seen by averaging (1).

Hence we see that

$$\delta_i'' = (1-\varepsilon)\delta_i' = J(t)\delta_i \quad (i=1,\ldots,N), \qquad (5)$$
$$J(t) = (1-\varepsilon)\,df/dx\,|_h\,.$$

This line is crucial. The factor $J(t)$ is common to all maps. *Thus, at every step in the chaotic transient process, the maps separated by the mean field never mix each other. This is the mechanism, which keeps the population unbalance unchanged before and after the chaotic transient process.* Numerically we have checked that the maps almost always move in two close clusters around the mean field. But sometimes the clusters split largely. This does not invalidate above resolution. The danger of the mixing occurs when they come close. Our mechanism gives a guarantee of no mixing in such an emergence. Another danger might occur at the cluster configuration $f(x_{av}^{+}) = f(x_{av}^{-})$. But during the chaotic transient steps the clusters evolve almost always together and we have checked that such a coincidence with large splitting is negligibl-

We close this note by showing a typical posi-nega switch ($N=50$, a=1.98, $\varepsilon=0.3$) in Fig. 4, which clearly exhibits the memory preserving mechanism by a print circuit pattern. The maps never mix across their mean field during the transient process and the posi-nega switch is a change in the count of the parity (even and odd) of iteration steps.

[1] K. Kaneko, Phys. Rev. Lett. **63**, p. 219 (1989).

[2] K. Kaneko, Physica (Amsterdam) **41D**, p. 137 (1990).

[3] T. Shimada, Technical Report of IEICE, NLP97-159, 71 (1998); T. Shimada and K. Kikuchi, Phys. Rev. **E62**, p. 3497 (2000).

[4] H. Fujigaki, M. Nishi and T. Shimada, Phys. Rev. **E53**, p. 3192 (1996); H. Fujigaki and T. Shimada, *ibid.* **55**, p. 2426 (1997).

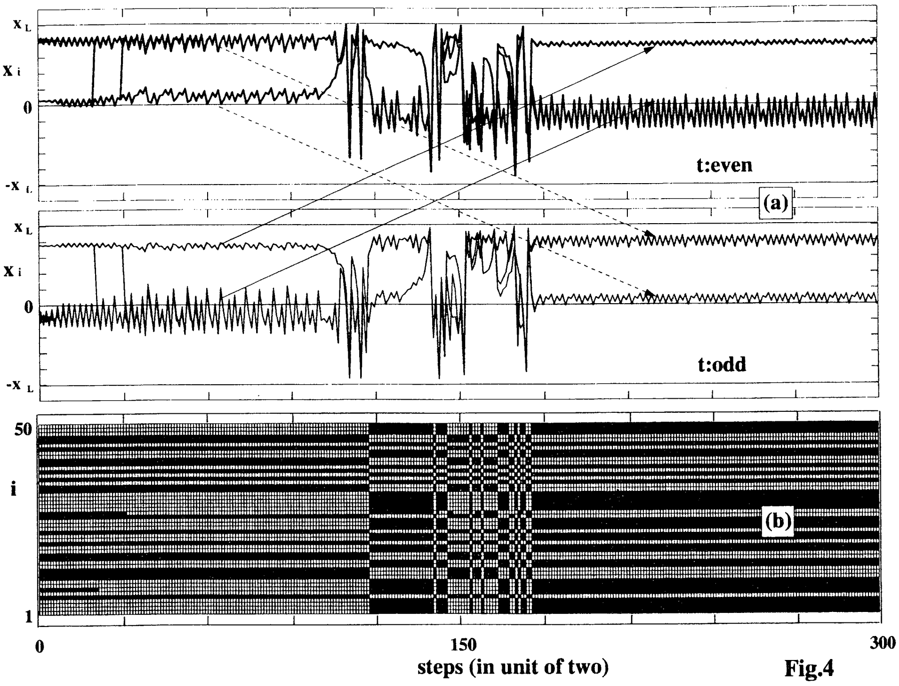


Fig.4

(a) The attractor at even (odd) steps is shown in the upper (lower) diagram. The attractor on the whole, the even and odd iteration steps together, is precisely the same before and after the chaotic transition. The arrows point out that the posi-nega switch is a change in the count of the parity (even and odd) of iteration steps.
(b) The maps are distinguished by their mean field. *The print-circuit shows no mixing of maps across the mean field.*

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Classification and Function Prediction of the Protein by using Data Compression

K. Sugawara and T. Watanabe
Graduate School of Information Systems
Univ. of Electro-Communications
Chofu, Tokyo, JAPAN.

Abstract

Protein has a complicated spatial structure and has chemical and physical functions which originate from the structure. It is important to predict the structure and the functions of the protein from DNA sequence or amino acid sequence from a viewpoint of biology, medical science, protein engineering and so on. But today there is no method to predict them accurately from the sequence. Instead, there are some approaches to estimate the functions based on a similarity retrieval of sequences approximately. In this paper, we propose a new method for similarity retrieval of amino acid sequence based on the concept of the homology retrieval using the data compression. Introduction of the compression by dictionary technique enables us to describe the text data as n-dimensional vector using n dictionaries which is generated by compessing n typical texts, and enables us to classify them based on their similarity. We examined the effectiveness of our proposal using real genome data.

1 Introduction

It is well known that structures and functions of living things are composed of many kind of proteins. Protein is originally one-dimensional chain of amino acids, but it has a complicated space structure by folding and has chemical and physical functions which originate from the structure. As the sequence of amino acids are generated according to the information in DNA, it is necessary to investigate the relation between structure/function of proteins and DNA sequence/amino acid sequence in order to understand the mechanism of living things. This is because the knowledge is useful from the viewpoint of biology, medical science, protein engineering and so on.

Today, we can encode DNA sequence into amino acid sequence, but there is no method to predict the structure and the function of unknown protein accurately from DNA sequence or amino acid sequence. Instead of that, there are some techniques to estimate the functions approximately[1].

The function estimation method is classified into two types. One is called "Homology retrieval", which estimates the function of the gene based on the general similarity of the arrangements. The other is called "Motif retrieval", which estimates the function using the pattern which commonly appears for functionally resembling arrangements.

As a technique for the homology retrieval, there is a classical algorithm called dynamic programming (DP)[2]. Recently, the genome database dramatically enlarges scale, and DP is disadvantageous in the retrieval of the whole array data base because of its calculation time. As the result, the high-speed retrieval algorithms such as BLAST method[3] or FASTA method[4] were developed in order to shorten the calculation time.

In this study, we propose a new method for similarity discriminant of amino acid sequence information based on the concept of the homology retrieval using the data compression.

2 How To Classify the Sequence

We use the text compression method for classification of the sequence. Approaches to text compression can be divided into two classes: statistical method and dictionary method[5]. The dictionary method was developed by Ziv and Lempel[6] and "LZ78" is one of the representatives. Dictionary method achieves compression by replacing groups of characters or phrases with indices in a dictionary. The dictionary is a list of characters or phrases that are expected to appear frequently. In general, the length of the indices are shorter than characters or phrases themselves, thereby we say the text is compressed.

There are two remarkable points in dictionary method.

● The created dictionary depends on the content of the text.

● The created dictionary is available for compressing other texts.

Now let's denote a text as T_i, a dictionary generated from T_i as D_i, and the compression ratio as $R(T_i, D_i)$.

$$R(T_i, D_i) = \frac{Compressed\,length\,of\,T_i\,by\,using\,D_i}{Original\,length\,of\,T_i}.$$

Assume we have three texts T_1, T_2 and T_x. If the content of T_x resembles to T_1, the following relation is obtained.

$$R(T_x, D_1) > R(T_x, D_2).$$

If the content of T_x does not resemble to both T_1 and T_2,

$$R(T_x, D_1) \sim R(T_x, D_2) >> 0.$$

Using this property, the characteristics of Text T_x can be expressed by a vector. Compression of n texts generates n dictionaries. It enabies us to express the feature of Text T_x as a n-dimensional vector $\vec{K_x}$.

$$\vec{K_x} = (R(T_x, D_1), R(T_x, D_2), ..., R(T_x, D_n)).$$

The space, the axis of the space, and the vector in this space are called as "Space of Data Compression(SDC)", "SDC basis", and "Feature vector", respectively [7]. We use this method for classification and recognition of the amino acid sequences. This method is simple and powerful, and it was applied to the recognition of audio data and image data[8, 9].

In this paper, we apply this method to the classification of amino acid sequences and shows its effectiveness. The procedure for classification and estimation is as follows.

(P1) n typical amino acid sequences are compressed by LZW and generate n basis dictionaries.

(P2) Proteins, of which function already known, are compressed by the dictionaries and expressed by feature vectors.

(P3) Unknown amino acid sequence is compressed by the dictionaries and expressed by a vector.

(P4) Function of unknown amino acid sequence is estimated by measuring the distance between the unknown feature vector and the known feature vectors in SDC.

Figurel is a schematic of the proposed method.

3 Experiment

We examine the effectiveness of our proposed method using real genome data. In this paper, we choose LZW[10] for text compression. LZW is an improved method of LZ78, which is one of the most famous compression method.

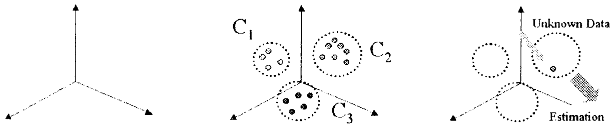
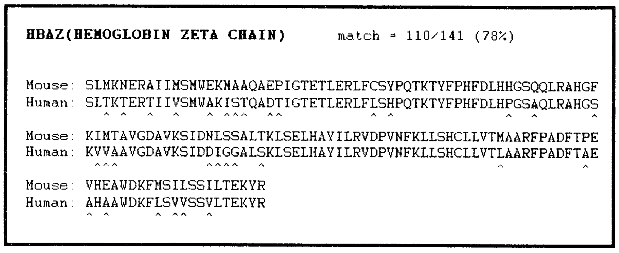


Figure 1: Schematics of the proposed method. (a)Creation of SDC, (b)Clustering of known specific vectors, (c)Estimation of unknown data.

The data are obtained from GenBank by NCBI (National Center for Biotechnology Information) of NIH[11].

3.1 Preliminary Experiment of Classification

Each protein have a specific amino acid sequence and have a specific function. There is a lot of common proteins in each species of animals, but it is known the amino acid sequences are not identical because of "defects", "insertion" and "substitution" in each species. Figure 2 shows an example of amino acid sequence which are obtained from human and mouse. Though the function of the protein is completely identical, you can find out that there are some differences between human and mouse.



```
HBAZ(HEMOGLOBIN ZETA CHAIN)        match = 110/141 (78%)

Mouse: SLMKNERAIIMSMWEKMAAQAEPIGTETLERLFCSYPQTKTYFPHFDLHHGSQQLRAHGF
Human: SLTKTERTIIVSMWAKISTQADTIGTETLERLFLSHPQTKTYFPHFDLHPGSAQLRAHGS
         ^ ^  ^  ^   ^ ^^^  ^^          ^ ^             ^  ^      ^

Mouse: KIMTAVGDAVKSIDNLSSALTKLSELHAVILRVDPVNFKLLSHCLLVTMAARFPADFTPE
Human: KVVAAVGDAVKSIDDIGGALSKLSELHAVILRVDPVNFKLLSHCLLVTLAARFPADFTAE
        ^^^          ^^^^  ^                           ^         ^

Mouse: VHEAWDKFMSILSSILTEKYR
Human: AHAAWDKFLSVVSSVLTEKYR
       ^ ^     ^ ^^  ^
```

Figure 2: An example of amino acid sequence of human and mouse.

In this section, we examine that the proposed method can classify functionally same poteins correctly even if there are some differences between species.

At first, we picked up 5 kinds of amino acid sequences obtained from *Rattus norvegicus*(rat) and generated five dictionaries. Proteins used here are (1) Hemoglobin α-chain (HBA), (2) Interleukin-1α (IL1A), (3) Rhodopsin, (4) Globulin, (5) Neurotrophin-3 (NGF). Number of dictionary corresponds to each of the above number. For example, Dic1 is a dictionary generated from Hemoglobin α-chan.

Test data are the amino acid sequences obtained from other species: (1)Human, (2)Monkey, (3)Mouse, (4)Pig, (5)Cat, (6)Dog, (7)Rabbit.

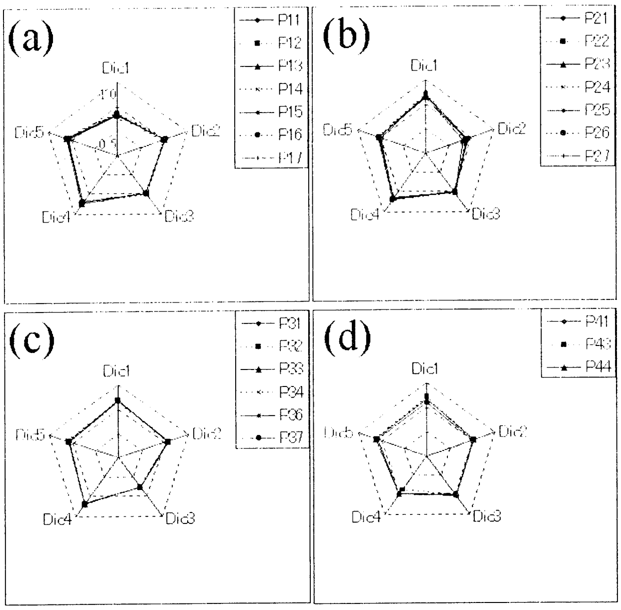


Figure 3: Radar charts of proteins. (a)feature of HBA, (b)IL1A, (c)Rhodopsin, (d)Globulin.

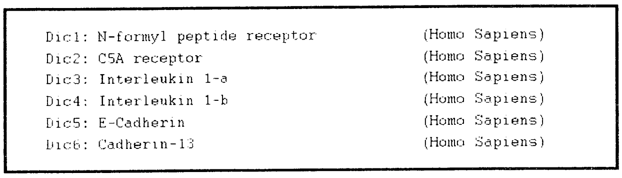
Figure 3 shows the result. Each axis means the compressibility of the protein by each dictionary. Pxy means the protein, in which x implies the kind of protein and y implies the species. For example, P13 is "Rhodopsin" obtained from "Mouse" and P41 is "Globulin".obtained from "Human."

We can find out two points from this result. One is that the same kind of proteins are compressed similarly in spite of the difference of species. The other is that the shape of the chart depends on the protein. These facts imply the proposed method classifies the proteins even if the structure differs a little.

3.2 Estimation of the Unknown Protein

Next, we will classify three kinds of protein families. The protein families used here are
(A) Seven-transmembrane type protein(32),
(B) Immunisatio mediator, Interleukin(35),
(C) Cell adhesion material, Cadherin(42).
The figure in the parenthesis indicates the number of data used in this experiment.

In this experiment, we choose six proteins for generating dictionaries. Figure 4 shows the name of proteins which are used here.



Dic1: N-formyl peptide receptor	(Homo Sapiens)
Dic2: C5A receptor	(Homo Sapiens)
Dic3: Interleukin 1-a	(Homo Sapiens)
Dic4: Interleukin 1-b	(Homo Sapiens)
Dic5: E-Cadherin	(Homo Sapiens)
Dic6: Cadherin-13	(Homo Sapiens)

Figure 4: Proteins for generating dictionaries.

There is not enough space to explain these proteins, but their information can be easily obtained from GenBank[11].

Figure 5 shows the the result. Each axis means the compressibility of the protein by each dictionary.

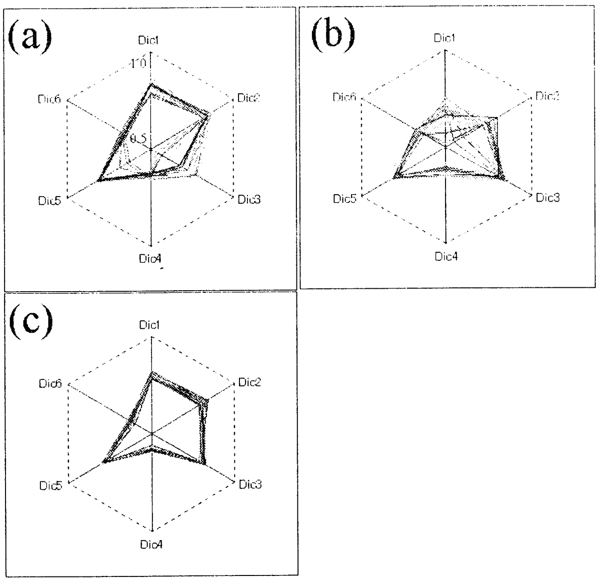


Figure 5: Result of experiment. (a)Seven-transmembrane type protein family, (b)Interleukin family, (c)Cadherin family.

As you can see, there are some types in each chart. Figure 6 is schematics of the classification. In a family of 7 transmembrane type protein, we can classify them into 4 typical types. In a family of immunisatio mediator interleukin, we can also classify them into 4 types. In a family of cell adhesion material cadherin, there is one type.

This analysis gives us the possibility to estimate unknown amino acid sequence data. The most easiest method for estimation is to measure the distance between the schematics and the unknown data. Figure 7 is the distances between the typical data and unknown data. Here, we choose nine data as unknown data and

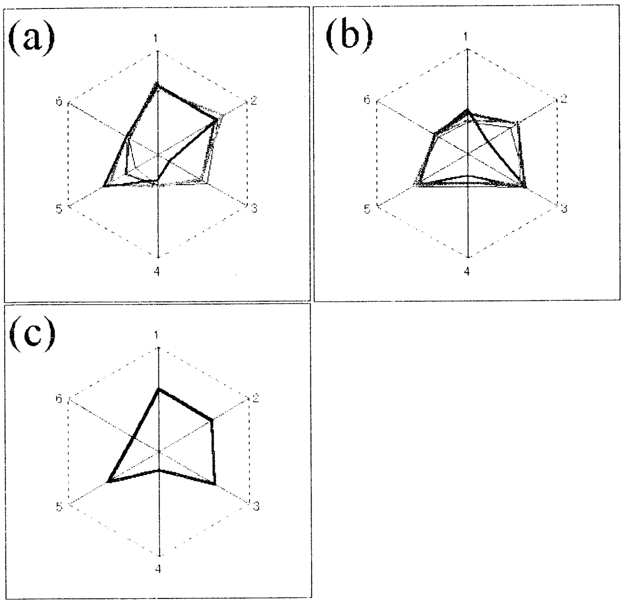


Figure 6: Typical feature of each protein.

measured the Euclidean distance in SDC. We know it is possible to estimate the the family which the unknown data should belong to.

4 Conclusion

In this study, we applied the concept of the Space of Data Compression to the amino acid sequence, and showed that it is possible to classify proteins into similar families. Here, we show the result of small-scaled experiment, but it is easy to make an experiment in a large scale. Moreover, we can easily expand the dimension of SDC for detailed classification.

This work was supported by Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] A. Konagaya、"Genome and Computer"、kyoritsu (2000)(in Japanese).

[2] S.B.Needleman and C.D.Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *J.Mol.Biol.*, 48, (1970), pp. 443-453.

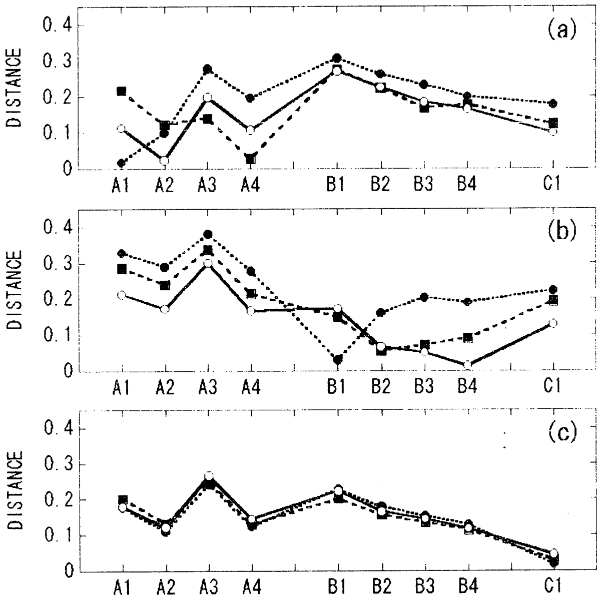


Figure 7: Distance between the schematics and unknown data.

[3] S.F.Altschul, et.al, "Basic local alignment search tool", *J.Mol.Biol.*, 215(1990), pp.403-410.

[4] D.J.Lipman and W.R.Pearson, "Rapid and sensitive protein similarity searches", *Science*, 227(1985), pp.1435-1441.

[5] T.C.Bell, J.G.Cleary and I.H.Witten, "TEXT COMPRESSION", Prentice-Hall (1990).

[6] J.Ziv and A.Lempel, "Compression of individual sequences via variable-rate coding", *IEEE Trans. on Infor.Theory*, Vol.24, No.5, (1978), pp.530-536.

[7] T.Watanabe, et.al, "A new universal pattern representation scheme using data compression", *submitted to IEEE PAMI.*

[8] K. Kondou, N. Kato and T. Watanabe, "OS-R: Online Sketch Recognition by Data Compression Technique", *J.IPSJ*,Vol.38, No.12, (1997) pp.2468-2478 (in Japanese).

[9] T. Watanabe、" UPRDC: Universal Pattern Representation by Data Compression", *proc. JSPRS*, (2000) pp.253-258 (in Japanese).

[10] T.Welch, "A Technique for High-Performance Data Compression", *IEEE Computer*, Vol.17, No.6, (1984), pp.8-19.

[11] http://www.ncbi.nlm.nih.gov/Genbank

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Spatial Frequency Specific Visual Adaptation May Cause Paradoxical Transient Improvements in Visual Acuity – Experimental and Simulation Studies –

○Y.Nagai*, K.Uchida+, K.Ueda+, H.Onodera+, T.Tanaka*, N.Sugie *
*Department of Electrical and Electronic Engineering
Graduate School of Science and Technology, Meijo University
1-501 Shiogamaguchi, Tempaku-ku Nagoya, 468-8502, Japan
sugie@meijo-u.ac.jp
+Nagoya Electric Works.Co.,Ltd.
29-1 Mentoku, Shinoda, Miwa-cho, Ama-gun Aichi, 490-1294, Japan
nagai@nagoya-denki.co.jp

Abstract

In the visual cortex of monkey and human, several types of processings are conducted; orientation-selective edge detection, and so on. It is also reported by Campbell and his associates that the cerebral visual cortex conducts processing consisting of spatial-frequency-and-orientation-specific multiple narrow-band channels. It is also reported that spatial-frequency-and-orientation-specific visual stimuli cause adaptations in the corresponding channels, resulting in the reductions of visual sensitivity around the frequency ranges.

In the course of the simulation studies using Landolt rings, to measure visual acuity, we happened to find spatial-frequency-specific adaptations will cause paradoxical transient improvements in visual acuity. Therefore we conducted perceptual experiments to see if the improvements in visual acuity take place after spatial-frequency-specific visual adaptation, with affirmative results.

The findings may be significant in designing road information boards for car drivers.

Key words: Spatial frequency characteristic, visual acuity, spatial frequency adaptation, multiple channel model, Landolt ring

1 Introduction

We study a visual system in order to implement the human visual information processing mechanism with a computer. A human being has various visual characteristics, and it is very difficult to model them all and integrate. So we examine the model which handled the spatial frequency characteristic as the first phase. This time, we focused our attention on the multiple channel theory that Campbell et al. proposed, and constructed the visual model that had more than one frequency channel[1]. So far computer implementations of the human visual-system have not dealt with the multiple channel model. It is possible that this model reproduces the virtual perceived image as an output. This model can generate outputs during spatial frequency adaptation. The coefficients of the model were set to be similar to the spatial frequency characteristic that Campbell et al. showed.

The evaluation used a Landolt ring for a general visual acuity test. Than as a result, the spatial frequency characteristic due to Campbell et al. was found to be about 1.2 in visual acuity. Moreover, we happened to find spatial-frequency-specific adaptations will cause paradoxical transient improvements in visual acuity, when the visual perceived image of a Landolt ring was reproduced during spatial-frequency adaptation. So we showed a visual stimulus of sinusoidal grating to examinees in order to confirm the phenomenon. We let them adapt to spatial frequencies, and measured visual acuity orientation-specific before and after adaptation and compared them. As a result we confirmed visual acuity change and it was almost similar to the prediction including orientation specificity.

2 Spatial Frequency Adaptation and Multiple Channel Theory

Various kinds of band-pass type spatial frequency filters that center frequencies exist in the human visual system. When one looks at a vertical or horizontal grating with a certain spatial frequency for a long time, the sensitivity of the corresponding filter or channel drops. In Fig.1A, the real line shows a typical spatial frequency characteristics of the human visual system (MTF). The abscissa denotes spatial frequency and the ordinate contrast sensitivity. After the adaptation to a grating of 7.1[cpd](indicated by the arrow), the contrast sensitivities get lowered around 7.1 [cpd] indicated by small solid squares[2]. Fig.1B shows one of the major evidences of the existence of narrow-band spatial frequency channel, the bandwidth of which is about one octave at the level of half of the maximal sensitivity. It is well accepted that the human visual system consists of a set of such channels. The envelope of channel characteristic as shown in Fig.1B constitutes the MTF in Fig.1A.

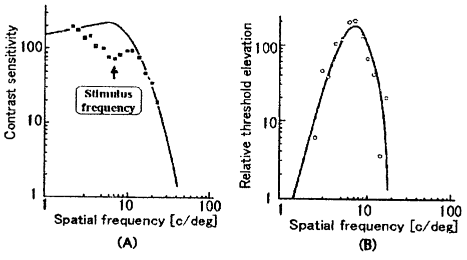


Figure 1: Effect of adaptation as an evidence of existence of narrow-band spatial frequency channel[3].

3 Model of Multiple Channel and Virtual Perceived Image

In Fig.2, in the human visual system, the visual stimuli are first received and preprocessed in the retina and then transmitted to the cerebral visual cortex via the lateral geniculate body. The propagation characteristic in retina and lateral geniculate body is a low-pass type filter of a comparatively wide bandwidth. It is known that this filter can be modeled with Eq.(1)[3].

$$\nabla^2 G(x,y) = \frac{-1}{\pi\sigma^4}(1 - \frac{x^2+y^2}{2\sigma^2}) \exp -\frac{x^2+y^2}{2\sigma^2} \qquad (1)$$

By the change of σ of Eq.(1), the frequency of the channel can be changed. The propagation characteristic in the cerebral visual cortex is the band-pass type filter of a comparatively narrow orientation selectivity. This filter was modeled with

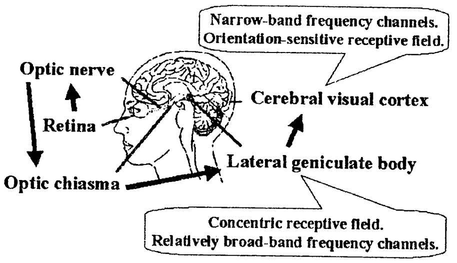

Narrow-band frequency channels.
Orientation-sensitive receptive field.
Optic nerve
Retina
Optic chiasma
Cerebral visual cortex
Lateral geniculate body
Concentric receptive field.
Relatively broad-band frequency channels.

Figure 2: Human visual system.

$$\begin{aligned} Dog(x,y) \quad &= \\ &(\frac{1}{\sqrt{2\pi}\sigma_{x,e}} \exp(-\frac{x^2}{2\sigma_{x,e}{}^2}) - \frac{1}{\sqrt{2\pi}\sigma_{x,i}} \exp(-\frac{x^2}{2\sigma_{x,i}{}^2})) \\ &\times \frac{1}{\sqrt{2\pi}\sigma_y} \exp(-\frac{y^2}{2\sigma_y{}^2}) \qquad (2) \end{aligned}$$

as Wilson et al. showed[3]. Eq.(2) added overemphasis to a difference of Gaussian functions (DOG) with orientation selectivity by manipulating the ratios among $\sigma_{x,e}$, $\sigma_{x,i}$ and σ_y.

In Fig.3 is shown the organization of our multiple channel model. The first stage corresponds to the processing in the retina and the lateral geniculate body. According to the neurophysiological studies[3], there are a set of low-pass type frequency channels each of which has its own specific cut-off frequency. We have implemented a set of four channels as described below. The processing at this stage used an Eq.(1).

The next stage corresponds to the processing in

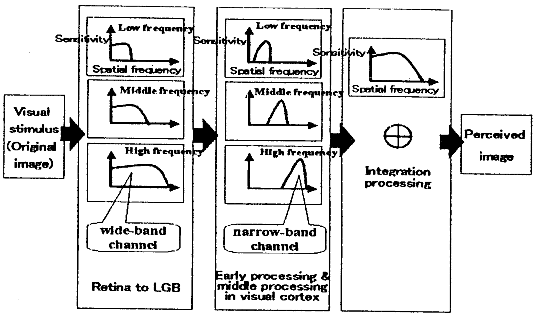


Figure 3: Organization of multiple channel model.

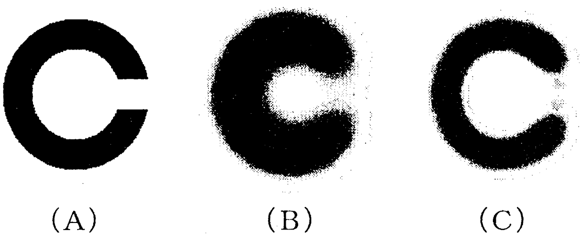


Figure 4: Virtual perceived image. (A) is a presentation stimulus. (B) is a normal perceived image. (C) is a virtual perceived image after adaptation.

cerebral visual cortex. An each channel at this stage is associated with orientation selectivity of 0,45,90,135[degrees].

The weight for 45,135[degrees] were set to half of those of 0,90[degrees] on the basis of experiment results of Heeley et al[5]. Therefore there are four spatial frequency specific channel at this stage, each of which consist of four orientation selective filters. The filter at this stage is implemented with Eq.(2).

We set the weight for each filter so that the overall characteristic at the integration stage cortex agrees with the MTF shown in Fig.1A. Thus given a visual stimulus as an input to the model, the virtual perceived image is obtained as the output. This model is able to handle various spatial frequency characteristics as a contrast sensitivity in after spatial frequency adaptation by letting the weight of corresponding channel change.

With the model, simulation studies were conducted. In Fig.4A is shown a Landolt ring as a stimulus. (B) is a virtual perceived image. The image size is 512*512 dots. The Landolt ring of visual acuity 1.0 is presented at a distance of 5[m]. (C) shows that after adaptation to 6[cpd] horizontal grating. As a result an appearance of a gap of a Landolt ring changes and shows that visual acuity may change for the better (compare Fig.4C with Fig.4B), when we adapted to a certain specific spatial frequency.

4 Spatial Frequency Adaptation and Change in Visual

We conducted experiments to confirm visual acuity improvement by after spatial frequency adaptation shown by simulation studies.

4.1 Methods and Preliminary Experiments

We executed the visual acuity test with a visual acuity test device ES-10 made to cope with JIS-7309 by Yagami Co., Ltd.. The experiment was executed after an examinee was adapted to ambient light. The experiment was done in the room that did not receive influence of the daylight. The brightness of the room by indirect lighting maintained an average of 300[lx]. The brightness of the region of the visual acuity test device and that of the adaptation stimulus exhibition device were set to become the same. We kept room temperature at 23 degrees. The sinusoidal grating of visual stimulus was displayed with 21 inch CRT at the distance of 1[m]. The maximum brightness (white) is about 106[cd/m^2], and the minimum brightness is about 4[cd/m^2]. Because there is a study report that an adaptation effect lasts for about 1 minute[4], the visual acuity test after adaptation was carried out just after having removed a visual stimulus. In addition, we carried out an extra experiment to confirm that a visual acuity change happens by not a brightness of a uniform visual stimulus but spatial frequency adaptation due to looking at sinusoidal grating.

16 subjects were examined. They consisted of 15 men and one woman. They were from 23 to 41 years old. We found that the visual acuities for 13 subjects did not change just after adaptation to a visual stimulus with uniform brightness. The visual acuities changed for the better with two subjects and worse with one subject. As a whole the visual acuity will not change after adaptation to uniform brightness stimuli.

4.2 Adaptation to a Grating of Single Spatial Frequency

We show the results in Fig.5. The visual acuities changed for the better with five out of ten subjects when we showed a vertical sinusoidal grating of 6[cpd] as a visual stimulus, and did not change with four sub-

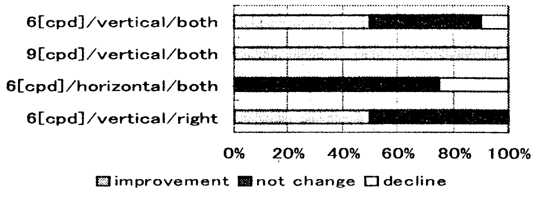


Figure 5: Results of experiments, adapted to some spatial frequencies. The vertical axis shows spatial frequency, grating orientation.

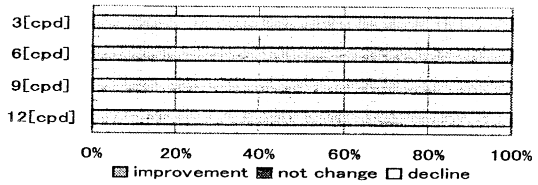


Figure 6: Results of experiments adapted to a range of frequencies.

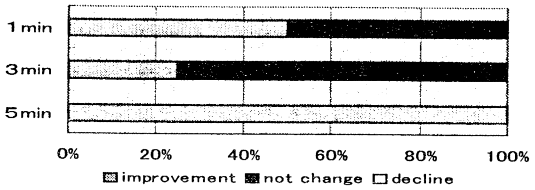


Figure 7: Results of experiments that let presentation time change.

jects. The visual acuity of the subjects that became good was less than 1.2. The visual acuity of the two subjects that did not change was more than 1.2.

Generally, as for the person that visual acuity is good, contrast sensitivity with high spatial frequency will be comparatively good. Therefore, since the spatial frequency of 6[cpd] let a subject of about 0.8 visual acuity improve, the subject with good visual acuity may improve if they adapt to higher spatial frequency. Actually we found that the visual acuity of all the members improved when we showed a vertical sinusoidal grating of 9[cpd] for six subjects whose visual acuities ware more than 1.0.

Three did not change when they were shown a horizontal sinusoidal grating of 6[cpd] as a visual stimulus for four subjects whose visual acuity improved by adaptation to a vertical sinusoidal grating of 6[cpd]. That is, the adaptation is orientation selective.

In Fig.6 is shown a result when we carried out an adaptation experiment that let spatial frequency change for four subjects who improved in both vertical sinusoidal gratings of 6[cpd] and 9[cpd]. Visual acuity of all the members improved in all spatial frequencies.

4.3 Influences of Duration of Adaptation

We carried out an experiment that let presentation time of a visual stimulus change for four subjects whose visual acuity improved to all spatial frequencies. Though the influence of presentation time was inconsistent for one minute and three minutes. Visual acuity of all the members improved for five minutes.

5 Consideration

According to the report of Nagano[4], a subject adapts in about 3 minutes. This difference is probably due to the difference in contrast and presentation method of a visual stimulus strong contrast and presentation method of a visual stimulus.

6 Conclusion

Simulation studies with our multi-channel model, it was suggested that spatial frequency adaptation will improve visual acuity. We confirmed the results by an experiment about a visual acuity change after adaptation. It was proved that our multiple channel model reproduced spatial frequency adaptation of a human visual system precisely. It was proved that spatial frequency adaptation gave influence on visual acuity.

Spatial frequency adaptation may happen to a driver running a highway. We intend to investigate the influence in the future. This study can be useful for examining road signs and for development of a human friendly road information board taking into considerate of adaptation.

References

[1] K.Uchida,et al.,"Designing Human Friendly Road Information Board for Highway Drivers from the Viewpoint of Multiple Spatial Frequency Channel Model of Human Visual System," *Proc. 1999 IEEE International Conference on System, Man, and Cybernetics*, Ⅳ, 607-611, 1999.

[2] F.W.Cambell and J.G.Robson,"Application of Fourier analysis to the visibility of gratings," *Journal of Physiology*, 197, 551-566, 1968.

[3] D.Marr,"Vision" Chapter 2, *W.H.Freeman*, 1982.

[4] T.Nagano,"Spatial Frequency Responses of the Human Visual System," *Circulars of the Electrotechnical Laboratory*, No.193, 1977.

[5] D.W.Heeley and B.Timney,"Meridional anisotropies of orientation discrimination for sine wave gratings," *Vision Research*, 28, 337-344, 1988

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Teleoperation of CAN-based Systems using a Java in the Internet

Jin-Woo Park
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea

Min-Sik Jeong
Dept. of Mechatronics
Pusan National University
Pusan, 609-735, Korea

Jang-Myung Lee
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea

{jinupark, jeongms, jmlee}@hyowon.pusan.ac.kr

Abstract

In this paper, we propose a method of constructing a CAN-based system, which supports teleoperation over the Internet. As a result, CAN-based control system is opened up for remote control, monitoring and maintenance using the Internet. Most of the system is controlled using Java language because it is small, powerful package and can easily be reused in the whole system on different platforms and also implemented in embedded system. The effectiveness· of proposed method is demonstrated through the experiments using CAN-based manipulator and active camera system with the two degree of freedom, which allow remote access.

Keywords : CAN, Remote control, Tele-robotics, Java API, JNI, JMF

1. Introduction

Remote monitoring, remote control, remote maintenance servicing and customer support are very important fields in automation industry today. Recently, one of interesting field for CAN-based system is allowing remote access via the Internet. It makes it possible to access the Internet without relation to geographical position.

Controller Area Network (CAN) is a serial bus system especially suited for networking "intelligent" devices as well as sensors and actuators within a system or sub-system. All CAN nodes are able to transmit data and several CAN nodes can request the bus simultaneously. The maximum transmission rate is specified as 1M bit/s. One of the outstanding features of the CAN protocol is its high transmission reliability. The CAN controller registers a stations error and evaluates it statistically in order to take appropriate measure [1]. Therefore it efficiently can be transmitted and received data for real-time data exchange. Furthermore it would be more attractive to access CAN-based system with standard tools and standard hardware with no need of proprietary technologies to the Internet. One of solutions is to use the Internet and the Java programming environment [2].

In this paper, we offer the method for connecting underlying CAN-based system and the Internet. Most of the system is written in Java because it is small, powerful package and can easily be reuse the whole system on different platforms and implemented in embedded system and JNI (Java Native Interface) is used for interfacing CAN-based system and Java. We make the Java-CAN API that represents the Java part of the CAN communication level. It provides some classes and interfaces to transparently access the hardware driver and therefore offers another level of abstraction to the underlying hardware. The major aim of this paper is to describe the developed software infrastructure to provide worldwide access to CAN-based system, with its focus on portability, reusability and little system requirements on the clients and the server side. The effectiveness of proposed method is demonstrated by experiences using CAN-based manipulator and active camera system with the two degree of freedom, which allow remote access. We use TCP/IP for control data transmission that provides a connection-oriented, reliable, full duplex, byte stream service and RTP (Real-time Transfer Protocol) for real-time camera image transmission that provides real-time data transmission.

This paper is organized as follows. In the section 2, we simply discuss about Java and CAN. Section 3 deals with proposed system architecture, and how to connect CAN-based control system to the Internet using JAVA and JNI. We apply proposed method to CAN-based manipulator and active camera system with the two degree of freedom. In the section 4, Experimental implementation and results is showed. Section 5 present conclusions drawn from this work and future work.

2. CAN and Java

In order to teleoperate the CAN-based system via the Internet, we use Java language because it is small, powerful package and can easily be reuse the whole system on different platforms and implemented in embedded system. We brief on CAN and Java at this section.

2.1 CAN

The CAN was originally developed to support cheap and rather simple automotive applications. However, because of its performance and low coat, it is also being considered in automated manufacturing and process control environments to interconnect intelligent devices such as modern sensors and actuators. One of the outstanding features of the CAN protocol is its high transmission reliability. The CAN controller registers a stations error and evaluates it statistically in order to take appropriate measure. Therefore it efficiently can be transmitted and received data for real-time data exchange [3].

2.2 Java

The object-oriented programming language Java was originally developed in a research project for electronic consumer devices with distributed control. A fundamental requirement was to design a programming language that supports the development of robust applications. A java platform does not only consist of a Java Virtual Machine (JVM), it additionally provides various standard application programming interface (API) which are uniform for any system. These include functionality for networking or graphical user-interfaces. JVM implementations are even embedded in Web-browers, which allow client systems to download HTML-pages with integrated Java programs (applets) via the Internet and to execute the Java applications in the browser environment. These applets are able to interact with applications on server systems [4].

3. System Architecture

We aim at teleoperating the CAN-based manipulator through the Internet using Java. Fig.1 shows the overall diagram of the system.

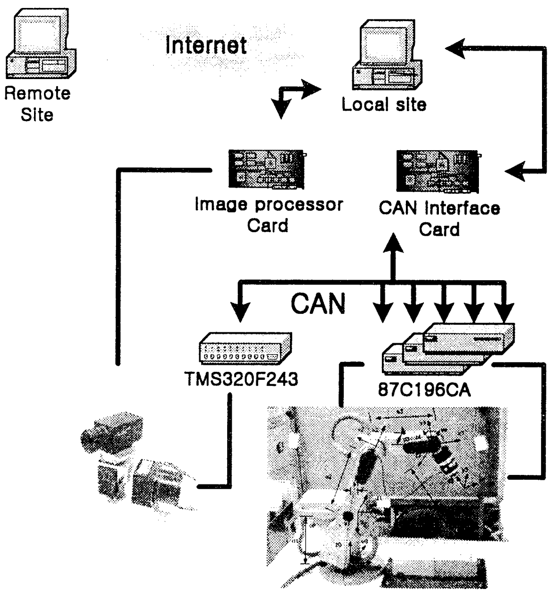


Fig. 1. Overview of the system architecture

3.1 Hardware Setup

3.1.1. Active camera system

In order to get a visual feedback of the current display status, we install two degrees of freedom of the camera system, which delivers image of manipulator.

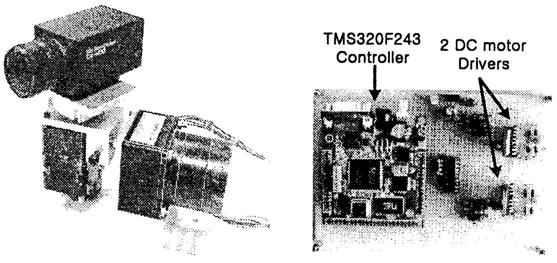


Fig. 2. Two degrees of freedom of the camera plantform (Left) and Controller (Left)

Active camera system consists of TMS320F243 DSP controller that supports CAN [5] and two motor drivers that are shown in Fig. 2.

3.1.2. 5-d.o.f manipulator

SCOBOT-ER VII is used for client system, visible in Fig 3. Originally, SCORBOT-ER VII that is made by Eshed Robotics in 1991 communicated with host computer through serial. It is impossible for implementing out project that is the development of teleoperation surgery system because serial communication between two controllers is too late for real-tome data exchange. So, we have also applied CAN system to robot controller for real-time communication and redesign hardware instead of old robot controller. It consists of 5 sub-controllers that are based on micro-controller (80196CA) and host controller (Windows NT). Host controller calculates trajectory generation of robot and sends calculated value to sub controller. Sub controller receives commands of host computer through CAN and controls each axis of robot and sends position, velocity information to host computer. It also has force-torque sensor for reflecting force [6].

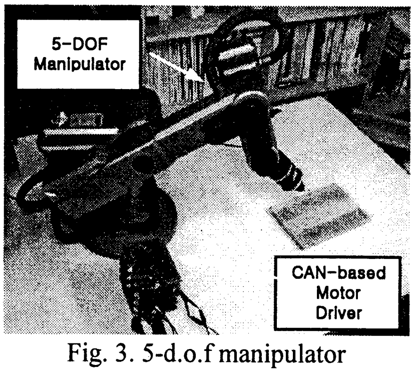


Fig. 3. 5-d.o.f manipulator

3.2 Software Setup (Interface and communication)

We use JNI(Java Native Interface) for interfacing CAN and Java and utilize TCP/IP for control data transmission that provides a connection-oriented, reliable, full duplex, byte stream service and RTP (Real-time Transfer Protocol) for real-time image transmission that provides real-time data transmission.

3.2.1 JNI (Java Native Interface)

Low-level CAN communication modules are built in Visual C++ and accessed through the JNI, as shown in Fig. 4. The system is not limited to CAN or a specific application layer protocol because TCP/IP is only used as a transport mechanism. CAN can be replaced by any other fieldbus offering a corresponding interface.

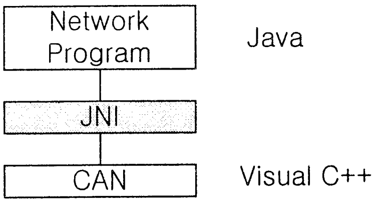
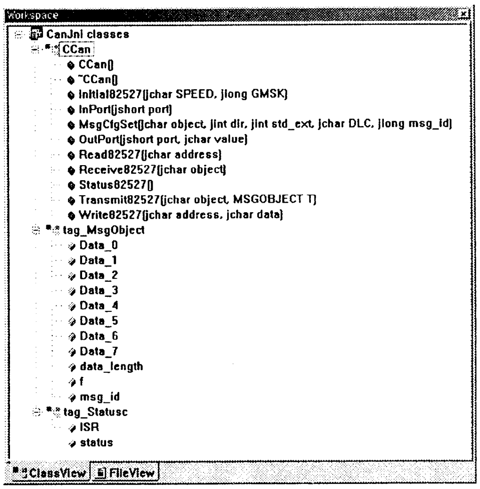


Fig. 4. Block diagram of Java-CAN Interface

The Java-CAN API is formed by the CCan package that is shown in Fig. 5. The CCan package provides Java classes and interface to access a PC-CAN interface card that that has been developed in the laboratory for Intelligent Robotics via JNI. The *MsgCfgSet* class encapsulates a generic CAN message with an 11 bit or 29 bit identifier, 8 bytes of data and the data length code at the Fig. 5. Fig. 6 shows the example of Java and CAN programming to access CAN to Java. (i.e. CanJni.cpp and CanJni.java are written by Visual C++ and Java respectively)



```
CanJni classes
  CCan
    CCan()
    ~CCan()
    Initial82527(jchar SPEED, jlong GMSK)
    InPort(jshort port)
    MsgCfgSet(jchar object, jint dir, jint std_ext, jchar DLC, jlong msg_id)
    OutPort(jshort port, jchar value)
    Read82527(jchar address)
    Receive82527(jchar object)
    Status82527()
    Transmit82527(jchar object, MSGOBJECT T)
    Write82527(jchar address, jchar data)
  tag_MsgObject
    Data_0
    Data_1
    Data_2
    Data_3
    Data_4
    Data_5
    Data_6
    Data_7
    data_length
    f
    msg_id
  tag_Statusc
    ISR
    status
```

Fig. 5. Classes for JNI

3.2.2 Graphics Display

Recently, development trends in teleoperatiors control technology are aimed at amplifying the advantages and alleviating the disadvantage of the human element in teleoperation through the development and use of computer graphics or virtual reality displays, and new computer-based human-machine interface devices and techniques. For this reason, we put emphasis on graphic display in teleoperation. And then, Task visualization is a key problem in teleoperation, because most of the operator's control decisions are based in visual or visually conveyed information [7]. In this paper, we use JMF™ 2.0 API (Java Multimedia Framework) for real-time image transmission. It enables the display, capture and streaming of multimedia data in Java technology-based applications and applets. As a standard extension to the Java platform, it delivers the ease-of-development and cross-platform benefits of the Java programming environment to developers incorporating media data.

```
<CanJni.cpp>
JNIEXPORT void JNICALL Java_CanJni_takeCanJni(JNIEnv* env, jobject obj, jchar id, jdouble value)
{
        CCan m_Can;
        m_Can.Initial82527(b50K,0xfffffff8); // (baud rate, identifier)
        m_Can.MsgCfgSet(1,RX,std,2,3);       // (object number, direction[RX/TX], standard/extent format[std/ext],
                                             //  data length code[0..8], identifier)
        m_Can.Write82527(P2_out,id); }       // (port, identifier)

<CanJni.java>
public class CanJni
{       public native double CanJni_takeCanJni(jobject obj, jchar id, jdouble value);
        static
        {
         System.loadlibrary("CanJni");     }
         public static void main(String[] args) {
             CanJni app = new CanJni();   } }
```

Fig. 6. Example of programming for JNI

such as audio and video into their applets and applications. Fig. 7 shows the JMF architecture that consists of controlling player, processor and installing new plug-in components [8].

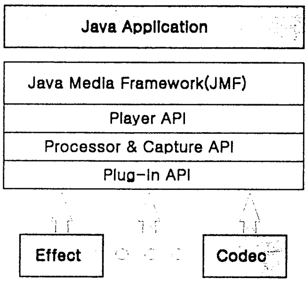


Fig. 7. The JMF structure

4. Experimental Implementation

Experimental black diagram of the system is depicted in Fig. 8. Remote site can confirm the status of manipulator using image transmission. The image is transmitted by real time protocol of JMF to Remote site. RTP is suitable to transfer huge data like image, sound.

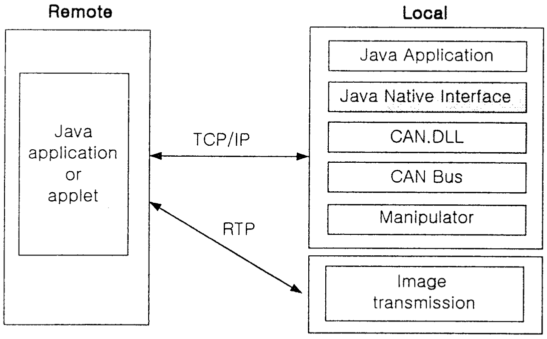
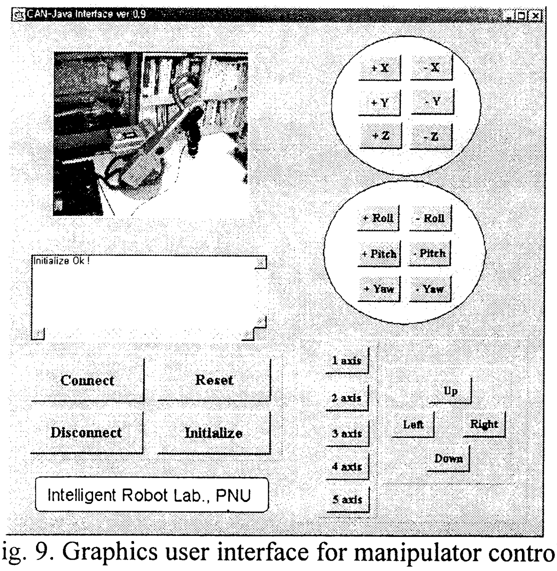


Fig. 8. Block diagram of the experimental setup



Initialize Ok !

Fig. 9. Graphics user interface for manipulator control

Fig. 9 shows the result of teleoperation of manipulator at the remote site. In order to provide an interactive interface to the manipulator, the provided form can be controlled interactively using the buttons *(+X, +Y, +Roll, +Pitch, 1 axis, 2axis, Up, Down* button etc.)

Conclusions

Currently, there is a great interest for distributed control systems in the automation part. Especially the Internet environment promises new application fields. However, we can see a gap between today's automation systems like CAN-based system and the Internet environment. Therefore, in this paper, we offered the method for connecting underlying CAN-based system and the Internet. Most of the system is written in Java because it is small, powerful package and the whole system can easily be reused on different platforms and implemented in embedded system. JNI (Java Native Interface) is used for interfacing CAN-based system and Java.

Notice that the growing developments in the Internet area and the increasing use of Intranets offer a good starting platform towards well supported, user-friendly, integrative and unique user interfaces to CAN as a part of industrial or control communication solutions, and will help to increase their acceptance.

As a future work, we need to more study the fields of user interface, network security and multiple accesses. It is also necessary to overcome time-delay that is occurred during transmission via the Internet.

References

[1] CAN-the technical introduction, CAN in Automation, http://www.can-cia.de/ican.htm.

[2] G.Gruhler, W.Kuchlin, and Th.Lumpp, "Accessing CAN-based automation systems via the Internet", *CAN Newsletter*, vol.1, pp.24-28, March 1998.

[3] W.Lawrenz, "CAN System Engineering From Theory to Pactical Applications", *Springer*, 1997.

[4] J. Gosling and H. McGilton., The Java Language Environment: A White Paper, *Addison-Welsey, Mountian View*, CA, 1995.

[5] Texas Instruments, http://www.ti.com.

[6] J.W. Park, D.G. Roh and J.M. Lee, "Design of CAN-based System for Distributed Control," *Proceedings of ITC-CSCC*, pp. 600-603, 2000.

[7] A.K. Bejczy, P. Fiorini, W.S. Kim and P. Schenker, "Toward Integrated Operator Interface for Advanced Teleoperation under Time-Delay," *Proc. of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pp. 563-570, 1994.

[8] Sun Microsystems, http://java.sun.com/products /java-media/jmf/index.html.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Self-Organization of Behavior-based World Model for Autonomous Mobile Robot

Satoru Ishigaki , Masaaki Ida , Osamu Katai
Graduate School of Informatics
Kyoto University
Yoshida Honmachi, Sakyo, Kyoto 606-8501, Japan

Abstract

In this paper, we present a world modeling system of behavior based mobile robot. Our robot control system is based on Subsumption Architecture which is a class of behavior-based control system. In addition to the advantages of the control system, we apply Self-Organizing Map to create internal symbols which represent external world based on time series of robot's behaviors. We present results of experiments in order to demonstrate the efficiency of our system.

1 Introduction

In design of control system of an autonomous mobile robot, it is an important subject to generate world models. However, there might be discordance between a model and the real environment. The problem is called Symbol Grounding Problem [1].

Subsumption Architecture (SA) is a class of behavior-based control system [2]. An SA control system is constructed by the set of behavior modules, i.e., each module executed asynchronously connects its input information to output information directly. By means of the advantages of its quick execution cycle and distributed control processing, SA control systems have been applied in various noisy and changeable environments. However, for more actual and complex planning problems such as global path planning or map generation, environment description, or world model, is indispensable for more intelligent SA control system.

In this research, we propose a learning system which generates a world model by self-organization based on behavior such that the world model is suitable for the behavior-based robot.

2 World modeling based on behavior

The fundamental concept of environmental modeling in this research is based on Sensory-Motor Coordination [3]. The idea of Sensory-Motor Coordination is that the model of external world is generated not only from sensory information but also from the relation of sensory information and action. In this context we proposed a method to form symbols which is suitable to be coupled with the behavior-based systems [4]. It is based on the idea that categories of objects in an environment are to be extracted from the concepts about the categories of the environmental structure.

In this research, the outputs of the behavior modules express the relation between sensory information and action. Using behavior modules, our system becomes possible to generate a world model which reflect the internal state of the robot which does not appear in robot's motor output. We adopt Self-Organizing Map (SOM) [5] to learn world model from the outputs of behavior modules.

SOM is a kind of a competitive-type neural network, and is a self-organization algorithm which clusters the input information in higher dimensional space to the state space of lower dimension. Self-organization by SOM makes it possible to obtain world model specific to the robot through the actual actions. Therefore Symbol Grounding Problem is expected to be resolved.

3 System architecture

The system proposed in this research consists of the following three stages (Figure 1):

(1) **Extraction of behavior states**
The activated state value is calculated for each behavior module while SA control system functioning.

(2) **Clusterization of behavior states (SOM1)**
Aggregating the values of all behavior modules, Self-Organizing Map (SOM1) improves its learning parameters to recognize similar and characteristic relations among the realized actions. The

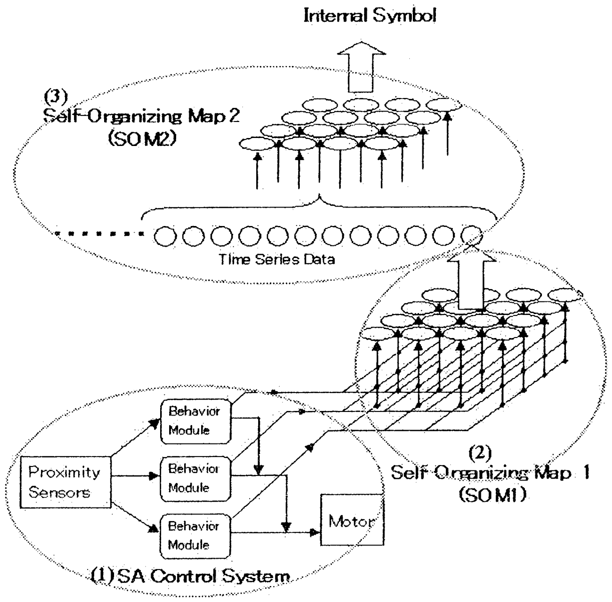


Figure 1: System architecture

sequential outputs from SOM1 are recorded as time series data after learning process.

(3) **Self-organization of internal symbols (SOM2)**
The parts of time series data in which continuous changes happen in a neighborhood of each data are extracted from data of SOM1. SOM2 executes learning from the outputs of SOM1 in the continuation data. The outputs of SOM2 after learning are regarded as being sets of recognized landmarks (internal symbols) representing the local landscapes.

The features of our proposed system are as follows:

- **Filtering:** Because of using behavior information, only the information which is meaningful for the robot's actions can be extracted out of input information.
- **Expandability:** The system is based on an unsupervised learning. Therefore in case some behavior modules are newly added or changed in the control system, the system only executes learning process in an adequate learning period for readjustment. By improving the robot's behavior, internal symbols are also improved according to the new behavior.
- **Adaptability:** The system confers the output information from existing behavior modules, therefore any particular type of environment is not presumed for our recognition system. Even when the robot is in different environment, the internal symbols which is suitable for the environment can be obtained by executing learning process again.
- **Subjective recognition:** There is no guarantee that the expression of world model is easy to be understood by human being, however it is possible to make a world model which is meaningful and rational for the robot which has restricted sensory and action abilities.

4 Experimental results

In this section, we present results of experiments in order to demonstrate the efficiency of our system. The dotted lines in each figure express the robot's trajectory, and each number expresses internal symbol which corresponds to the local position. We note that the numbers in each figure are unrelated to the numbers in other figures.

4.1 Change of the internal symbol by addition of behavior modules

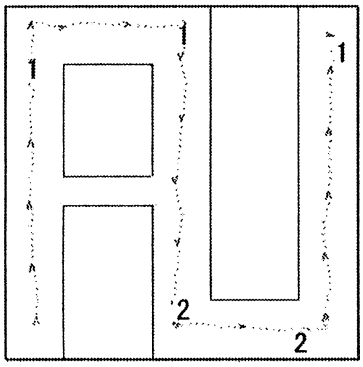


Figure 2: Behavior of a simple architecture

Figure 2 shows the result of the experiment by the control system which consists of four behavior modules (Avoid obstacle, Move along left wall, Move along right wall, Move forward). SOM1 consists of 5 × 5 units and SOM2 consists of 2×2 units. It can be seen that internal symbol 1 corresponds to right turn, and internal symbol 2 corresponds to left turn. Only the landscape feature which has an effect on robot's action is filtered and is reflected in the internal symbol.

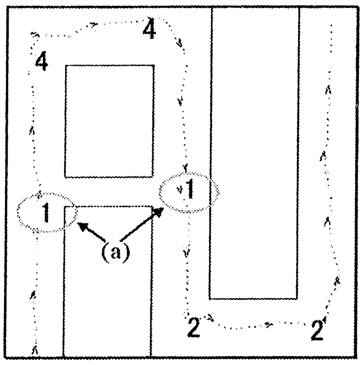


Figure 3: Behavior of an augmented architecture

Next, system is expanded by adding two behavior modules (Left turn, Right turn). The function of new behavior modules is to alter its direction towards the side where the wall disapears. Figure 3 shows the result when experimenting by the control system which consists of a total of six behavior modules. The system architecture is not changed from previous experiment without the modification of behavior modules. In addition to turning position of either side, the place where the robot can go by a narrow passage has also been recognized as an internal symbol. These internal symbols are indicated as (a) in Figure 3. The result means that the new internal symbols are generated according to the expanded action.

4.2 Adaptation in different environment

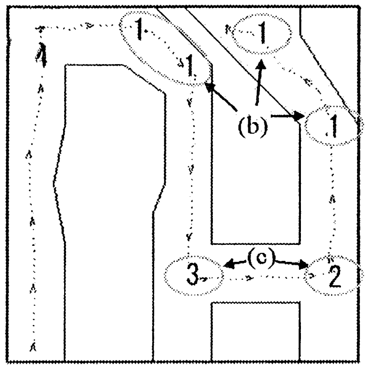


Figure 4: Application to a new environment

Figure 4 shows the result of experiment in more complicated environment. The behavior control system is the same as previous experiment. Internal symbols 2 and 3 in the figure (c) correspond to turning left. Different internal symbols such as 2 and 3 correspond to the place where the robot turned left in a similar manner. The reason is that because of using the output information of behavior modules, the difference in the internal state in the process of determining action is reflected to internal symbol although the resultant action is the same.

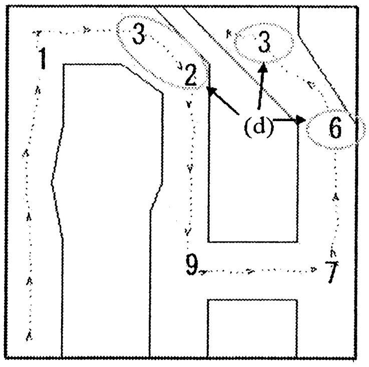


Figure 5: Increase of unit number

Internal symbol 1 in the figure (b) corresponds to the gradual corner which does not exist in the former environment. The differences between these gradual corners are not distinguished because of the shortage of output units of SOM2.

In order to obtain the internal symbols suitable for complicated landscape features, the number of units in SOM2 is increased to 3 × 3 (Figure 5). The gradual corners are distinguished as internal symbols 2, 3, and 6 in the figure (d) according to these landscape difference.

This experiment shows that the internal symbols which adapt to a new environment can be introduced also with using the same system architecture. This is because the internal symbol is made out of not only the sensory information but also of the relation of sensory information and action.

4.3 Formation of the internal symbol in complicated environments

Figure 6 and Figure 7 are the results by experiments in a complicated larger environment. The system architecture is the same as the above experiment. The internal symbols obtained as a result of learning are roughly divided into two groups. Internal symbols 2, 3, 5 and 7 correspond to the positions of turning left, and internal symbols 4 and 9 correspond to the positions of turning right. Strict distinction cannot be obtained although there is different tendency for each

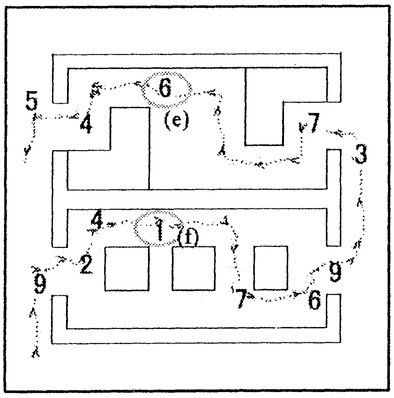


Figure 6: Application to complex environment 1

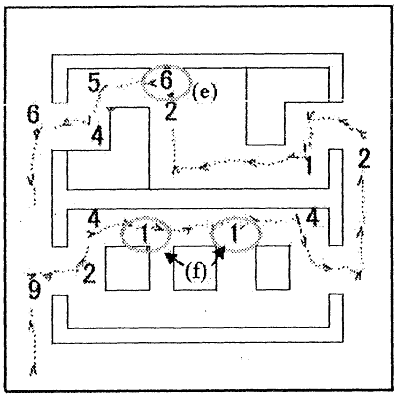


Figure 7: Application to complex environment 2

internal symbols which seems to correspond to the same landscape feature. The reason why landscape feature and the internal symbol do not correspond to each other in a one-on-one is that we use reflective actions of a behavior-based robot. It is considered that this problem can be resolved by associating the internal symbols corresponding to the same landscape feature.

As other characteristic internal symbols, internal symbol 1 in the figure (f) corresponds to passing by side of a narrow path, and internal symbol 6 in the figure (e) corresponds to advancing position into a narrow path from large space. Although specific actions were not set up to such landscape features, actions which adjusted to the landscape features take place because of the interaction with the environment. Therefore the internal symbols which are adapted to the landscape features are generated. Thus, the feature of the behavior-based robot that it can accommodate flexibly to the environment is reflected also in the generation of the internal symbols.

5 Conclusion

The internal symbols obtained by the above experiments express the important landscape features which the robot extracted through its actions, so do not necessarily correspond to the landscape feature which we human beings pay attention to. Therefore, in some cases the different internal symbol may corresponds to the landscape features which seems to be similar by recognized by us, or in the other cases, different landscape features may correspond to the same internal symbol. Thus, it is considered more natural that how landscape features recognized changes according to subjects who recognize the landscape. In many conventional researches, the robot is imposed by hard burdens and restrictions as the result of being made to recognize landscape features similar to the interpretation by human being. In this research, our proposed method abandons the interpretation by human being and adopts internal symbols specified to the robots' style recognition. There it is regarded as a meaningful world model for the robot itself or the robots of the same type. Although it is difficult to use an internal symbol for interaction of a robot and human being directly, it is suitable for robots' planning and communication.

References

[1] S. Harnad: "The Symbol Grounding Problem" *Physica, D 42*, pp. 335–346, 1990

[2] R. Brooks: "A Robust Layered Control System for A Mobile Robot" *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, 1986

[3] R. Pfeifer and C. Scheier: "Sensory-Motor Coordination: the metaphor beyond" ***Robotics and Autonomous Systems, Special Issue on "Practice and Future of Autonomous Agents"20***, 1997

[4] A. Fukayama, M. Ida, and O. Katai: "Behavior-based Fuzzy Control System for a Mobile Robot with Environment Recognition by Sensory-Motor Coordination" *Proceedings of Fuzz-IEEE99*, pp. 105–110, 1999

[5] T. Kohonen: "The Self-Organizing Map" *Proceedings of the IEEE*, 78(9) pp. 1464–1480, 1990

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Behavior Patterns Emerging among Mobile Robots with a Diversity of Personalities Cooperating in Collection Cleaning-up Tasks

○E.Uozumi*, Y.Sagawa * *, N.Sugie +
Department of Electrical and Electronic Engineering
Graduate School of Science and Technology, Meijo University
1-5-1 Shiogamaguchi, Tempaku-ku, Nagoya, 468-8502, Japan
*c3992003@meijo-u.ac.jp, **sagawa@meijo-u.ac.jp
+sugie@meijo-u.ac.jp

Abstract

We investigate the behavior patterns emerging among autonomous mobile robots with a diversity of personalities. They cooperate one another in collection cleaning-up tasks. First we emphasize the rationale for introducing personalities into robots. Second we present the assumptions on the common functions of the robots; the visual function, the mobility, and the capturing-and-conveying function of target objects. Third we define a function specific to each robot; personalities. Fourth we explain the collision avoidance behavior which is very much dependent on personalities. Finally we present the results of simulation studies of robots with a diversity of personalities cooperating in collection cleaning-up tasks. We discuss the influence of the distribution of various personalities on the performance of tasks.

Key words: Mobile robot, multiple agents, cooperation, personality, collision

1 Introduction

Many studies have been done on the behaviors of multiple robots cooperating or competing with one another.

In [5], a joint intentions theory is proposed for better team work.

In [4], each robot is controlled by its own controller. There are a leader robot and the follower robots.

In contrast, homogeneous implementation is assumed in [2].

To cope with interference, a dominance hierarchy is introduced allowing only one of robots to act at one time. Proposed scheme is caste arbitration implementation, which may remind us of social insects such as ants.

In this paper, we introduce personality of robot. By varying personalities of robots that perform a task in group, they can avoid collisions and perform the task effectively. As humans [3], some robots are more positive and others are rather reserved. We define the degree of positivity to represent the personality of each robot. When two robots are going to collide with each other, the less positive robot will concede and detour. Thus collision avoidance is implemented straightforwardly without introducing complicated procedure.

We report some of the results of preliminary simulation studies with a diversity of personality distribution and discuss the significance of various rather than single personalities.

2 Tasks

Robots are located in a room. The task subjected to them is to collect target objects in the room and transport them to a certain spot (hereafter called 'Origin'). It may happen that two or more robots move toward one and the same object resulting in collisions, if some proper measure is not taken. But when it detects the risk of collision with another robot, it try to avoid it in a way described later.

The performance of the task is evaluated by the time needed to transport all the target objects to the Origin.

3 Functions of Robots

Each robot has the following functions.
(1) Each robot has an omnidirectional 2D visual field.
(2) Each robot can move in any 2D direction, with one and the same speed.
(3) The motion trajectory of each robot consists of

piecewise linear lines.

(4) When a robot detects target objects, it moves toward the object in the shortest distance from the robot to capture it.

(5) Each robot can capture and transport only a single object at once.

(6) After a robot captures a target object, it moves with the object straight ahead toward the Origin

(7) Each robot tries to capture and transport objects as long as there are objects left uncollected.

Moreover, we define "personality" which characterizes each robot. Personality is a rather complicated and controversial concept[3]. Here we cite the definition by R.B.Cattell: personality enables us to predict what a human of a certain personality would do in given situations.

We will describe in short order how to implement "personality" into robots.

4 Personality and Collision Avoidance Behavior

We define the personality as the degree of positivity of each robot. If one robot is more positive than the other, the former will concede the way and detour just before collision.

We assume that the degree of positivity of other robots can be judged only when they come close enough within the sphere of observation of the robot in concern, which will be defined in short order.

4.1 Judging Collision

Fig.1 shows how collision is detected. A circle around the robot(Fig.1(a)) indicates the sphere of observation of the robot. When another robot is going to invade the sphere of observation, collision judgement starts(Fig.1(b)). Each robot is associated with its motion vector (an arrow).

The motion of the robot2 relative to the robot1 can be obtained by subtracting the motion vector , or adding the vector depicted as a dotted arrow, of robot1 from that of robot2.

The circle shown in Fig.1(b) is a different circle from the circle shown in Fig.1(a). The circle indicates the sphere of influence of the robot. The robot does not permit an invasion of the sphere by any other robots.

If the direction of the relative motion vector of robot2 goes through the sphere of influence of robot1, it conclude that robot2 will collide with it. If the relative motion vector of robot2 does not go through the sphere of influence of robot1, no collision is detected (Fig.1(c)).

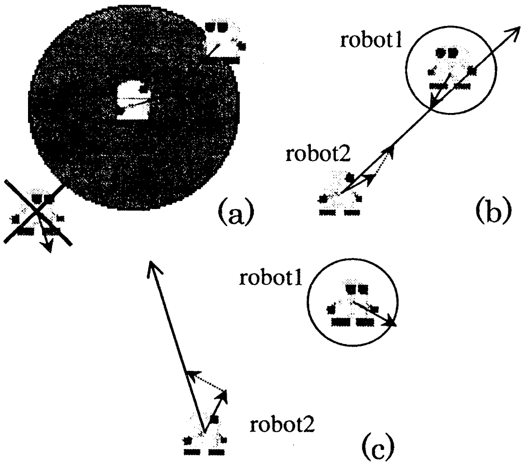


Figure 1: Sequence of collision judgement

(a) a situation where a robot entering the sphere of observation of another robot.
arrows: motion vectors
(b) the relative motion vector–long straight arrow–goes through the sphere of influence predicting a foreseeable collision.
dotted arrow: inverse motion vector of robot1
(c) case of no collision foreseeable collision

4.2 Avoiding Collision

Fig.2 shows the procedure of collision avoidance. In Fig.2(a), two tangent lines, emanating from robot2, to the sphere of influence of robot1 are shown.

The circle shown in Fig.2(b) is formed from the end point of the arrow which is opposite to that for robot1 in order to form the vector for avoiding collision. Note that the radius of the circle is equal to the speed of motion of robots. Then referring to Fig.2(b), two relative motion vectors are obtained to avoid collision. These two vectors are used to determine two candidates for the motion vector.

Fig.2(c) shows these two candidates of motion vectors of robot2, out of which one is chosen from the standpoint of less chance of collision.

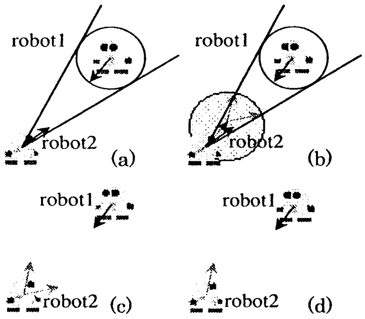


Figure 2: Sequence of collision avoidance procedure

(a) determining tangent lines
(b) obtaining relative motion vectors avoiding collision
(c) two candidates for the motion vector of conceding robot
(d) finally determined motion vector

5 Performance Dependent on Personality Distribution

We report the results of two simulation studies, where robots are asked to carry objects to the Origin. In 5.1, robots behave in accord with the rules described above. In 5.2, each robot behaves on the basis of the predicted behaviors of other robots in addition to the basic rules in 5.1.

5.1 Basic Performance

In Fig.3 are shown the trajectories of each robot when five robots collected five objects at the Origin. The degrees of positivity of robot0, robotl, robot2, robot3 and robot4 are 4, 3, 1, 5and 2, respectively. The degree 5 is the most positive, and the degree 1 is the least positive.

In Fig.3, robot0, robotl, and robot2 move toward the object in the center of the room. Since robot0 is more positive than robotl, robot2 captures the object and carries it to the Origin. On the other hand, when robotl moves toward the object in the center of the room, it notices halfway robot0 moving toward the object and concedes to avoid. Thus robotl moves upward for a very short while, and then robotl takes the object at the upper-left, and carries it to the origin.

As for robot2, it notices halfway robot0 and robotl moving toward the object in the center of the room and thus concedes to avoid collision with robot0 and robotl. Now at first robot2 moves upward and downward for a while to avoid collision and then robot2 takes the object at the lower-left and carries it to the Origin.

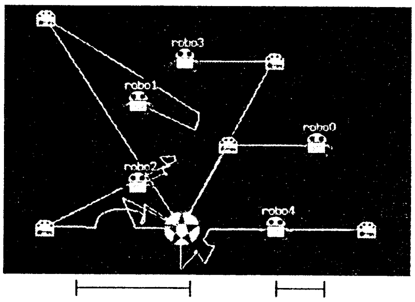


Figure 3: Trajectories of robots 0,1,2,3,and 4, whose degrees of positivity are 4,3,1,5,and 2, respectively. Two line segments, from left to right, below the figure denote the radii of the spheres of observation, and that of influence, respectively.

	Degree of positivity of robots 0,1,2,3,4	Steps required	Average traversed distance of each robot
Maximum steps required	3,5,4,2,1	1665	896.63
Minimum steps required	4,3,1,5,2	983	435.50
	4,3,2,5,1	983	436.96
	5,3,1,4,2	983	435.50
	5,3,2,4,1	983	436.96
Average steps of 120 trials		1216.28	601.68

Table 1: The cases where each robot is different from others in the degree of positivity.

robot3 moves toward the object on its right, and robot4 moves toward the object on robot4s right. And then, robot3 and robot4 capture each object and carry it to the Origin.

In Tablel is shown the simulation result for various personality distributions.

5.2 Performance with the Cooperation Work

In the next simulation, robots perform cooperative work in which the robot may concedes the object by predicting behavior of other robots. Whether each robot concedes the object or not depends on the degree of positivity of robot as well as relation of distances to objects from robots.

The robots shown in Table2 have the sphere of observation that is bigger than that shown in Fig.1(a). When other robots invade the sphere of observation, each robot investigates whether the object it is going to capture is the same as the object which other robot approacheds. If the object is the same, the robot may have to concede the object to other robots. It depends on the comparison of the personality distribution of robots and on the comparison of distances from robots.

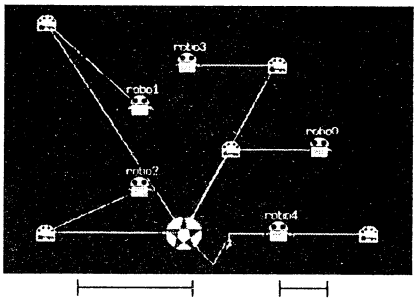


Figure 4: Trajectories of robots 0,1,2,3,and 4, whose degrees of positivity are 5,4,2,3,and 1, respectively. Two line segments, from left to right, below the figure denote the radii of the spheres of observation, and that of influence, respectively.

	Degree of positivity of robots 0,1,2,3,4	Steps required	Average traversed distance of each robot
Maximum steps required	3,4,1,5,2	1450	649.36
	3,5,1,4,2	1450	648.16
Minimum steps required	4,2,3,5,1	934	430.15
	5,2,3,4,1	934	442.29
Average steps of 120 trials		1128.30	518.15

Table 2: The cases where each robot predicts the behavior of other robots.

	Degree of positivity of robots 0,1,2,3,4	Steps required	Average traversed distance of each robot
Maximum steps required	2,3,1,5,4	2089	998.33
Minimum steps required	5,4,2,3,1	732	321.72
	5,4,3,1,2	732	319.94
	5,4,3,2,1	732	321.72
Average steps of 120 trials		1010.72	443.50

Table 3: The cases where each robot predicts the behavior of other robots with a very large sphere of observation.

When the steps required and average traversed distance of each robot of the fifth row of Table2 are compared with those of the sixth row of Table1, each value of Table2 is smaller than that of each value of Table1.

Moreover, the robots in the case shown in Table3 have the sphere of observation that contain all the territories shown in Fig.4. In Fig.4 are shown the trajectories of each robot of Table3 with 732 steps. This is the case showing the best performance of all cases.

6 Summary

We proposed a scheme of collision avoidance among mobile robots and obstacles. We introduced a concept of personality related to the degree of positivity, which is exploited to decide who should concede and detour to avoid collision.

We conducted simulation studies based on the above scheme and found rather interesting results. Additionally, we found the predominance of the work efficiency by the cooperation of the robots from that result.

What happens if the tasks are performed by robots that can understand all the other's behavior simultaneously and cooperate each other?

Another issue may be related to the distribution of personalities. In evolutionary computing [1], an individual with the best performance is obtained through genetic algorithm. However, we humans are endowed with a diversity of personalities, the distribution itself, rather than individual, of which might be related to the performance index of human being.

References

[1] D.B.Fogel, "What is Evolutionary Computation?," *IEEE Spectrum*, pp.26-32, February 2000.

[2] D.Goldberg, and M.J.Mataric, "Interference as a Tool for Designing and Evaluating Multi-Rogot Controlleres," *Proc. AAAI-99*, pp.637- 642, Orlando, Florida, July 1999.

[3] Heibonsha, "The Encyclopedia of Psychology , New Edition," *Heibonnsha Publishing Company*, Tokyo, Japan, 1994.

[4] K. Kosuge, T.Oosumi, K.Chiba, "Decentralized Control of Multiple Mobile robots Handling a Single Object in Coordination," *J. of Robotics Society of Japan*, Vol.16, No.1, pp.87-95, January 1998.

[5] M. Tambe, "Agent Architectures for Flexible, Practical Teamwork," *Proc. AAAI-99*, pp.22-28, Providence, Rhode Island, July 1997.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyc Japan, 15-17, January, 2001,

Adaptive and Economic Data Representation in Control Architectures of Autonomous Real World Robots

J. Fischer
GMD (AIS.ARC)
National Research Center
for Information Technology
Autonomous Intelligent Systems
Sankt Augustin, 53754, Germany
joern.fischer@gmd.de

R. Breithaupt
GMD(AIS.ARC)
National Research Center
for Information Technology
Autonomous Intelligent Systems
Sankt Augustin, 53754, Germany
ralph.breitha@gmd.de

Dr. M. Bode
Cortologic AG Berlin
& Westfaehlische Wilhelms
University of Muenster
Departement of Applied Physics
Muenster, 48149, Germany
bode@cortologic.de

Abstract

Learning algorithms of autonomous robots in complex, real world environments usually have to deal with many degrees of freedom and continuous state spaces. Reinforcement learning is a promising concept for unsupervised learning but common algorithms suffer from huge storage and calculation requirements to construct an internal model by estimating a value-function for every action in every possible state. In our approach to approximate this function with lowest costs we introduce a flexible method that focuses upon states of highest interest and interpolate between them with a fast and easy to implement algorithm. In order to provide highest accuracy to any predefined limit we enhanced this algorithm by a fast converging multi-layer error-approximator.

1 Introduction

Reinforcement learning (RL) algorithms are very successful in unsupervised learning tasks of agents acting in a completely unknown environments with an unspecific scalar reward signal given sporadically in certain states as the only error-feedback. The basic idea is to extrapolate a value function on an internal state space model for every action in every possible state of the agent defining the most promising behavior in order to maximize the total reward. Not knowing how the state space must be discretized to be able to solve the task satisfactory common RL-algorithms usually have to face the problem of very high storage demands and many exploration efforts learning mostly unimportant or already known information[1]. In our recent work we introduced very promising methods that enable the agent to learn its state space discretization efficiently by itself by using a vector quantizer (VQ) to classify areas of similar Q-values [2][3]. Unfortunately, the economic use of state representations leads to coarse and unsteady approximations of the ideal value function. That spoils the idea of a value-gradient pointing to the goal and often results in sub-optimal behavior of the agent.

To enhance this algorithm we now introduce a simple but efficient interpolation method. The approximated value-functions become steady and generally even fewer prototypes are needed with a lower mean-approximation error. For several Reinforcement Learning algorithms [4][5] it is essential to know the value-function as precisely as possible especially in its minima and maxima. To achieve this goal an adapted prototype learning rule is initiated which leads to minimum and maximum representation and an even more economical one layer approximation method. Further enhancement leads to multiple layer error-function approximation in the same manner, which makes it possible to reduce the error down to a predefined minimum. This is supported by thc capability to expand the informational capacity of the VQ [5][6] without destroying already stored information. Advantages of this concept are demonstrated in a reinforcement learning robot control simulation. Further on we qualitatively compare this approximation method in the context of reinforcement learning to associative memory and backpropagation in neural networks.

2 A Vectorquantizer as a Function Approximator

Discretizing a continuous state space means to make use of similarities on certain areas due to the goal of the agent. A well explored way to find such areas is to use a vector quantizer, where each similarity-class subspace is represented by a single prototype-vector [2]. Defining a metric on the state space, each possible state of the agent is attached to the "nearest" prototype. In that way the state space is divided into Voronoi-cells [7]. One of the possible learning rules to find the right positions for the prototypes is to move each "winning" prototype vector in the direction of the presented state vector if it is of the appropriate class. Thus the prototypes approximate the center of their respective class. This implies that the "used" space, where states are often presented, is resolved better, than the "unused" space. Other learning strategies are e.g. proposed by Kohonen [8][9].

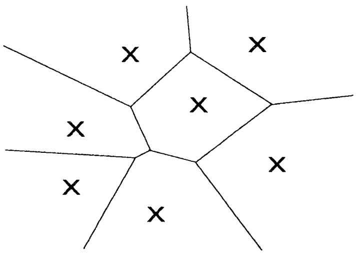


Fig.1: Prototypes in a 2-dimensional input space and their respective Voronoi-cells.

Treating the Voronoi-cells just like the states of a common reinforcement Q-learning algorithm, each prototype has to store Q-values for every action, computed by one of the existing methods of Q-value estimation. E.g. in [3] we presented a VQ that automatically found a minimal number of prototypes needed to solve the given tasks in combination with Q-learning. The economic use of VQ-prototypes is always a compromise between the minimized expenses for storage and learning speed and the accuracy (resolution) of the value-function approximation which is of course unsteady at the transitions of the Voronoi-cells. So the behavior of the agent is satisfactory but mostly not optimal.
In this paper we first want to focus more generally on the principles of approximating an optimal value-function on the state space. To obtain a universal basis of value-function approximation we want to allow functions with several minima and maxima.
With a common VQ often many prototypes are needed to get a proper approximation. Fig.2 shows an example of a value function of a 2-dimensional state space, approximated by 1200 randomly distributed prototypes (here learning the mean values of their respective areas). Even with the considerable number of prototypes the approximation of this simple example (Fig.2a) is still very rough and unsteady. In higher dimensions much more prototypes are necessary to approximate even simple functions with high accuracy.

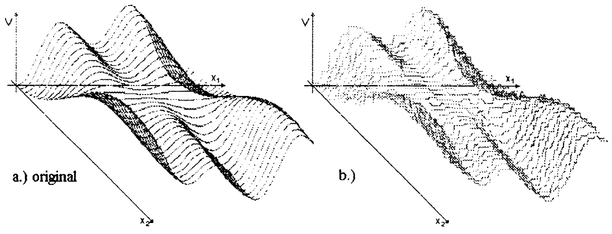


Fig. 2: A simple, non-monotonous function example (a) approximated by 1200 VQ-prototypes (b).

Besides, the time for learning the function rises proportional to the number of prototypes. In other words this procedure is not very efficient and we will see, that there are ways to reduce the number of prototypes significantly without increasing the error.

3 Interpolation

For conventional function approximators like several neural networks steady functions are used as a basis for an approximation. This seems plausible especially if these functions should have the generalizing aspect that similar input vectors should lead to similar output.
A way to get smooth approximated functions is to interpolate between several checkpoints. An easy to implement and fast approximation method is the following:
First we calculate the Euclidean distance D_i from the input vector $\bar{x}$ to each prototype-vector $\overline{xs_i}$.

$$D_i = (x_1, x_2, \dots, x_n) = \sqrt{\sum_{j=1}^{n} (x_j - xs_{ij})^2}$$

We transform and normalize the vectors of all distances so that

$$d_i = \frac{(e^{cD_i^2} - 1)^{-1}}{\sum_j (e^{cD_j^2} - 1)^{-1}} \qquad \forall D_i \neq 0, \tag{1}$$

($c > 0$ const.) and:

$$\sum_i d_i = 1 \tag{2}$$

Then we calculate the approximated function as follows:

$$F(x_1, x_2, \dots, x_n) = \sum_i f_i \cdot d_i \qquad D_i \neq 0 \forall i \tag{3}$$

$$F(x_1, x_2, \ldots, x_n) = f_i \qquad forD_i = 0$$

where f_i is the mean value of all learned values that were presented while prototype number i was the winning one. The function F is steady by definition, because with any input vector $(x_1, x_2, ..., x_n)$ and prototype i and the distance $D_i \longrightarrow 0$ there is:

$$F = \sum_i f_i \frac{(e^{cD_i^2}-1)^{-1}}{\sum_j (e^{cD_j^2}-1)^{-1}} \tag{4}$$

$$= \sum_i f_i \frac{(e^{cD_i^2}-1)^{-1}}{(e^{cD_i^2}-1)^{-1} + \sum_{j\neq i}(e^{cD_j^2}-1)^{-1}}$$

$$= \sum_i f_i \frac{1}{1+\sum_{j\neq i}\frac{e^{cD_i^2}-1}{e^{cD_j^2}-1}}$$

For $D_i \to 0$ we obtain $F = f_i$.

Let us have a look on the principles of this approximation method. Fig.3 shows a one dimensional example of an approximation between 4 prototypes: Fig.4 shows an approximation example with 300 randomly distributed prototypes. The function seems smooth, but the gradient of the approximated function mostly does not fit. It seems that the positions of the prototypes are not at all optimal. We notice, that the gradient in all locations of the prototypes is zero by definition. The function is not only steady but differentiable near the checkpoints. The fact that the gradient in any prototype is zero is a very useful effect for reinforcement tasks as we will show later on and leads us to another fundamental enhancement of our algorithm.

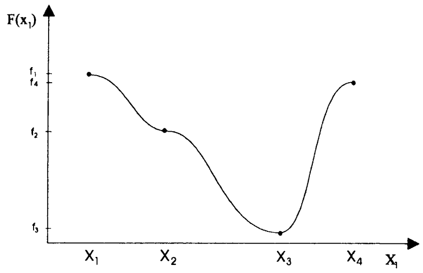


Fig.3: Scheme of the interpolation between 4 prototypes.

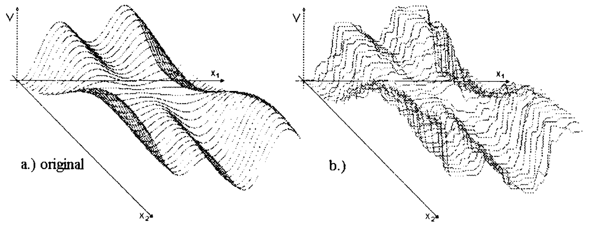


Fig.4: The same original function (s. Fig.2a) approximated by an interpolation between 300 prototypes

4 Minimum-Maximum Detection

If the gradient of a function is zero at certain points it indicates that there are maxima, minima or saddle points. In reinforcement learning maxima and minima of the value-function occur mostly in states of "high interest" respectively where direct external reward is available. These states are containing the only "sure" information about the environment and should therefore be represented perfectly by a serious value-function approximation method. With this information we can construct a promising learning rule for the positions of our interpolating prototypes. First we introduce two different kinds of prototypes: max-prototypes that should run into local maxima of the function and min-prototypes that run into local minima. We change the prototype selection rule to the following: we look for the nearest max-prototype and the nearest min-prototype. If the presented function value is higher than the value of the max-prototype or lower than the min-prototype's value, the corresponding prototype relaxes one step to the input vector:

Without noise on the input e.g. it is possible to set the corresponding prototype directly to the location and value of the "best" known input vector . Fig.5 shows an example of the approximation of the function also shown in Fig.2, but instead of 1200 used Prototypes we received a function approximated by only 10 prototypes. To achieve a result like the one shown above we started with 50 randomly distributed prototypes and reduced them in cases when two or more prototypes tried to run to the same extremum dependent on a predefined minimal allowed distance

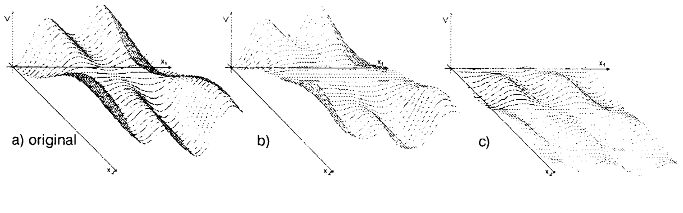
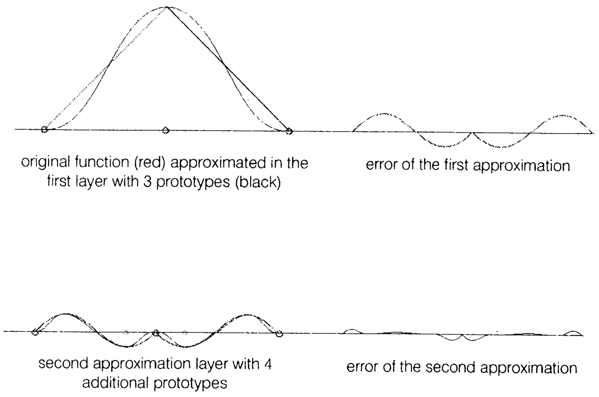


Fig. 5: Original function, approximated by 10 Min-Max-prototypes (b) and its residual error-function (c)

ε between them. It seems that we have found a very well performing approximation algorithm with suitable properties for real-world-reinforcement learning tasks: The number of prototypes is automatically minimized reducing storage capacity and calculation time. Finding minima and maxima of the value-function is one of the easiest and fastest ways to handle sporadically given external rewards witch additionally are represented absolute precisely. The interpolation between the prototypes is computed rapidly, smooth in any point and estimates reasonable (generalizing) predictions for unexplored states based on the experience of the agent. If the approximation of a given value-function should be accurate in every point in the state space there is still one problem left. With a fixed interpolation method the approximation-error between the extrema can generally not be driven down to zero in most cases (as shown in Fig.5c). So in part 5 we extend the algorithm to an approximation algorithm with the capability of an error convergence down to any predefined error-limit.

5 Multy-Layer-Error-Function Approximator

If the type of a possible value-function is unknown it is not very useful for real-world-agents to spend lots of efforts on complicated and slow interpolation methods. It is more reasonable to use a rather fast one that considers the few important demands of a RL-environment, as the one we introduced in the last chapter, and handle the remaining error in just the same way. All we know about this error function is, that it is (by construction)zero at the points, where the prototypes of the "first approximation layer" are located. Now, to approximate the error-function in a "second layer" we can use them as fixed zeros and only have to add new prototypes which like before move towards the minima and maxima. In this way the greatest errors of the first layer are handled perfectly in the second layer. To reduce the error even more, any numbers of layers can be added. Each layer approximates the error of its predecessor by using the prototypes of all former layers as zeros and add some more to find the remaining maxima and minima. The whole approximation can now be obtained easily by summing up all layers. It is of course advisable to run the layers on different time scales so e.g. the second layer waits until the first doesn't change significantly. This can be achieved easily by starting with a number of prototypes in the first layer and let the prototypes which should be removed fall into lower layers. So a fixed resource of prototypes is distributed automatically to several layers where they are needed most. Fig.7 shows an example of a two layered function approximation.


original function (red) approximated in the first layer with 3 prototypes (black)

error of the first approximation

second approximation layer with 4 additional prototypes

error of the second approximation

Fig. 6: This example of a triangle-function demonstrates how accurate such an unsteady function can be approximated by only two layers of aou algorithm.

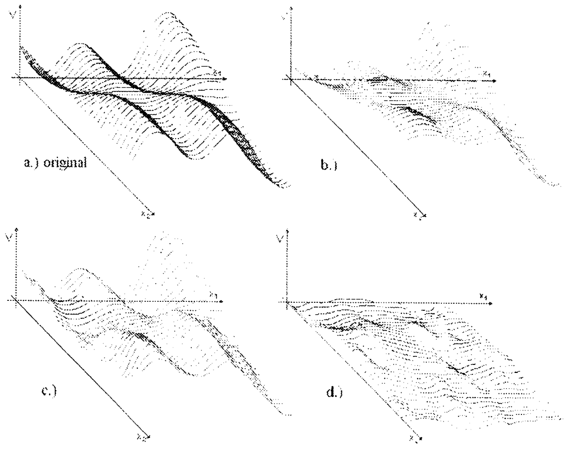


Fig. 7: Result: the original function (a) is approximated in the first layer by 7 prototypes. The combination of the first(b) and second layer approximation, with 7 + 20 prototypes results in a much better approximation (c) with a very low remaining error-function (d).

6 Robot Control Example

To demonstrate the flexibility of our concept, we implemented a modified version of the algorithm in a GMD-rover robot "Kurt II"[10]. The simple learning task was to explore the environment in order to find a way from the starting point (upper left) to the goal in the upper right corner Fig.8 avoiding the walls in between. The robots sensory input was reduced to the global coordinates of the robot (provided by wheel encoders), its orientation (internal compass) and 6 bumpers indicating collisions with a wall. The coordinates were not very accurate, but sufficient to solve the problem. Hitting a wall was punished by a negative reward, while reaching the goal was rewarded positively. In this implementation the concepts of the function approximator are changed a little to handle a realistic task where no value-function is explicitely given: We only use one layer to approximate the value function and insert new prototypes where unexpected external reward is given, because states of external reward define our extrema of the value function. The robots reinforcement agent receives the global two dimensional coordinates, where a learned value-function as a function of the robots position can be directly plotted. In Fig.8a we can see the labyrinth with a successful trial of the robot to get from the start to the goal. Fig.9 shows the learned negative value-function in dependence of its x-and y-coordinates. The robot-control now is simple. After very few exploration efforts the robot can follow the smooth slope of his value-function directly "down" to the goal extrapolating his gained experience.

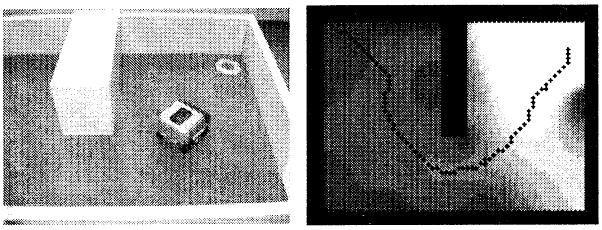


Fig.8: In these pictures we can see how KurtII found his way by building an internal state representation with positive(bright) and negative(dark) reward (right).

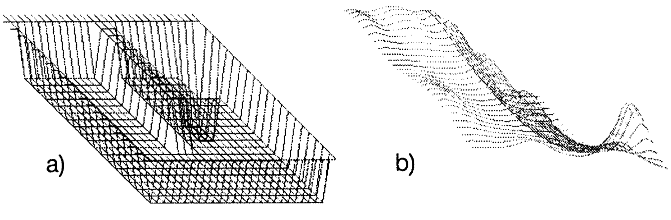


Fig.9: The left picture (a) shows the external reward signals given in the simple labyrinth. On the right we can see the approximated value-function (b). The negative function is plotted so it is easier to think of the robots virtual moves to find the ideal path as the moves of a marble rolling down towards the lowest point, the goal.

7 Conclusions

Our interpolation method allows us to focus precisely on interesting points in the state space. For the robot control system these interesting states are those where the value function has extreme-values (minima or maxima), corresponding to states which the robot should either avoid or head for. In RL these are the only points in the state space where an external reward signal leads to exact values of the value function. Hence, an internal approximation of the value-function should represent in particular these points with high accuracy, a goal which is met by our algorithm. This goal is neither achieved by a standard backpropagation (like TD-Gammon[11]) algorithm nor by an associative memory function approximator (like CMAC [4]). Both kinds of function approximators are often used in context with reinforcement learning. Backpropagation and associative memory function approximation are neural network constructions which can hardly be expanded, when the informational capacity is not sufficient to solve a reinforcement problem satisfactorily. An extensibility may be decisive, because a too large neural network in general has low generalizing capabilities, while a too small neural network may disregard important details. The presented approximator in contrast should guarantee convergence (not formally proved yet). It begins with more generalizing function approximation and reaches a predefined error limit without the problem of getting stuck in local minima, which backpropagation algorithms sometimes do. Because of normalizing the distance-functions our algorithm is independent of the scaling of the function to approximate. Backpropagation and associative memory networks in generally do have problems to approximate high gradients, because usually the output function of each neuron is predefined and static. A problem with our algorithm by now is, that learning time may rise, if many prototypes first move towards the highest maximum, then fall into the next approximation layer and move to a little lower maximum , then fall into the next approximation layer etc.. Consequently many layers are needed and the moving prototypes need too much time to get to their "own maximum or minimum". A solution to this problem will soon be presented (title of the paper: "Adaptive state space quantisation for reinforcement learning agents"), where we construct an algorithm with the capability of dividing prototypes to raise the informational capacity where it is needed most indicated by the variance of Q-values. In addition a hierarchical version with less calculation expense will be developed especially for the implementation in complex real world robots.

References

[1] Sutton R S, Barto A G, Reinforcement Learning: An Introduction, MIT Press Cambridge, Massachusetts, London, England, 1998

[2] Fischer J, Strategiebildung mit neuronalen Netzen, Diplomarbeit der Angewandten Physik, WWU-Muenster, 1999.

[3] Breithaupt R, Adaptive und kooperative Automaten in statischen und dynamischen Umgebungen, Diplomarbeit der Angewandten Physik, WWU-Muenster, 1999.

[4] Sutton R S, Santamaria C J, Ram A, Experiments with Reinforcement Learning Problems with Continuous state and Action Spaces, University of Massachusetts Amherst, (UM-CS-1996-088), 1996

[5] Kroese Ben J A, van Dam J W, Adaptive space quantisation for reinforcement learning of collision-free navigation, Faculty of mathematics and computer Science, University of Amsterdam Kruislan 403, NL-1098 SJ Amsterdam, 1992

[6] Samejima K, Omori T, Adaptive internal state space construction method for reinforcement learning of a real world agent, Faculty of Engeneering, Tokyo University of Agriculture and Technology, Nakachi 2 24-26 Koganei, Tokyo, in Neural Networks, Vol.12,No.7-8,p1143-1156,1999

[7] Okabe A, Boots B, Sugihara K, Chiu S N, Concepts and Applications of Voronoi Diagrams, (second edition) , Chichester: John Wiley, 2000

[8] Kohonen T, Self-Organization and Associative Memory - Springer Series in Information Sciences, Springer, Third edition, 1989

[9] Herz, J., Krogh, A., & Palmer, "Introduction to the Theory of Neural Computation." Addison-Wesley, 1991

[10] Hertzberg, J., Kirchner, F. "A prototype study of an autonomous robot platform for sewerage system maintainance", In: Autonomous Robots Journal, 4, pp. 319-331, Kluwer Academic Publishers, 1997

[11] Tesauro G, Temporal Difference Learning of Backgammon Strategy, Machine Learning, ed. by D. Sleeman and P. Edwards. San Mateo, Morgan Kaufmann Pub Inc., pp. 451-57., 1992

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Implementation of a Virtual Manufacturing Line for Agile Assembly

Ju Yong Choi*, Sang Min Cha*, Sok Ha Kim*, Hee Chang Lee**,
Jong Il Bae***, Man Hyung Lee****

* Graduate School of Mechanical Engineering, Pusan National University
** Department of Industrial Engineering, Yangsan College
*** Department of Electrical Engineering Pukyong National University
**** School of Mechanical Engineering Pusan National University
Changjun-dong, Kumjung-gu, Pusan, 609-735, Korea

E-mail : jychoi75@kr.qrio.com, mahlee@pusan.ac.kr

Abstract

A virtual manufacturing system is developed in order to accomplish rapid changeover from the assembly of one product to the assembly of another product. In the virtual system we propose a new database model which is based on the hierarchical and the relational model. Teaching and modification of tasks by off-line program enables the agility and the productivity to increase. This virtual system is available on designing a layout of a factory, making a time scheduling, and linking the factories by TCP/IP. The developed system is made by Visual C++ 6.0 of Microsoft and OpenGL of the Silicon Graphics.

Keywords : Agility, Virtual manufacturing, Off-line program, TCP/IP

1. Introduction

In recent industrial world to improve productivity factory automation through robot manipulators' adoption has been proceeding in overall industrial parts. Also as the globalization of the robot's utilization, the demand of the convenience of usage with its excellent performance is increasing. Furthermore the turbulent and changing environment has a great impact on all the manufacturing-related activities such as order, planning, shop floor control, and other management-related activities such as personnel, marketing, services, etc.

Agile manufacturing has been emerging as a new paradigm which incorporates high productivity and quality [1],[2]. In [3] Agile manufacturing is defined as the ability to accomplish rapid changeover between the manufacture line of different assemblies by using the essentially same manufacturing one which is consist of several robots and devices.

In this paper a virtual manufacturing system is developed to accomplish rapid changeover from the assembly of one product to the assembly of another product. We suggest a scheme that heightens the agility for the diversity of manufacturing models by using a hybrid database system which consists of an internal and an external database [4]. A static model or hierarchical model related with the dynamic and kinematical features of the components is represented by tree-typed lower components in the internal database system. And a relational model pursues dynamic connection with the data in the external database according to the modules which is constructed by user without predetermining connections between data [5]. The developed virtual manufacturing system performs the forward-inverse kinematical analysis, trajectory planning and its simulation for the robot manipulators. Also, Task time for each work-cell, delay time between modules, and speed of conveyor are monitored [6]. In the virtual manufacturing system, tasks taught and modified by off-line program are more agile and productive [7]. This virtual system is available on not only designing a layout of a factory, making a time scheduling but also linking the factories by TCP/IP. The developed system is made by Visual C++ 6.0 of Microsoft to supply graphic-interface environment and by OpenGL of the Silicon Graphics to embody 3D graphic environment.

This paper is organized as follows. A hybrid database system is proposed in section 2 and a virtual manufacturing line is presented in section 3. In section 4 TCP/IP is described and section 5 concludes.

2. Proposition of a Database System

To make an efficient virtual manufacturing process it is necessary to systematically construct graphic modeling data [8]. So as to the fact the proposed structure of database is designed taking a significant consideration into the composition of manufacturing processes that a user makes. When the database exchanges data of users interface as a users necessary work process, a virtual manufacturing line system should be actuated to be effective and standard. Because of the necessity of this system the hybrid database model adopting a hierarchical model having tree structure and a relational model to be proper for frequent change of data connections was developed in this study.

2.1 a hierarchical model

Generally a database system based on a hierarchical model can be

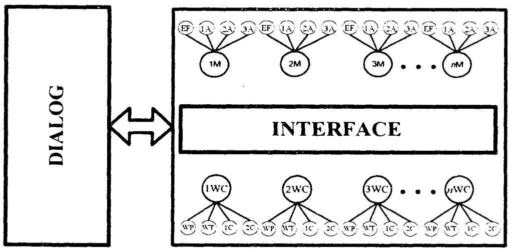


Fig. 1 the proposed database system

defined as a structure without ambiguity. In the aspect of data process technique, modeling data organization is efficient as the access path is determined. But this type is not proper to move along the tree structure that the accessible paths to all the manufacturing components are predetermined. Hierarchical model constrains this kind of connection between a upper component and another lower component that belongs to other upper components. The reason is that ambiguous relationships between manufacturing components cannot be represented by the direct hierarchical model. For example, the data of pallets position and task positions, etc have common relationships to be difficult to have clear relationships. To compensate this fault we adopted relational model that can construct a user-supportive program or harmoniously alternate modules.

2.2 a relational model

Ordinarily relational model unlikely to other models is not predetermined for the relationships between manufacturing modules and the lower data. This model is based on a two-dimensional table that explains relationships. The main feature of a relational model the determination of data connections and data accessible paths is not defined beforehand, and as a user alternates manufacturing modules, so to speak, as only dynamic state, the connection structure is determined. This model can be applied as a work process is frequently changed or many alternations for lack of experience are predicted

2.3 a hybrid database system

Therefore to meet the necessity of this database model we propose a structure of a concrete database system model as following. The developed database system based on the two models preferred above, as shown in Fig 1, arranges arbitrarily n

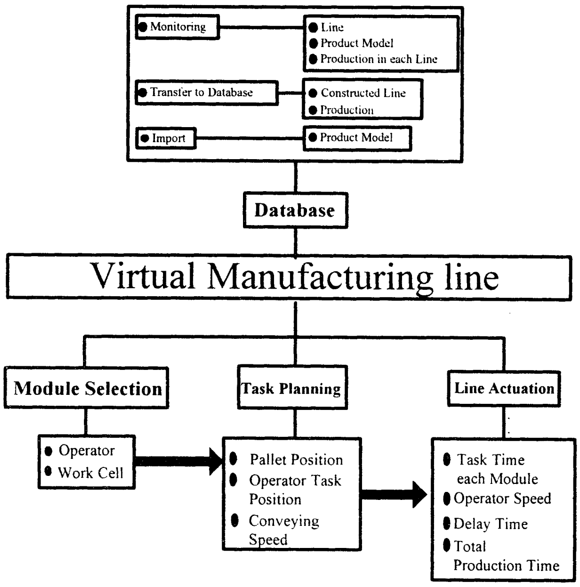


Fig. 2 the concept of virtual manufacturing line

operator and work cell modules as upper components of the data and have kinematical, dynamic, and various relating modeling components as their lower components. The lower modeling data of each operator module presented in Fig 1 are considered as kinematical features including joint angles of an operator and joint length and the dynamic features including the weight, moment of links, and the torque values of each joint, and the lower modeling data of each work cell module as features of the absolute coordinate position of a work cell, conveying speed, and the other peculiar cell features, etc. Also each upper module data and its lower modeling data adopts hierarchical model, a kind of tree structure. There are common data of each manufacturing component on which manufacturing modules is composed as a users intention based when relational model is adopted that each upper data, or an operator and a work cell are connected on the interface. These components are the absolute coordinates of cells, absolute coordinate positions of operator equipped with cells, relative coordinates between cell base and end-effectors of manipulators, working pallets position, task time each cell, conveying time from the edge of a cell to a work position, delay time for the difference of the task and conveying time of modules, the process schedule, and pallets position, and proceeding status in the whole line. Operators and work cells have common

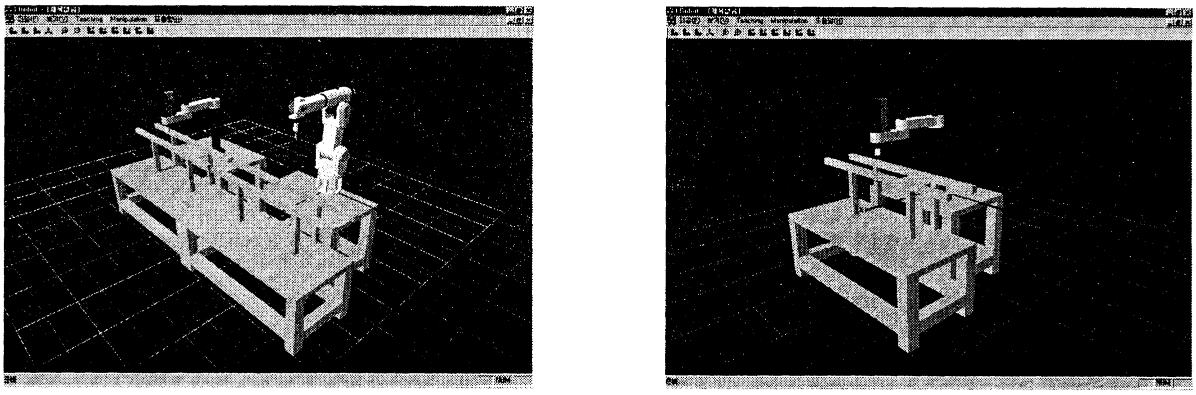


Fig. 3 A virtual construction of a module selected in the selection dialog

modeling components and are included as one module. As the number of the module, as we called, modules get a series of natural number of the data in the interface to share common data for each module. As this methodology the composition of database system shares the data of the interface to connect a dialog window that a user uses. After that, when a user actuates a manufacturing line in the virtual simulation, the dialog window exchanges the data with the composed database system.

3. Virtual Manufacturing Line by the hybrid database system

In this paper we develop a virtual manufacturing line on Window 98/NT environment of Microsoft, operating system using of the greater part of PC user. Window programs have a merit on making GUI environment that programmer can use without the expert knowledge about hardware. Also because of supporting integration environment to conveniently manage codes and resources, Visual C++ 6.0 makes tasks easy and has a strong point of the high efficiency of tasks because of supporting various developing methodology. Fig. 2 shows the construction of the developed virtual line which consists of a manufacturing components and the user module selection part where a operator module and a work cell module is selected to make tasks proper. Each task module constituted in this way is sequentially connected to make a virtual manufacturing line. Therefore as the palette position installation belonging to task planning, operator task position determination, and conveying speed installation, etc are inputted to a manufacturing line made by manufacturing modules constituted in the Module Selection, the line simulation is accomplished. And as the result, we can monitor the task time for each module, operator speed, the delay time caused by the difference of conveying speed, and the total processing time for one product to be produced by the process from the first module to the last one constructed line. We can select a unit module of a robot manipulator and a work cell. The dialog box for task planning makes a user to teach tasks for each process module as end-effect coordinate values of a robot, pallets position, and conveying speed to simulate the composed virtual simulate by a virtual manufacturing line.

As seen above, a virtual simulator program makes a user to select manufacturing components, operators and work cells in need of each module and upload them to main window. The left-side picture of Fig. 3 shows the shape uploaded on main window after selecting a work cell and a SCARA robot in the first module. On each unit module, selecting an operator and a work cell, a line construction can be proceeding in this way. The right-side picture of Fig. 3 presents the figure in the middle of composing a line from the modules paired with an operator and a work cell selected from the selection dialog. Fig. 4 is the figure to make up the whole manufacturing line to compose modules from the selected manufacturing components. It can input the information of manipulator task positions and conveying speed and pallet position coordinates on the basis of work cell coordinates, conveying time and conveying speed of a work cell from the selection dialog of task planning using of a virtual manufacturing line and virtually actuate the whole line. After the simulation we can monitor the time from the edge of a cell to task point, total working time of each module, and delay time caused by the different conveying time between modules, and the total time of coming out a part product through the whole line.

4. TCP/IP among Virtual Manufacturing Lines

At the Database in the menu, which place on a virtual program a user can access the data about products models in all the factories included as well as the other factories information the line composition, production in each line and so on. And a user uses this database through the menu to transfer its own line construction, production to the database system and to import the information that is the requested products model. Fig. 5 shows the concept picture of linkage among the factories involved in the production by TCP/IP so that a user can watch the status of the production in each factory and order a products model.

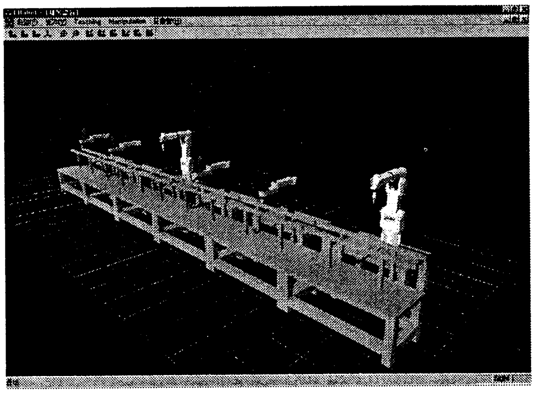


Fig. 4 A virtual construction of the whole manufacturing line

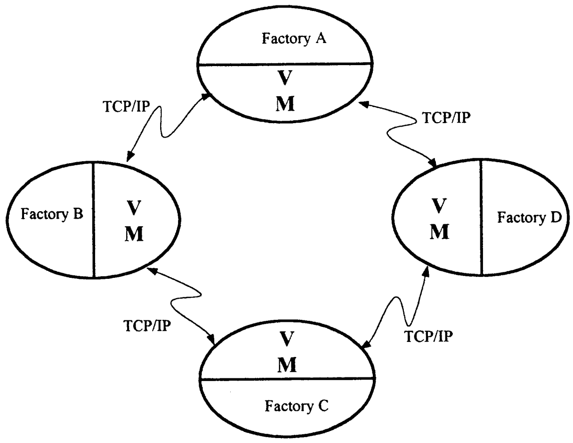


Fig. 5 A virtual manufacturing linked by TCP/IP

5. Conclusion

To cope with the reduction of products life-cycle as the variety of products along with the various demands of consumers, a virtual simulator was developed to make the changeover of manufacturing line efficient to embody a virtual simulation similar to a real manufacturing line. We developed a virtual simulator that can easily change manufacturing process as proper selection of manufacturing components that put into the line process. We realized a virtual manufacturing line using various kinds of operators and work cells and other perimetric equipments through the developed virtual simulator. It became known that one can change work cell and operator components for each unit module, and create a dialog window to manage separate work process for each module so that pallet's position, task position, and pallet speed of cells could be determined. To make efficient and standard flow of the manufacturing component and process data relational model was adopted dynamically connecting an operator and a work cell with the data in the interface by a user's intention by putting each manufacturing component, and the related modeling data as lower data to hierarchically connect them and putting common data in the interface so that the data of manufacturing process could effectively flow. And it is thought that this simulator made a user to properly select manufacturing components, operators and work cells to be necessary to a manufacturing line and to change the position installation of a pallet, the task position determination of an operator, and pallet's conveying speed for each module including task planning in the dialog box and actuate a virtual manufacturing line before the real production so that the alternation time of a manufacturing line could be reduced. And we developed the simulator that can embody a virtual manufacturing which can produced products by the way of distributed factories linked with each other by TCP/IP so that it can be shown as one alternative that the classical manufacturing methodology can be converted to the agile manufacturing system by adding the agility.

Acknowledgment

This paper is supported from G7 in Korea. Here I thank Professor ManHyung Lee for his advice.

References

[1] M. A. Uoussef, "agile Manufacturing: a necessary condition for competing in global markets," *Indust. Engng*, pp. 18-20, 1992.

[2] T. Kidd, *Agile Manufacturing: Forging New Frontiers*, Addison-Wesley, 1994.

[3] Roger D. Quinn, Greg C. Causey, and Frank L. Merat *et al* "Design of an Agile Manufacturing Workcell :for Light Mechanical Applications," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 858-863, 1996.

[4] Hyunbo Cho, and Moyoung Jung, "Enabling Technologies of Agile Manufacturing and its related activities in Korea," *Computers ind. Engng*, Vol. 30. No. 3, pp. 323-334. 1996.

[5] S. Kojima and H. Hashimoto, "3-D CAD Data Oriented Self-planning of Assembly Robot Cell Systems," *IEEE/ASME Int. Conf. On Advanced Intelligent Mechatronics*, September 19-23, pp. 484-489, 1999.

[6] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons, 1989.

[7] K. Son, M, C. Lee, J. M. Lee, and S. H. Han *et al*, "Real-time Evaluation of an Off-line Programming System for SCARA robot," *Proceedings of the 2nd ASCC*, no. 1, pp. 89-92, 1997.

[8] B. Scholz-Reiter, *CIM INTERFACES Concepts, standards and problems of interfaces in Computer-Integrated Manufacturing*, CHAPMAN & HALL, 1992.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Statistical Analysis of Subjective Evaluations of Mental Commit Robot

Takanori Shibata*1 Teruaki Mitsui*1 Kazuyoshi Wada*1 Li Yan*1
Akihiro Touda*2 Kazuo Tanie*1

*1 Bio-Robotics Division, Robotics Department
Mechanical Engineering Laboratory, AIST, MITI
*2 Sankyo Aluminum Industry Co.

Abstract: Recent advances in robotics have been applied to automation in industrial manufacturing, with the primary purpose of optimizing practical systems in terms of such objective measures as accuracy, speed, and cost. This paper describes research on mental commit robot that seeks a different direction that is not so rigidly dependent on such objective measures. The main goal of this research is to explore a new area in robotics, with an emphasis on human-robot interaction. In the previous research, we categorized robots into four categories in terms of appearance. Then, we introduced a cat robot and a seal robot, and evaluated them by interviewing many people. The results showed that physical interaction improved subjective evaluation. Moreover, a priori knowledge of a subject has much influence into subjective interpretation and evaluation of mental commit robot. In this paper, 785 subjects evaluated the seal robot and the results were analyzed by multivariate analysis.

1. Introduction

Machines are artifacts. Different from the aesthetic objects, machines have been designed and developed as tools for human beings while being evaluated in terms of objective measures [2]. Machines are passive because human beings give them goals. Machines will not be active as long as they are tools for human beings.

However, if a machine were able to generate its motivation and behave voluntarily, it would have much influence to an interacting human. At the same time, the machine would not be a simple tool for humans nor be evaluated only in terms of objective measures. Subjective evaluation is important. For a human, multi-modal stimulation should be influential. People interacting with the machine or observing the interaction may consider the machine as an artificial creature. Behaviors of the machine can be interpreted as emotional.

There are many studies on human-machine interaction. Here, we don't discuss studies on human factors in controlling machines used as tools. In some studies, machines recognize human gestures or emotions by sensory information, and then act or provide some information to the human. However, modeling gestures or emotions is very difficult because these depend on the situation, context and cultural background of each person.

Concerning action by a machine toward a human, an artificial creature in cyber space can give only visual and auditory information to a human. A machine with a physical body is more influential on human mind than a virtual creature.

Considerable research on autonomous robots has been carried out. Their purposes are various such as navigation, exploration and delivery in structured or unstructured environments while the robots adapt to the environments. In addition, some robots have been developed to show some emotional expressions by face or gestures. However, although such robots have physical bodies, most of them are not intended to interact physically with a human.

We have been building animal type robots as examples of artificial emotional creatures [1, 3-7]. The animal type robots have physical bodies and behave actively while generating motivations by themselves. They interact with human beings physically. When we engage physically with an animal type robot, it stimulates our affection. Then we have positive emotions such as happiness and love, or negative emotions such as anger and fear. Through physical interaction, we develop attachment to the animal type robot while evaluating it as intelligent or stupid by our subjective measures. In this research, animal type robots that give mental value to human beings are referred to as "mental commit robot"

The chapter 2 describes previous research and development of mental commit robot while explaining subjective interpretation and evaluation of robot through physical interaction. The chapter 3 describes the way of evaluation of a seal robot by 641 and the results of multivariate analysis. Then, we will show importance of physical interaction in subjective evaluation on the seal robot. Finally, the chapter 4 concludes this paper.

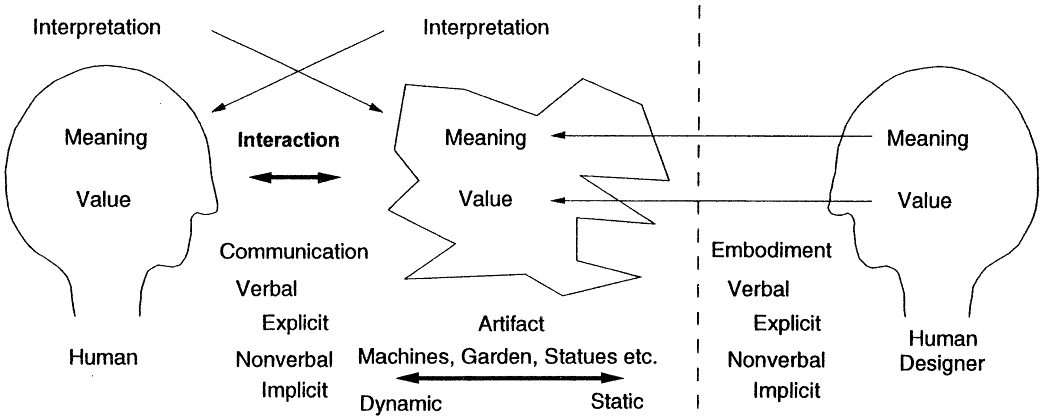


Fig. 1 Subjective Interpretation and Evaluation through Interaction

2. Previous Research on Mental Commit Robot

When a human interacts with a robot, he perceives it by his sense organs; vision, audition, touch, taste, olfactory, and so on. He interprets meaning of robot's behavior by depending on his senses and by using his memory and knowledge (Fig. 1). Depending on his subjective interpretation, he evaluates the robot.

In a case of a robot in computer graphics (simulation), a human perceives the robot by his vision and audition. Even though precise expression of the robot in computer graphics were presented to the subject, only two modalities (vision and audition) of a subject could be stimulated. In order to improve subjective evaluation, the number of modalities as well as quality should be increased.

As for real robot, it has a physical body. When a human interact with a robot physically, the human senses the robot in terms of multiple modalities. In the previous research, we investigated subjective interpretation of robot's behaviors by psychological experiments. In the experiment, a picture of a dog was equipped with a tail with one DOF, and subjects were asked to interpret emotions of the dog by watching wagging tail [4]. Then, a simple tactile sensor was added to the system and the tail wagged depending on stroking the tactile sensor by subjects. In the first experiment, subjects interpreted meaning of wagging by visual and auditory information. In the second one, subjects had tactile information in addition to vision and audition. As the results, the second experiment was much more impressive for most subjects because of physical interaction with tangibility. In addition, interpretations of emotions were various based on knowledge of dogs; for example, some had experience of owning dogs. Therefore, multiple modalities are important in human-robot interaction.

In addition, a priori-knowledge influences subjective interpretation. In the previous research, we categorized robots into four categories in terms of appearance:

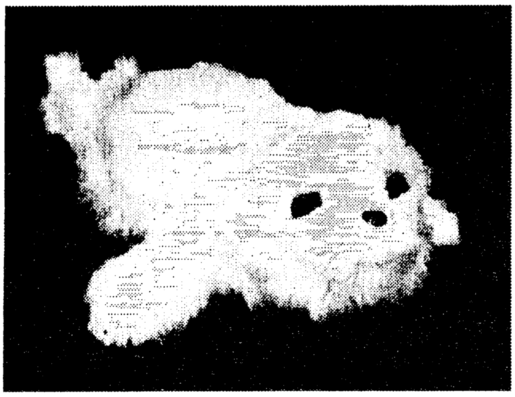
Category 1: Human
Category 2: Familiar Animal as Pet (pet animals: ex. cat and dog)
Category 3: Non-familiar Animal as Pet (ex. seal, penguin, and whole)
Category 4: New Character (artificially designed character: ex. AIBO [8], R100)

We have developed dog robot that belongs to category 4, seal robot that belongs to category 3, and cat robot that belongs to category 2.

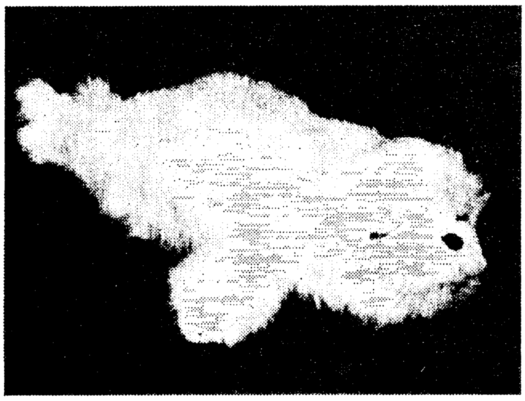
We have evaluated cat robot and seal robot by asking many subjects before and after having physical interaction with robots. Then, we compared the results of subjective evaluation of cat robot with those of seal robot. In the case of cat robot, most subjects out of 88 evaluated the cat robot as high value while increasing it after having physical interaction. However, some subjects had negative comments in terms of structure and reaction of the cat robot. They compared the cat robot with image of real cat in their mind.

On the other hand, in the case of seal robot, most subjects out of 40 didn't know details of real seal nor have experience of physical interaction with real seal. In the subjective evaluation, most subjects had high value on the seal robot both before and after physical interaction with it.

In this paper, we describe further evaluation on seal robot by more number of subjects (785 people). The next chapter describes the way of evaluation and statistical analysis of the results.



(a) Active



(b) Winking Eyes

Fig. 2 Seal Robot Version 3

3. Subjective Evaluations of Mental Commit Robot and Statistical Analysis

3.1 Specifications of Seal Robot

Its appearance is from a baby of harp seal that has white fur for three weeks from its born. In cooperation with Sankyo Aluminum Industry, we built seal robot version 3. As for perception, seal robot has tactile, vision, audition, and posture sensors beneath its soft white artificial fur. For tactile sensors, ten balloon sensors were applied. As for action, it has six actuators; one for eyelid, two for neck, one for each front fin, and one for two rear fins. Weight of seal robot was 3.4 [kg].

The robot has a behavior generation system that consists of hierarchical two layers of processes: proactive and reactive processes. These two layers generate three kinds of behaviors: proactive, reactive, and physiological behaviors.

3.2 Ways of Evaluations

There was an exhibition in Tokyo, Japan, from July 21 to Aug. 6, 2000. The total number of participants to the exhibition was 1,124,728. We had a booth to represent mental commit robots (seal and cat robots). Master of ceremony explained what is mental commit robot, aim of the robot, functions of the seal robot, and so on to people about five min., and then we asked them to interact with a seal robot. After having interaction about 10 min., we asked the people to answer questions as subjective evaluation. Through the exhibition, we collected answers by 785 subjects. Their age and sex are shown in Fig. 3.

There were 15 questions concerning robots as semantic differential (SD) method. All the questions were explanatory variables. In order to analyze answers, principal component analysis and factor analysis as multivariate analysis were applied. As the principal component analysis had better cumulative contribution than the factor analysis, this paper adopted the former. Results by 641 subjects out of 785 were used in the analysis because they had complete answers to 15 questions and other didn't.

3.2 Results of Subjective Evaluation

As the result of principal component analysis, two factors which eiganvalues were more than 1 were extracted. After varimax rotation operation with Kaiser normalization, two factors characterized factor loading shown in Fig. 4 were obtained.

The first factor is characterized by "natural," "texture of skin," "comfortable by touching," and "calm by touching." Its contribution is 33.4%. The second factor is characterized by "lovely," "want to touch," and "want to talk." Its contribution is 28.3%.

The first factor relates to "contact with a robot," and the second factor does to "view to a robot." This means that physical interaction have the most influence in subjective evaluation and the appearance have the second most influence. Therefore, when we design a robot that interacts with a human physically, we should consider those factors.

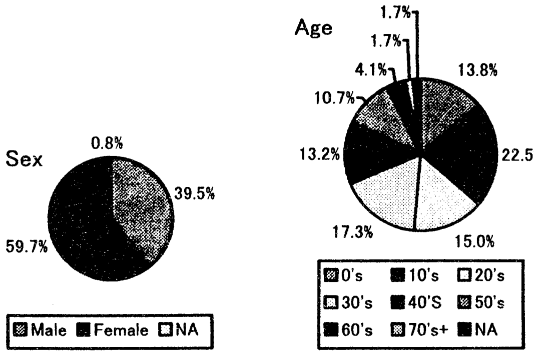


Fig. 3 Sex and Age of Subjects

Table 1 Question and Factor Loadings for Two Factors

Question	Factor1	Factor2
lovely	0.328	0.801
want to touch	0.172	0.856
want to talk	0.283	0.752
lively	0.561	0.377
friendly	0.505	0.586
affective	0.630	0.451
natural	0.799	0.193
texture of skin	0.769	0.201
fun by touching	0.630	0.557
comfortable by touching	0.782	0.348
calm by touching	0.766	0.325
like	0.541	0.659
necessary	0.472	0.421
want to own	0.589	0.498
want to present	0.373	0.369
contribution(%)	33.399	28.263

4. Conclusions

This paper analyzed evaluation of a seal robot by 641 subjects. The results show that physical interaction/contact had the most influence in subjective evaluation and the appearance of the robot had the second most influence.

We will analyze the results more while considering scores of evaluation.

References

[1] T. Shibata, et al., Emotional Robot for Intelligent System - Artificial Emotional Creature Project, Proc. of 5th IEEE Int'l Workshop on ROMAN, pp. 466-471 (1996)

[2] H. Petroski, Invention by Design, Harvard University Press (1996)

[3] T. Shibata and R. Irie, Artificial Emotional Creature for Human-Robot Interaction - A New Direction for Intelligent System, Proc. of the IEEE/ASME Int'l Conf. on AIM'97 (Jun. 1997) paper number 47 and 6 pages in CD-ROM Proc.

[4] T. Shibata, et al., Artificial Emotional Creature for Human-Machine Interaction, Proc. of the IEEE Int'l Conf. on SMC, pp. 2269-2274 (1997)

[5] T. Tashima, S. Saito, M. Osumi, T. Kudo and T. Shibata, Interactive Pet Robot with Emotion Model, Proc. of the 16th Annual Conf. of the RSJ, Vol. 1, pp. 11, 12 (1998)

[6] T. Shibata, T. Tashima, and K. Tanie, Emergence of Emotional Behavior through Physical Interaction between Human and Robot, Procs. of the 1999 IEEE Int'l Conf. on Robotics and Automation (1999)

[7] T. Shibata, T. Tashima, K. Tanie, Subjective Interpretation of Emotional Behavior through Physical Interaction between Human and Robot, Procs. of Systems, Man, and Cybernetics, pp. 1024-1029 (1999)

[8] M. Fujita and K. Kageyama, An Open Architecture for Robot Entertainment, Proc. of Agent'97 (1997)

[9] T. Shibata, K. Tanie, Influence of A Priori Knowledge in Subjective Interpretation and Evaluation by Short-Term Interaction with Mental Commit Robot, Proc. of the IEEE Int'l Conf. On Intelligent Robot and Systems (2000)

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

A Framework of Deliberative Decision Making in "Conscious" Software Agents

○ Ravikumar Kondadadi
Department of Mathematical Sciences
University Of Memphis.
Memphis, TN 38152,USA
001-901-678-2320
kondadir@msci.memphis.edu

Stan Franklin
The Institute for Intelligent Systems
University Of Memphis.
Memphis, TN 38152,USA
001-901-678-3142
stan.franklin@memphis.edu

ABSTRACT:

When we humans are faced with a problem to solve, we try to create in our mind different strategies or possible solutions. We imagine the effects of executing each strategy or trial solution. Eventually, we decide upon one strategy or trial solution and try solving the problem using it. In 1890 William James proposed a model that describes this voluntary decision-making calling it the ideo-motor theory. According to this theory at instance of time our mind is a seat of many ideas related to each other either favorably or antagonistically. Whenever an idea prompts an action, the antagonistic ideas may object it and try to block that action, while the favorable ideas may support it trying to help the action to take place. While this conflict is going on among several ideas we are said to be "deliberating". Software agents, so equipped, should be able to make voluntary decisions like humans. This paper describes a computational implementation for this deliberation process in a software agent, including James' ideo-motor theory of voluntary action.

Keywords:

Deliberation, voluntary action, Scenario creation, ideo-motor theory, IDA, "conscious" software agents

1. INTRODUCTION

Deliberation is the process of debating with oneself concerning alternative courses of action. Humans do deliberate and make decisions by deliberation. Humans often think about the pros and cons of several courses of action before taking a decision. We think about various alternatives for solving a problem and finally decide on one alternative. For example when we are hungry, we may think of several alternatives, like going to a Japanese restaurant, a Mexican restaurant or a Chinese restaurant. A fondness for Sushi may finally push us to choose the Japanese restaurant.

The sequence of steps leading to a possible solution is called a *scenario.* Building a scenario and evaluating it to see if it is really a feasible solution is a significant part of the deliberation process. If a scenario is successful we consider the course of action (idea) it represents as a possible alternative for solving the current problem. These ideas compete with each other in the decision-making battle. We finally take the action suggested by the idea that won the battle. That action is called a "voluntary action" because it occurred as a result of the deliberation process.

One of the main goals of Artificial Intelligence is to produce intelligent software that can think and act like humans. One step in this direction is to produce intelligent software capable of deliberating about alternative courses of action for solving a problem, and of choosing a good one.

An autonomous agent is a system situated within, and a part of, an environment. The system senses that environment, and acts on it, over time, in pursuit of its own agenda. It acts so as to possibly effect what it senses in the future [7]. In this paper we describe an implementation of the deliberation process in an autonomous software agent called IDA (Intelligent Distribution Agent).

The rest of the paper is organized as follows: In Section 2 we describe the autonomous software agent IDA. In Section 3 we describe the process of scenario creation. Section 4 is devoted to ideo-motor theory and voluntary action. Section 5 contains conclusions drawn.

2. IDA

IDA is a "conscious" software agent (to be defined below) being designed and implemented by the "Conscious" Software Research Group at the University of Memphis. It's being developed for the US Navy. Each year thousands of Navy personnel are assigned new jobs at a cost of some $600 million dollars in moving expenses. This process of directing the movement of individuals to fill vacancies in ships, submarines etc. are called distribution [8]. The Navy employs people called *detailers* to perform this distribution task. Detailers try to keep both the sailors and Navy satisfied by keeping track of sailors' preferences while conforming to Navy policies. IDA is designed to completely automate the job of a detailer. As IDA's main goal is to be positioned as an alternative to a detailer, it must be able to communicate with sailors in natural language, access personnel and job databases, calculate the fitness of each job with respect to sailor preferences and Navy policies, deliberate about the temporal possibility of a job, select one or two jobs, and compose and send an email offering the selected jobs.

Like a human detailer, IDA must deliberate about the jobs to offer to the sailor. This deliberation should be done "consciously." "Conscious" software agents [4,5,8] are cognitive agents that implement global workspace theory, a psychological theory of consciousness [1,2]. Global workspace theory postulates that a mind is composed of small special processes, which are usually unconscious. It has two main constructs; this set of distributed unconscious processes and a global workspace or blackboard.

Baars used the analogy of a collection of experts, seated in an auditorium who can solve different problems. We do not know who can solve which problem. The global workspace strategy is to make the problem available to everyone in the auditorium by putting it on the blackboard. Then the expert who is expert on this particular problem can identify the problem and solve it. One of the main functions of consciousness is to recruit resources to deal with a novel or problematic situation. When a novel situation occurs, an unconscious process broadcasts that to all other processes in the system by trying to put it on the blackboard, which causes it to become conscious. Only one problem can be on the blackboard at any time. So Baars theory explains why consciousness is a serial system.

We call each of these unconscious processes *codelets*. A codelet is a small piece of code capable of performing some basic task (analogous to an expert in the above analogy). An attention codelet is a kind of codelet whose task is to push some information into "consciousness".

At any time in the mind different processes (attention codelets) could be competing with each other, each trying to bring its information to "consciousness". Processes may form coalitions depending on the associations between them. The activation of a process measures the likelihood of it becoming "conscious". The coalition with highest average activation finds its way into "consciousness."

Like we humans, voluntary decision-making in "conscious" software agents is done via deliberation. Different attention codelets, each representing a possible solution, compete with each other to come into "consciousness." That is, they compete with each other to be offered as a solution to the current problem.

3. SCENARIO CREATION

During deliberation we think about different possible solutions to a problem. These solutions take the form of scenarios, interpreted broadly. A familiar example for scenario creation in humans might be creating scenarios of different routes to a destination. When we think about different routes from one place to another, images of scenes along those different routes will be formed in mind. We start from the source location and try to build the travel route to the destination step by step. After building a scenario we evaluate the scenario to see if it can be a possible solution. If the route fails to reach destination we discard the scenario and starts a new one. Finally we select the best route among the successful routes. The example above mentioned is an example of a spatial scenario. There are different types of scenarios that we generally create; temporal scenarios, spatial scenarios, temporal and spatial scenarios and causal scenarios.

Scenario creation in IDA follows Barsalou's perceptual symbol systems [3,6]. Scenarios are composed of scenes. Each scene depicts a particular place, object or event in whole scenario.

IDA starts deliberation with the job with highest fitness. This fitness value depends on many factors like sailor's preferences, Navy policies, sailor's qualifications etc. While deliberating about the jobs to offer to the sailor, IDA creates temporal scenarios about each job. Before offering a job to the sailor, IDA checks to see if the estimated date of departure from the current job falls before the estimated date of arrival for the next job. IDA builds scenarios for each job by selecting a detach month for the current job, and then adding leave time, travel time and proceed time (extra time for a shore to sea move). Scenarios are constructed and stored as frames in IDA's working memory. Figure 1 shows a typical temporal scenario in IDA. In case if a job needs training IDA has to check if the sailor can take the training by adding leave and travel time to detach month and checks if the resultant date falls before the training start date.

Gap is defined as the difference between the resultant date after adding leave time, travel time and proceed time to the detach month of the current job, and the estimated arrival date for the next job. A scenario in IDA is considered to be successful if the gap is acceptable. Otherwise the scenario is adjusted to have an acceptable gap. The detach month of the current job can be adjusted within the PRD (Projected rotation date) window which is 3 months early and 2 months late. If the gap is still unacceptable even after adjusting the detach month within the PRD window, the job will be discarded, and IDA will start creating a scenario for the job with the next highest fitness in the list.

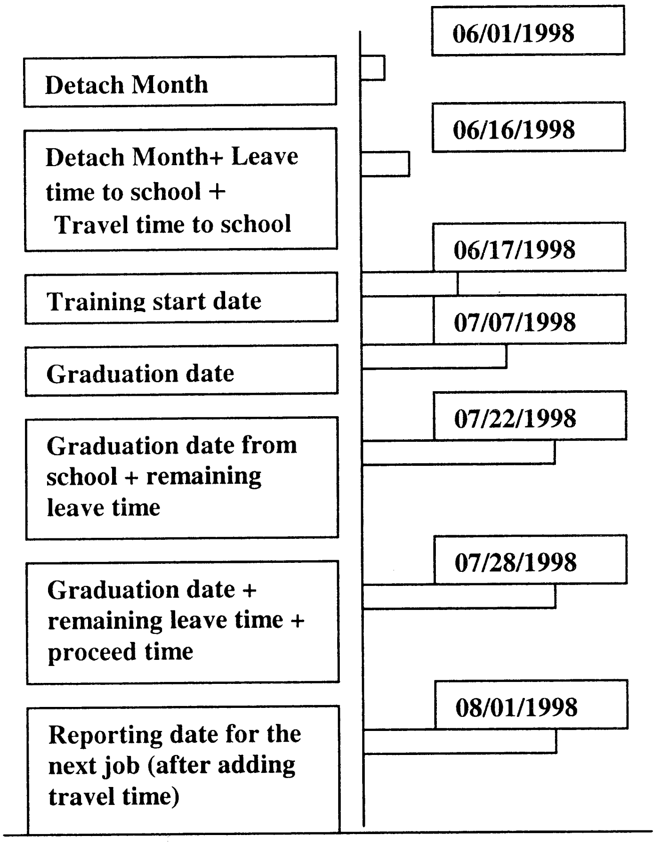

Detach Month
Detach Month+ Leave time to school + Travel time to school
Training start date
Graduation date
Graduation date from school + remaining leave time
Graduation date + remaining leave time + proceed time
Reporting date for the next job (after adding travel time)
06/01/1998
06/16/1998
06/17/1998
07/07/1998
07/22/1998
07/28/1998
08/01/1998

Figure 1: A typical scenario in IDA

4. VOLUNTARY ACTION

William James proposed ideo-motor theory [9] to explain the phenomenon of voluntary action in humans. Whenever a person thinks of an action, an idea of it will be formed immediately in his mind. If the action follows immediately the notion of it in the mind, we have ideo-motor action. The person is aware of nothing between conception and execution of the act. According to ideo-motor theory, an act will emerge successfully if the conscious image of that action can exist for sometime without competing images or intentions. If there are no antagonistic intentions, the action occurs spontaneously and without any inner-struggle. If there are some antagonistic ideas against an idea then these different ideas or images compete with each other, each becoming conscious in some order. In this case, the person is said to be in a state of "indecision". This state could last for hours, days or even months. At some point of time, some idea will be lucky enough to have no strong opponents and will immediately produce the appropriate motor effect. James called this act of metal consent to movement "fiat". There is no need to explain this. We all know about this and have experienced it.

When IDA deliberates about a job, she thinks of different attributes of the job, such as the moving costs, the job's priority, its fitness and the gap. Each of these attributes may have a corresponding idea in IDA's subconscious, which, if the job is a good one from that particular attribute's perspective, tries to push it into "consciousness". These ideas are implemented by attention codelets representing each of these attributes that are looking at scenarios in working memory in order to propose jobs.

Whenever scenario creation is done for a job, the attention codelets looking at this scenario may propose or object the job depending on the particular attribute's value about which that particular attention codelet is concerned. The strength of a proposal or objection depends on the activation of the codelet, which proposed or raised the objection. That activation, in turn, depends on the value of the attribute or, perhaps, of several attributes. In our current implementation we have five different activation levels for each attention codelet; propose with very high activation, propose with less activation, do nothing, object with high activation and object with very high activation.

If an attention codelet proposes a job with enough activation, it is able to bring the job into "consciousness". In response to this proposal, other codelets may favor the job, may object it or may propose a different job. If an attention codelet does not like the proposed job, it may look at the previous scenarios created and may propose a different job, if it can find one it likes better. If it cannot find any suitable job to propose, it just raises an objection against the job currently being considered.

If a codelet likes the proposed job, it may send some activation to the codelet that proposed the job so that the proposed codelet has higher activation. Every time a codelet comes to "consciousness", it looses some of its activation. Also the activation of the attention codelets decay constantly. After some point of time they completely loose their activation and die.

According to James, every idea awakens some actual movement (motor effect) to some degree and awakens it in a maximum degree when it is not kept from doing so by other ideas. An idea should remain in "consciousness'" for some time to be selected as the winner in the deliberation process. To keep track of the time a job is in "consciousness", we have implemented a codelet called the *timekeeper* in IDA. Whenever an attention codelet proposes a job, the timekeeper starts its clock. If there are no objections for the job for a certain amount of time, timekeeper marks that job as "to be offered to the sailor". The amount of time, λ, the timekeeper should wait before marking a job depends on the number of codelets currently running in the system and should be determined via tuning. While IDA is in a state of "indecision" the timekeeper loses patience. Each time another idea becomes "conscious" λ is reset, but reset to a lower value than the default value at which it started. If the timekeeper's patience wears out, it sends activation to the attention codelet that initiates scenario construction. Thus IDA is likely to continue with scenario creation for the next job in the list. After selecting one job, the whole process is repeated and IDA may select a second job. The deliberation generally terminates after IDA finds 2-3 jobs for the sailor. If IDA cannot find any good jobs to offer to the sailor, she'll ask the sailor to wait until the next requisition list of jobs appears.

Our current implementation has four different attention codelets for such voluntary decision-making, one each for moving costs, priority, gap and fitness. Each acts to propose jobs it likes and to oppose those it doesn't. Whenever a scenario is successfully created, these codelets look at the corresponding attributes of the job in the working memory and may try to propose or object the job with some degree of activation. The activation depends on the value of the attribute. If the activation of a proposal or objection is high enough, it may come to "consciousness" and be broadcast. For example, assume that the moving costs and the priority of the current job is very low. The attention codelet concerned with moving cost may like the job and get its proposal into "consciousness." Since the job has a very low priority the attention codelet concerned mostly with priority may search among the jobs with completed scenarios for job with higher priority. If it finds one, it may propose that job by bringing it to "consciousness." So there is a conflict here between the moving cost codelet and the priority codelet. Each may continue to propose its choice, but it does so with less activation each time. The conflict may continue until one of them wins because the other hasn't enough activation to make it into "consciousness." In this case, the winner still has enough activation to get into "consciousness" and to remain there without any objections for sufficient amount of time. Or the timekeeper may loose patience and the scenario creator may get enough activation to start creating a scenario for the next job.

5. CONCLUSIONS

This paper describes a conceptual framework for deliberative decision-making in humans, and its computational implementation in a "conscious" software agent, IDA. Due to the limitation of space we cannot show the results of experiments. Our preliminary results are very encouraging. Our approach works in a way reminiscent of what each of us has experienced in our own human decision-making. To the best of our knowledge, this is the first implementation of James' "ideo-motor" theory. Our current implementation is very specific to IDA. Our future work involves building a generic toolkit for the deliberation process in "conscious" software agents.

6. ACKNOWLEDGEMENTS

This research was supported by ONR grant N00014-98-1-0332. It was performed with essential contributions from the Conscious Software Research Group including Art Graesser, Sri Satish Ambati, Ashraf Anwar, Myles Bogner, Arpad Kelemen, Irina Makkaveeva, Lee McCauley, Aregahegn Negatu, and Uma Ramamurthy. The authors would like to particularly thank Lee McCauley for his contribution to some of the ideas presented in this paper.

7. REFERENCES

[1] B.J. Baars, A Cognitive Theory of Consciousness (Cambridge University Press, Cambridge, 1988).

[2] B.J. Baars, In the Theater of Consciousness (Oxford University Press, Oxford, 1997).

[3] L.W. Barsalou, Perception Symbol Systems, Behavioral and Brain Sciences 22 (1999) 577.

[4] M. Bogner, Realizing "consciousness" in software agents (Ph.D. Dissertation, University of Memphis, Memphis, TN, USA, 1999).

[5] M. Bogner, U. Ramamurthy, S. Franklin, "Consciousness" and conceptual learning in a socially situated agent, Human Cognition and Social Agent Technology, ed. K. Dautenhahn (John Benjamins, Amsterdam, 2000) p. 113.

[6] S. Franklin, Deliberation and Voluntary Action in "Conscious" Software Agents, Neural Network World 10 (2000) 505.

[7] S. Franklin, A.C. Graesser, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Intelligent Agents III (Springer Verlag, Berlin, 1997).

[8] S. Franklin, A. Kelemen, L. McCauley, IDA: A Cognitive Agent Architecture, IEEE Conf on Systems, Man and Cybernetics (IEEE Press, 1998) p. 2646.

[9] W. James, The Principles of Psychology (Harvard University Press, Cambridge, MA, 1890).

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Object Recognition Using the Stereo Vision for Underwater Robots

○Hidekazu Tanaka, Etsuro Shimizu, and Masanori Ito
Dept. of Electronic and Mechanical Engineering
Tokyo University of Mercantile Marine
2-1-6 Etchujima, Koto-ku, Tokyo 135-8533, Japan
htanaka@ipc.tosho-u.ac.jp

Abstract

In order to develop the robot eye system, the measurement and the recognition of three-dimensional (3-D) objects are one of important problems. The aim of this research is to recognize the object in the water with the information which can get from the stereo vision system. Since influence of refraction of light is large in the water, a large error occurs in the measurement of the object in the water. The error arises only in the distance from the camera to the object. Therefore the distance to the object is estimated as a linear by the least squares method from the actual distance of the object and the result of measurement in the water. The object is recognized by comparing the features of each object extracted from the image with the features of the registered object. Then, the position of the object is decided. In order to verify the accuracy of the object recognition, the experiments using the underwater robot is made.

Key Words : Object Recognition, Stereo Vision, Underwater Robots

1. Introduction

A man gets about 80-90% information of the outside world from the vision. Therefore the vision is most important in sensing functions. Since the ocean, the universe and so on are a dangerous environment to work for human, robots which can work in those environment instead of human are expected. Those robots should be equipped functions such as the object recognition and the collision avoidance autonomously. It is considered that the vision system is most suitable system to equip those functions. In order to develop the vision system for the robots, the measurement and the recognition of 3-D objects are one of important problems.

There are two methods to recognize the 3-D object. One is the method using the two-dimensional feature which extracted from the image. The other is the method which is compared 3-D model to the 3-D information of the scene[1][2]. There are also two methods to get the 3-D information of the object. One is the active method which irradiates the light wave, radio wave, sound wave on the object. The other is the passive method based on an image which is obtained by the camera. The stereo vision is the representative of the passive method which based on the triangulation, and easy to understand. The information about the position and the shape of object directly. Since it is difficult to use the light and the radio wave in the water[3], it is considered that the stereo vision is effective method as the vision system of the underwater robot. The stereo vision system using this research consists of a three-camera module and a software system that performs range measurement, and transforms all pixels of the input image into the 3-D coordinate value.

The aim of this research is to recognize the object in the water with the 3-D data which can get from the stereo vision system. Since the influence of refraction of light is large in the water, and it becomes the cause of large error in the measurement of the object. In this research, we examined about the characteristics of an error under the stationary flow. The object is recognized with comparing the features of each object extracted from the image with the features of the registered object. In order to detect the feature of each object from the image, an algorithm is construct. The experiments using the underwater robot is made to verify the accuracy of the object recognition.

2. Stereo Vision System

The stereo vision system used in this research is consisted of a three-camera module and a software system that performs range measurement. The three-camera module is shown in Fig.1. The principle of this stereo vision system is shown in Fig.2. Three cameras are arranged to take two stereo pairs. These pairs are a top-and-right camera and a right-and-left camera. In the stereo vision with a right and left camera,

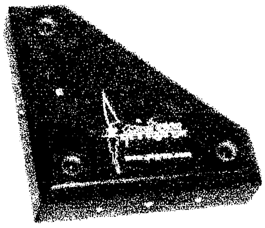


Fig.1 The three-camera module

the disparity of X direction is brought corresponding to the distance from camera to the object.

The algorithm to calculate the distance is as follows.

(1) The partial image of the image of Right camera is taken out.

(2) The partial image is matched on the images of Left camera.

(3) A position on the image of Left camera which is most similar to the partial image is looked for.

(4) A disparity at that time is assigned to a partial image.

Even a top-and-right camera does a similar calculation. The disparity is Y direction in stereo vision with a top-and-right camera.

An occlusion occurs in the stereo vision, and there is a territory which does not look common with either of images. Many occlusions occur on the edge in the vertical direction with a right-and-left camera.

However, the three-camera module has two baselines. These are the vertical direction and the horizontal direction. Even if occlusion occurs in the base line of one side, it may be measured by the other side. The stereo vision system is reducing the points which can not measurement [4].

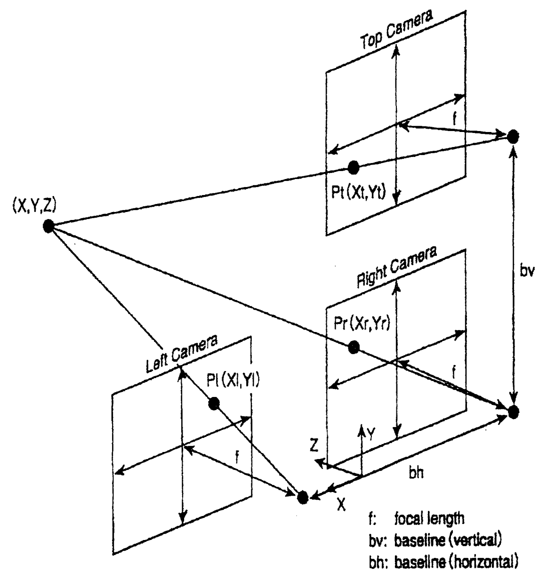


Fig.2 The principle of stereo vision system

3. Positioning of the Object

The influence of the background is large for the image processing. Therefore a background is covered with black curtain to reduce the noise of the image. A contrast with the object and the background becomes clear by covering a background with black curtain. The corresponding point can be decided easily in matching between the camera images of the right-and-left camera and the top-and-right camera.

3.1. The Rectification of an Error

The influence of refraction of light is large in the water. Therefore a large error occurs in the measurement of the object. The error arises only in the distance from the camera to the object.

A result in consideration of the refractive index of the water is shown in Fig.3. A rectification error was large.

A distance to the object is estimated as a linear function by the least-squares method from the distance of the actual object and the result of measurement of the distance in the water. A result of a rectification by the least-squares method is shown in Fig.4.

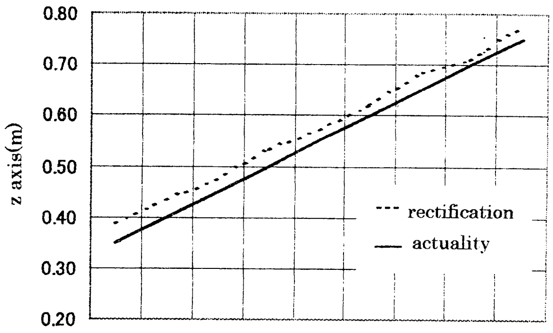


Fig.3 A result of the refractive index in the water

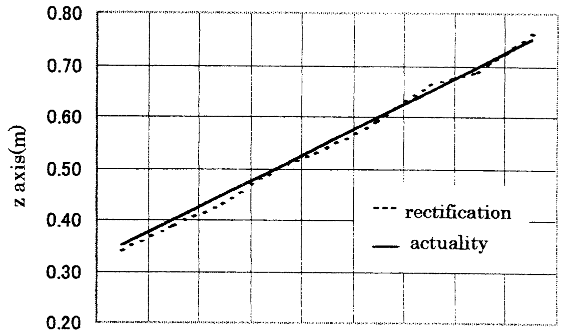


Fig.4 A result of a rectification by the least-squares method

3.2. The detection of the object

The flow of processing to detect the object is shown in Fig.5. The feature of target object is registered in advance. The target object is shown in Fig.6.

(1) The image is acquired by using the stereo vision system, and all pixels of the image are transformed into the 3-D coordinate value. The right image is saved from ,the right camera. The right image is shown in Fig.7.

(2) The image which obtained by the right camera is made binarization.

(3) It is checked whether a pixel is connected, and divided into each object.

(4) The features of each object are calculated. The

features are the area and the circumference.

(5) The features of each object are compared with the features of the registered object. The most corresponding object is extracted. A result of the recognition is shown in Fig.8.

(6) The position coordinate of the detected object is decided with the 3-D data which can get from the stereo vision system.

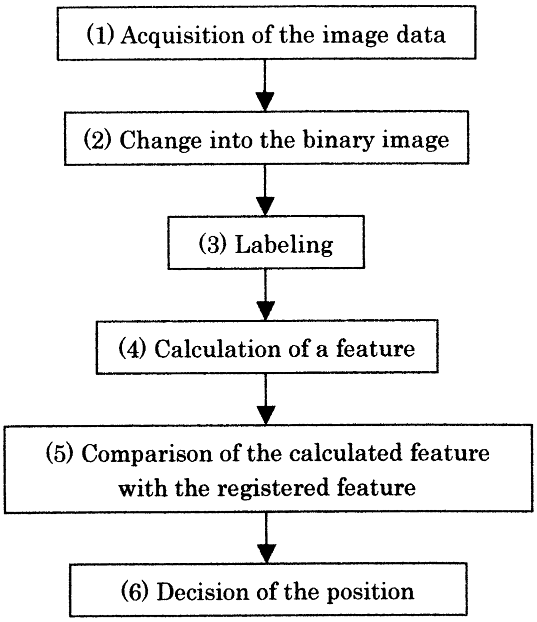

(1) Acquisition of the image data
(2) Change into the binary image
(3) Labeling
(4) Calculation of a feature
(5) Comparison of the calculated feature with the registered feature
(6) Decision of the position

Fig.5 The flow of processing in order to detect the object

4. Experiment

In order to verify the accuracy of the object recognition, the experiments that the underwater robots catch the object was made.

4.1. Underwater robots

In this research, the underwater robots that the dual robot arm and a hand are moved based on the visual information and made to do the work of the purpose.

4.2. Result of experiment

First, an experiment was made under the atmosphere environment. A robot hand could take the object in the high probability as that result. Therefore, as for the measurement and the recognition of the object under the atmosphere environment, it verified that reliability was high. Next, the experiment was made in the water which was the purpose of this research. The probability of success of the experiment was low compared with under the atmosphere environment. A recognition rate of object was unchanged with under the atmosphere environment. But, the error of position coordinate is large. This cause is that an error could not be corrected completely due to the lighting condition and the position of the object.

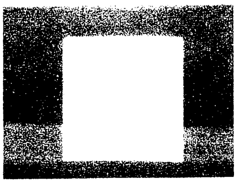


Fig.6 Target object

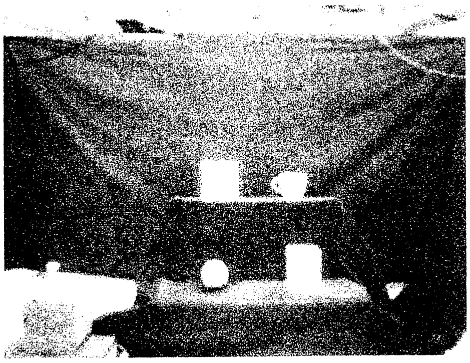


Fig.7 Right camera image

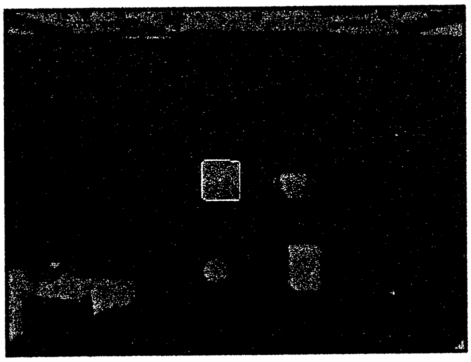


Fig.8 Recognition result

5. Conclusion

As first step of the object recognition for underwater robots, this paper described the measurement and the recognition of object as the robot eye system which stays in the water using the stereo vision system. The conclusion of this research is the following.

- This system has the difficulty that the influence of water is not so small. To compensate the influence, we rectified the error using the least-squares method.
- We proposed the algorithm which recognize the object using the two-dimensional feature extracted from the image.
- We confirmed the performance of this system with typical experiment.

The algorithm which was proposed in this paper could recognize the object that a shape was different. However it is difficult to recognize the object of the similar figures.

Therefore we need to construct new algorithm which can recognize the object of the similar figures. The algorithm can construct using the distance data which can get from the stereo vision system. As for that algorithm, we describe the next paper.

References

[1] Sumi Y and Tomita F (1997), Three-Dimensional Object Recognition Using Stereo Vision. IEICE Transactions D-II Information Processing, vol.J80-D-II, No.5, pp.1105-1112.

[2] Funabashi J, Matsuo H, Iwata A (1997), 3-D Object Recognition with Adaptive Multi-Scale Tree. IEICE Transactions D-II Information Processing, vol.J80-D-II, No.5, pp.1113-1121.

[3] Inoguchi S and Sato K (1990), 3-D Imaging Techniques for Measurement. shokodo.

[4] Okutomi M (1998), Difficulties in Stereo Vision. Journal of the Robotics Society of Japan, vol.16, no.2, pp.773-777.

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15–17, January, 2001,

Humanized Robot (Hubot) with K-Artificial Brain

Y.G.Zhang
Institute of Systems Science, Academia Sinica, Beijing China, 100080
yzhang@iss01.iss.ac.cn

Masanori Sugisaka
Department of Electric and Electronic, Oita University, Japan, 870-11
msugi@cc.oita-u.ac.jp

Abstract

In this paper authors explain the development of KANSEI technology briefly and show a basic design diagram for K-artificial brain. It has three main parts. First is KANSEI information sensing, second is parallel multi-mode information processing, third is KANSEI knowledge base. The prospective is to create humanized robot. It will be more advanced than human-shaped robot.

Key words: KANSEI technology; Computer sensing; Artificial Brain; Humanized robot.

I. Introduction

The development of robot has already passed more than 40 years, from robot hand to special functional robot, then programming control robot. In recent ten years the autonomous robot and intelligent robot arose. In fact, people want robot to have human-shape and intelligence human-like. Human-shaped is relatively easy; we have found from TV that a Japanese company has already created a human-shaped robot, which can move like human, it can freely upstairs and downstairs. This progress is very exciting the people. Behind it, as a matter of fact, the computer sensing technology backs it up. For example, computer vision (including advanced image processing), computer touching (including mechatronic) and so on. However, this is not the final object to human being, people want not only human-shaped, but also more humanized robot, here we want to create a new terminology that **Hubot**. In fact, humanized means robot has an **artificial brain by using KANSEI technology**. (We call it K-artificial brain). KANSEI technology has 3 main parts. Computer sensing is a necessary base, it is the all sources of KANSEI information; second is parallel multi-mode KANSEI information processing, it needs specially designed structure and algorithm; furthermore, KANSEI knowledge base is very important, what is the structure? How to express a KANSEI knowledge? These will be the hard research works in the future. In this paper authors try to show a basic design idea for this kind artificial brain by using KANSEI technology. However, we cannot imagine that a high intelligent K-artificial brain were only a "static head", it has no body and cannot move and do nothing except having intelligence. We hope it can serve human being. Thus, artificial brain would be coevolution with its "body". Obviously we can suppose that it should be a necessary and important part of new type of robots or any life-like agents. Tem years ago, there is a TV drama in the U.S., it tell us a story that a computer engineer created a machine girl who has human shape and a computer was build in, she can speak, talk with human, move like human. She is very honest for the instructions. Thus she caused a many interesting story making people grin from ear to ear. Ten years ago that is a science tale, but now it is a near future.

II. Developing Of "KANSEI" Technology

2.1) What is "KANSEI" technology?

KANSEI technology is a way to make computer have more functions similar to human. It depends on various sensors and parallel computing, which are similar to the "sensing organs" of human body, and also real way of human brainwork. Briefly, to install "eyes", "eras", "nose", and "tongue" for computer, and to process multi-mode KANSEI information obtained from those "sensing organs". Also, computer could form KANSEI experiences from KANSEI information and then KANSEI concepts. Furthermore, those are transferred into KANSEI knowledge. Thus, computer could have more artificial intelligence in a higher level.

2.2) Sensing input and progress

We category KANSEI input as the following six manners:

a) Computer Vision;
b) Computer Hearing;
c) Computer Tasting;
d) Computer Smelling;
e) Computer Touching; and
f) Computer digital input without sensors.

So far, there are various progresses for the above KANSEI input information. Basically, computer vision has little fast progress than others due to digital image processing technology. Now we can capture and store an image in digital format directly. Computer can read letters, characters and recognize some special patterns now, and discriminate different poison of simple objects in 3-D. That means computer has already had very primitive vision, which similar to insets vision. The difficult jobs are to capture and understanding moving images dynamically.

Computer hearing sensing has also obtained many progresses; due to high quality receiver computer can hear all sound in a very wide spectrum that human has. Now, computer can distinguish two different but very closed voices in spectrum. Sound recognition technology has made experiments to apply to Banking saving business. In addition, the voice synthesis devices can make electronic speaker sounds like human talk. The drawback is that the sentences it said is limited within those human designed, it cannot speak freely like human conversation. Now the audio direct input technology and language translation technology will help computer to understand sentences said by human.

Computer vision and hearing are two main KANSEI information, we could predict that in the near future those will achieve more progresses to fit the demand of KANSEI technology.

As to "smelling" and "tasting" of the KANSEI information, so far only a few work about them. Related research has started in SVBL (Satellite Venture Business Laboratory) of Muroran Institute of Technology, Japan.

Computer "touching" is only in a start stage; some robot hands can touch object *softly* and keep it up or down. However, human's touching sensing could sense objects, temperature, and feels different materials without vision. So, it is more difficult than vision and hearing.
From above we can say that KANSEI information detection is still developing, further more works are needed.

2.3) Parallel processing of KANSEI multi-mode information
The essentials part of KANSEI technology should be the **parallel processing of KANSEI multi-mode information.** We can show a simple example, while you are waiting for your friend to drink together. Meanwhile you heard the sound of knocking door and opened it, you found a person. At this time you received multi-mode KANSEI information: vision senses a familiar face, hearing familiar greeting voice from him, maybe he hold a box of delicious meat and you could be smelling the food fragrance. Your KANSEI knowledge tells you that your friend comes. You will have some reactions; you shake his hand and say some greeting words. During this period you have vision, hearing, touching and smelling senses and you process all this parallel multiple-mode information in the same time. The KANSEI knowledge base in your brain should be understanding the information and making compare with all old memories. Finally, you have some reaction to this information.

2.4) KANSEI knowledge base
Another related research should be **KANSEI knowledge base**. This is not original knowledge base, which is not in a parallel multiple-mode form. The questions is that how to express the KANSEI knowledge? What is the standard structure of KANSEI knowledge base? Usually, based on successive KANSEI incentives, KANSEI input information forms experiences and concepts, and then they are emerged as KANSEI knowledge. The access to KANSEI knowledge base is also different from the original knowledge base.

III. K-Artificial Brain

In a long term human wants to make resemble artificial brain by electronic or bio-electronic devise. Before the progress of hardware was taken the Artificial Neural Network (ANN) algorithm was born. A very famous work in ATR is the Cellular Automata Brain (CAB), which is a simulator of ANN by Cellular Automata in software at the first stage and then implementing in hardware devise. This is a very advanced and brave experiment. Whatever it is successful or not, it told us clearly that human will make resemble of human brain earlier or later. Of course, finally, resemble artificial brain maybe not in electronic form but in some bio-electronic form due to new type of biological material arise. One of authors of this paper, Professor Sugisaka, proposed a primitive artificial brain in 1998. He designed a more friendly and autonomous robots and calls it **Liferobot** (see Fig.1).
Another way is to use KANSEI technology to implement artificial brain. Maybe it is a way more practical. The authors of this paper proposed a

modified design to artificial brain, which is shown in the Figure 2 in the last page of this paper. We call it K-artificial brain. Anyway, people will in two ways to implement artificial brain.

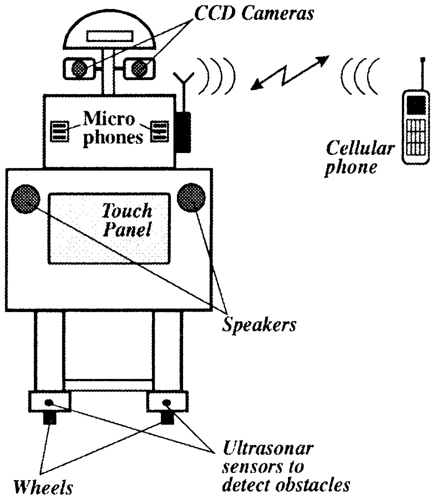


Fig. 1 The Liferobot in Oita university

IV. Coevolution Of Brain And Body

In the long long evolutionary process of human there are coevolution of both brain and body, that means not only brain become complicated more and more, but also the body had many change to adapt the change in environment, and body became more functional. The specific division of organs appeared and they evolved in very precise and tiny. As the developing of the structure of computer hardware it has more specific division and become precise more and more. Of course, if people want computer to be more intelligent its software has also to become more complicated. KANSEI technology would make computer into an artificial brain. Of course, the KANSEI technology is just in its earlier stage and has a long way to go over. However, we cannot imagine that a high intelligent artificial brain with KANSEI technology were only a "static head", it has no body and cannot move and do nothing except having intelligence. We hope it can serve human being. Thus, artificial brain would be coevolution with its "body". Obviously we can suppose that it should be a necessary and important part of new type of robots. People will develop very flexible arms and legs for this new type of robots and make them human-body-like. They could be on duty in houses or offices, could be monitors of specific places, could be nurses to take care of patients and have conversation with patients. This is not a tale, but a near future.

V. The Future, Artificial Life Engineering And Humanized Robots (Hubot)

From the description above we could see a prospect that we are doing a very attractive work that to build up an artificial system with higher intelligence and humanization by using of KANSEI technology in computer. We believe that a new direction in Artificial Life (ALife) field is arising, we try to name it as **Artificial Life Engineering** (briefly, ALile Engineering). Its main intent is to build up a new type of robots, which with KANSEI knowledge and higher intelligence, it is humanization of computer with KANSEI technology. We also can We also can name this new type of robots as **Humanized Robots**, or briefly as **Hubot.**

The development of computer is not only to improve the hardware and software; finally it will become the extension of human brain and body.

REFERENCES

[1]Zhang, Y.G. and Sugisaka, M., "A Review to the International Symposium on Artificial Life and Robotics (AROB)", **Proceedings of 5th International Symposium on Artificial Life and Robotics (AROB'00)**, pp. I-7, Oita, Japan , 2000

[2] Zhang, Y.G., Nagashima, T. and Shimohara, K., "Humanization of computer with KANSEI technology", Invited talk, also in **The proceedings of Third International Conference on Human and Computer(HC'2000)**, pp. 273-277. Sepetember 6-9, 2000, Aizu university, Aizuwakamatsu, Japan.

[3]Sugisaka, M, "Neurocomputer control in an artificial brain for tracking moving objects", **Artificial Life and Robotics**, Vol.1(1), pp 47-51, 1997.

[4] Sugisaka, M, "Design of an artificial brain for robots", Artificial Life and Robotics, Vol.3(1), pp 7-14, 1999.

[5] Sugisaka, M, "Behavioral control of artificial life robot, **Proceedings of The Fourteenth International Coference on Systems Engineering (ISCE 2000)**, pp 524-531, 12-14 September 2000, Coventry University, UK.

[6] Sugisaka, M, "Cognitive and behavioral artificial liferobot", **The Proceedings of 2000 International Symposium on Mechatronics and Intelligent Mechanical System for 21 Century (ISIM2000)**, Oct.4-7, 2000, KyongSangNam-Do, Korea.

[7] Sugisaka, M, "Artificial liferobot for human welfare" **The 6th Bellman Continuum, International Workshop on Intelligent Systems Resolutions,** Dec.11-12, 2000, National Tsing Hua University, Taiwan.

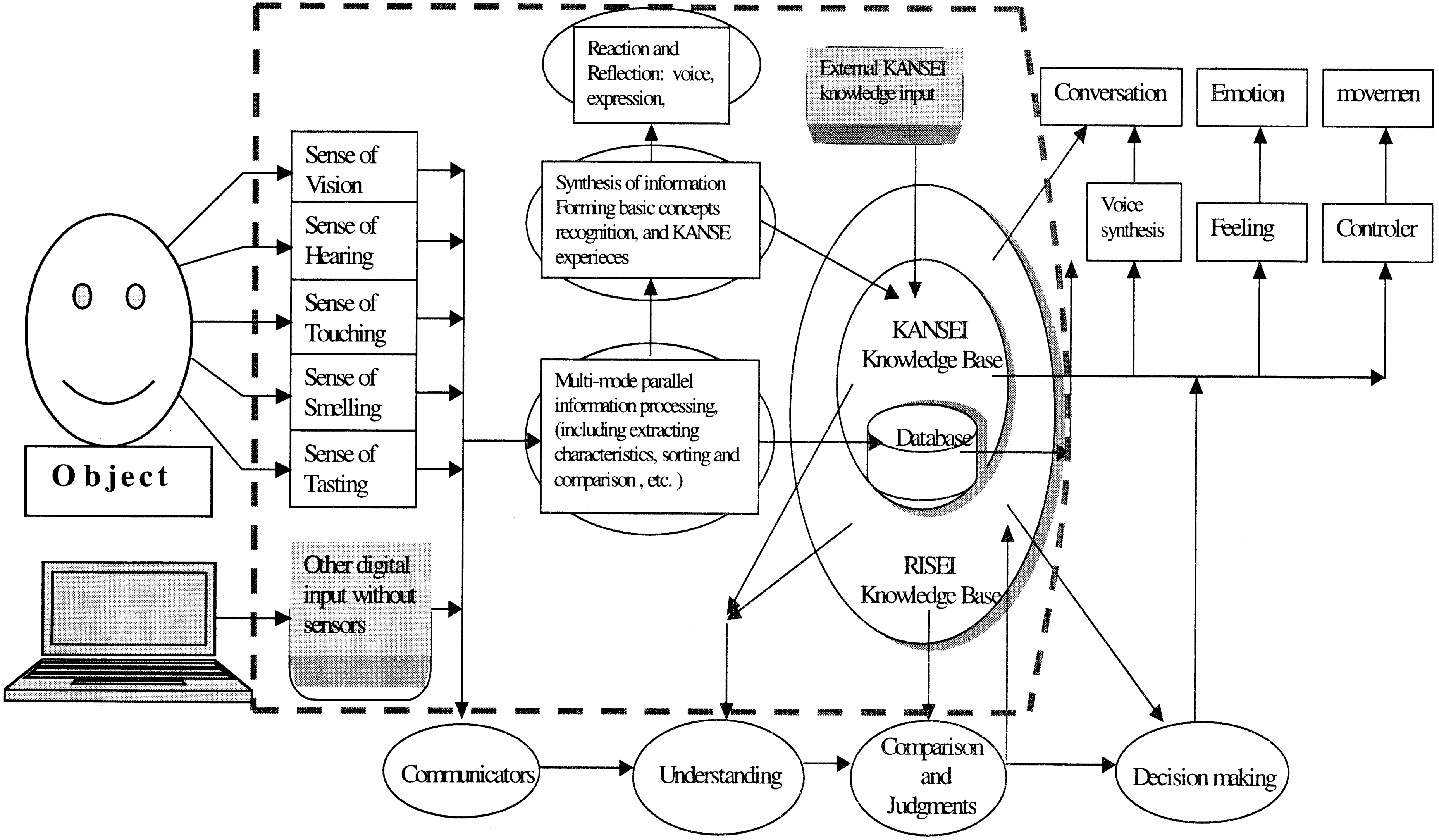

Object
Sense of Vision
Sense of Hearing
Sense of Touching
Sense of Smelling
Sense of Tasting
Other digital input without sensors
Reaction and Reflection: voice, expression,
Synthesis of information Forming basic concepts recognition, and KANSE experieces
Multi-mode parallel information processing, (including extracting characteristics, sorting and comparison , etc.)
External KANSEI knowledge input
KANSEI Knowledge Base
Database
RISEI Knowledge Base
Conversation
Emotion
movemen
Voice synthesis
Feeling
Controler
Communicators
Understanding
Comparison and Judgments
Decision making

Figure 2 The diagram of Artificial Brain with KANSEI technology

Proc. of The Sixth Int. Symp. on Artificial Life and Robotics (AROB 6th' 01)
Tokyo Japan, 15-17, January, 2001,

Investigation Summary of Physiological and Psychological Influence of Animals on People

Kazuyoshi Wada[*1], Takanori Shibata[*2], Teruaki Mitsui[*2], Li Yan[*2], Kazuo Tanie[*1, 2]

*1 Institute of Engineering Mechanics
University of Tsukuba
1-1Tennodai, Tsukuba, Ibaraki, 305-8577 Japan
mv305@mel.go.jp

*2 Bio-Robotics Div., Robotics Dept.
Mechanical Engineering Laboratory, AIST, MITI
1-2 Namiki, Tsukuba, Ibaraki, 305-8564 Japan
{shibata, mitsui, li, tanie}@mel.go.jp

Abstract

It has been pointed out that animals give people mental effects of enjoyment, relief and so on. We investigate various researches on physiological and psychological influences that animals exert on people. We consider the influences as functions of animals, and model them. We suggest a hypothesis putting forward, which will lead to the application to mental commit robot.

1. Introduction

We have researched and developed the artificial emotional creature [1][2]. It influences human mind and is a machine that attaches importance to subjective evaluation like real pet animals. It exists as something more than the daily necessity. The artificial emotional creature interacts with a human physically. It acts autonomously with its purposes and motives. As a result, the human subjectively interprets the artificial emotional creature as it has feelings, and he receives the own emotional change such as enjoyment and relief.

We have researched and developed mental commit robot as an application of artificial emotional creature [1]-[4]. It enriches a person's mind, prevents from mental sicknesses and gives enjoyment at leisure time. After carrying out basic psychological experiments, we have developed animal type robots (Figs. 1 and 2), while developing sensory systems and behavior generation algorithms.

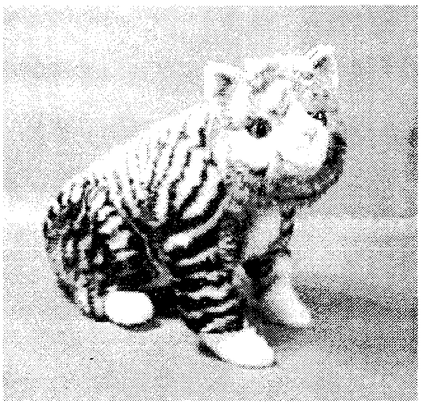


Fig. 1 Cat Robot (developed with OMRON)

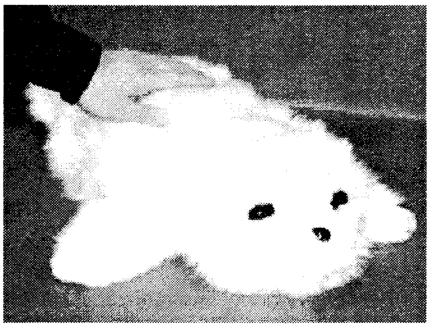


Fig. 2 Seal Robot
(developed with Sankyo Aluminum Industry)

It is known that the animals bring people mental effects by interaction. The effects are classified into three categories: physiological advantage, psychological advantage, and social advantage [5]. For instance, a sick recovery and the adaptation [6] belong to physiological advantage. Energy and motives increase [8], and relaxation [9] belongs to psychological advantage.

Connecting people [7] belongs to social advantage. We aim to investigate physiological and psychological influences that animals exert on people from these researches. We consider the influences as animal's functions and model them in order to apply to mental commit robot.

The chapter 2 describes investigation method in this research, the chapter 3 describes the investigation results about relation between animal's effects on people and interactions, the chapter 4 concludes this paper and describes the future perspectives on this research.

2. Investigation Method

There are the following three kinds of papers concerning physiological and psychological influence that animals exert on the people.

1) "Anecdotes" [9]-[12]

2) Investigation by "questionnaires" [7]

3) Physiological and psychological research [6]

1) Anecdotes are stories of the effects that animals bring people several mental and physical situations. The characteristics of "anecdote" are many kinds of effects and descriptions of interactions between people and animals. However, due to subjectivity of an author and a person who interacts an animal in an anecdote, objectivity of data is missing.

2) As for "questionnaires," we obtain overall tendency about a large number of people. The characteristics of questionnaires are various mental effects by subjective investigations. However, description about interactions between person and animals are scarce and objectivity of data is missing.

3) From physiological and psychological research, we obtain physiological indices data such as blood pressure and heart beat rate. Therefore, the effects of animals and interactions between people and animals are limited. However, objectivity of data is the highest of the three kinds.

As the results, we choose "anecdotes" for investigation from the standpoint that this research focuses on modeling animal's functions.

Next, physiological and psychological effects of animals are influenced by the following elements – age, mental and physical conditions, animal's kind, and interaction between people and animals. Therefore, we extract these elements from anecdotes. Due to a variety of expressions about animal's effects and actions of a person and animals in the anecdotes, we arranged the animal's effects by the classification defined by Yokoyama[5]. Moreover, we classified and arranged the actions of people and animals by how to interact as follows.

1) Verbal Communication

① Speak, talk

(a) A person speaks to an animal.

(b) Surrounding people speak to an animal.

(c) A person talks with the surrounding people.

2) Touch /contact

② Touch, fondle

(a) Touch/contact between a person and an animal

(b) Touch/contact between a person and the surrounding people.

(c) Touch/contact between an animal and the surrounding people.

(d) Touch/contact between a person, an animal and the surrounding people.

3) Exercise

③ Walk, walk together, come close

(a) Walking exercise by a person and an animal.

(b) Walking exercise by a person and the surrounding people.

(c) Walking exercise by an animal and the surrounding people.

(d) Walking exercise by a person, an animal and the surrounding people.

④ Play, ball play

(a) Playing exercise between a person and an animal.

(b) Playing exercise between a person and the surrounding people.

(c) Playing exercise between an animal and the surrounding people.

(d) Playing exercise between a person, an animal, and the surrounding people.

- The following is regular interactions between a person and an animal.

⑤ Taking care of and training an animal.

⑥ Horse back riding (horse therapy).

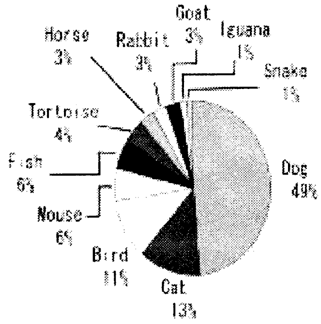


Fig. 3 Composition of Animals

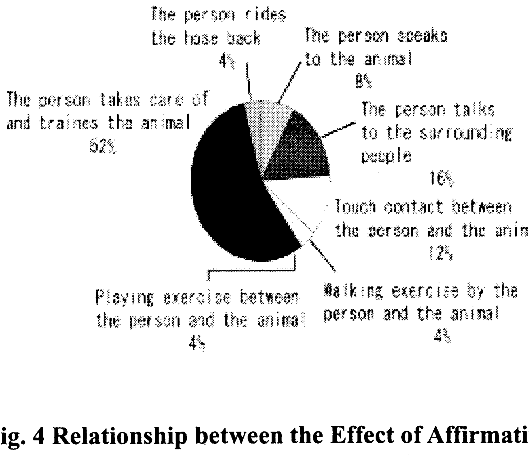


Fig. 4 Relationship between the Effect of Affirmative Feeling and Kinds of Interactions

3. Investigation results

Fifty-two anecdotes were marked in investigation. As for sex, 60% of anecdotes dealt with men, 25% did with women, and 15% were unknown. As for age, 27% of anecdotes dealt with adults, 42% did with children, and 31% were unknown. As for mental and physical conditions of the person, 36% of anecdotes dealt with disabled persons, 60% did with non-disabled, and 4% were unknown. The number of animals amounts to 70 in total because there were some anecdotes contained two or more animals. 49% of animals in anecdotes were dogs, 13% were cats, 11% were birds, and 11% were other animals as shown in Fig. 3.

We investigated the relation between a way of interaction and its effects from animals to people. Two or more effects and actions existed in one anecdote. It was thought that one action influenced two or more effects. We didn't examine the relation between an action and an effect simply. We assumed that all actions influenced all effects in an anecdote. There were certain actions that belonged to one anecdote, which was assumed as frequency of actions. We obtained the total number of frequency of actions in each effect through all anecdotes. In addition, we investigated the number of frequency of an action occupied in the total number of frequency of actions.

The effect of affirmative feelings is one of the animal's effects on people. The effect of affirmative feelings means that a person acquires self-esteem, useful feeling, a sense of responsibility, and a sense of achievement by interacting with an animal. Fig. 4 shows the relation between the effect of affirmative feelings and kinds of interactions. 52% of the effects of affirmative feelings were caused by taking care of and by training an animal, and 16% were by talking with the surrounding people. We can say that it is important for people to take care of and train animals for the effect of affirmative feelings.

Consequently, we think that animal's functions such as obeying instructions by a person and demanding care from a person for food and excrete bring the effect of affirmative feelings.

In order for mental commit robot to have similar effect to rise affirmative feelings in a human, the following three functions are important:

(1) To demand care from a person by demanding charge and making light troubles,

(2) To understand instructions of a person

(3) To learn the instructions

4. Conclusions

We investigated fifty-two anecdotes of interaction between animals and people. The effect of affirmative feelings is appeared as the result of investigation. Other effects were not appeared. One of the reasons is that the number of investigated anecdotes was not enough. A further investigation is necessary to investigate other effects.

It is thought that the result is affected by the fact that the rate of dogs in anecdotes occupies a half in all animals. Especially, the function of obeying to a person is thought to be a unique function of dogs. The function is an opposite function in the case of cats that are another familiar pet animals.

Therefore, we should consider whether the model is based on relation between ordinal people and animals or the model is based on a special case between a specific person and an animal. Then, we should consider whether the model is suitable for modeling physiological and psychological influence.

We will investigate further to improve present model and apply to mental commit robot.

Reference

[1] T. Shibata, et al, "Emotional Robot for Intelligent System – Artificial Emotional Creature Project," *Proc. of 5th IEEE Int'l Workshop on ROMAN*, pp466-471, 1996.

[2] T. Shibata, et al, "Artificial Emotional Creature for Human-Machine Interaction," *Proc. of the IEEE Int'l Conf. on SMC*, pp2269-2274, 1997.

[3] T. Shibata, "Mental Commit Robot for Healing Human Mind (in Japanese)," *JRSJ*, Vol.17, No.7, 1999.

[4] T. Shibata, "Subjective Interpretation and Value Created through Physical Interaction between Human and Robot (in Japanese)," *Proc. of 17th Conf. on JRSJ*, pp1119-1120, 1999.

[5] A. Yokoyama, "アニマル・セラピーとは何か (What is Animal therapy.)", *NHK BOOKS*, 1996.

[6] E. Friedmann. "The value of pets for health and recovery," *Pets Benefits and Practice*, December. 1990.

[7] J. A. Serpell. "Evidence for long term effects of pet ownership on human health," *Pets Benefits and Practice*, December. 1990.

[8] R. Green. "PETS IN PUBLIC HOUSING," *People, Animals, Environment*, Spring, 1989.

[9] P. Maggitti. "Nor Iron Bars a Cage The Story of Pets in Prison," *The ANIMALS' AGENDA*, July/August, 1988.

[10] M. Kale. "Kids & ANIMALS," *Inter Actions*, Vol.10, No.3, 1992.

[11] B. J. Carmack. "The role of companion animals for persons with AIDS/HIV," *HOLISTIC NURSING PRACTICE*. JANUARY. 1991.

[12] L.V. Remoorlene. "The Therapeutic Effects Animals for Children With Special Needs," *People-Animals-Environment*. Spring. 1988.

Author Index