PROCEEDINGS OF THE SEVENTH INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB 7th '02)

Vol. 1

Jan. 16-Jan. 18, 2002
B-Com Plaza, Beppu, Oita, JAPAN

Proceedings of The Seventh International Symposium on

ARTIFICIAL LIFE AND ROBOTICS

(AROB 7th '02)

for Information Technology

January 16-18, 2002
B-Con Plaza, Beppu, Oita, Japan

Editors: Masanori Sugisaka and Hiroshi Tanaka

THE SEVENTH INTERNATIONAL SYMPOSIUM
ON
ARTIFICIAL LIFE AND ROBOTICS

(AROB 7th '02)

ORGANIZED BY

Organizing Committee of International Symposium on Artificial
Life and Robotics   (Department of Electrical and Electronic
Engineering, Oita University, Japan)

CO-SPONSORED BY

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

COOPERATED BY

ISICE, IEICE, IEEE Japan Council, JARA

SUPPORTED BY

The Commemorative Association for Japan World Exposition
(1970)
Asahi shimbun
Beppu City Hall
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 NISHINIPPON
TOS Broadcast Company
The Mainichi Newspapers
THE YOMIURI SHIMBUN

ADVISORY COMMITTE CHAIRMAN

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

GENERAL CHAIRMAN

M. Sugisaka (Oita University, Japan)

CO-GENERAL CHAIRMAN(PROGRAM)

H.Tanaka (Tokyo Medical & Dental University, Japan)

CO-CHAIRMAN

J.L.Casti (Santa Fe Institute, USA)

ADVISORY COMMITTEE

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

INTERNATIONAL ORGANIZING COMMITTEE

K.Aihara (The University of Tokyo, Japan)
W.B.Arthur (Santa Fe Institute, USA)
W.Banzhaf (University of Dortmund, Germany)
C.Barrett (Los Alamos National Laboratory, USA)
M.Bedau (Reed College, USA)
Z.Bubnicki (Wroclaw University of Technology, Poland)
J.L.Casti (Santa Fe Institute, USA)
H.S.Cho (KAIST, Korea)
J.M.Epstein (Santa Fe Institute, USA)
T.Fukuda (Nagaya University, Japan)
H.Hashimoto (The University of Tokyo, Japan)
D.J.G.James (Coventry University, UK)
S.Kauffman (Santa Fe Institute, USA)
K.Kosuge (Tohoku University, Japan)
C.G.Langton (Santa Fe Institute, USA)
M.H.Lee (Pusan National University, Korea)
J.J.Lee (KAIST, Korea)
M.Nakamura (Saga University, Japan)
G.Obinata (Nagoya University, Japan)
S.Rasmussen (Santa Fe Institute, USA)
T.S.Ray (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)
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)
Z.Bubnicki (Wroclaw University of Technology, Poland)
T.Christaller (GMD-German National Research Center for
    Information Technology, Germany)
H.Asama (RIKEN, Japan)
I.Harvey (University of Sussex, UK)
H.Hashimoto (The University of Tokyo, Japan)
  (Co-chairman)
H.Hirayama (Asahikawa Medical College, Japan)
H.Inooka (Tohoku University, Japan)
K.Ito (Tokyo Institute of Technology, Japan)
J.Johnson (The Open University, UK)
Y.Kakazu (Hokkaido Unuversity, Japan)
O.Katai (Kyoto University, Japan)
S.Kawaji (Kumamoto University, Japan)
S.Kawata (Tokyo Metropolitan University, Japan)
S.Kitamura (Kobe University, Japan)
T.Kitazoe (Miyazaki University, Japan)
S.Kumagai (Osaka University, Japan)
H.H.Lund (University of Southern Denmark, Denmark)
H.Miyagi (Ryukyu University, Japan)
M.Nakamura (Saga University, Japan)
R.Nakatsu (ATR, Japan)
S.Omatsu (University of Osaka Prefecture, Japan)
R.Pfeifer (University of Zurich-Irchel, Switzerland)
T.E.Ray (University of Oklahoma, USA) (Co- chairman)
T.Sawaragi (Kyoto University, Japan)
T.Shibata (MITI, MEL, Japan)
K.Shimohara (ATR, Japan)
M.Sugisaka (Oita University, Japan)
H.Tanaka (Tokyo Medical & Dental University, Japan)
    (Chairman)
N.Tosa (ATR, Japan)
K.Ueda (Kobe University, Japan)
K.Uosaki (Tottori University, Japan)
H.Wakamatsu (Tokyo Medical & Dental University, Japan)
K.Watanabe (Saga University, Japan)

LOCAL ARRANGEMENT COMMITTEE

M.Hara (Oita University, Japan)
K.Nakanishi (Oita University, Japan)

K.Nakano (University of Electro-Communications, Japan)
K.Okazaki (Fukui University, Japan)
K.Shibata (Oita University, Japan) (Chairman)
K.Shigemitsu (Oita Industrial Research Institute, Japan)
M.Sugisaka (Oita University, Japan)
Y.Suzuki (Tokyo Medical & Dental University)
X.Wang (Oita Institute of Technology, Japan)
I. Yoshihara(Miyazaki University, Japan)

HISTORY

This symposium was founded in 1996 by the support of Science and International Affairs Bureau, Ministry of Education, Science, Sports and Culture, Japanese Government. Since then, this symposium was held every year at B-Con Plaza, Beppu, Oita, Japan except Tokyo, Japan (AROB 6th '02). The Seventh symposium will be held on 16-18 January 2002, at B-Con Plaza, Beppu, Oita, Japan. This symposium invites you all to discuss development of new technologies concerning Artificial Life and Robotics based on simulation and hardware in twenty first century.

OBJECTIVE

The objective of this symposium is the development of new technologies for artificial life and robotics which have been recently born in Japan and are expected to be applied in various fields. This symposium will discuss new results in the field of artificial life and robotics.

TOPICS

Artificial brain research
Artificial intelligence
Artificial life
Artificial living
Artificial mind research
Bioinformatics chaos
Brain science
Cognitive science evolutionary computations
Complexity
Computer graphics
DNA computing
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
Multi-agent systems
Nano-biology
Nano-robotics
Neural networks
Neurocomputers
Neurocomputing technologies and its application for hardware
Pattern recognition
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 fir the consideration o publication in an international journal ARTIFICIAL LIFE AND ROBOTICS (Springer) and APPLIED MATHEMATICS ANDCOMPUTATION (North-Holland). All correspondence related to the symposium should be addressed to AROB Secretariat

Dept. of Electrical and Electronic Engineering,
Oita University
700 Dannoharu, Oita 870-1192, JAPAN
TEL +81-97-554-7841 FAX +81-97-554-7818
E-MAIL arobsecr@cc.oita-u.ac.jp
Home Page http://arob.cc.oita-u.ac.jp/

MESSAGE

Masanori Sugisaka
General Chairman of AROB
(Professor, Oita University)
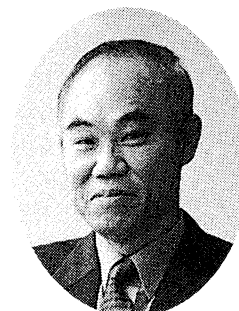
It is my great honor to invite you all to the upcoming International Symposium on Artificial Life and Robotics. The first symposium was held in February (18-20) 1996, B-Con Plaza, Beppu, Oita, Japan. That symposium was organized by Oita University under the sponsorship of the Japanese Ministry of Education, Culture, Sports, Science and Technology (Monkasho), and co-sponsored by Santa Fe Institute (USA), SICE, RSJ, and IEEJ, (Japan). I would like to express my sincere thanks to the Science and International Affairs Bureau, Monkasho, Japanese Government, for their repeated support.

This symposium is supported by The Commemorative Association for Japan World Exposition (1970) and other institutions. The symposium invites you to discuss the development of new technologies in the 21st century, concerning Artificial Life and Robotics, based on simulation and hardware.

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.

M. Sugisaka

December 20, 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 Seventh International Symposium on Artificial Life and Robotics (AROB 7th '02). 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. The accomplishment of HGP(Human Genome Project) has published the special issue of Nature, and 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. Thus, in the "post-genomic era", 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 Beppu, Oita.

H. Tanaka

December 20, 2001

TIME TABLE

	RoomA	RoomB	RoomC	RoomD
1/15(Tus.) 8:00				
13:00	Registration (Registration Desk)			
17:00				
1/16(Wed.)8:00	Registration (Registration Desk)			Workshop1 WS1
9:00	Chair Y. Zhang GS1　　　　　6	Chair S. Rasmussen GS2　　　　　6	Chair P. Sapaty GS3　　　　　6	
10:30				
10:40	Coffee Break			Opening Ceremony
11:00				Plenary Talk Chair H. Tanaka PT1　Dr. Parker
12:00	Lunch			
13:00	Chair S. Kumagai S1　　　　　5	Chair N. Homma S2　　　　　5	Chair M. H. Lee GS4　　　　4	
14:00			Chair Z. Bubnicki	
14:15	Chair A. Dorin GS5　　　　3	Chair D. Green GS6　　　　3	GS7　　　　4	
15:00				
15:10	Chair H. Tanaka GS8　　　4(3)	Chair S. Kawata S3　　　　4	Chair G. James GS9　　　　4	
16:10	Chair K. Aihara S4　　　　5	Chair C. Taylor GS10　　　4	Chair G. Wang GS11　　　4(5)	
17:25				

GS1:	Intelligent Control and Modeling-1
GS2:	Artificial Life-1
GS3:	Robotics-1
GS4:	Mobile Vehicle-1
GS5:	Complexity
GS6:	Artificial Life-2
GS7:	Mobile Vehicle-2
GS8:	Artificial Brain Research & Artificial Intelligence
GS9:	Mobile Vehicle-3 & Micro-Machines
GS10:	Artificial Life-3 & Innovative Computation
GS11:	Genetic Algorithm

S1:	Bioloical Complexity: From Neural Network to Genetic Network
S2:	Complex Neural Systems
S3:	Machine Learning
S4:	Nonlinear Neurodynamics: Analysis and Application

	RoomA	RoomB	RoomC	RoomD
1/17(Thurs.)8:00	Registration (Registration Desk)			
9:00	Chair M. Bedau & H. Hashimoto S5　　　　7	Chair H. Ishibuchi GS12　　　　7	Chair J. J. Lee GS13　　　　7	Workshop2 WS2
10:45 / 11:00	Coffee Break			Plenary Talk Chair M. Sugisaka PT2 Dr. Hashimoto
12:00				
13:00	Lunch			
	Chair Y. Suzuki S6　　　　5	Chair K. Watanabe S7　　　　5	Chair M. Nakamura GS14　　　　5	
14:15 / 14:30	Coffee Break			
	Chair N. Homma S8　　　　5	Chair W. Wells GS15　　　　5	Chair J. J. Lee GS16　　　　5(4)	
15:45	Chair A. Hirst GS17　　　　5	Chair M. Bedau S9　　　　5(4)	Chair L. Parker GS18　　　　5	
17:00				
18:00	Banquet (Reception Hall)			
20:00				

GS12: Fuzzy Control and Evolutionary Computation

GS13: Robotics-2

GS14: Robotics-3 & Related Fields

GS15: Intelligent Control and Modeling-2

GS16: Human-Machine Cooperative Systems & Control
Human-Welfare Robotics

GS17: Micro-Robot World Cup Soccer Tournament
& Related Fields

GS18: Pattern Recognition

S5: The Origin and Evolution of Life &
Interaction and Intelligence

S6: Dynamical Networks

S7: Machine Intelligence and
Robotic Control

S8: Complex Dynamics in
Recurrent Neural Networks

S9: Recent Artificial Life in Australia

		RoomA	RoomB	RoomC	RoomD
1/18(Fri.)	8:00	Registration (Registration Desk)			
	9:00	Chair T. Kitazoe S10　　　　　6(4)	Chair M. Kono S11　　　　　6(5)	Chair K. Uosaki S12　　　　　6(4)	
	10:30	Coffee Break			
	10:45 11:00	Chair .J. Casti GS19　　　　　5	Chair T. Yamawaki GS20　　　　　5	Chair H. Hirayama GS21　　　　　5	
	12:00	Lunch			
	13:00	Chair J. Johnson GS22　　　　　6(4)	Chair G. James GS23　　　　　6		
	14:30	Farewell Party			
	15:30				

GS19: Multi-Agent Systems-1

GS20: Neural Networks-1

GS21: Molecular Biology, Nano-biology,

GS22: Multi-Agent Systems-2

GS23: Neural Networks-2 & Computer
Graphics

S10: Intelligent Systems of Autonomous Mobile Robots
for Collective Behavior and Nursing

S11: Control and Perception for Uncertain, Under Actuated
Systems

S12: Optimization and Simultaneous Perturbation

TECHNICAL PAPER INDEX

January 16 (Wednesday)

Room D

11:00~12:00 PT-1 Plenary Talk1
Chair H. Tanaka

January 17 (Thusday)

11:00~12:00 PT-2 Plenary Talk2
Chair M. Sugisaka

January 16 (Wednesday)

8:00~10:40 WS1 Workshop 1

January 17 (Thusday)

9:00~10:20 WS2 Workshop 2

January 16 (Wednesday)

Room A

9:00~10:30 GS1 Intelligent control and modeling 1
Chair: Y. Zhang

13:00~14:15 S1 Biological complexity: from neural network to genetic network
Chair: S. Kumagai (Osaka University, Japan)
Co- Chair: S. Doi (Osaka University, Japan)

J, Matsuoka, Y. Sekine, K. Saeki (Nihon University, Japan)
K. Aihara (The University of Tokyo, Japan)

14:15~15:00 GS5 Complexity
Chair: A. Dorin

15:10~16:10 GS8 Artificial brain research & Artificial intelligence
Chair: H. Tanaka

16:10~17:25 S4 Nonlinear neurodynamics : analysis and application
Chair: K. Aihara
Co-Chair: H. Suzuki

Room B

9:00~10:30 GS2 Artificial life 1
Chair: S. Rasmussen

13:00~14:15 S2 Complex neural systems
Chair: N. Homma
Co-Chair: K. Abe

16:10~17:25 GS10 Artificial life 3 & Innovative computations
Chair: C. Taylor

Room C

9:00~10:30 GS3 Robotics 1
Chair: P. Sapaty

I. Shim, J. Yoon (Pusan National University, Korea)

13:00~14:00 GS4 Mobile vehicle 1
Chair: M. H. Lee

14:00~15:00 GS7 Mobile vehicle 2
Chair: Z. Bubnicki

15:10~16:10 GS9 Mobile vehicle 3 & Micro-machines
Chair: G. James

A. A. Loukianov, M. Sugisaka (Oita University, Japan)

16:10~17:40 GS11 Genetic algorithms
Chair: G. Wang

January 17 (Thursday)

Room A

9:00~10:45 S5 The origin and evolution of life & Interaction and intelligence
Chair: M. Bedau
Chair: H. Hashimoto

Chair: N. Homma
Co-Chair: K. Gohara

15:45~17:00 GS17 Micro-robot world cup soccer tournament & Related fields

Chair: A. Hirst

Room B

9:00~10:45 GS12 Fuzzy control & Evolutionary computations

Chair: H. Ishibuchi

13:00~14:15 S7 Machine intelligence and robotic control
Chair: K. Watanabe
Co-Chair: K. Izumi

14:30~15:45 GS15 Intelligent control & modeling 2
Chair: W. Wells

15:45~17:00 S9 Recent artificial life in Australia
Chair: M. Bedau

A. Dorin (Monash University, Australia)

Room C

9:00~10:45 GS13 Robotics 2
Chair: J. J. Lee

13:00~14:15 GS14 Robotics 3 & Related fields
Chair: M. Nakamura

14:30~15:45 GS16 Human-machine cooperative systems &
Control human-welfare robotics
Chair: J. J. Lee

15:45~17:00 GS18 Pattern recognition
Chair: L. Parker

M. Sugisaka, X. Fan (Oita University, Japan)

GS18-5 *Development of a face recognition system for the liferobot* ······538
       M. Sugisaka, X. Fan (Oita University, Japan)

January 18 (Friday)

Room A

9:00~10:30 S10 Intelligent systems of autonomous mobile robots for collective behavior and nursing
Chair: T. Kitazoe
Co-Chair: M. Tabuse

S10-1 *Autonomous collective movement of mobile robot khepera* ······542
       T. Horita, T. Shinchi, M. Tabuse, T. Kitazoe (Miyazaki University, Japan)

S10-2 *Khepera robots applied to highway autonomous mobiles* ······546
       A. Todaka, M. Tabuse, T. Shinchi, T. Kitazoe (Miyazaki University, Japan)

S10-3 *An Autonomous mobile robot and camera sensing* ······550
       Y. Kitamaru, K. Sugihara, M. Tabuse, T. Kitazoe (Miyazaki University, Japan)

S10-4 *The intelligent systems for improving the quality life of wheelchair users* ······554
       K. Kusano, T. Kitazoe, M. Tabuse, T. Shinchi (Miyazaki University, Japan)

10:45~12:00 GS19 Multi agent systems 1
Chair: J. Casti

GS19-1 *Multi-agent developmental model of plants* ······556
       X. Xu, C. S. Zhang (Tsinghua University, P. R. China)

GS19-2 *A Multi-agent method for texture synthesis* ······560
       T. Jiao, C. S. Zhang (Tsinghua University, P. R. China)

GS19-3 *The artificial market model considering agent's preference on market selection* ······564
       J. Shibata, T. Tanaka, K. Okuhara (Hiroshima Prefecture University, Japan)

GS19-4 *Network security via multilple agent systems* ······568
       H. S. Seo (Sungkyunkwan University, Korea)
       S. M. Jung, H. W. Kim (Shamyook University, Korea)

GS19-5 *Desirable-event subgoal theory in the design of swarm robot control systems* ······572
       J. Johnson (Open University, UK)

M. Sugisaka (Oita University, Japan)

13:00~14:30 GS22 Multi agent systems 2
Chair: J. Johnson

Room B

9:00~10:30 S11 Control and Perception for Uncertain, Underactuated
Systems
Chair: M. Kono
Co-Chair: M. Yokomichi

10:45~12:00 GS20 Neural Networks 1

Chair: M. T. Yamawaki

13:00~14:30 GS23 Neural networks 2 & Computer graphics
Chair: G. James

Room C

9:00~10:30 S12 Optimization and Simultaneous Perturbation
Chair: K. Uosaki

10:45~12:00 GS21 Molecular biology & Nano-bioligy & DNA computing
Chair: H. Hirayama

Advances in Multi-Robot Systems

Lynne E. Parker

Center for Engineering Science Advanced Research

Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge TN 37831-6355, USA

email: ParkerLE@ornl.gov

Abstract— As research progresses in distributed robotic systems, more and more aspects of multi-robot systems are being explored. This article describes advances in multi-robot systems and surveys the current state of the art. Our focus is principally on research that has been demonstrated in physical robot implementations. We have identified eight primary research topics within multi-robot systems — biological inspirations, communication, architectures, localization/mapping/exploration, object transport and manipulation, motion coordination, reconfigurable robots, and learning – and discuss the current state of research in these areas. As we describe each research area, we identify some key open issues in multi-robot team research. We conclude by identifying several additional open research issues in distributed mobile robotic systems.

I. INTRODUCTION

The field of distributed robotics has its origins in the late-1980's, when several researchers began investigating issues in multiple mobile robot systems. Prior to this time, research had concentrated on either single robot systems or distributed problem-solving systems that did not involve robotic components. The topics of particular interest in this early distributed robotics work include:

- *Cellular (or reconfigurable) robot systems*, such as the work by Fukuda, *et al.* [20] on the Cellular Robotic System (CEBOT) and the work on cyclic swarms by Beni [8],
- *Multi-robot motion planning*, such as the work by Premvuti and Yuta [38] on traffic control and the work on movement in formations by Arai, *et al.* [2] and Wang [48], and
- *Architectures for multi-robot cooperation*, such as the work on ACTRESS by Asama, *et al.* [4].

Since this early research in distributed mobile robotics, the field has grown dramatically, with a much wider variety of topics being addressed. This paper examines the current state of the art in autonomous multiple mobile robotic systems. The field of cooperative autonomous mobile robotics is still so new that no topic area within this domain can be considered mature. Some areas have been explored more extensively, however, and the community is beginning to understand how to develop and control certain aspects

of multi-robot teams. Thus, rather than summarize the research into a taxonomy of cooperative systems (see Dudek [18] and Cao [12] for previous related summaries), we instead organize this research by the principal topic areas that have generated significant levels of study, to the extent possible in a limited space. As we present the review, we identify key open research issues within each topic area. We conclude by suggesting additional research issues that have not yet been extensively studied, but appear to be of growing interest and need in distributed autonomous multi-robot systems.

II. BIOLOGICAL INSPIRATIONS

Nearly all of the work in cooperative mobile robotics began after the introduction of the new robotics paradigm of behavior-based control [10], [3]. This behavior-based paradigm has had a strong influence in much of the cooperative mobile robotics research. Because the behavior-based paradigm for mobile robotics is rooted in biological inspirations, many cooperative robotics researchers have also found it instructive to examine the social characteristics of insects and animals, and to apply these findings to the design of multi-robot systems[1].

The most common application of this knowledge is in the use of the simple local control rules of various biological societies – particularly ants, bees, and birds – to the development of similar behaviors in cooperative robot systems. Work in this vein has demonstrated the ability for multi-robot teams to flock, disperse, aggregate, forage, and follow trails (e.g., [30], [15], [17]). The application of the dynamics of ecosystems has also been applied to the development of multi-robot teams that demonstrate emergent cooperation as a result of acting on selfish interests [32]. To some extent, cooperation in higher animals, such as wolf packs, has generated advances in cooperative control. Significant study in predator-prey systems has occurred, although primarily in simulation [7], [21]. Competition in multi-

---

[1] For a more detailed analysis of various types of biological systems – what Tinbergen called *differentiating* (e.g., ants) and *integrative* (e.g., birds) – and their relationship to cooperative robotics work (i.e., "swarm" vs. "intentional"), see [35].

robot systems, such as found in higher animals including humans, is beginning to be studied in domains such as multi-robot soccer [46], [29].

These areas of biological inspiration and their applicability to multi-robot teams seem to be fairly well understood. More recently identified, less well understood, biological topics of relevance include the use of imitation in higher animals to learn new behaviors, and the physical interconnectivity demonstrated by insects such as ants to enable collective navigation over challenging terrains.

III. COMMUNICATION

The issue of communication in multi-robot teams has been extensively studied since the inception of distributed robotics research. Distinctions between implicit and explicit communication are usually made, in which implicit communication occurs as a side-effect of other actions, or "through the world", whereas explicit communication is an specific act designed solely to convey information to other robots on the team. Several researchers have studied the effect of communication on the performance of multi-robot teams in a variety of tasks, and have concluded that communication provides certain benefit for particular types of tasks [27], [6]. Additionally, these researchers have found that, in many cases, communication of even a small amount of information can lead to great benefit [6].

More recent work in multi-robot communication has focused on representations of languages and the grounding of these representations in the physical world [22], [23]. Additionally, work has extended to achieving fault tolerance in multi-robot communication, such as setting up and maintaining distributed communications networks [51] and ensuring reliability in multi-robot communications [34]. While progress is being made in these more recent issues of communication, much work remains to enable multi-robot teams to operate reliably amidst faulty communication environments.

IV. ARCHITECTURES, TASK PLANNING, AND CONTROL

A great deal of research in distributed robotics has focused on the development of architectures, task planning capabilities, and control. This research area addresses the issues of action selection, delegation of authority and control, the communication structure, heterogeneity versus homogeneity of robots, achieving coherence amidst local actions, resolution of conflicts, and other related issues. Each architecture that has been developed for multi-robot teams tends to focus on providing a specific type of capability to the dis-

tributed robot team. Capabilities that have been of particular emphasis include task planning [1], fault tolerance [36], swarm control [31], human design of mission plans [26], and so forth.

A general research question in this vein is whether specialized architectures for each type of robot team and/or application domain are needed, or whether a more general architecture can be developed that can be easily tailored to fit a wider range of multi-robot systems. Relatively little of the previous work has been aimed at unifying these architectures. Perhaps an all-encompassing architecture would be too unwieldy to implement in practical applications. It remains to be seen if a single general architecture for multi-robot teams can be developed that is applicable to a much wider variety of domains than is currently possible with existing architectures.

V. LOCALIZATION, MAPPING, AND EXPLORATION

An extensive amount of research has been carried out in the area of localization, mapping, and exploration for single autonomous robots. Only fairly recently has much of this work been applied to multi-robot teams. Almost all of the work has been aimed at 2D environments. Additionally, nearly all of this research takes an existing algorithm developed for single robot mapping, localization, or exploration, and extends it to multiple robots, as opposed to developing a new algorithm that is fundamentally distributed. One exception is some of the work in multi-robot localization, which takes advantage of multiple robots to improve positioning accuracy beyond what is possible with single robots [42], [19].

As is the case with single robot approaches to localization, mapping, and exploration, research into the multi-robot version can be described using the familiar categories based on the use of landmarks [14], scan-matching [11], and/or graphs [40], [39], and which use either range sensors (such as sonar or laser) or vision sensors. While the single robot version of this problem is fairly well understood, much remains to be studied in the multi-robot version. For example, one question is the effectiveness of multi-robot teams over single-robot versions, and to what extent adding additional robots brings diminishing returns. This issue has begun to be studied (see [39]), but much much remains to be determined for the variety of approaches available for localization, mapping, and exploration.

VI. OBJECT TRANSPORT AND MANIPULATION

Enabling multiple robots to cooperatively carry, push, or manipulate common objects has been a long-standing, yet difficult, goal of multi-robot systems. Many research projects have dealt with this topic area;

fewer of these projects have been demonstrated on physical robot systems. This research area has a number of practical applications that make it of particular interest for study.

Numerous variations on this task area have been studied, including constrained and unconstrained motions, two-robot teams versus "swarm"-type teams, compliant versus non-compliant grasping mechanisms, cluttered versus uncluttered environments, global system models versus distributed models, and so forth. Perhaps the most demonstrated task involving cooperative transport is the pushing of objects by multi-robot teams [43], [45]. This task seems inherently easier than the carry task, in which multiple robots must grip common objects and navigate to a destination in a coordinated fashion [49], [24]. A novel form of multi-robot transportation that has been demonstrated is the use of ropes wrapped around objects to move them along desired trajectories [16].

Nearly all of the previous work in this area work involves robots moving across a flat surface. A challenging open issue in this area is cooperative transport over uneven outdoor terrains.

VII. MOTION COORDINATION

A popular topic of study in multi-robot teams is that of motion coordination. Research themes in this domain that have been particularly well studied include multi-robot path planning [52], traffic control [38], formation generation [2], and formation keeping [5], [48]. Most of these issues are now fairly well understood, although demonstration of these techniques in physical multi-robot teams (rather than in simulation) has been limited. More recent issues studied within the motion coordination context are target tracking [37], target search [25], and multi-robot docking [33] behaviors.

Nearly all of the previous work has been aimed at 2D domains, although some work has been aimed at 3D environments [52]. One of the most limiting characteristics of much of the existing path planning work is the computational complexity of the approaches. Perhaps as computing processor speed increases, the computational time will take care of itself. In the meantime, this characteristic is a limiting factor to the applicability of much of the path planning research in dynamic, real-time robot teams.

VIII. RECONFIGURABLE ROBOTICS

Even though some of the earliest research in distributed robotics focused on concepts for reconfigurable distributed systems [20], [8], relatively little work has proceeded in this area until the last few years. More recent work has resulted in a number of actual physical robot systems that are able to reconfig-

ure. The motivation of this work is to achieve function from shape, allowing individual modules, or robots, to connect and re-connect in various ways to generate a desired shape to serve a needed function. These systems have the theoretical capability of showing great robustness, versatility, and even self-repair.

Most of the work in this area involves identical modules with interconnection mechanisms that allow either manual or automatic reconfiguration. These systems have been demonstrated to form into various navigation configurations, including a rolling track motion [53], an earthworm or snake motion [53], [13], and a spider or hexapod motion [53], [13]. Some systems employ a cube-type arrangement, with modules able to connect in various ways to form matrices or lattices for specific functions [9], [54], [44], [47].

Research in this area is still very young, and most of the systems developed are not yet able to perform beyond simple laboratory experiments. While the potential of large numbers of robot modules has, to some extent, been demonstrated in simulation, it is still uncommon to have implementations involving more than a dozen or so physical modules. The practical application of these systems is yet to be demonstrated, although progress is being made in that direction. Clearly, this is a rich area for continuing advances in multi-robot systems.

IX. LEARNING

Many multi-robot researchers believe that an approach with more potential for the development of cooperative control mechanisms is autonomous learning. While a considerable amount of work has been done in this area for multi-agent learning [50], somewhat less work has been accomplished to date in multi-robot learning. The types of applications that are typically studied for this area of multi-robot learning vary considerably in their characteristics. Some of the applications include predator/prey [7], [21], box pushing [28], foraging [31], multi-robot soccer [46], [29], [41], and cooperative target observation [37].

Particularly challenging domains for multi-robot learning are those tasks that are *inherently* cooperative – that is, tasks in which the utility of the action of one robot is dependent upon the current actions of the other team members. Inherently cooperative tasks cannot be decomposed into independent subtasks to be solved by a distributed robot team. Instead, the success of the team throughout its execution is measured by the combined actions of the robot team, rather than the individual robot actions. This type of task is a particular challenge in multi-robot learning, due to the difficulty of assigning credit for the individual actions of the robot team members. Multi-robot learning in

general, and inherently cooperative task learning in particular, are areas in which significant research for multi-robot systems remains.

X. ADDITIONAL OPEN ISSUES IN DISTRIBUTED AUTONOMOUS MOBILE ROBOTICS

It is clear that since the inception of the field of distributed autonomous mobile robotics less than two decades ago, significant progress has been made on a number of important issues. The field has a good understanding of the biological parallels that can be drawn, the use of communication in multi-robot teams, and the design of architectures for multi-robot control. Considerable progress has been made in multi-robot localization/mapping/exploration, cooperative object transport, and motion coordination. Recent progress is beginning to advance the areas of reconfigurable robotics and multi-robot learning. Of course, all of these areas have not yet been fully studied; we have identified key open research challenges for these areas in the previous sections.

Several other research challenges still remain, including:
- How do we identify and quantify the fundamental advantages and characteristics of multi-robot systems?
- How do we easily enable humans to control multi-robot teams?
- Can we scale up to demonstrations involving more than a dozen or so robots?
- Is passive action recognition in multi-robot teams possible?
- How can we enable physical multi-robot systems to work under hard real-time constraints?
- How does the complexity of the task and of the environment affect the design of multi-robot systems?

These and other issues in multi-robot cooperation should keep the research community busy for many years to come.

REFERENCES

[1] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multi-robot cooperation in the Martha project. *IEEE Robotics and Automation Magazine*, 1997.

[2] T. Arai, H. Ogata, and T. Suzuki. Collision avoidance among multiple robots using virtual impedance. In *Proceedings of the Intelligent Robots and Systems (IROS)*, pages 479–485, 1989.

[3] Ronald C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[4] Hajime Asama, Akihiro Matsumoto, and Yoshiki Ishida. Design of an autonomous and distributed robot system: ACTRESS. In *Proceedings of IEEE RSJ International Workshop on Intelligent Robots and Systems*, pages 283–290, Tsukuba, Japan, 1989.

[5] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, December 1998.

[6] Tucker Balch and Ronald C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):1–25, 1994.

[7] M. Benda, V. Jagannathan, and R. Dodhiawalla. On optimal cooperation of knowledge sources. Technical Report BCS-G2010-28, Boeing AI Center, August 1985.

[8] Gerardo Beni. The concept of cellular robot. In *Proceedings of Third IEEE Symposium on Intelligent Control*, pages 57–61, Arlington, Virginia, 1988.

[9] H. Bojinov, A. Casal, and T. Hogg. Emergent structures in moduluar self-reconfigurable robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1734–1741, 2000.

[10] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[11] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 476–481, 2000.

[12] Y. Uny Cao, Alex Fukunaga, Andrew Kahng, and Frank Meng. Cooperative mobile robotics: Antecedents and directions. In *Proceedings of 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95)*, pages 226–234, 1995.

[13] A. Castano, R. Chokkalingam, and P. Will. Autonomous and self-sufficient conro modules for reconfigurable robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[14] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[15] J. Deneubourg, S. Goss, G. Sandini, F. Ferrari, and P. Dario. Self-organizing collection and transport of objects in unpredictable environments. In *Japan-U.S.A. Symposium on Flexible Automation*, pages 1093–1098, 1990.

[16] B. Donald, L. Gariepy, and D. Rus. Distributed manipulation of multiple objects using ropes. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 450–457, 2000.

[17] Alexis Drogoul and Jacques Ferber. From Tom Thumb to the Dockers: Some experiments with foraging robots. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 451–459, 1992.

[18] Gregory Dudek et al. A taxonomy for swarm robots. In *Proceedings of 1993 IEEE International Conference on Intelligent Robots and Systems (IROS '93)*, pages 441–447, 1993.

[19] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Collaborative multi-robot exploration. *Autonomous Robots*, 8(3), 2000.

[20] T. Fukuda and S. Nakagawa. A dynamically reconfigurable robotic system (concept of a system and optimal configurations). In *Proceedings of IECON*, pages 588–595, 1987.

[21] Thomas Haynes and Sandip Sen. Evolving behavioral strategies in predators and prey. In Gerard Weiss and Sandip Sen, editors, *Adaptation and Learning in Multi-Agent Systems*, pages 113–126. Springer, 1986.

[22] L. Hugues. Collective grounded representations for robots. In *Proceedings of Fifth International Conference on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[23] David Jung and Alexander Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, 8(3), July 2000.

[24] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal. Vehicle/arm coordination and mobile manipulator decentralized cooperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 546–553, 1996.

[25] S. M. LaValle, D. Lin, L. J. Guibas, J-C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *submitted to 1997 International Conference on Robots and Automation*, 1997.

[26] D. MacKenzie, R. Arkin, and J. Cameron. Multiagent mission specification and execution. *Autonomous Robots*, 4(1):29–52, 1997.

[27] Bruce MacLennan. Synthetic ethology: An approach to the study of communication. In *Proceedings of the 2nd interdisciplinary workshop on synthesis and simulation of living systems*, pages 631–658, 1991.

[28] S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. In *Proceedings of AAAI-91*, pages 8–14, 1991.

[29] S. Marsella, J. Adibi, Y. Al-Onaizan, G. Kaminka, I. Muslea, and M. Tambe. On being a teammate: Experiences acquired in the design of RoboCup teams. In O. Etzioni, J. Muller, and J. Bradshaw, editors, *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 221–227, 1999.

[30] Maja Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In J. Meyer, H. Roitblat, and S. Wilson, editors, *Proc. of the Second Int'l Conf. on Simulation of Adaptive Behavior*, pages 432–441. MIT Press, 1992.

[31] Maja Mataric. *Interaction and Intelligent Behavior*. PhD thesis, Massachusetts Institute of Technology, 1994.

[32] David McFarland. Towards robot cooperation. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 440–444. MIT Press, 1994.

[33] B. Minten, R. Murphy, J. Hyams, and M. Micire. A communication-free behavior for docking mobile robots. In *Proceedings of Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[34] P. Molnar and J. Starke. Communication fault tolerance in distributed robotic systems. In *Proceedings of Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[35] L. E. Parker. Adaptive action selection for cooperative agent teams. In Jean-Arcady Meyer, Herbert Roitblat, and Stewart Wilson, editors, *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 442–450. MIT Press, 1992.

[36] L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[37] L. E. Parker. Multi-robot learning in a cooperative observation task. In *Proceedings of Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[38] Suparerk Premvuti and Shin'ichi Yuta. Consideration on the cooperation of multiple autonomous mobile robots. In *Proceedings of the IEEE International Workshop of Intelligent Robots and Systems*, pages 59–63, Tsuchiura, Japan, 1990.

[39] N. S. V. Rao. Terrain model acquisition by mobile robot teams and n-connectivity. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[40] I. Rekleitis, G. Dudek, and E. Milios. Graph-based exploration using multiple robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[41] P. RIley and M. Veloso. On behavior classification in adversarial environmentts. In *Proceedings of Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[42] S. Roumeliotis and G. Bekey. Distributed multi-robot localization. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[43] D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 235–242, 1995.

[44] D. Rus and M. Vona. A physical implementation of the self-reconfiguring crystalline robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1726–1733, 2000.

[45] Daniel Stilwell and John Bay. Toward the development of a material transport system using swarms of ant-like robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 766–771, 1993.

[46] P. Stone and M. Veloso. A layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, 12:165–188, 1998.

[47] C. Unsal and P. K. Khosla. Mechatronic design of a modular self-reconfiguring robotic system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1742–1747, 2000.

[48] P. K. C. Wang. Navigation strategies for multiple autonomous mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 486–493, 1989.

[49] Z. Wang, Y. Kimura, T. Takahashi, and E. Nakano. A control method of a multiple non-holonomic robot system for cooperative object transportation. In *Proceedings of Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[50] Gerhard Weiss and Sandip Sen, editors. *Adaption and Learning in Multi-Agent Systems*. Springer, 1996.

[51] A. Winfield. Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In *Proceedings of Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

[52] A. Yamashita, M. Fukuchi, J. Ota, T. Arai, and H. Asama. Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3d environment. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3144–3151, 2000.

[53] M. Yim, D. G. Duff, and K. D. Roufas. Polybot: a modular reconfigurable robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 514–520, 2000.

[54] E. Yoshida, S. Murata, S. Kokaji, and K. Tomita dn H. Kurokawa. Micro self-reconfigurable robotic system using shape memory alloy. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000 – this volume)*, 2000.

Intelligent Space: - Interaction and Intelligence -

Hideki Hashimoto

Institute of Industrial Science, University of Tokyo
PRESTO, JST
hashimoto@iis.utokyo.ac.jp
http://dfs.iis.u-tokyo.ac.jp

Abstract

The intelligent space is a space where we can easily interact with computers and robots, and get useful service from them. In such a space, our life could be more comfortable and we can get more satisfaction. In this paper, the technologies to achieve intelligent space are introduced and problems in using these technologies are discussed. Then I will introduce existing research relating to intelligent space. Finally, I will conclude this paper with some argument about the future of intelligent space.

1. What is making space intelligent?

Making space intelligent can be defined as a space with functions that can provide appropriate services for human beings by capturing events in the space and by utilizing the information intelligently with computers and robots [1].

Robots having partial intelligence become more intelligent through interaction with the space. Moreover, robots can understand the requests (e.g. gestures) from people, so that the robots and the space can support people effectively. This space using intelligence is called the Intelligent Space. The concept of the Intelligent Space is shown in Fig. 1. DIND (Distributed Intelligent Network Device) has a sensing function through devices such as a camera and microphone that are networked to process the information in the Intelligent Space.

The Intelligent Space can physically and mentally support people using robot and VR technologies; thereby providing satisfaction for people. These functions will be an indispensable technology in the coming consumption society.

This commentary introduces technologies to realize the Intelligent Space, and discusses an associating issue, the current research regarding the Intelligent Space and its future.

2. Distributed Intelligent Network Device (DIND)

This chapter describes the function of the Intelligent Space, especially that of the DIND.

2.1. The concept of DIND

In the Intelligent Space, the DIND understands events in the space and provides appropriate services for people by using devices such as robots, displays, and speakers. A DIND is composed of sensors, a processor for the information from the sensors, a network for information interchange, and a power source. It is microminiaturized and a low-cost device. Since networking is a prerequisite, functions such as a high level of security, self-diagnosis, and function sharing are essential [2]. Fig. 2 shows the concept of DIND. As constructed, DIND needs MEMS (Micro Electro Mechanical System) and nanotechnology for its miniaturization.

Figure 1. The Intelligent Space

DIND with MEMS Device
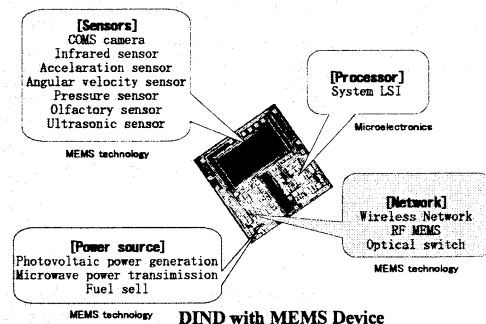
Figure 2. The concept of DIND

2.2. The basic function of DIND

Each DIND inherently has an intelligence limit because of its size, however networking decentralized DINDs in the space can realize a high level of intelligence through their autonomous cooperation.

The basic functions of networked DIND are as follows:

• Observation of events in the space

A function to observe events in the space via sensors such as visual, infrared, radio frequency and ultrasonic sensors, high-sensitivity microphone, and laser radar.

• High level processing of the obtained data

A function to process locally obtained data and to output results to the network in a sensor-independent format.

• Intelligent decision

A function to suppose the events in the space by utilizing information from networked DINDs and past data, and to make an appropriate decision (a cooperative activity in networked DINDs).

• Offering of appropriate services

A function to issue commands to robots and/or manipulators for physical support.

2.3. DIND network

The intelligence, placed spatially by DINDs, connects the physical and digital spaces, and realizes the understandings of people's intentions and the appropriate services for them. The Intelligent Space is a place where various technologies merge into one on a DIND base, and thereby evolves as a platform.

Fig. 3 shows the concept of a network system. The DIND network connects outdoor and indoor spaces seamlessly, forming a large Intelligent Space. Various and diverse data such as family healthcare, indoor child monitoring and outdoor traffic monitoring are shared and/or processed by the Intelligent Space as a platform to realize a wide variety of services.
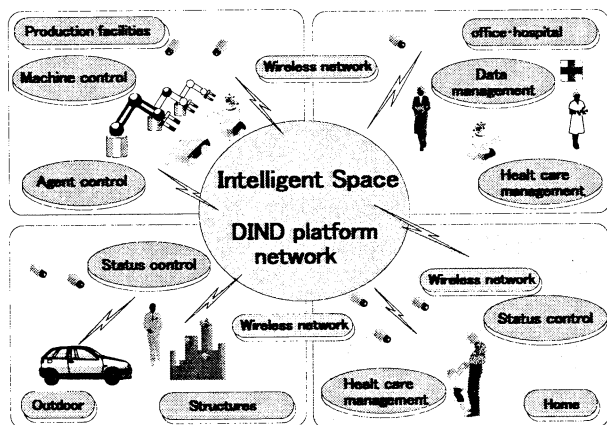
Figure 3. DIND network system

At the present time, a form of Intelligent Space could be a linkage of a network with cellular phones, microcellular phones, and mobile computing devices.

3. Elements of technology supporting the Intelligent Space

This section explains elements of the technology required for an Intelligent Space. DIND sensor and network technology, data processing technology to transmit the obtained data, and the actuator technology that is essential for robots.

3.1. Sensor technology

Sensor technology must be hypersensitive and miniaturized.

Sensors types are visual, infrared, olfactory and gustatory sensors, and microphones,.

3.1.1. Visual sensors. Current mainstream is image processing by CCD. However, CMOS camera-integrated Image Processing Chips will be the mainstream in the future and will be built into DINDs as an IP.

3.1.2. Microphones. High sensitivity microphones can distinguish human voices for fulfilling people's requests. Technologies for microphones are the capturing of voice and the locating of that person.

3.1.3. Infrared sensors. Infrared sensors are widely applicable; e.g. object detection, thermometry, communication, and image display. In object detection, for example, infrared lasers can be used to detect distance, velocity, and direction of moving objects, be applied to the safe running of automobiles and takeoff/landing control of aircraft. Infrared sensors in the Intelligent Space are used for locating, thermometry, and hygrometry.

3.1.4. Olfactory sensors. Olfactory sensors are used to measure gases in air; some of them can be used to monitor the air condition in houses and production facilities, and the quality of food. Presently the intelligent sensor that integrates many ceramic gas sensors can distinguish more than 60 kinds of odors.

3.1.5. Gustatory sensors. Unlike other sensors, gustatory sensors cannot express tastes just by measuring certain physical quantity. This technology can realize this by mimicking the actual gustatory recognition process of human beings.

3.2. Network technology

This section introduces the network technology required for the Intelligent Space.

3.2.1. MEMS (Micro Electro Mechanical System).
MEMS is a general term of micro-systems in which micro-sensors, actuators, and control circuits are integrated by micromachining technology on the basis of IC manufacturing processes [3]. This technology enables the miniaturization of mechanical systems and designs that can compact the entire system into a grain-of-sand size. This technology is also essential for the miniaturization of DINDs. Moreover, this technology will play an important role in future networks because optical switches based on MEMS technology further improve the performance of broadband using optical fiber.

3.2.2. Wearable computing.
Wearable computers, which are miniaturized and power-thrifty wearable mobile computers, have gained much attention. These computers have a potential to extend human memory and sensibilities by integrating them into future garments, and to drastically change society and culture [4]. They have already been used industrially for inspection and maintenance of aircraft and ships.

Wearable computers are network access points in the Intelligent Space and perform the function of enhancing human ability and transferring computation to the human side. For example, all of what human beings recognized and/or memorized are stored in the computer and can be read out anytime. Sensors such as GPS, cameras, microphones can be equipped with these computers.

3.2.3. Ubiquitous computing.
An ubiquitous computing consists of many computers that are allocated, so as to be invisible, in various areas of the living space so as not to hinder human activity [5]. Examples are pressure sensors in chairs and small infrared emitting badges that indicate the location of the person wearing it.

In many cases the present allocation of computations use rooms two-dimensionally, leaving the ceiling and floor spaces. Sensors and devices may be embedded in these spaces in the future, so that architectural and designing consideration needs to be given for realizing the Intelligent Space.

3.2.4. Network robotics.
Telerobotics, a technology to control robots by remote manipulation, has been researched and developed to handle hazardous materials in nuclear facilities, to work under the sea and to deal with dangerous objects such as explosives. Nowadays the application to surgical operations has been researched and tried. Network robotics is considered as a concept that links this telerobotics with the growing Internet. This technology is important for the use of robots and agents in the Intelligent Space by remote manipulation.

Telerobotics is a one-on-one system where a specific operator manipulates a specific robot via communications paths, while network robotics, as with the Internet, is a system that many and unspecified persons can use at anytime and anywhere by accessing a control server. A variety of software required for the robot system can be decentralized by leveraging the Internet. OMG (Object Management Group) is developing its specification and has proposed CORBA (Common Object Request Broker Architecture) [6].

At present, some robots are connected with and controlled on the Internet [7]. Research and development is ongoing in fields including remote health monitoring and remote manipulation; the fusion of network and robotics has just begun and further developments are expected.

3.3. Actuator technology

With the advancing intelligence of robots, functional improvement of their hardware is necessary. For this actuator technology has been essential.

Against this backdrop, practical applications of new actuators [14][15][16] that are alternatives to electric and fluid actuators are awaited.
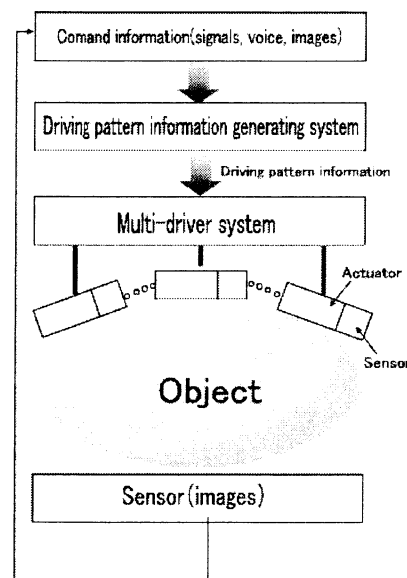
Figure 4. Schematic concept of actuator fusion

The concept of actuator fusion is proposed [8]. Fig. 4 shows the concept of actuator fusion. The concept of actuator fusion is "to realize the movements and actions requested by the coordinated performance of many actuators". A typical example of the application of the actuator fusion concept is microstructural actuators. Microstructural actuators can be made in various forms by combining many micro-sized actuator modules that are produced by micromachining. In this case, their maximum degree of freedom is equal to the number of modules. Higher degrees of freedom seem to be

associated with more control difficulties; however, a sufficient precision of force and position can be attained only by the on-off control of each module. This is one advantage of using micro-modules. The goal in the development of microstructural actuators is to produce robots consisting of an interior main skeleton and an outer coat surfaced completely with micro-modules.

3.4. Other technologies

This section explains other technologies related to the Intelligent Space.

3.4.1. Interface technology. Interfaces between human beings and the Intelligent Space may be voice and gestures that are natural for human beings, with a five senses based interface such as VR technology. An interface utilizing brain waves has a potential to be developed as the ultimate interface. Our laboratory is investigating the interface that returns a sense of force to human beings by using a haptic interface [9].

3.4.2. Remote learning. Remote learning using cameras and microphones has begun however current applications are by simple touch panels, remote images, and their explanations. The Intelligent Space can realize higher level of support by DIND, VR, and network technologies. For example, in the remote learning of sports, DIND can closely monitor the actions of users and give appropriate advice in real time. The training of actions requiring the sense of force such as remote work, operation, and manipulation can be experienced in the VR space using a haptic interface that can transmit the sense.

Moreover, DIND and VR technology in the Intelligent Space can extend a feeling of shearing the space for virtual lectures and/or meetings.

3.4.3. Recognition technology. The present interaction between human beings and computers is conducted mainly via a keyboard and/or mouse, however a simpler and more natural way is needed in the Intelligent Space. For that purpose, in the Intelligent Space where both verbal communication and nonverbal information (expressions and gestures) are needed to respond to requests from people in the space, to recognize the speaker by the human recognitive function of a camera, and further to identify the location of the speaker only by the conversation is necessary.

Expression recognition presently targets just the 6 basic expressions (surprise, fear, aversion, anger, happiness and sadness). In the recognition of gestures, there are many problems, including mix-ups between one gesture and another, and between an unconscious gesture and a certain request, due to the low competent of computer vision. Further studies are expected to solve most of the problems associated with human interfaces. However, no matter how much performance

is improved, it is doubtful that it will become perfect. Therefore the correction and addition of information is simultaneously needed, as the circumstances demand, from the closer interactions that are generally observed among human beings.

3.4.4. AI (Artificial Intelligence) technology. Artificial intelligence (AI) is a research area to realize machines having powers of understanding and intelligence, however AI like "Astro Boy" and "HAL9000" is far from being a reality at this stage; many AI researchers dedicate themselves to the development of machines that have only limited expert knowledge [10]. The expected application of AI in the Intelligent Space is the development of agents that can comprehend natural language as an interactive interface, understand images for object recognition, and work intelligently and coordinately. The agent here is an autonomous object that can solve problems cooperatively with other agents. Human beings and robots can be included in these agents in a broad sense. The interactive interface using natural language comprehension and the image recognition using image understanding in the Intelligent Space can be realized to some extent by the cooperation of many agents.

4. Present state of the Intelligent Space

This chapter introduces some examples of research into the Intelligent Space, apart from ours.

4.1. EasyLiving

The EasyLiving project by Microsoft is developing architecture and technologies for the intelligent environment [11]. This intelligent environment is defined as the space that provides access to information and services for users in the space. The purpose of this project is to develop an architecture that provides easy and user-friendly technology utilizing various devices. Their research covers software to support many decentralized devices called middleware, the geographic modeling in order to specify the device that users want to use, and the format to describe perception and services.

4.2. Intelligent Room

The Intelligent Room studied by MIT is an experimental environment to realize a natural interaction between human beings and computers by embracing computation in daily life activities [12]. Many computer visions and voice/gestures recognition systems are installed here to locate people and to detect what they are doing and saying in the room. The user interfaces in this system are not a mouse or keyboard but gestures, voice, and actions. A purpose of this

research is to make computers user-friendly, and moreover, essentially invisible to the users.

4.3. Oxygen Project

The Oxygen Project studied by MIT aims to develop computation systems that can be used anywhere, as if we took in oxygen from air [13]. In the future, people will not need to have a personal device, and will be able to use computation services anywhere by utilizing devices integrated into the environment.
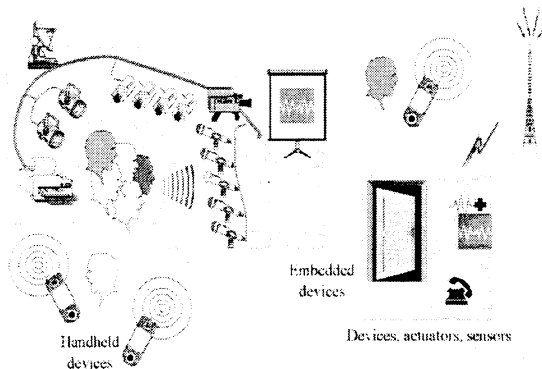
Figure 5. Oxygen Project

Fig. 5 shows the concept of the Oxygen Project. Natural and simple interactions including voice and gestures are practicable, so that people do not need to use a keyboard/mouse and to learn how to use the computer. Computer use is easy.

4.4. Interactive Workspace Project

The Interactive Workspace Project by the University of Stanford is researching the potential of cooperative working between human beings and computers/interactive devices in the space integrated with many technologies [14]. There are PDAs and notebook computers connected by wireless LAN, high-resolution displays built in walls or placed on tables in this space. Specialized I/O devices such as LCD tablets, microphones, and active cameras are also positioned.

4.5. Robotic Room

"In the cooperation between human beings and machines/robots, machines/robots must compliment the weak points of human beings and work for people's convenience." From this point of view, Robotic Room is studied to realize a natural support by using a room as a robot, sensing the people's living activity and performing actuation for people [15]. This room treats the behavior of human beings as media and is intended to support people on the basis of media. This system is intended to study a robotic sickroom in order to give wide support to people in medical and welfare areas, and more recently in daily life.

4.6. Smart Dust

The University of California, Berkeley, is developing a miniaturized device based on MEMS technology mainly for sensing and information transmission [16] (See Fig. 6). The function objectives are:
• Sensing function for the temperature, humidity, air pressure, light intensity, inclination/oscillation, and magnetic field.
• Two-way communication (within 20 meters)
• Long-distance information transmission by laser
• Micro processing
• Power source (for a week of continuous use.)
• Miniaturization (It was miniaturized to 100 mm^3 in July 1999, via "macro motes (a 10-yen coin size)"; however it is still functionally less than perfect. It should have been miniaturized to 1 mm^3 by July of this year.)

Smart Dust Components
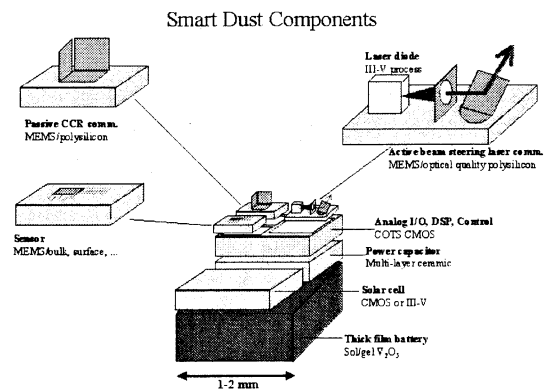
Figure6 : Smart Dust

Applications of Smart Dust include:
1. Military use (e.g. to search for enemy and their deployment on the battlefield)
2. Virtual keyboard (an input device that senses the finger movement via Smart Dust worn on the finger tip).
3. Quality control of items and inventory management in production facilities and warehouses
4. Smart Office (to maintain comfortable temperature by attaching Smart Dust to each person's clothes, measuring body and ambient temperature and communicating with the air conditioner in the space. A project adopting the wireless technology to measure and control the environment in a building is now in progress in the Center for Built Environment in Berkeley and Smart Dust is planned for use as a device to realize this project.)
5. For physically handicapped people or in the places with low visibility

5. Conclusion

The Intelligent Space is a platform where various technologies can be fused on the basis of network technology [17]. Thus an Intelligent Space that adopts and adapts technologies in various ways will be formed in the future. Once the core of the Intelligent Space (network technology, MEMS, and nanotechnolgy) has been highly developed, DINDs will be scattered worldwide, and a global intelligent space will be able to be realized. Indoor and outdoor systems will be fused seamlessly by networks and everyone will be able to instantly access anywhere in the world, though physical distance will still exist.
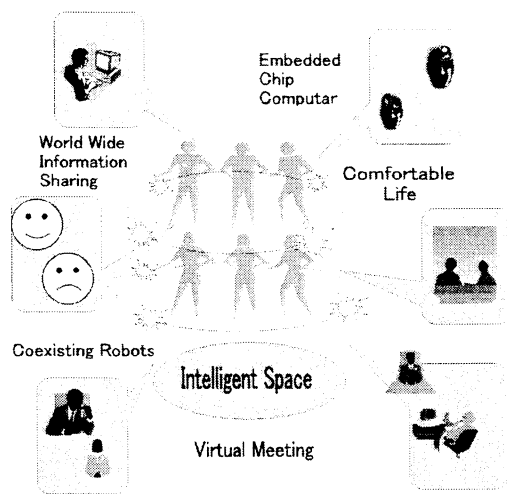
Figure 7. The Intelligent Space in the future

The Intelligent House itself will be a center of life and work: people in the Intelligent Space can attend realistic virtual meetings and lectures without going to the office or school. This is a primary role of the Intelligent Space linking physical and digital spaces. Human type and/or pet type robots will be in each home according to their specific roles. People's life should be more comfortable (Fig. 7)

References

[1] Hideki Hashimoto , "Intelligent Space" (in Japanese), Proceedings of 36th SICE'97, 103C-8, pp.45-46 , 1997.7

[2] Hideki Hashimoto, Naoto Kobayashi, Toru Yamaguchi, "Intelligent Interactive Space" (in Japanese), The Japan Society of Mechanical Engineering, Robotics and Mechatronics, 2BI4,1998

[3] http://www.mems-exchange.org/, "MEMS Exchange"

[4] Rei Inasaka, "Wearable Computer" (in Japanese), Journal of the Virtual Reality Society of Japan Vol.5, No.1 pp.13-16, 2000.1

[5] http://www.ubiq.com/hypertext/weiser/UbiHome.html, "Ubiquitous Computing"

[6] http://www.omg.org/

[7] S. Goldberg, G. Bekey, Y. Akatsuka, "DIGIMUSE:an interactive telerobotic system for remote viewing of three-dimensional art objects", Preprints of IEEE IROS'98 workshop 'Robots on the Web',pp55-59,1998

[8] Toshi Takamori, "What is Claimed to the Actuators in Robotics, Now -Hoping for the New Actuator-" (in Japanese), Journal of the Robotics Society of Japan , Vol.15 No.3, pp.314-317, 1997

[9] Noriaki Ando, Kouhei Gonda, Masahiro Ohta, Hideki Hashimoto, Taiichi Kusano, Isao Kamiyama,"Micro Work Cooperative System with Haptic Interface" IFAC-Conference on MECHATRONIC SYSTEMS (Mechatronics 2000), Vol.3, pp.1085-1090, 2000.9, Darmstadt, Germany

[10] Kenji Sugawara, "Artificial Intelligence" (in Japanese), Morikita Shuppan. 1998

[11] B. Brumitt, B. Meyers, J. Krumm, A. Kern and S. Shafer, "EasyLiving: Technologies for Intelligent Environments", Proceedings of the International Conference on Handheld and Ubiquitous Computing, September 2000

[12] Michael H. Coen MIT Artificial Intelligence Laboratory, "A Prototype Intelligent Environment", http://www.ai.mit.edu/projects/hci/

[13] http://oxygen.lcs.mit.edu/

[14] http://graphics.stanford.edu/projects/iwork

[15] T. Sato, Y. Nishida and H. Mizoguchi: Robotic room: symbiosis with human through behavior media; Robotics and Autonomous Systems, Vol.18, pp.185-194 1996

[16] http://www-bsac.eecs.berkeley.edu/~pister/SmartDust/

[17] Hideki Hashimoto, "Development of Intelligent Mechatronics" (in Japanese), Journal of the Japan Society for Technology of Plasticity Vol.40 , No.456 pp.18-24 , 1999.1

Constructing artificial life in the laboratory

S. Rasmussen (Los Alamos National Laboratory and
Santa Fe Institute, USA)

The Inside Story on Systems, Minds and Mechanisms

JOHN L. CASTI

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

AND

Santa Fe Institute
Santa Fe, NM 87501, USA

Abstract

Do the laws governing a system look different when you're inside the system from those you see when you look at the system from the outside? This is the central question of what's come to be termed, "endophysics." Here we examine this question for the case when the system is the human mind. More specifically, we consider the problem of "strong" AI, which asks if a computing machine can *duplicate* the cognitive capacity of the human mind—in principle, at least. Looked at from the outside, this question reduces to the familiar Turing test for a thinking machine. But from the endophysics point of view, the matter becomes far more problematical, leading to some of the strongest critiques against strong AI.

Following consideration of the strong AI problem, the paper concludes with a discussion of the issue of a system observing itself. Endophysically speaking, this situation leads immediately to all the familiar problems associated with self-reference, tangled loops and paradox, both logical and geometric. Our final conclusion is that the only way to break out of these loops is to think endophysically, which means recognizing explicitly that, generally speaking, the laws of nature **do** look different depending upon where you stand.

Fantastic Voyages

ASIDE FROM SERVING AS THE CINEMA DEBUT OF RAQUEL WELCH, the 1966 film *Fantastic Voyage* was notable for its stunning special effects depicting what the inside of the human body might look like if you were the size of a blood cell. Interesting as the film was from the standpoint of speculation about how the human body might look from the inside, this experiment in "human nanotechnology" missed a golden opportunity to ask

what today we might term the fundamental question of endophysics: Are the laws of the system any different for an observer standing **inside** the system (e. g., the human body) than for an observer looking at the system from the outside? Put more prosaically, would Raquel Welch in her form-fitting white jumpsuit see a different law governing the flow of electrical impulses in the brain than her full-sized human counterpart would see measuring those same impulses from the outside with, say, a microtip electrode?

So there is a fundamental question to be settled here between "physics from within" and "physics from without," a distinction that reflects one of the most basic dichotomies in nature and in life: the difference between being inside and outside something. To fix the general idea as it shows up in science, let's briefly consider a few examples.

• *Relativity Theory*—when asked how he came to discover the theory of relativity, Einstein replied that he thought about how he would see the world if he were riding on a beam of light. This is a perfect example of endophysical thinking, which in this case led Einstein to the startling conclusion that physics *would* look different if you were on a light beam. Specifically, you would see yourself traveling at with a definite, finite velocity, which in turn would engender all the by now familiar time and space contractions in other systems that you observe *outside* the light beam. But from the light beam itself, you see nothing unusual—just "normal" speed-of-light movement from one location to another.

• *Gödel's Theorem*—the key message of Gödel's famous incompleteness result is that within any consistent, logical deductive system powerful enough to express any statement about whole numbers, there are statements that can be made—but which cannot be proved either true or false using the rules of inference of that system. Nevertheless, by jumping outside the system ("jootsing"), we can see that such statements are actually true. They just cannot be proved using the rules of inference contained in that logical framework. From an endophysics point of view, what Gödel is saying is that arithmetic is incomplete—when looked at from the inside. But it can be completed if we look at the same system exophysically. So the laws of arithmetic do look different, depending on whether you look at them from inside or outside the system.

• *The Genetic Code*—one of the seminal achievements in modern molecular biology was the unraveling of the genetic code by Francis Crick and Sydney Brenner in the late 1950's. This code translates three-letter combinations of the nucleotide bases A, G, C and T into the twenty amino acids that make up the proteins forming all known life forms on this planet. The translation process begins by the passive copying of a fragment of the DNA chain into a strand of messenger RNA (mRNA). This mRNA is then "read" by a ribosome that moves along the strand much like a tape-recorder head traveling along a reel of tape. As it moves from one end of the mRNA strand to the other, the ribosome reads the symbols written on the strand of mRNA, three letters at a time, capturing the called-for amino from pieces of transfer RNA floating about in the cellular cytoplasm. This, then, in a nutshell is a statement of the processes of genetic transcription and translation.

Now let's mimic Einstein and think endophysically about this situation. Suppose we hitch a ride on the ribosome as it passes from one end of the mRNA strand to the other. What would we see? Well, not much. More precisely, what we certainly would *not* see is any clear-cut distinction between the operations of passive copying in the transcription phase and meaningful decoding in the translation mode. In both cases, all that would be seen from a "molecule's eye view" would be purely syntactic chemical operations. Only by jootsing can we see the semantic distinction between the two operations. From the inside, there is only biochemistry; from the outside, there is a whole world of meaningful proteins.

• *Behavioral and Cognitive Psychology*—in the 1920's, John Watson made the radical suggestion that human behavior patterns do not have mental causes. Put more precisely, Watson's assertion was that it is unscientific to assert the existence of unmeasurable mental states of the brain that give rise to observable behavior. Thus arose the "behavioral" school of psychology, which focuses on externally-observed input/output or stimulus/response behavior patterns as the raw material upon which theories of the mind and behavior are to be based.

By way of contrast, cognitive psychologists claim that it is only by postulating internal states of the brain that we have any hope of constructing a genuine scientific theory of human behavior. The cognitive view asserts that it is the job of the psychologist to infer

somehow the nature of these internal states from observed behavior, and to then construct a predictive theory of human behavior on the basis of the relationships linking the internal states of the mind.

So from an endophysical standpoint, we see the hardware of the brain reconfiguring itself into different states. These states, in turn, give rise to thoughts and behaviors. But exophysically speaking, there are no such states upon which to base the "laws of thought." This distinction mirrors perfectly competing claims in the artificial intelligence game, in which we have pro-AI forces arguing what, in effect, is the exophysical position that if it behaves like a brain, then it **is** a brain. Anti-AI proponents reply that you can't judge a brain by its cover. Since this topic is of considerable interest at the moment, let me take a few pages to sketch more fully these diametrically opposed positions.

Into the Heart of the Mind

IN 1950, ALAN TURING PUBLISHED THE PAPER "Computing Machinery and Intelligence," which sparked off a debate that rages to this day: Can a machine think? In addition to its seminal role in drawing attention to the question of machine intelligence, Turing's paper was notable for its introduction of an operational test for deciding whether or not a machine really was thinking—human-style. This criterion, now termed the "Turing test," is unabashedly behavioristic in nature, involving the machine fooling a human interrogater into thinking it is actually a human. Turing's rationale for proposing what he called the Imitation Game, was that the only way we have for deciding whether or not other humans are thinking is to observe their behavior. And if this criterion is good enough to decide if humans are thinking, then fairness to machines dictates that we should apply the same criterion to them.

Amusingly, on November 8, 1991 the Boston Computer Museum held the world's first hands-on Turing test, in which eight programs conversed with human inquisitors on a restricted range of topics that included women's clothing, romantic relationships and Burgundy wine. At the day's end, the judges awarded first prize to a program called *PC*

Therapist III, which was designed to engage its questioner in a whimsical conversation about nothing in particular.

For example, at one point the program suggested to a judge: "Perhaps you're not getting enough affection from your partner in the relationship."

The judge replied, "What are the key elements that are important in relationships in order to prevent conflict or problems?"

"I think you don't think I think," responded the terminal.

This kind of interchange did little to fool the judges, most of whom said they were able to spot the "commonsense"-type of mistakes that were an immediate giveaway singling out the computer programs from the humans. Nevertheless, the overall conclusion from this historic experiment is that perhaps the Turing test isn't as difficult as many people originally thought, since even the primitive programs in this contest managed to fool some of the judges.

From an exo- vs. endo-physics point of view, it's clear that the Turing test represents an exophysics perspective on human intelligence. Standing outside the system, the test is designed to discern human intelligence in a machine by observing only the behavioral output of the machine. The Turing test says nothing about the internal constitution of the machine, how its program is structured, the architecture of the processing unit or its material composition. In Turing's view of intelligence, only behavior counts. And if you have the "right stuff," then you are a thinking machine.

Pro-AI enthusiasts have developed two quite different approaches to the creation of human intelligence in a machine, lines of attack that we might conveniently label "Top-Down" and "Bottom-Up." It's instructive to examine the basics of each approach, as they shed some light on the question of the "laws of thought" from both an exo- and an endo-physical perspective.

– *Top-Down AI* –

The central thesis of Top-Down advocates is that the essence of human intelligence does not depend on the material composition and structure of the human brain. Rather, human

cognitive capabilities are the result of rule-based symbol processing, and that any material substrate in which the symbols can be stored and manipulated is as good as any other. Put compactly, matter doesn't matter. Consequently, the primary thrust of the Top-Down research program is directed toward identifying the symbols and rules that the brain uses, and then translating those objects into corresponding symbols and algorithms within the confines of a digital computer. In short, the program tries to "skim off" the rules of thought and re-create those rules in a machine. Most importantly, Top-Downers claim that there need be no similarity at all between the internal architecture of the brain, i.e., the neuronal wiring diagram, and the architecture of the computer. This is clearly an exophysical perspective on the matter of what constitutes human intelligence.

Sad to say, the Top-Down view of the world has not been faring too well of late, despite its air of initial plausibility. The problem appears to be that even if there is such a thing as a high-level rule of thought responsible for shoving symbolic structures around in the brain, it's devilishly hard to distill the essence of that rule into a computer program. The human mind seems to be capable of all sorts of "slippage" in its cognitive contemplations, and this slippage from one category to another has so far proved to be an elusive will-'o-the-wisp to Top-Down researchers.

A good example of this slipperiness arises in Top-Down attempts to mimic human language translation. Put crudely, the Top-Down approach to the problem involves stuffing large dictionaries of the source and target languages into the machine, along with extensive rules of grammar, idiomatic phrases, exceptions to the rules and so forth. And while some of these efforts have resulted in translation programs that do a fair job of translating material involving very restricted contexts like technical manuals or academic research specialties, the performance of programs designed for translating general source material like a daily newspaper or an everyday conversation is pretty abysmal. For example, one early Russian ↔ English program translated the English idiom, "Out of sight, out of mind," into Russian and then re-translated the Russian "equivalent" back into English. What resulted was the shocker, "Blind and insane"!

Thus does the Top-Down approach flounder on the commonsense barrier noted in the Boston Computer Museum Turing test experiment. In thinking about something, humans never see a context-free situation. So, for instance, if I think about a key, I never see the key as simply a piece of machined metal. Rather, it always comes accompanied by a context: my front door, a safe, an automobile or maybe even a machined piece of metal if the context were my local locksmith's shop. But any thought I have of the key is always accompanied by some context. And it's just this kind of "context attachment" that Top-Down AI programs seem to lack. Despite more than thirty years of looking by Top-Down advocates there appears to be no easy way to identify high-level rules that can be skimmed off from human behavior in order to supply this sine qua non of human thought. And so we turn away from this exophysical attack on the thinking machines question to look at the situation endophysically.

– Bottom-Up AI –

Looking at the brain from the other end of the telescope, Bottom-Up proponents argue that the physical hardware of the brain does matter when it comes to human cognition. And if we're to have any hope of duplicating that kind of intelligence in a machine, it behooves us to explicitly account for the structure of that hardware in our programs. What this means is that we need to take an "insider's" perspective, looking at how the brain is actually wired up and how that structure serves to generate the kind of observable behavior we deem "intelligent."

The features of a machine characterizing a Bottom-Up approach to the strong-AI question are that the machine is: (1) composed of a large number of relatively primitive processors, that are (2) operating in parallel and that are (3) relatively unprogrammed. So instead of dictating the rules of thought by fiat, a Bottom-Up man or woman sees whatever rules of thought the machine embodies as arising in an emergent way from the re-configuration of the pattern connecting the many processors as the machine learns to survive in its environment.

In this essentially endophysical view of cognition, the machine starts out initially with very little knowledge of the world, but with a hardware configuration that's pretty plastic. As time goes on and the machine interacts with its environment, certain patterns of thought prove more effective at solving the problems of daily life than others. These successful patterns then ultimately get wired into the connections linking the machine's processors. It is these connections then that determine whatever rules we can say the machine uses in arriving at its actions and decisions.

The one thing that both Top-Down and Bottom-Up devotees are in agreement on is the claim that there is no obstacle, in principle, to the duplication of human cognitive capacity in a machine. The point of dispute revolves only about how to actually go about doing it—exophysically or endophysically. Interestingly enough, the arguments of the anti-AI forces can also be separated into the same exo- and endo- categories. Let's briefly see how.

– Phenomenological Anti-AI –

One of the most popular spokesmen for the anti-AI cause a few years back was Hubert Dreyfus, a philosopher at the University of California at Berkeley. Dreyfus, along with his brother Stuart, also a professor at Berkeley, argued against the possibility of strong-AI by appealing to the works of the phenomenological philosophers Heidegger, Husserl and Merlau-Ponty. These giants of modern continental philosophy claimed that there are many human cognitive activities that simply cannot be thought of as the following of a set of rules. A favorite example of the Brothers Dreyfus in this regard involves learning how to drive an automobile.

According to the Dreyfuses, gaining expertise at driving a car involves a successive passage through five identifiable stages:

• *Novice:* At this lowest skill level, context-free rules for good driving are acquired. Thus, one learns at what speed to shift gears and at what distance it's safe to follow another car at a given speed. Such rules ignore context-sensitive features such as traffic density and weather conditions.

- *Advanced beginner:* Through practical on-the-road experience, the novice driver learns to recognize concrete situations that cannot be described by an instructor in objective, context-free terms. For instance, the advanced beginning driver learns to use engine sounds as well as the context-free speed as a guide for when to shift gears, and learns to distinguish the erratic behavior of a drunk driver from the impatient actions of an aggressive driver in a hurry.

- *Competence:* The competent driver begins to superimpose an overall driving strategy upon the general rule-following behavior of the novice and the advanced beginner. He or she is no longer merely following rules that permit safe and courteous operation of the car, but drives with a goal in mind. To achieve this goal, the competent driver may now follow more closely than normal, drive faster than is allowed, or in other ways depart from the fixed rules learned earlier.

- *Proficiency:* At the previous levels, all decisions were made on the basis of deliberative, conscious choices. But the proficient driver goes one step further and makes decisions on the basis of a feel for the situation. There is no deliberation; things just happen. So, for example, the proficient driver when attempting to change lanes on a busy freeway may instinctively realize that there's another car coming up on the blind side and delay making a move. This instinctive reaction may arise out of experience in similar situations in the past and memories of them, although it may appear as an unexplainable "lucky guess" to an outside observer. Somehow there is a spontaneous understanding or "seeing" of a plan or strategy.

- *Expert:* An expert driver no longer sees driving as a sequence of problems to solve, nor does he or she worry about the future and devise plans. He simply becomes one with his car, and experiences himself as just driving rather than as driving a car. Thus, an expert driver has an intuitive understanding of what to do in a given setting. He doesn't solve problems and he doesn't make decisions; he just does what normally works.

The moral of this fable in five parts is that there is more to intelligence and expertise than mere calculative rationality. Expertise doesn't necessarily involve inference; the expert

sees what to do *without* applying rules. This is the essence of the Dreyfus argument against the possibility of a rule-based program's ever achieving anything that even remotely approximates genuine human intelligence.

I think it's clear from this example that the Dreyfus position against strong-AI is essentially an exophysics-based argument against rule-based behavior. The claim is that simply by looking at the external behavior of a human being, we can see cognitive activity that cannot be the result of following a set of rules. Thus, the Dreyfuses would certainly claim that it would be impossible to program a computer to drive a car in a manner indistinguishable from an expert human driver. In short, there are no high-level rules that human drivers follow in guiding their vehicles through traffic congestion and high-speed freeway traffic. And since machines can only follow the rules encoded into their programs, the full experience of even such a relatively simple human task as driving a car cannot be captured within the confines of a computer program. Ergo, machines cannot think.

As an aside, it's of some significance here to note that Hubert Dreyfus admitted to me in a private conversation that the main thrust of his exophysics-based argument is directed against the Top-Down approach to strong-AI, and that it may well be possible to duplicate human capacity in driving and everything else by following a Bottom-Up approach. But to avoid an unnecessarily long digression into how this might be done, let's pass on to consideration of endophysically-based arguments against strong-AI.

– Gödelian Anti-AI –

We have already spoken of the inherently endophysical character of Gödel's famous incompleteness theorem. One of the most influential arguments against the possibility of strong-AI was advanced in the 1960's by Oxford philosopher John Lucas, who appealed to Gödel's result saying, in effect, that since there exist arithmetic truths that we humans can see to be true but that a machine cannot prove, the capacity of the human mind must transcend that of any machine. This is clearly an endophysical argument, in which Lucas looks at the machine *from the inside,* concluding that from that inside perspective the machine has limitations that do not exist for the human mind. Recently, Roger Penrose,

another Oxford don, appealed to much the same line of argument in his well-known book *The Emperor's New Mind,* to also conclude that machines cannot think like you and me.

Since the publication of Lucas's paper in 1961, the arguments have raged hot and heavy against his line of reasoning, and I don't want to bore the reader by going into them again here. They can all be found in the sources cited in the References. Let it suffice for now to say only that Gödel's theorem involves certain hypotheses, most importantly that the formal system (i.e., computer program) be logically consistent. It's not at all clear that this condition is satisfied in the case of the human mind, as I think we can all remember instances when we behaved in a demonstrably inconsistent manner. And if the system is inconsistent, all bets are off as far as appeals to Gödel's result are concerned.

Perhaps the strongest argument yet launched by the philosophers against strong-AI is that given by John Searle, a colleague of Hubert Dreyfus from Berkeley. Searle's argument is essentially a Gödelian appeal, as well, based as it is on the endophysical idea that what goes on inside a computing machine when it moves symbolic representations around in accordance with a program is pure syntax. But, Searle argues, no amount of syntax, i.e., symbol shuffling, can ever give rise to semantics. In other words, the computer can have no understanding of the *meaning* of the symbols it manipulates. And without meaning there is no intelligence.

To dramatize this endophysical perspective, Searle constructed the colorful analogy of what is now called the "Chinese Room" test. This involves imagining a man ignorant of Chinese, who is locked up in a closed room with a dictionary that contains only Chinese ideographs and a set of cards each of which has a Chinese character printed on it. The man receives cards having Chinese characters printed on them through a slot in the door to the room. He then looks up the character in his dictionary, and passes back out through the slot the card containing the character called for by the dictionary. So from the perspective of the man in the room, there is no understanding of Chinese here at all, i.e., there is no semantics. There is only pure syntactic shuffling of cards in and out through the slot in accordance with the rules dictated by the dictionary. But from the third-person perspective of a native Chinese speaker outside the room, the sequence of cards in and out of the slot

may well represent a perfectly meaningful written conversation in Chinese about, say, tomorrow's weather, the state of the stock market or the end of the world. Searle's point here is that the actions of the man in the room duplicate exactly what happens inside a computer as it goes about its business of transforming input symbol strings into output strings.

Just as with Lucas's appeal to Gödel, Searle's arguments have been subjected to detailed scrutiny by the pro-AI community, a summary of which can be found in the books and papers noted in the References. For our purposes here, what's important about Searle's claim is that it shifts attention from the outsider's, third-person perspective of the Turing test to the insider's, first-person view given by thinking of what it would be like to be inside the computer pushing symbols around to the dictates of a program.

From this extended discussion of the strong-AI question, we see that it's perfectly possible to imagine an investigator inside a system seeing an entirely different set of laws of nature than an observer looking at the same system from the outside. But so far we've considered only the case when the observer is distinct from the system being observed—regardless of whether the observing is done from the inside or the outside. But what if these two systems are one and the same? Can a system observe itself from both the inside and the outside? And, if so, does it see itself in two different ways? The next section briefly considers this point.

Tangled Loops

THE ANCIENT PARADOX OF THE LIAR, ONE VERSION OF WHICH IS "This sentence is false," serves as the prototypical example of the kind of logical difficulties that arise when a system looks at itself. One of the easiest ways to see what's going on with such "tangled loops" is to consider the case of a self-amending law.

Article V of the US Constitution gives conditions under which the Constitution itself may be amended. Specifically, it states that whenever at least two-thirds of both the House and the Senate agree, a consitutional amendment may be enacted, subject to ratification by three-fourth's of the state legislatures. Now consider whether Article V may authorize

its own amendment or repeal. Can a rule that allows the changing of other rules also admit its own change? Note that this is an "endo-paradox," in that it pertains to changing the rules of the system by using a rule which itself is part of the system.

Looking at this situation a bit more closely, we discover that it is a legal version of the so-called *paradox of omnipotence,* involving the issue of whether the Congress has the power to make any law at any time. If the Congress does have such a power, then can it limit its own power to make law? If it can, then it can't—and conversely. So in the legal version we can say that either there is a law that the Congress cannot make or a law that it cannot repeal.

Clearly, if we allow the assumption of an omnipotent Congress to stand, then we come to a genuine paradox. But such a postulate implies the affirmation and the negation of the idea that Congress can limit its own power irrevocably. Thus, because this assumption implies a contradiction, it is false. Moreover, its falsehood does not imply its truth. Consequently, we can call the postulate false with finality.

By the view that such an omnipotent Congress cannot exist as defined, a clause that authorizes its own amendment or that acutally limits itself by self-amendment is a contradiction. This leads us to conclude that amendment clauses are immutable except by illegal or extra-legal means like a revolution. In other words, they can only be amended if we take an "exo-legal" view of the situation, allowing, in effect, the system to examine itself from the outside.

The foregoing legal "tangle" is symptomatic of a large number of paradoxes that arise when a system looks at itself from the inside, and which can only be unravelled by jootsing. Gödel's Theorem is another instance of this kind of situation. These sorts of tangled loops are taken up in much greater detail in volumes cited in the References. Now let's try to wrap up this brief excursion into the inner and outer worlds of logical and physical systems.

Eternally Unbridgeable?

IN HIS IMMENSELY ENTERTAINING SHORT STORY *The Seventh Sally, or How Trurl's Perfection Led to No Good,* Stanislaw Lem tells the tale of how the robot Trurl takes

pity on an exiled despotic king, building him a complete miniature world as a substitute kingdom upon which to exercise his diabolical whims. Upon returning to his homebase, Trurl recounts this act of charity to his robotic comrade-in-arms Klapaucius, who sees Trurl's action as anything but kindly and charitable. The ensuing debate between the two constructing robots captures admirably the distinction between what you see if you're inside the system as opposed to what you see if you look at it from the outside.

In their debate, Trurl took the outsider's, exo-view, regarding the inhabitants of his constructed world as being mere toys for the king to push around at his leisure. The basis of his argument with Klapaucius was that while the people of this artificial world gave every appearance of being living, thinking, feeling beings, this was just an illusion. In actual point of fact, the people of Trurl's world were just very clever toys, with no more humanness than an extremely sophisticated doll. Klapaucius, of course, looked at the world from the insider's, first-person perspective, arguing that if the beings did not have a full complement of genuine feelings and emotions, then the entire exercise would be tainted. Basically, Klapaucius argued that if the king didn't believe that his subjects were really feeling pain when he tortured them or joy when he declared a national holiday, then Trurl's efforts would have been in vain. And since the inhabitants of Trurl's world gave every appearance of possessing these traits, Klapaucius concludes that we must believe that they are completely human in every respect but size. Hence, Trurl's good intentions led to no good—at least for the inhabitants of the miniature world.

Since there is really no logical way to break out of this impasse, Lem's story is a particularly graphical way of coming to the conclusion that how you see things really does depend on whether you're inside or outside the system. There is no such thing as a universally valid law, independent of your vantage point. As the quantum physicist John Archibald Wheeler once put it, "the only law is that there is no law."

References

The first explicit consideration of what we would now call "endophysics" appears to be in the work of James Clerk Maxwell. For a discussion of Maxwell's ideas, as well as for an account of subsequent developments in the endophysics business, see the paper

Rössler, O. "Endophysics," in *Real Brains, Artificial Minds*, J. Casti and A. Karlqvist, eds., Elsevier, New York, 1987, pp. 25–46.

The term "endophysics" appears to have been introduced by the quantum physicist David Finkelstein, who also gave an early example of a finite-state machine (computer) whose internally evaluated state is different from that existing exophysically. For an account of this device, see the article

Finkelstein, D. and S. Finkelstein. "Computational Complementarity," *Intl. J. Theoretical Physics,* 22 (1983), 753–779.

For a complete summary of the artificial intelligence debate, together with copious references to the competing claims of the various schools of thought, see

Casti, J. *Paradigms Lost: Tackling the Unsolved Mysteries of Modern Science,* Morrow, New York, 1989 (paperback edition: Avon, New York, 1990).

The Turing test conducted by the Boston Computer Museum is described in some detail in

"Can Machines Think? Humans Match Wits," *The New York Times,* November 9, 1991.

An extensive discussion of self-reference, in general, as well as the problem of self-amendment, in particular, is found in the volumes

Self-Reference: Reflections on Reflexivity, S. Bartlett and P. Suber, eds., Martinus Nijhoff, Dordrecht, Netherlands, 1987,

Suber, P. *The Paradox of Self-Amendment,* Peter Lang, New York, 1990.

For a full account of other types of tangled loops and self-referential paradoxes, the following popular books by Douglas Hofstadter are hard to beat:

Hofstadter, D. *Gödel, Escher, Bach: An Eternal Golden Braid,* Basic Books, New York, 1979,

Hofstadter, D. *Metamagical Themas,* Basic Books, New York, 1985.

The tale of Trurl's miniature kingdom is found in

Lem, S. "The Seventh Sally, or How Trurl's Perfection Led to No Good," in *The Cyberiad,* Martin Secker and Warburg, London, 1975.

Another Lemian gem in this same connection is the story

Lem, S. "Non Serviam," in *A Perfect Vacuum,* Martin Secker and Warburg, London, 1979.

Quantifying Darwinism: methods for measuring adaptive innovations in artificial and natural evolving systems

M. Bedau (Reed College and University of Oklahoma, USA)

Mobile Robots' Intelligent Control in Autonomous Decentralized FMS

Hidehiko Yamamoto
Mechanical and Systems Engineering,
Gifu University
1-1 Yanagido, Gifu, Japan, 501-1193

Yoshiyuki Tsujimoto
Graduate School of System Engineering,
Wakayama University
930 Sakaedani,Wakayama, Japan, 640-8570

Abstract

In this paper, we propose the hypothetical reasoning for the production plan of a small and medium-quantity system used for multiple varieties, in FMS factory, by an autonomous decentralized system. And a path planning, which drives AGVs to avoid mutual collisions in the autonomous decentralized FMS factory, is needed. Therefore, we propose a system, which senses possibility of this mutual collision, and avoid it by itself. That's to say, the on-line process.

1 Introduction

In order to manufacture the products which satisfy consumers demand, the products need not only a function but also the added value like a design. Under these circumstances, the factory automation system has been changed from the mass production for a single variety into the small and medium production for multiple varieties.

In this paper, we deal with Flexible Manufacturing System (FMS) as a factory automation system of the small and medium production for multiple varieties. Especially we develop the robots moving control in an autonomous decentralized FMS. The autonomous decentralized FMS doesn't need a central control and has the flexibility for a factory layout or a production scheduling.

The robots moving among Machining Centers (MCs), and mobile robot are in danger of mutual collision. In order to avoid the collisions, a sensor has been given for each moving robot and the host computer has been watching all actions and directions of the robots. This method is too heavy a burden for a host computer. In order to solve the problem, we propose the system that robots foresee the possibility of the mutual collisions and avoid them by themselves.

2 Autonomous Decentralized FMS Factory

2.1 Autonomous Decentralized System

There's a limit to improve the functions of systems. We put plural systems as subsystems which have a certain amount of intelligence and functions to divide the operations among them. As for a method to control them, there're two ways. It's centralistic control type and the other is autonomous decentralized one.

The former does distribution or adjustment of operations by the mere host computer intensively. Under the unchangeable and firm circumstances, it is efficient. Accordingly it has a problem for taking steps to meet the situation. If once great fluctuation occurs, it's too difficult to make many modifications.

On the contrary, in the latter system, the latter make subsystems have a certain degree of autonomy and they can take steps autonomously ; thus the flexibility for meeting the situation is happened. We deal with the FMS which uses an autonomous decentralized system, in this paper.

2.2 Autonomous Decentralized Factory of FMS[1]

It's very difficult to analyse a system like the autonomous decentralized FMS. We evaluate and analyse the problems in operation by simulations of a computer. The autonomous decentralized FMS we simulate here is a small-medium quantity production system for variety of parts and it consists of;

- Moving robot:makes a hypothelical reasoning, moves and decides which part is taken. It's called Automatic Guided Vehicle(AGV).

- MC:processes the parts taken by AGVs.

- Storehouse : store the parts and products.

Each component of the above three has the 4 functions, the functions of self recognition, communication, making a decision and the one of performance.

2.3 Hypothetical Reasoning

This paper adopts hypothelical reason to make a decision for AGVs decentralized functions. Figure 1 shows the hypothetical reasoning. In Figure 1, the value of step 8 is decided as 2. The algorithm is as follows;
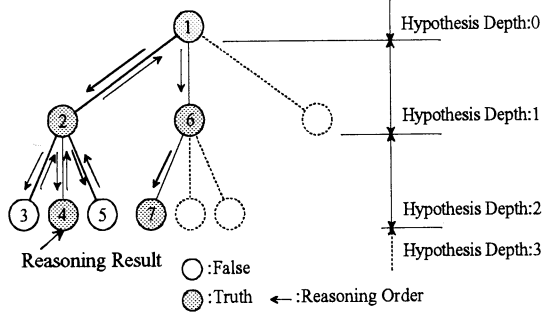
Figure 1 Hypothetical Reasoning

Step1: Present hypothotical depth is defined as 0.

Step2: Among plural selections at the next depth to 0, that's not identified whether true or not yet, we take the leftest node is decided as true.

Step3: Simulate the node.

Step4: Verify whether the node is true or not after the simulations.

Step5: Go to Step 8 when it's true. Go to Step 6 when it's false.

Step6: Consider the next node corresponding the node, which is decided as false, as true and go to Step 3. When there's not a selection at the same depth as one taken first, go to Step 7.

Step7: Raise the depth level by 1 and go to Step 6.

Step8: When the depth is over a certain depth, go to Step 9. If not, go to Step 2.

Step9: We consider it as true and hypothetical reasoning is finished.

The characteristic of the hypothetical reasoning is that it can deal with a selection which may be false. Because of the characterustic, imperfect knowledge can be dealt with as hypothesis.

2.4 Unsystematic Selection in Hypothetical Reasoning

As stated in section 2.3, we can get a noncontradictory selection. But this is not enough to realize high efficient plan for path and production.

For example, in the case of Figure 1, when both selections of no.4 and 7 are true, the selection no.2, which follows no.4, is supposed to be chosen. If the selection no.7 ought to be given priority over no.4 to realize high efficient plan, this gets a problem.

To solve this, AGVs rearrange selections at same depth. According to progress in the production, they give priority to the parts that are in delayed line, or among the several MCs of same kind, they also give priority to the one those rate of operation is low. By these rearrangements, the leftest selections, that are given priority, are supposed to be chosen.

In the case of Figure 1, check which selection no.2 or 6 ought to be chosen earlier and rearrange them. And also check no.3, 4 and 5·then rearrange. As for the others, do as the same. Therefore, Figure 1 should be changed to be Figure 2.
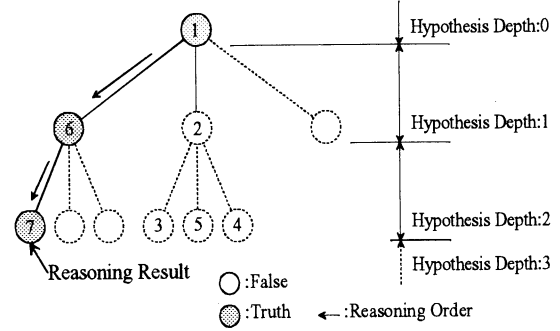
Figure 2 Hypothetical Reasoning(Selected Unsystematically)

Unsystematic selection in hypothetical reasoning realizes the high efficient path plan based on production plan, in an autonomous decentralized system.

2.5 Simulations

In our study, it 's AGVs to convey parts around the floor with reasoning own operation plan hypothetically.

And in the simulation space of autonomous decentralized FMS[2] in Figure 3, we simulate its plan. By this simulation, we verified not only realization of operation plan but also the peculiarity of autonomous decentralized system, such as flexibility, adaptability and reliability. We raise and reduce the kinds of productions for verification of flexibility and do the numbers of MCs for adaptability.

For reliability, we set special 4 conditions.

1. Normal situation.
2. Random 10% longer time for process.
3. 3 times random breakdowns of each AGV for 5 minutes within 24 hours.
4. Under the conditions where both of 2 and 3 happen together.

Figure 3 Simulation Area

Table 1 The Reslut of Simuration

Special Condition	1	2	3	4
PV	461	448.1	459.6	446
ARO	46.2058	44.5588	45.3918	44.2860
MRO	16.5871	16.8273	16.5794	16.7625

PV: Production Volume (pieces)
ARO: AGV's Rate of Operation (%)
MRO: MC's Rate of Operation (%)

Table 2 Disparity among The Conditions

Special Condition	1	2	3	4
PV	0	-12.9	-1.4	-15
ARO	0	-1.6470	-0.8140	-1.9198
MRO	0	-0.2402	-0.0077	-0.1754
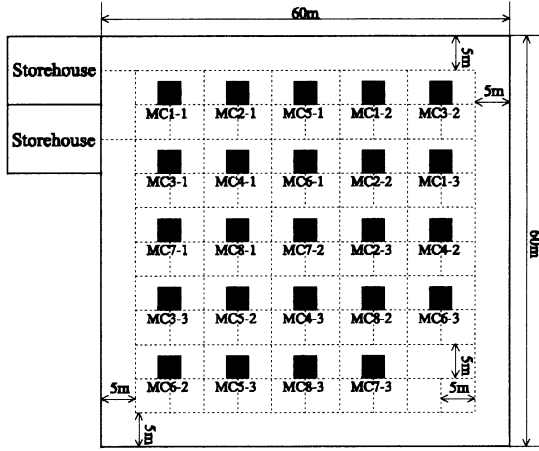
2.6 Results

We simulated under the conditions below to verify flexibility.

1. Maximum of 3 procedures for 3 kinds of productions.
2. Maximum of 5 procedures for 6 kinds.
3. Maximum of 8 procedures for 9 kinds.

They resulted in definite process without a problem. Also under any condition, each ratio of production that is fixed by a production plan can be realized.

For adaptability, we changed the number and kind of MCs. The kind was from 3 to 8. Then, a result confirmed that MCs could cope with the change, which means they have adaptability.

As for reliability, we had a simulation under the special conditions stated in Section 2.5. Furthermore we added conditions set for verifications of flexibility and adaptability. After that, whenever a production plan changes or the number and kind of MCs vary, the subsystems work with reliability. Down of systems or a big fall in efficiency were never happened.

We show one of results at Table 1. 9 kinds of productions were processed under the special conditions and through a maximum of 8 procedures with 8 MCs, 5 AGVs per MC within 24 hours.

As for special conditions 2~4, the average of 10 times of simulations is shown. The discrepancy between 1 and 2~4 can be shown in percentage at Table 2.

No downtime of whole factory happened. Under the 2, because of the random delay in process, a big fall in efficiency seemed to be happened though it didn't. Subsystems changed their operation plans flexibly. Under the 3, the AGVs' average rate of operation was supposed to be dropped by 1.04% but it 's held by 0.814% decrease. They covered against breakdowns.

3 AGV's Intelligent Control

So far, we haven't taken into consideration the possibility of AGVs' mutual collisions.

However there's danger of them in real factorys, where each AGV is put the sensor on and monitored every actions by a host computer that directs him. This is a heavy burden on a host computer and it grows more when subtle discrepancies in the operation plan occur under the special conditions of section 2.5 and others.

Therefore we also propose the autonomous decentralized system for path plan of AGVs, which makes them communicate and judges by themselves to avoid collisions in real-time.

3.1 Situations

It's a problem how to pass each other after avoidance of collisions in a factory that has 1 lane for AGVs.

The road traffic in our life has traffic regulations that avoid most car collisions. But when we catch a forward car up and compromise on narrow roads, we must judge the situations and express intentions.

For example, on a narrow crossing, I want to go straight and an oncoming car's getting on it faster. I'm sure to watch his winkers to know which direction he wants to go in. Wait until he turns right or left, run aside each other to let us pass.

3.2 Simulation of Collision Avoidance

We simulated in the space of Figure 3. AGVs run on the dotted lines that don't afford passing each other but for both ways.

Take a point where vertical and horizontal lines cross as a crossing. MCs are on the lines and AGVs

give and receive parts on a crossing in front of MCs, where they stop to do.

3.3 Collision Patterns

In the simulation space that has limitations, collisions can be classified into 6 patterns by situations.

1. Rear-end collision.
2. On the crossing.
3. Head-on collision.
4. At the storehouses.
5. During stopping to give and receive parts.
6. At a Deadlock.

1 case happens when an AGV catches a forward one up on a same path. 2 when they come on it simultaneously. 3 oncoming AGVs run against. 4 when more than 1AGV come to storehouses where are at a deadlock. 5 stopping blocks the others' path. 6 deadlock means that many types of collisions happen simultaneously; AGVs that are avoiding a head-on collision are got close by backward one, which could hit

3.4 Communication

AGVs communicate with each other to perceive danger of collisions and judge to classify them. They have each measure to cope with each pattern as knowledge, which make them avoid collisions on the spot autonomously without a host computer.

The communication between subsystems is not like a conversation but a broadcast. Send the information to every subsystem in a factory and the only subsystem that needs it takes it. Thus any communication is possible, which means no partial stop (of a subsystem) influences whole factory. Also there's no need to consider a burden on a server that's not necessary for a broadcast.

AGVs in our study plan operations to go to a destination spreading own path plans as informations. They go through crossings and spread their paths as they pass there. The information's about a present crossing, the next and the other after the 2nd with the passing time at the present one.

On taking it, any AGV can know the others' path to avoid collisions.

3.5 Collision Avoidance

Decide which AGV has priority for coming first on the point where a collision could be. The point is a crossing or between the crossings.

Accordingly they can act on each measure to cope with each collision pattern.

As for 1, they slow to keep a certain distance between themselves.

2, they're given numbers on and the smaller has priority. The larger makes itself come later.

3, spread informations afford a time for passing 3 crossings. The prior turns or comes straight first, the other waits for him to finish turning or change own path by turning at another crossing.

4, they don't come into the line that reaches storehouses until the other goes away.

5, the stopping has less possibility to move soon and it has difficulty avoinding collisions. The other searches for another path.

6, they can see the others do avoidance forward and act not to hit against them. When the forwards are prior, avoid a head-on collision with the one left that did avoidance at forward case. When the forwards are not, move not to hit them, which means when the forwards begin moving then begin moving,too. When the forwards come back then come back, too. In that case, they have to be careful because there's possibility of another AGV, which is made to come back together.

3.6 Results

The simulation done at 2.5 showed that collisions of 5 patterns at 3.4 could be avoided. Still, there's a need to decide which is prior over the others in the 6th pattern. Because of the plural AGVs that act avoidance respectively, which need an order.

4 Conclusions

We proposed an autonomous decentralized system applied to an operation plan of AGVs in an autonomous decentralized FMS factory. This makes a real-time operation plan have flexibility, adaptability and reliability.

We also proposed an Autonomous Decentralized system applied to a path plan of AGVs, which can avoid collisions in real-time.

References

[1] Yamamoto H, "Tecnology Toward Auttonomous decentralized FA Factory," *System, Control and Information.* Vol. 44, pp. 2000, 259-265.

[2] Yamamoto H, "AGV's Action Decision by Future Foresee Reasoning in Decentralized Autonomous FMS," *Transactions of The Japan Society of Mechanical Engineers. C,* Vol. 65, pp. 1999, 1281-1287.

Modeling of Motor Control on Manual Tracking for Developing Hand Movement Compensation Technique

T. Sugi*, M. Nakamura**, J. Ide[†] and H. Shibasaki [‡]

*Department of Electrical and Electronic Engineering, Saga University, Saga 840-8502, Japan

** Department of Advanced Systems Control Engineering, Saga University, Saga 840-8502, Japan

[†] Department of Commerce, Seinan Gakuin University, Fukuoka, 814-8511, Japan

[‡] Departments of Neurology and Human Brain Research Center, Kyoto University, Kyoto 606-8501, Japan

E-mail : sugi@cntl.ee.saga-u.ac.jp

Abstract

Skillful motor control of human is achieved by the appropriate motor commands generating from the central nerve system. Internal model in the cerebellum may have great contribution for realizing accurate motor control. In this paper, the model for human motor control on manual tracking to moving visual target was constructed based on the experimental data of the visual tracking test for normal healthy adults and patients with cerebellar ataxia. The compensation method for improving the motor function of patients were also developed by using the model.

Keywords: Motor Control, Visual Tracking Test, Motor Function, Modeling, Predictive Control, Movement Disorders

1 Introduction

Generation of appropriate motor commands by central nerve system practices the realization of dynamic and skillful motion of human. Several regions in human brain are concerned with the achievement of motor control. In particular, the cerebellum has an important role in motor control. Serious defects in motor control occur when the cerebellum has injured. Movement disorder originating from the cerebellum such as cerebellar ataxia develops a lack of accurate movement and troubles in motor learning.

In the past, we have studied the motor functions of the human hand movement through the visual tracking test [3]. We have previously reported the quantitative evaluation on the effect of motor learning [3], and so on.

In this paper, the model for human motor control on manual tracking to moving visual target was constructed in accordance with the acutual data of visual tracking test for patients. First, the characteristics of the motor functions on visual tracking test

were analyzed through the data recorded from normal healthy adults and patients with cerebellar ataxia. Then, the model of motor control apparatus was constructed. The central nerve system for generating the motor commands to a hand was represented by the Smith predictor, and the parameters in the model were determined by reflecting the characteristics of motor functions on visual tracking test. Finally, Hand movement compensation technique was constructed by using the model information of both normal subject and patient with cerebellar ataxia. The effect of hand movement compensation was investigated through the simulation.

2 Analysis of Human Hand Movement by Visual Tracking Test

2.1 Subjects and data acquisition

The measurement equipment for visual tracking test used in this study is shown in Figure 1. The system consisted of personal computer (NEC PC-9801BX), CRT monitor, digitizer (ADAPTEC KD4030) and magnetic pen [3]. Computer and digitizer was connected by RS-232C cable. Subject holds on a pen by his/her hand and moves it on dizitizer, just then the corresponding green rectangular point (pursuit point) is moved on CRT monitor. Visual target with red circle is also displayed on same monitor. Subject carries out the task to pursue the visual target as correctly and speedily as possible by moving one's hand.

Visual target used in the test moves 3 sec long from the upper part to the lower part of the monitor straightforwardly with a constant speed of 3 cm/s. Number of trials of visual tracking test was 20 times

for each subject. Number of subjects were 3 healthy normal adults and 3 patients with cerebellar ataxia.
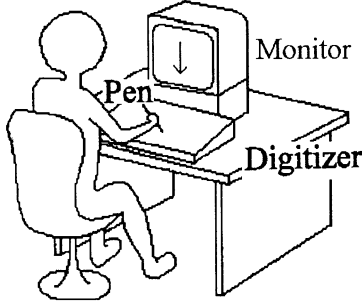
Figure 1: Measurement equipment for manual tracking to visual target.

2.2 Data analysis

Parameters for extracting the characteristics of the motor functions on visual tracking test were calculated from the recorded data (experimental results were seen in Figure 4). Three parameters were adopted in this study [3]. First, the position error E_p was defined as the difference between the vertical movement of hand tip position (pursuit point) and visual target. Secondly, standard deviation of velocity for vertical direction V_{SD} was calculated. Thirdly, the reaction delay L_E was defined by the time difference of the begining of movement between visual target and pursuit point.

The above three characteristic parameters were adopted in this study.

2.3 Experimental results

Figure 2 shows the result of parameters transition for each group. Averages of three parameters E_p, V_{SD} and L_E calculated from the respective groups were displayed from the top of the figure. In the position error E_p, normal subjects took smaller value as compared with patients with cerebellar ataxia. In the case of standard deviation of velocity V_{SD}, patients with cerebellar ataxia took higher value. This fact suggests that the patients with cerebellar ataxia could not achieve a smooth pursuit for the visual target. In case of reaction delay L_E, result of normal subjects was smaller than patients with cerebellar ataxia.
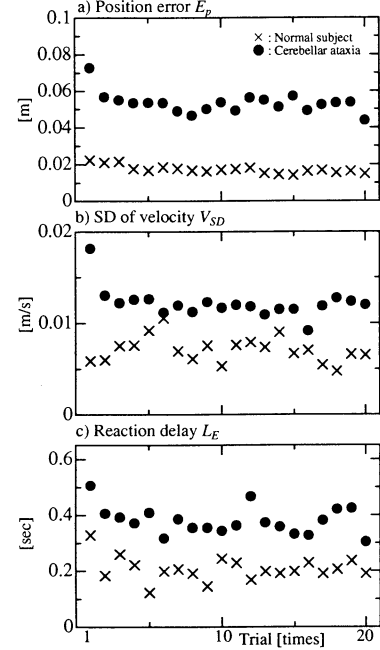
Figure 2: Quantitative evaluation of hand movement on visual tracking test. Comparison between normal subjects and patients with cerebellar ataxia.

3 Modeling of Motor Function on Visual Tracking Test by Smith Predictor

3.1 Block diagram

Human can achieve a skillful motion in spite of the central nerve system has a time delay. This fact may be regarded as that the control paradigm in human brain has a kind of predictive control strategy. Then the motor control model was constructed as shown in Figure 3(a). In this study, the central nerve system for generating the motor commands was represented by the Smith predictor as one of model predictive control techniques. Signal flow in motor control on visual tracking test may be considered as the closed-loop system. Detail explanation was described in the following sections.

3.2 Control objective

Accurate representation of dynamics on human hand as controlled objective is complex so that the detail human hand model can not be obtained easily. However, the principal properties of human hand dynamic will be able to express the simple mathematical

equations. In this study, dynamic of hand movement was simply defined as linear first order system as

$$G(s) = \frac{K}{1 + Ts} \qquad (1)$$

where gain K and time constant T were determined as 1.0 and 0.2, respectively. Actual motor control loop in human includes the reaction delay caused by recepter, axonal conductance, cognitive processing and so on, and the reaction delay is not negligible short. Thus the reaction delay was also considered in the control objective as e^{-Ls}.

(a)

(b)
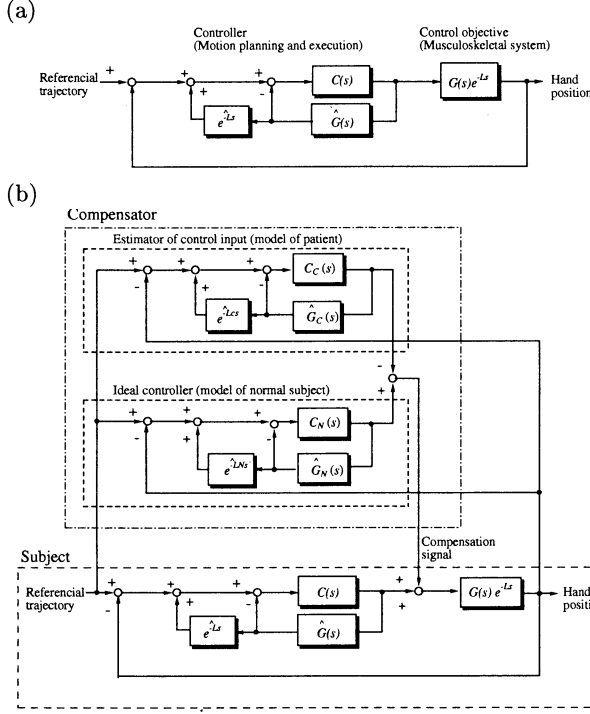
Figure 3: (a) Model representation of motor control on manual tracking to moving visual target. Central nurve system for generating motor commands was expressed by using Smith predictor. (b) Structure of hand movement compensator by using model information.

3.3 Generation of motor commands

Several models for representing motor control, especially for the cerebellum, were proposed in the past [1] [2]. In this study, we adopted the Smith predictor as representing the central nerve system for generating the motor commands. Smith predictor has a internal model and its feedback loop with the time delay. Internal model in the Smith predictor ($\hat{G}(s)$ in Figure

3(a)) was assumed to be the same as eq.(1) as

$$\hat{G}(s) = \frac{\hat{K}}{1 + \hat{T}s} \qquad (2)$$

Controller $C(s)$ in Figure 3(a) for generating motor commands was simply designed by PID controller as

$$C(s) = K_P(1 + \frac{1}{T_I s} + K_D s) \qquad (3)$$

where K_P, T_I and T_D were gain for propotional, integral and derivative, respectively.

3.4 Experimental data and model output

In order to determine all the parameters in the model, characteristic parameters E_p and V_{SD} described in 2.2 were used for optimization. Characteristic parameters were also calculated from the output of the constructed model. Then, the cost function was created and minimized by using least squares method.

Figure 4 shows the verification of the constructed motor control model. Model output was indicated by broken line. Left side of the figure shows the result for normal subject and right one displays that for patient with cerebellar ataxia. The upper row shows the variation of hand tip position and the lower row display the velocity. Obtained model outputs of all subjects were in a good agreement with the measurement data of actual hand momement.
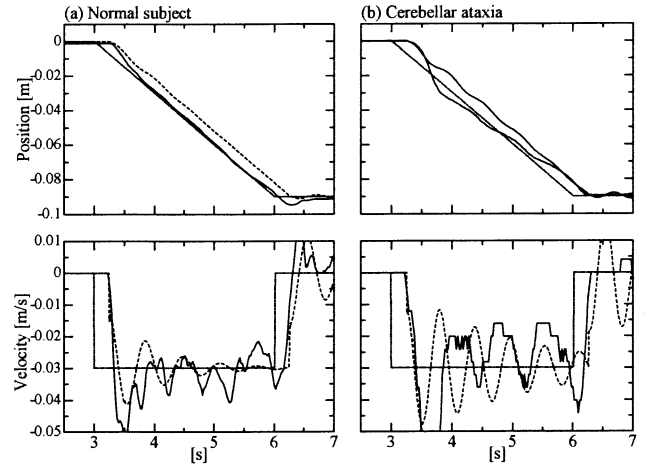
Figure 4: Comparison of experimental data (solid line) and model output (broken line). (a) Normal subject, (b) Patient with cerebellar ataxia.

4 Development of Hand Movement Compensation Technique by Using Model Information

4.1 Structure of compensator

To Improve the human hand movement is the final goal of this study. The characteristics of hand movement for patients with cerebellar ataxia was reflected to the model structure which was obtained from the actual experimental data. Therefore, the compensation technique for improving the hand movement of patients can be developed by using the model information. Figure 3(b) shows the block diagram for hand movement compensation. Compensator in this figure is possible to construct by the information of both the model for the patient and the model for healthy adult.

Estimation of control input for a subject (patient) is done by the model obtained from a patient with cerebellar ataxia. Equations of estimator in Figure 3(b) are written as

$$C_C(s) = K_{PC}(1 + \frac{1}{T_{IC}s} + K_{DC}s) \qquad (4)$$

$$\hat{G}_C(s) = \frac{\hat{K}_C}{1 + \hat{T}_C s} \qquad (5)$$

The ideal controller for achieving the smooth motion is designed by the model for normal subject. Equations of ideal controller are expressed as

$$C_N(s) = K_{PN}(1 + \frac{1}{T_{IN}s} + K_{DN}s) \qquad (6)$$

$$\hat{G}_N(s) = \frac{\hat{K}_N}{1 + \hat{T}_N s} \qquad (7)$$

Compensation signal can be generated by the difference between two models.

4.2 Simulation result

The fundamental effect of compensator was investigated through the simulation study. Figure 5 shows the simulation result for improving the hand movement in patient. Broken line means the model output of a patient with cerebellar ataxia without compensator. Solid line expresses the result by using the compensator. Amplitude of osscilation in velocity was decreased by the compensator, and the smooth motion in the patient with cerebellar ataxia was achieved.

Simulation result shown in Figure 5 was obtained under the ideal condition. However, there are more than a few modeling errors in the constructed model,

and the modeling error often deterirates the control preformance. Influences of the modeling error for robustness or stability must be verified.
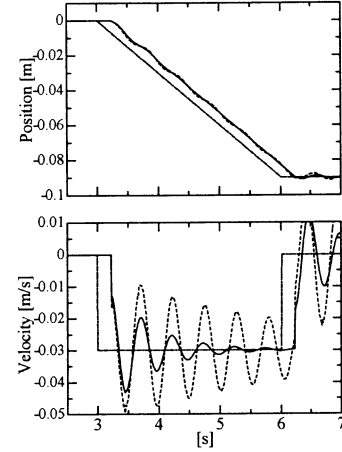
Figure 5: Simulation result of hand movement compensation.

5 Conclusion

Based on the actual data, the motor control model was constructed by using the Smith predictor. The method for hand movement compensation was proposed by using the model information obtained from a normal subject and a patient with cerebellar ataxia. Possibility for improving the hand motion in the patient was confirmed through the simulation. Improvement of this study expects the recovery of hand movement of patients with various movement disorders.

References

[1] Kawato M, Furukawa K, Suzuki R (1987) "A hierarchical neural-network model for control and learning of voluntary movement," *Biological Cybernetics*, 57, pp.169-185.

[2] Miall RC, Weir DJ, Wolpert DM, Stein JF (1993) "Is the cerebellum a Smith predictor ?," *Journal of Motor Behavior*, 25-3, pp.203-216.

[3] Ide J, Sugi T, Nakamura M, Shibasaki H (2000) "Quantitative evaluation of learning effect for different movement disorders by use of manual tracking to moving visual target (in Japanese)," *The Commercial Review of Seinan Gakuin University*, 46-3,4, pp.347-361.

Motor Control System with Skill by Construction of Internal Model

Koji Okuhara,
Dept. of Management
and Information Sciences
Hiroshima Prefectural Univ.
Shobara, 727-0023, Japan

Takeshi Matsui,
Dept. of Management
and Information Sciences
Hiroshima Prefectural Univ.
Shobara, 727-0023, Japan

Toshijiro Tanaka
Dept. of Management
and Information Sciences
Hiroshima Prefectural Univ.
Shobara, 727-0023, Japan

Abstract

In this paper we will present a mechanism of skill of task in a motor control system. For the purpose Knowledge Option network (KOnet) is proposed to acquire impedance in coefficient matrix of a state equation. After estimating suitable impedance for a task, the motor control system comes to generate motor planning instruction from using a feedback control with sensory information to using feedfoward control with a prediction at internal model.

key words Motor Control Planning, Skill, Construction, Internal Model

1 Introduction

In voluntary movement control of human action, it seems to become smooth from clumsy movement after some iterative exercise. It can be thought that motor planning is changed over a forward control by constructing an internal model with *Sensory Feedback Information* at the cerebellum. We propose such mechanism to incorporate in a motor control system.

In order to realize a desired movement, it is necessary to control a position and force of object according to the external environment. Thus we must acquire the adequate *Motor Impedance* for the work. The internal model for skill of motor planning is constructed by learning through exercises. It is known that neural network is usefull to acquire the impedance for task [1]. Thus we propose *KOnet* to a learning part for realizing skillful movement.

We take into account the mechanism that the motor planning using the internal model orders a slow and small movement by referencing sampling of sensory feedback information when the internal model does not have a high conviction [2]. As the reliability of the internal model is improved, the motor planning will order a fast and large movement without referencing sampling of sensory feedback information [3].

In this paper, we will think that a skill for a work is a transition from a feedback control to a forward control which is according to the degree of conviction for the internal model, and it is an active realization of the adequate motor impedance for the work. For the uncertain action and noise, therefore, we set a stiffness higher by adding power to motor at initial state, and change its stiffness lower during learning.

2 Outline of Internal Model

2.1 Structure of Internal Model

We consider a linear discrete state and observation equation with controllabilty and observability.

$$\mathbf{x}_{t+1} = \mathbf{A}_t\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t + \mathbf{D}_t\mathbf{w}_t, \qquad (1)$$

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}_t + \mathbf{v}_t \qquad (2)$$

where $\mathbf{x}_t \in \Re^n$ denotes a state vector, $\mathbf{u}_t \in \Re^r$ is an input vector and $\mathbf{y}_t \in \Re^m$ is a observation vector. $\mathbf{w}_t$ and $\mathbf{v}_t$ are assumed to be white gaussian noise. $\mathbf{A}_t \in \Re^{n \times n}$ and $\mathbf{B}_t \in \Re^{n \times r}$ in state equation are coefficient matrix. $\mathbf{C}_t \in \Re^{m \times n}$ in observation equation is a observation matrix.

Figure 1 is a conceptual figure of evolutionary in human voluntary movement. The setting of target is done at association area (A.A.) and its output is sent to motor area (M.A.). In motor area it outputs a motor planning instruction after receiving an input from association area and a motion corrective instruction from a cerebellum intermediate region (C.I.R.). It is assumed that cerebellum intermediate region outputs its motor corrective instruction from somatosensory mainly. Furthermore motor planning instruction gives to musculoskeletal system after receiving motor corrective instruction from sensory information which is mainly acquired by vision at cerebellum lateral region (C.L.R.). It is known that cerebellum is concerning

with motor controls requiring a skill. This mechanism is similar to Kalman filter.
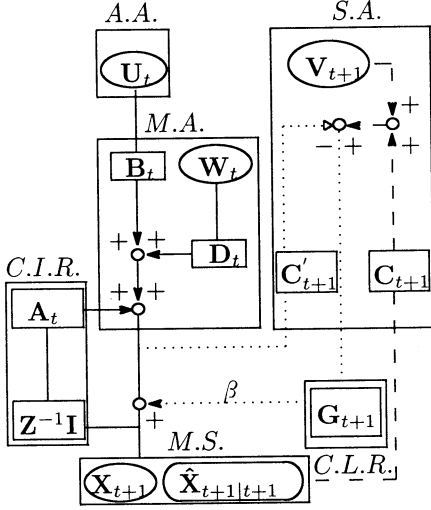
Figure 1 Conceptual Figure of Internal Model

Now we describe skill of motor control by using internal model. For motor planning, in case we don't have enough experience or uncertain disturbance, we control our movement by using feedback information from sensory area. At that time since time delay occurs for necessity of sensory information $\mathbf{y}_t$ of vision mainly, we can't execute smooth movement. However, as learning proceed motor area and cerebellum acquire impedance for task. By using a prediction $\hat{\mathbf{x}}_{t+1|t}$ instead of the sensory information $\mathbf{y}_t$ smooth movement can be executed without a motor corrective instruction from cerebellum lateral region.

2.2 Dynamics of KOnet for Skill

Assuming that the sampling data consist of explanatory vector $\mathbf{x}_s \in \Re^n$ and explained vector $\mathbf{y}_s \in \Re^m$ and we describe a joint vector of them by vector $\mathbf{z}_s = [\mathbf{x}_s \mathbf{y}_s]^T \in \Re^d$ where $d = n + m$. Figure 2 shows the dynamics of KOnet to acquire the impedance for task.

The kth input neuron ($k = 1, 2, \cdots, K$) has parameters ϕ_k and w_k. The parameter ϕ_k consists of set of vector and matrix $\{\mathbf{m}_k, \Sigma_k\}$, where $\mathbf{m}_k$ is $[m_k^1, m_k^2, \cdots, m_k^d]^T \in \Re^d$ and Σ_k is a $d \times d$ positive define and symmetric matrix with σ_k^{ij} as the ijth element. The vector $\mathbf{m}_k$ consists of vectors $\mathbf{m}_k^x \in \Re^n$, $\mathbf{m}_k^y \in \Re^m$ which relates to $\mathbf{x}_s$, $\mathbf{y}_s$ respectively,
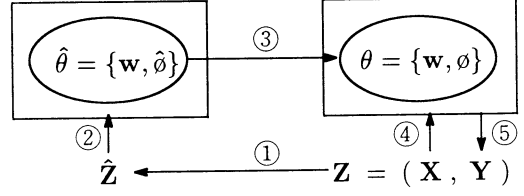
Figure 2 Structure and Dynamics of KOnet

it is given by

$$\mathbf{m}_k = [\mathbf{m}_k^{x^T} \ \mathbf{m}_k^{y^T}]^T. \tag{3}$$

The matrix Σ_k has matrices $\mathbf{C}_k^x \in \Re^n \times \Re^n$, $\mathbf{C}_k^y \in \Re^m \times \Re^m$ concerning vector $\mathbf{x}_s$, $\mathbf{y}_s$ respectively, and matrices $\mathbf{C}_k^{xy} \in \Re^n \times \Re^m$, $\mathbf{C}_k^{yx} \in \Re^m \times \Re^n$ which is mutual concerning with vectors $\mathbf{x}_s$, $\mathbf{y}_s$. Thus it is

$$\Sigma_k = \left[\begin{array}{cc} \mathbf{C}_k^x & \mathbf{C}_k^{xy} \\ \mathbf{C}_k^{yx} & \mathbf{C}_k^y \end{array} \right]. \tag{4}$$

n dimensional sth input vector $\mathbf{x}_s \in \Re^n$, ($s = 1, 2, \cdots, S$) are transmitted to all input neurons. S denotes sampling number of data. kth input neuron outputs

$$\mathbf{y}_k = \mathbf{m}_k^y + \mathbf{D}_k^y(\mathbf{x}_s - \mathbf{m}_k^x) \tag{5}$$

where matrix $\mathbf{D}_k^y \in \Re^m \times \Re^n$ is given by

$$\mathbf{D}_k^y = \mathbf{C}_k^{yx} \mathbf{C}_k^{x^{-1}} \tag{6}$$

and $\mathbf{C}_k^{x^{-1}}$ denotes an inverse matrix of $\mathbf{C}_k^x$.

kth input neuron transmits its output through a weight α_k^x to output neuron. By adding up these inputs the output neuron generates

$$\begin{aligned} \mathbf{y} &= \sum_{k=1}^{K} \alpha_k^x \mathbf{y}_k \\ &= \sum_{k=1}^{K} \alpha_k^x \{\mathbf{m}_k^y + \mathbf{D}_k^y(\mathbf{x}_s - \mathbf{m}_k^x)\} \end{aligned} \tag{7}$$

where

$$\alpha_k^x = \frac{w_k p_k(\mathbf{x}_s|\phi_k)}{\sum_{k=1}^{K} w_k p_k(\mathbf{x}_s|\phi_k)} \tag{8}$$

and

$$\sum_{i=1}^{K} w_i = 1, \tag{9}$$

$$p_k(\mathbf{x}_s|\phi_k) = N_n(\mathbf{x}_s, \phi_k^x). \tag{10}$$

The parameter ϕ_k^x denotes a set of $\{\mathbf{m}_k^x, \mathbf{C}_k^x\}$ and

$$N_n(\mathbf{x}_s, \phi_k^x) = \frac{1}{(2\pi)^{n/2} \|\mathbf{C}_k^x\|^{1/2}}$$

$$\times \exp\left\{-\frac{1}{2}(\mathbf{x}_s - \mathbf{m}_k^x)^{\mathrm{T}} \mathbf{C}_k^{x^{-1}} (\mathbf{x}_s - \mathbf{m}_k^x)\right\} \quad (11)$$

$\|\cdot\|$ denotes determinant of matrix.

From mentioned above, let $\mathbf{w}$ be a set $\{w_1, w_2, \cdots, w_K\}$, ϕ be a set $\{\phi_1, \phi_2, \cdots, \phi_K\}$ and θ be $\{\mathbf{w}, \phi\}$. It is understood that this internal model generates its output depending on their parameter θ.

3 Learning in Internal Model

3.1 Formulation by Probability

For modelling a mapping from the explanatory vector $\mathbf{x}_s$ to the explained vector $\mathbf{y}_s$, we must estimate the value of parameter θ. In this subsection, we formulate the dynamics by probability for the purpose we will derive a learning algorithm for acquiring the impedance. So we regard the vector $\mathbf{m}_k$ and matrix $\mathbf{\Sigma}_k$ as the mean vector and covariance matrix, then we can easily understand that the output $\mathbf{y}_k$ of kth input neuron can be rewritten as

$$\begin{aligned} \mathbf{y}_k &= \int_{\Re^m} \mathbf{y} p_k(\mathbf{Y}|\mathbf{X}_s, \phi_k) d\mathbf{y} \\ &\equiv \mathrm{E}_k[\mathbf{Y}|\mathbf{X}_s] \end{aligned} \quad (12)$$

where $\mathrm{E}_k[\mathbf{Y}|\mathbf{X}_s]$ represents conditional expectation value of a random vector $\mathbf{Y}$ for a random vector $\mathbf{X}_s$ at kth input neuron. The output $\mathbf{y}$ of system is rewritten as

$$\begin{aligned} \mathbf{y} &= \sum_{k=1}^{K} \alpha_k^x \int_{\Re^m} \mathbf{y} p_k(\mathbf{Y}|\mathbf{X}_s, \phi_k) d\mathbf{y} \\ &= \int_{\Re^m} \mathbf{y} \sum_{k=1}^{K} \alpha_k^x p_k(\mathbf{Y}|\mathbf{X}_s, \phi_k) d\mathbf{y}. \end{aligned} \quad (13)$$

We can consider the following probability

$$p(\mathbf{Y}|\mathbf{X}_s, \theta) = \sum_{k=1}^{K} \alpha_k^x p_k(\mathbf{Y}|\mathbf{X}_s, \phi_k) \quad (14)$$

by assuming that a selection probability of kth neuron is given by w_k, that is

$$\begin{aligned} \mathbf{y} &= \int_{\Re^m} \mathbf{y} p(\mathbf{Y}|\mathbf{X}_s, \theta) d\mathbf{y} \\ &\equiv \mathrm{E}[\mathbf{Y}|\mathbf{X}_s]. \end{aligned} \quad (15)$$

Therefor it is easily understood that the output vector $\mathbf{y}$ of system is a conditional expectation value of a random vector $\mathbf{Y}$ for a random vector $\mathbf{X}_s$.

The conditional probability density function $p_k(\mathbf{Y}|\mathbf{X}_s, \phi_k)$ at kth input neuron obey a m dimensional normal distribution with the mean vector $\mathrm{E}_k[\mathbf{Y}|\mathbf{X}_s] \in \Re^m$ and a following covariance matrix $\mathbf{C}_k^{y'} \in \Re^m \times \Re^m$

$$\mathbf{C}_k^{y'} = \mathbf{C}_k^y - \mathbf{D}_k^y \mathbf{C}_k^x \mathbf{D}_k^{y\mathrm{T}}, \quad (16)$$

then the conditional probability density function $p(\mathbf{Y}|\mathbf{X}_s, \theta)$ of system can be derived by

$$p(\mathbf{Y}|\mathbf{X}_s, \theta) \equiv \sum_{k=1}^{K} \alpha_k^x N_m(\mathbf{X}_s, \varphi_k) \quad (17)$$

where φ_k is given by $\{\mathbf{y}_k, \mathbf{C}_k^{y'}\}$.

The random vector $\mathbf{Z}_s$ at kth input neuron obeys the following probability density function

$$p_k(\mathbf{Z}_s|\phi_k) \equiv N_d(\mathbf{Z}_s, \phi_k) \quad (18)$$

with the mean vector $\mathbf{m}_k$ and the covariance matrix $\mathbf{\Sigma}_k$, thus we can find that the random vector $\mathbf{Z}_s$ for the total system obeys the following mixture normal distribution

$$p(\mathbf{Z}_s|\theta) \equiv \sum_{k=1}^{K} w_k N_d(\mathbf{Z}_s, \phi_k). \quad (19)$$

This means that we can apply the Expectation Maximization algorithm to derive parameter θ in learning of KOnet.

3.2 Learning Algorithm

As the learning algorithm of KOnet, though we can derive the value of parameter $\theta = \{\mathbf{w}, \phi\}$ directly by applying EM algorithm. However, in order to avoid an oscilation which is caused when we apply the algorithm to non-diagonal elements of covariance matrix $\mathbf{\Sigma}_k$, we will make KOnet learn a part of parameter ϕ.

In this paper, we standardize observed data to be its mean 0 and variance 1 by deriving a mean m^i and variance σ^{ii} for each ith variable. After obtaining a correlation matrix of the standardized observation data we transform it into

$$\hat{\mathbf{z}}_s = \mathbf{E}_{\mathbf{v}} \left\{ \mathbf{\Sigma}^{-1/2} (\mathbf{z}_s - \mathbf{m}) \right\} \quad (20)$$

by using a matrix $\mathbf{E}_{\mathbf{v}} = [\mathbf{E}_1, \mathbf{E}_2, \cdots, \mathbf{E}_d]^{\mathrm{T}} \in \Re^d \times \Re^d$ composed of eigen vector $\mathbf{E}_i \in \Re^d$, $(i =$

$1, 2, \cdots, d)$ of the correlation matrix. Here $\mathbf{m} = [m^1, m^2, \cdots, m^d]^{\mathrm{T}} \in \Re^d$ and $\boldsymbol{\Sigma}$ is $d \times d$ diagonal matrix with σ^{ii} as iith element. In learning mode KOnet uses the orthogonalized vector $\mathbf{z}'_s$.

From the result, the random vector $\hat{\mathbf{Z}}_s$ at kth input neuron obeys the probability density function with the mean vector $\hat{\mathbf{m}}_k = \mathbf{B}^{\mathrm{T}}(\mathbf{m}_k - \mathbf{m})$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_k = \mathbf{B}^{\mathrm{T}}\boldsymbol{\Sigma}_k\mathbf{B}^{-1}$, where $\mathbf{B}^{\mathrm{T}} = \left(\mathbf{E_v}\boldsymbol{\Sigma}^{-1/2}\right)$. $\hat{\boldsymbol{\Sigma}}_k$ is a $d \times d$ diagonal matrix having variance $\hat{\sigma}_k^{ii}$ as iith element.

Let $\hat{\phi}_k$ be a set $\{\hat{\mathbf{m}}_k, \hat{\boldsymbol{\Sigma}}_k\}$, $\hat{\phi}$ be a set $\{\hat{\phi}_1, \hat{\phi}_2, \cdots, \hat{\phi}_K\}$ and the parameter $\hat{\theta}$ be a set $\mathbf{w}$ and $\hat{\phi}$. The estimation of parameter $\hat{\theta}$ can be obtained by maximizing the following log-likelihood function:

$$L(\hat{\theta}) = \sum_{s=1}^{S} \log p(\hat{\mathbf{Z}}_s | \hat{\theta}). \tag{21}$$

The update rule for the parameter $\hat{\theta}$ can be derived from

$$\begin{aligned} Q(\hat{\theta}|\hat{\theta}^{(t)}) &= \mathrm{E}[L(\hat{\theta}, k)|\hat{\mathbf{z}}_s, \hat{\theta}^{(t)}] \\ &= \sum_{s=1}^{S}\sum_{k=1}^{K} h_k^{(t)}(\hat{\mathbf{z}}_s) \log p(\hat{\mathbf{z}}_s, k|\hat{\theta}) \end{aligned} \tag{22}$$

and

$$\frac{\partial Q(\hat{\theta}|\hat{\theta}^{(t)})}{\partial \hat{\theta}} = 0. \tag{23}$$

We finally apply a Deterministic Annealing EM algorithm [4] to learning algorithm of KOnet. It is represented by

$$w_k^{(t+1)} = \frac{1}{S}\sum_{s=1}^{S} h_k^{(t)}(\hat{\mathbf{z}}_s) \tag{24}$$

$$\hat{\mathbf{m}}_k^{(t+1)} = \frac{\sum_{s=1}^{S} \hat{\mathbf{z}}_s h_k^{(t)}(\hat{\mathbf{z}}_s)}{\sum_{s=1}^{S} h_k^{(t)}(\hat{\mathbf{z}}_s)} \tag{25}$$

$$\hat{\boldsymbol{\Sigma}}_k^{(t+1)} = \frac{\sum_{s=1}^{S}(\hat{\mathbf{z}}_s - \hat{\mathbf{m}}_k^{(t)})(\hat{\mathbf{z}}_s - \hat{\mathbf{m}}_k^{(t)})^{\mathrm{T}} h_k^{(t)}(\hat{\mathbf{z}}_s)}{\sum_{s=1}^{S} h_k^{(t)}(\hat{\mathbf{z}}_s)} \tag{26}$$

where

$$h_k^{(t)}(\hat{\mathbf{z}}_s) = \frac{\{w_k^{(t)} N_d(\hat{\mathbf{z}}_s, \hat{\phi}^{(t)})\}^{\beta}}{\sum_{s=1}^{S}\{w_k^{(t)} N_d(\hat{\mathbf{z}}_s, \hat{\phi}^{(t)})\}^{\beta}}. \tag{27}$$

In the learning algorithm of KOnet, the parameter $\theta = \{\mathbf{w}, \phi\}$ is not derived directly but a part of the parameter $\hat{\phi}$ of ϕ is derived by DAEM algorithm. Thus covariance matrix Σ'_k is made to be $\mathrm{diag}(\sigma_k^{ii})$, learning become stable compared with considering non-diagonal elements.

4 Mechanism for Skill

In special case such as $K = 1$, the inverse model of KOnet is represented by

$$\mathbf{x}_t = \mathbf{m}_1^x + \mathbf{D}_1^x(\mathbf{y}_t - \mathbf{m}_1^y) \tag{28}$$

where

$$\mathbf{D}_1^x = \mathbf{C}_1^{xy}\mathbf{C}_1^{y^{-1}}. \tag{29}$$

Now that, we regard $\mathbf{x}_t$ as $\hat{\mathbf{x}}_{t|t}$, $\mathbf{m}_1^x$ as $\hat{\mathbf{x}}_{t|t-1}$ and $\mathbf{m}_1^y$ as $\hat{\mathbf{y}}_{t|t-1} = \mathbf{C}_t\hat{\mathbf{x}}_{t|t-1}$, then we have

$$\mathbf{C}_t = \mathbf{m}_1^y\mathbf{m}_1^{x+} \tag{30}$$

$$\mathbf{P}_{t|t} = \mathbf{D}_1^x(\mathbf{C}_t^{\mathrm{T}})^+ \tag{31}$$

$$\begin{aligned}\mathbf{A}_{t-1}\mathbf{P}_{t-1|t-1}\mathbf{A}_{t-1}^{\mathrm{T}} \\ = \mathbf{P}_{t|t}(\mathbf{I}_n - \mathbf{D}_1^x\mathbf{C}_t) - \mathbf{I}_n \end{aligned} \tag{32}$$

$$\mathbf{B}_{t-1} = (\mathbf{m}_1^x - \mathbf{A}_{t-1}\mathbf{x}_{t-1})\mathbf{u}_{t-1}^+ \tag{33}$$

where $\mathbf{x}^+ = \mathbf{x}^{\mathrm{T}}(\mathbf{x}\mathbf{x}^{\mathrm{T}})^{-1}$ denotes a generalized inverse matrix. $\mathbf{I}_n \in \Re^{n \times n}$ is a unit matrix.

5 Summary

In this paper, we proposed Knowledge Option network (KOnet) to acquire impedance in a coefficient matrix of a state equation. After estimating suitable impedance for a task, we presented a mechanism to generate motor planning instruction by using KOnet. from using a feedback control with sensory information to using feedfoward control with a prediction at internal model.

References

[1] T. Tsuji, M. Nishida and K. Ito, "Iterative Learning of Impedance Parameters for Manipulator Control Using Neural Networks", *SICE*, Vol 28, No. 12, pp. 1461-1468, 1992.

[2] S. Mitsuyama, J. Nishii, Y. Uno and R. Suzuki, "A Control Model for Arm Movements with a Stiffness Adjusting Mechanism", *Tech. Report of IEICE*, NC92-122, pp. 213-220, 1993.

[3] Y. Sakaguchi and K. Nakano, "Motor Planning According to Reliability of Internal Model," *Proc. of IJCNN-93-Nagoya*, Vol. 2, pp. 1321-1324, 1993.

[4] S. Ueda and R. Nakano, "A Deterministic Annealing EM Algorithm and Its Application to Probabilistic Neural Networks," *Tech. Report of IEICE*, NC95-10, pp. 73-80, 1995.

Direct-Vision-Based Reinforcement Learning in a Real Mobile Robot

Masaru IIDA   Masanori SUGISAKA   and Katsunari SHIBATA
Dept. of Electrical and Electronic Engineering, Oita Univ.
700 Dannoharu, Oita, Japan. email:iida@cc.oita-u.ac.jp

Abstract

In this paper, it was confirmed that a real mobile robot with a simple visual sensor could learn appropriate actions to reach a target by Direct-Vision-Based reinforcement learning (RL). The learning was done on-line without any advance knowledge and any helps of humans. In Direct-Vision-Based RL, row visual sensory signals are put into a layered neural network directly, and the neural network is trained by Back Propagation using the training signal that is generated based on reinforcement learning. By a character of the visual sensor, the robot can not distinguish the target object from the background when the target is located on the right of and just in front of the robot. However, the robot could obtain the actions to avoid such states and to reach the target.

1 Introduction

Reinforcement learning is an attractive method as an autonomous learning for autonomous robots, and is utilized to obtain the appropriate mapping from state space to action space. By combining reinforcement learning and a neural network, continuous states and actions can be dealt with because non-linear functions with continuous input and output values can be approximated by the neural network. This combination has been applied to non-linear control tasks[1][2] and games[3].

Among many kinds of sensors for a robot, a visual sensor has a lot of sensory cells, and gives huge pieces of information about environment to the robot. Our human also depends deeply on the visual information to know the environment state. Asada et al. applied reinforcement learning to real soccer robots with a visual sensor[4]. In this case, the state space was divided into some discrete states by pre-processing the visual sensory signals, and the robot learned appropriate actions for each state by Q-learning.

On the other hand, in Direct-Vision-Based reinforcement learning (RL), the whole process from sensors to motors are computed by a layered neural network[5]. Raw visual sensory signals are put into a layered neural network directly, and the neural network is trained by Back Propagation. However, since the training signal is generated autonomously inside the robot based on reinforcement learning, the learning is still autonomous. By this learning, not only the motion planning, but also a series of processes from sensors to motors including recognition, can be learned synthetically. It is reported that when a robot learns actions to reach a target, spatial information is represented adaptively on the hidden layer after learning. Moreover, it was examined that the learning is faster and more stable than that when pre-processed signals are utilized as inputs[6].

The effectiveness of Direct-Vision-Based RL mentioned above has been confirmed only on some simulations. In this paper, it is shown that a real mobile robot with a monochrome visual sensor can learn appropriate motions from scratch without any advance knowledge in "going to a target task".

2 Actor-critic architecture

Here, actor-critic architecture[7] is employed, and actor(action command generator) and critic(state evaluator) are composed of one layered neural network. This means that the hidden layer is used by both actor and critic. This architecture is the same as the simulations in [5]. TD(Temporal Difference) is applied for the learning of the critic. TD error is defined as

$$\hat{r}_t = r_t + \gamma P_t - P_{t-1}, \qquad (1)$$

where γ : a discount factor, r_t : reward, P_t : state evaluation value. The evaluation value at the previous time P_{t-1} is trained by the training signal as

$$P_{s,t-1} = P_{t-1} + \hat{r}_t = r_t + \gamma P_t, \qquad (2)$$

where $P_{s,t-1}$ is the training signal for the evaluation value. On the other hand, the motion command of the robot is the sum of the outputs of $\mathbf{a}_t$ and random numbers $\mathbf{rnd}_t$ as trial and error factors. The motion command $\mathbf{a}_{t-1}$ is trained by the training signal as

$$\mathbf{a}_{s,t-1} = \mathbf{a}_{t-1} + \hat{r}_t \ \mathbf{rnd}_{t-1}. \qquad (3)$$

The neural network is trained by Back Propagation according to Eq(2) and (3). By this learning, motion commands are trained to gain more evaluation value. Here, the layered neural network has one hidden layer

and 3 output units. One of the outputs is for critic, and the other two are for actor. The output function of each hidden or output neuron is sigmoid function whose output range is from -0.5 to 0.5.

3 Experimental system and environment

Fig.1 shows the robot with a monochrome visual sensor (Khepera and K213 Vision Turret) used in this paper. The specifications of Khepera and K213 Vision Turret are as follows.

Height : **55mm**
Diameter : **33mm**
Interface with PC : **RS232C(serial port)**
Transmission rate : **38400 bps**
Sensor cell : **64**
Resolution : **256 gray scale**
Visual field : **36 degree**

This visual sensor is composed of two parts (Fig.2), image perception optics and light intensity detector optics. The light optics detects light intensity around the robot at first, and then image perception optics adjusts image sensory outputs according to the light intensity. Therefore, when the light intensity is not strong enough, all the pixel values become almost white, and as a result, the robot could not distinguish bright points and dark points. In the other words, when the target is located just in front of and on the right side of the robot, the robot loses the target.

Fig.3 shows experimental environment. The action area has 70×70cm which is surrounded by a height of 10cm white paper wall, and a fluorescent light is set to keep enough light intensity. The target in the task stands 8cm tall with a diameter of 2.5cm which is wrapped black paper around.

4 Application to a real robot

4.1 Coping with a time delay

When Direct-Vision-Based RL is applied to a real robot, a time delay should be considered, while it does not have to be considered in simulations. PC receives visual sensory signals from the real robot through RS232C serial port, and its transmission rate is not fast enough. The necessary time to execute each command is as follows.

Transmission of visual sensory signals : **90msec**
Transmission of action commands: **10msec**
Computation of neural network : **less than 1msec** (for both forward and learning)

Considering the measurement interval of the visual sensor, sampling time is set to be 300msec. If computation of the neural network and transmission of the
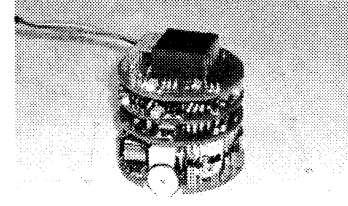
Figure 1: A picture of Khepera with K213 Vision Turret
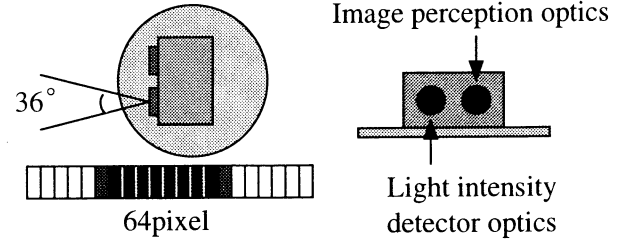
Figure 2: K213 Vision Turret
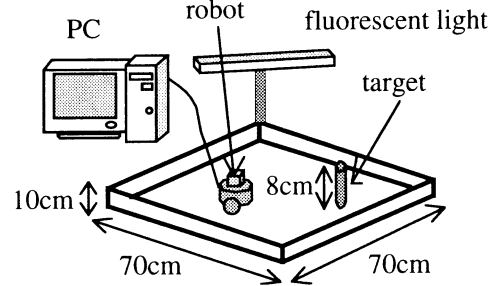
Figure 3: Experimental environment

action command are done just after the transmission of the visual sensory signals, the robot continues to move with the previous action commands during the transmission of the visual sensory signals. Then, the robot location obtained from the visual sensory signals is different from the robot location when the next action command is transmitted. Here, in order to reduce this influence, the visual sensory signals are transmitted just after the action command. Fig.4 shows the timing chart of updating of each value and system events in this experiment. Therefore, P_t is influenced by the action a_{t-2} on behalf of a_{t-1}. The learning of critic is done by Eq(2) that is the same as the simulation. On the other hand, the action command at two steps before is trained by the training signal as

$$a_{s,t-2} = a_{t-2} + \hat{r}_t \ \mathbf{rnd}_{t-2} \qquad (4)$$

on behalf of Eq(3).
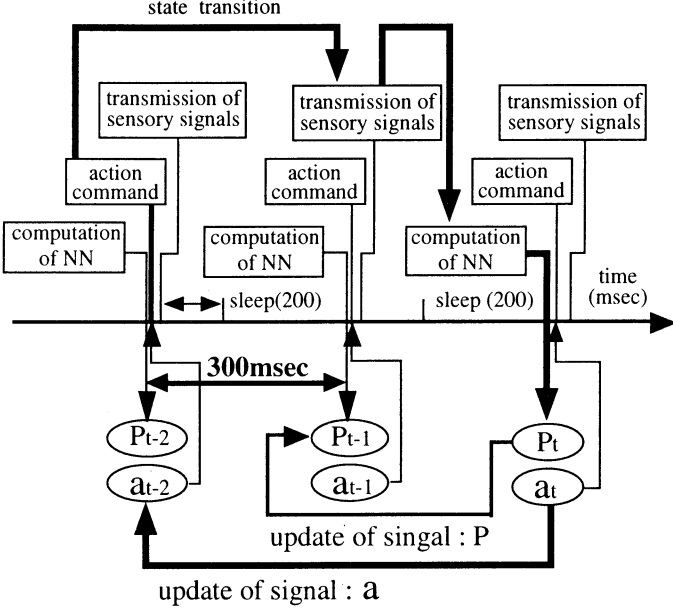
Figure 4: Timing chart of the learning for the real robot

Figure 5: Distribution of evaluation value and the trajectory of the robot. These figures are drawn on the coordinates of *center pixel* and *width*.

4.2 Discrete actions

Since the action command for each wheel of Khepera should be an integer, the continuous action value is divided into an integer as

$$speed_t = (int) \; 4 \cdot (2 \cdot \mathbf{a}_t + \mathbf{rnd}_t), \qquad (5)$$

$$if(speed_t \leq -3) \; speed_t = -3$$
$$if(speed_t \geq 3) \; speed_t = 3$$

$-3 \leq speed_t \leq 3$, $-0.5 \leq output_t \leq 0.5$, $-0.4 \leq rnd_t \leq 0.4$, where *speed* : action command for the robot.

5 Experiment

5.1 Task

In this paper, the task that the real mobile robot with monochrome visual sensor reaches a target, is employed. Here, 3-layered neural network has 64 input units, 30 hidden units, and 3 output units. One of the outputs is for critic, and the other two are for actor. Before learning, the input-hidden connection weights are small random numbers, and all the hidden-output connection weights are 0.0. After the transmission of the visual sensory signals, each of them is binalized with the boundary value of 85, and the number of pixels of the dark area is defined as *width* of the target in the robot's view. The central pixel number of the dark area is defined as *center pixel*. In learning, initial *width* and *central pixel* is chosen randomly from

$5 \leq width \leq 29$ and $5 \leq center \; pixel \leq 59$. From the initial position, the robot can always get the whole object on its visual sensor. Then, the robot can go to the initial position by itself according to a given program. At the beginning of learning, since the robot moves only according to the random numbers, the robot is located within the range that is close to the target. As the learning progresses, the range of the initial robot location becomes wider gradually. When $30 \leq width$ and $21 \leq center \; pixel \leq 41$, the state evaluation output is trained to be 0.4 as a reward. When the target disappears out of the visual field, it is trained to be -0.4 as a penalty. Otherwise each trial is stopped at 150 time step even if the robot can not reach the target object. The evaluation value is the sum of the evaluation output and 0.5, and the discount factor is 0.99.

5.2 Learning result

Fig.5 shows the state evaluation value after learning. This figure is drawn by computing the outputs off-line for 35 sample sets of visual sensory signals. The vertical axis indicates the *width* and the horizontal axis indicates the *center pixel* of the target object on the robot's view. It is seen that as the learning progress, the distribution of the state evaluation value is formed gradually. As shown in Fig.5, the state evaluation value becomes larger when the *width* of the target becomes larger on the visual sensor. It is smaller
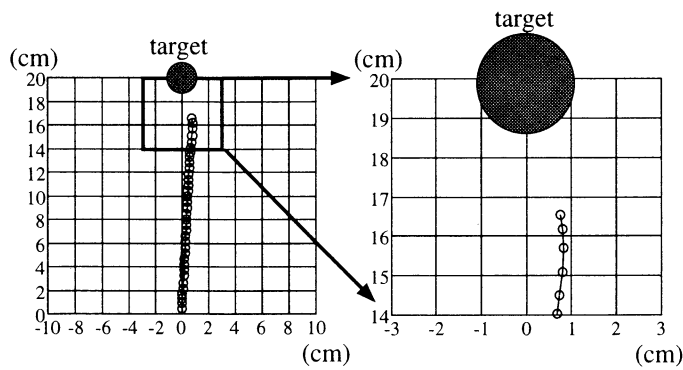
Figure 6: The locus of the robot on the field after 4000 trials

when the target is shown on the right side of the visual sensor than on the left side. The reason is that the robot misses the target when it is shown on the right side of the sensor as mentioned in the section 3. Fig.5 (c), (d) show the locus of the target object in the robot's view, and the robot approaches the target while looking at it on the left side of the visual sensor.

Next, Fig.6 shows the locus of the robot on the absolute coordinates after 4000 trials (the same trial as Fig.5 (d)). When the robot comes near the target object, the robot catches the target object on the center of the robot's view by rotating counterclockwise. Fig.7 shows the photo of the robot to that the robot reaches the target object without missing. As the result of the learning, the robot could obtain the action to overcome the defect of the visual sensor.

6 Conclusion

It was shown that the real robot with a monochrome visual sensor could obtain reaching actions to a target object through the learning from scratch without any advance knowledge and any helps of humans by Direct-Vision-Based reinforcement learning. The robot could obtain the actions to avoid missing of the target that happens due to a sensor character.

Acknowledgements

References

[1] Anderson, C. W. (1989) "Learning to Control an Inverted Pendulum Using Neural Networks", *IEEE*
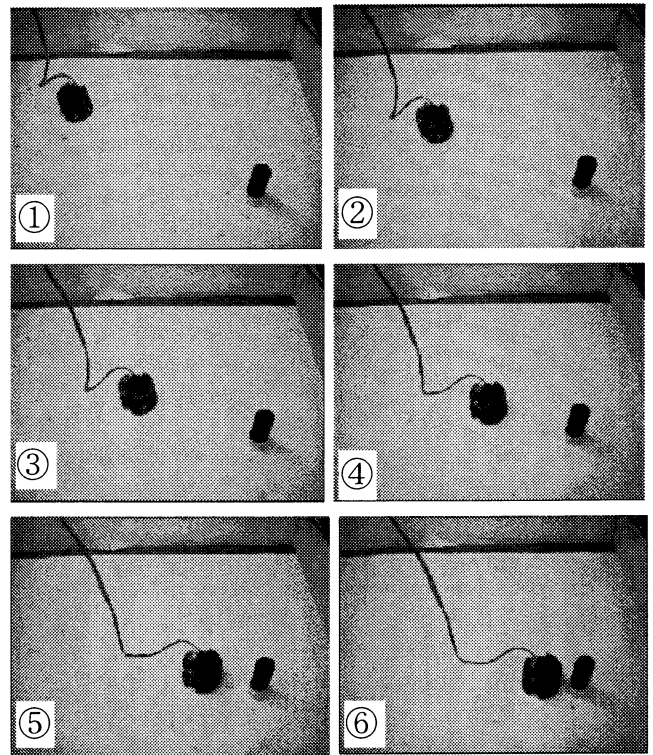
Figure 7: The robot succeeded in reaching a target object

Control System Magazine, Vol. 9, pp.31-37

[2] Morimoto, J. and Doya, K. (2000) "Acquisition of Stand-up Behavior by a Real Robot using Hierarchical Reinforcement learning ", *5th Robotics Symposia*, pp.397-402 (in Japanese)

[3] Tesauro, G. J. (1992) "Practical Issues in temporal difference learning", *Machine Learning*, 8, pp.257-277

[4] Asada, M., Noda, S., Tawaratsumida, S. and Hosoda, K. (1996) "Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning",*Machine Learning*, Vol.24, pp.279-303

[5] Shibata, K., Ito, K. and Okabe, Y.(1998) "Direct-Vision-Based Reinforcement learning "Going to a Target" Task with an Obstacle and with a Variety of Target Sizes", *Proc. of NEURAP'98*, pp.95-102

[6] Shibata, K., Sugisaka, M. and Ito, K. (2001) "Fast and Stable Learning in Direct-Vision-Based Reinforcement learning", *Proc. of AROB 6th*, Vol.1, pp.200-203

[7] Barto, A. G., Sutton, R. S. and Anderson, C. W. (1983) "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems", *IEEE Trans. SMC-13*, pp.835-846

Effectiveness of Sensory Motion
in the Learning of Capturing Task of a Moving Object

Shin'ichi MAEHARA, Masanori SUGISAKA, Katsunari SHIBATA

Dept. of Electrical and Electronic Engineering, Oita Univ. 700 Dannoharu, Oita 870-1192, Japan

Abstract

Acquisition of actions based on prediction, which needs some context information, is important for a robot in dynamic environment. It is considered that a sensory motion is useful to avoid missing a moving object and to know its motion easily. In this paper, a capturing task of a moving object is employed as an environment, in which both robot and target move. Here, the appropriate actions in this task are learned by the combination of Elman-type recurrent neural network[1] and reinforcement learning. In this paper, the effect of the sensory motions is focused on. Three kinds of sensory motions, (1) looking to a constant direction in absolute coordinates, (2) keeping the object in the center, and (3) fixed on the robot, are employed, and the learning results are compared. Simulation result is shown that the robot captured the object faster in the case of (1) and (2) than (3).

1 Introduction

Our humans can know a motion of a target, and obtain appropriate actions based on prediction, even if both ourselves and target move. For example, when a baseball player catches a fly, he detects the motion of the ball from his visual sensors, at first, then goes beforehand to the falling point of the ball, and catches the ball finally. Acquisition of such actions is important for a robot in dynamic environment.

Since the prediction is done to realize more appropriate actions towards some purpose, it can be regarded as "a way to achieve a certain purpose". Then, it is considered that such actions can be obtained according to the necessity by learning of the action for the purpose. On the other hand, in the conventional prediction methods, such as Kalman filter for liner systems, and some other methods using fuzzy or neural network, prediction itself is an object, and what should be predicted has been given by humans.

Recently, reinforcement learning that can learn a series of actions towards some purpose has been focused. By the combination of reinforcement learning and neural network, continuous inputs and outputs can be utilized. Also, it has expected that a series of

processes from sensors to motors can be synthetically learned by this combination.

When both robot and target move, it is difficult for a robot to know the target movement. The robot may miss a moving object because there is a gap between the movement direction and the target direction. If the robot miss an object, it cannot know the present state of the target. Moreover a robot must compensate its notion to know the target motion, because the signals from the visual sensor is influenced by robot motions. Thus the appropriate sensory motions are necessary.

In this paper, it is shown in a capturing task of a moving object that actions based on prediction can be obtained through learning. Then, the sensory motion is focused on, it is verified what sensory motion is more effective.

2 Reinforcement learning
using recurrent neural network

The environment, in which the present states cannot be identified only from the present observations, can be regarded as POMDP(Partially Observable Markov Decision Process). Only from the present visual sensory signals, the location of the target can be observed, but its velocity cannot be obtained. POMDP cannot be utilized in reinforcement learning unless some context information.

Some architecture to combine reinforcement learning and recurrent neural network have been proposed already[2][3]. There are two points to classify these methods. One of them is whether the architecture of reinforcement learning is Q-learning or Actor-critic. Q-learning needs to quantize its action, and Actor-critic doesn't need it. The other point is whether the model is introduced to extract the some context information or not. When the model is employed, huge amount of calculation is necessary, and the most part of the extracted context may not be required by the given task. If an essential part of the information, which used as inputs and outputs of the model, are chosen by humans, then autonomous learning ability is spoiled.

Thus, in this paper, Actor-Critic Elman Network (Fig.1) is employed, and model is not used.
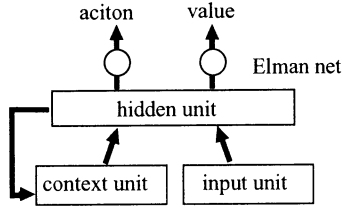
Figure 1: Actor-Critic Elman Network.

3 The learning of capturing actions of moving target

3.1 Acquisition of actions based on prediction

For acquisition of actions based on prediction, some context information is needed. If a robot can know a target motion using some context information, it is expected that the robot turns according to the target velocity at first, then goes straight to the place where the robot will get the target (Fig.2(a)). Nishioka has shown that a mobile robot with a visual sensor can obtain appropriate actions using Actor-Critic Elman Network[4]. However, the robot in that paper can move to any directions without any constraints. Since the direction of the sensor is always fixed to one direction on the absolute coordinates, there is no concept of rotation. Normally, a robot body rotates according to the change of its moving direction. If a visual sensor is fixed on the robot, the direction of the sensor changes together with the robot.

3.2 The effect of sensory motion

When both robot and target move, the robot must compensate its motion on the visual sensor. If the knowledge about the change of the visual signals according to the robot motion is not given in advance, it is difficult for a robot to know the target movement. Furthermore the robot may miss the moving target as mentioned above when the robot moves based on prediction (Fig.2(b)). Thus appropriate sensory motions are required. Then, it is verified that what sensory motion is more effective.

Some eye's motions can be observed in humans. VOR(Vestibulo-Ocular Reflex) and OFR(Ocular Following Response) can be considered to be useful to obtain a stable vision image by removing the motion of the background. While, smooth pursuit eye movement is useful to keep the object in the center of the sensor.

By getting a hint from these eye movements, three kinds of sensory motions, (1)to keep a constant direction in absolute coordinates (Fig.2(c-i)), (2)to keep the object in the center (Fig.2(c-ii)), and (3)fixed on

(a) The forestall actions based on prediction

(b) Missing of the target in the fixed visual sensor

(c-i) Looking to a constant direction

(c-ii) Keeping the object in the center

(c-iii)Fixed on the robot

(c) Three sensory motions in this paper
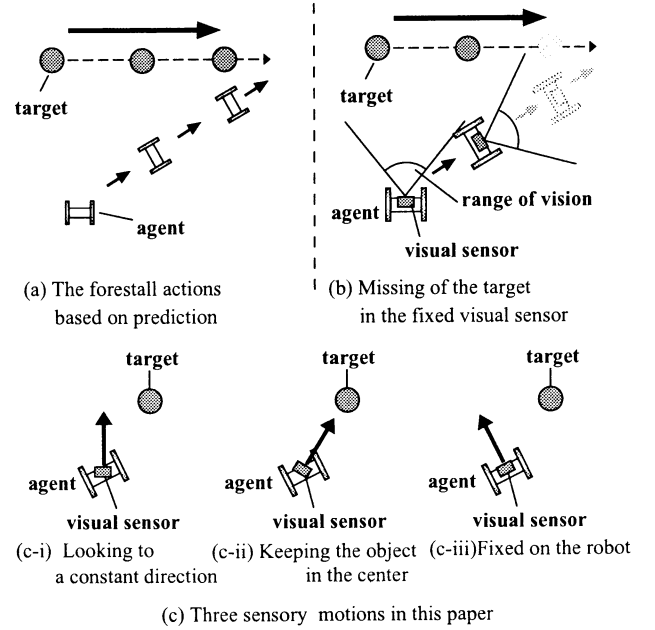
Figure 2: Actions based on prediction.

the robot (Fig.2(c-iii)), are employed here. (1) is employed to obtain a stable vision image, and this sensory motion is the same as that of Nishioka. These three sensory motions are used in a capturing task of a moving object, and learning results are compared.

4 Simulation

4.1 Task setting

In the task here, an object moves with a constant velocity in the direction of x axis, and the wheel-type mobile robot with visual sensor is requiring to get the object as soon as possible. The robot can go straight and rotate, but cannot go laterally without rotation. Simulation environment is shown in Fig.3. The wheelbase diameter of the robot is 2.0, its visual sensor is composed of 50 sensory cells, and the sensor covers all directions. Therefore, the robot never misses the target. Since there is no overlaps of sensory cell, receptive field of one sensory cell is 7.2 degrees. The target is located on $x = 0.0$, $0.01 \le y \le 5.0$ at time step $t = 0$, it moves with a constant velocity v, which is chosen from randomly within the range of $-0.25 \le v \le 0.25$ at every trial. Since the maximum speed of the robot is 0.4, it can always capture the target even if the robot runs after the target. However, in the case that the robot rotates and goes along the short cut, the robot can reach the target faster than in the case that it runs after the target. The robot doesn't move for the first 3 time steps to detect the target motion. It is assumed that the robot reaches the target and gets 0.4

visial sensor
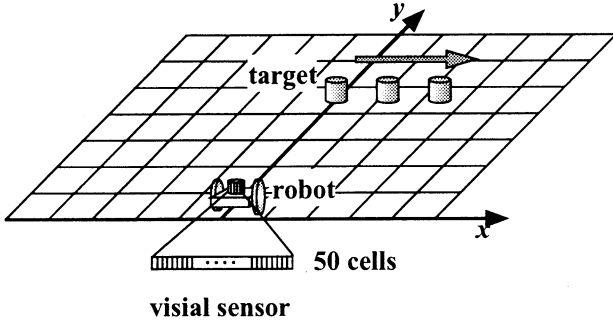
Figure 3: Simulation environment of "capturing a target" task for a robot with a visual sensor.

Figure 4: Robot's routes after 200,000 trials of learning. The initial position is $(x, y) = (0.0, 4.0)$, the velocity of the target $v = \pm 0.20$.

as a reward when the robot goes through the center of target.

The action is learned by an Actor-critic Elman Network. In this neural network has only one hidden layer that is composed of 30 neurons. This neural network is used for both actor and critic, One of the three output units is for the value(Critic), and the other two are for the motion command of the wheels(Actor). The system learns using BPTT(Back Propagation Through Time)[5]. The steps back in time for BPTT is 30. Initial values of all the hidden unit are 0.0 before the learning. It has been shown that the learning is faster and more stable by using row visual sensory signals as inputs of a neural network[6]. Then, the visual sensory signals are put into the neural network directly. However, the robot cannot know its motions only from visual inputs, then the angle between the robot direction and the y axis, and the speed of both wheels at 1 step before are given. In the case of sensory motion to keeps the object on the center of visual sensor, the target motion cannot be obtained from the visual sensory signals. Then, the angle between the direction of visual sensor and the y axis is also given. It has been shown that Sigmoid-based neural network can approximate easily strong non-linear functions by using localized signals as inputs[7]. Therefore, the each of continuous inputs is localized by 10 RBF units, and then they are put into the neural network.

4.2 Result

Robot's routes after 200,000 trials of learning are shown in Fig.4. The initial position of the target is $(x, y) = (0.0, 4.0)$, and the velocity of the target $v = \pm 0.20$. Fig.4 shows that the robot could get the target faster when the sensor is moved than when the sensor is fixed on the robot. In the case that the sensor is moved, the route is almost a straight line. It shows that the robot took the short cut and got the target at $\triangle$ or $\bigcirc$. While, in the case that the sensor is fixed on the robot, the route is curved. It shows that the
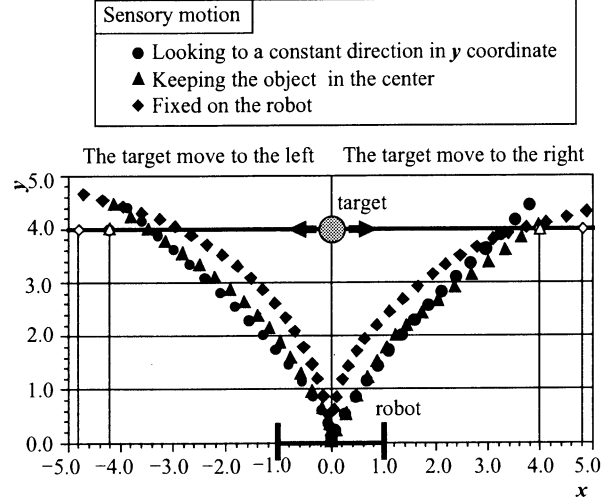
robot ran after the target and it got the target at $\Diamond$.

Motion commands to the wheels when the robot took the trajectory are shown in Fig.4. In the case that the sensor is moved, the sign of motion commands are different each other at the first time step, and then both motion commands become large. It shows that the robot turned according to the target velocity at first action, and then went straight short cut. While, in the case that the sensor is fixed on the robot, both commands always have a same sign, and one of them cannot be the maximum value at the early steps.

5 Discussion

When both robot and target move, the robot must compensate its motion to know the target motion. In the case that the visual sensor is fixed on the robot, the change of the visual sensory signals is large when the robot rotates. Especially, if the distance between the robot and target is large, this change is much larger than the target motion. Thus the detection of the target motion becomes difficult. This is supported by the learning result that the robot could go straight and captured the target when the target was located near the robot. While, in the case that the sensor is moved, the influence of rotation isn't so serious.

Since there is a gap between the movement direction and the target direction, the robot may miss the target when the visual sensor is fixed on the robot as shown in Fig.2(b). In the case of the sensory motion to keep a constant direction, the robot misses the target when a gap between the direction of the visual sensor and the target direction is large. While, in the case of the

(a)Looking to a constant direction

(b)Keeping the object in the center

(c)Fixed on the robot
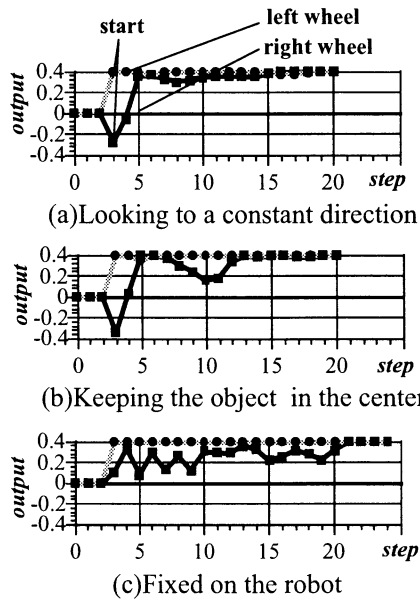
Figure 5: Motion commands to the wheels when the robot took the trajectory in Fig.4. The velocity of the target $v = 0.2$.

sensory motion to keep the object in the center, the robot doesn't miss the target even if the target moves any directions.

In the visual sensor of humans, sensory cells are densely assigned in part of the center on the retina, and we can move this part to appropriate place. This motion is useful to obtain the information of the target effectively. To introduce such distribution, the sensory motion to keep the object in the center is effective.

When the sensory motion to keep the object in the center is used, the robot cannot know the target position only from the sensory signals, because the visual sensory signals are decided only according to the distance to the target from the robot. However, the learning result is almost the same as in the case of the sensory motion to keep constant direction. In this sensory motion, the angle between the target direction and absolute coordinates is utilized to know the target location. Thus the robot motion dosen't influencehe this angle, and it might be useful to know the target position. However, this angle cannot be obtained easily in the real environment. Then, the angle between the robot direction and the sensor direction should be used. Since this angle changes according to the robot motion, the detection of the target motion becomes more difficult. However, the distance to the target from the robot is not represented on the angle signals but on the visual signals. Thus, It is expected that the detection of the target motion is easier than when the sensor is fixed on the robot.

Finally, the effectiveness of fixed sensor on the robot

is discussed. When the robot avoids the obstacle that located on the route to its goal, the sensory motion is useful to detect the obstacle. Thus, it is considered that this function and other sensory motion are used alternately.

6 Conclusion

The paper presented that a sensory motion is important to obtain the actions based on prediction when both robot and target move. Through a capturing task of a moving object, it was shown that the performance was better when the sensor was moved than when the sensor was fixed on the robot. The potential of the sensory motion to keep the object in the center was also shown.

Acknowledgements

References

[1] J. L. Elman : Finding Structure in Time, Technical Report CRL 8801, Center for Research in Language, Univ. of California, San Diego (1998)

[2] S. D. Whitehead and L. J. Lin : Reinforcement learning of non-Markov decision process, Artificial Intelligence 73, pp. 271-306 (1995)

[3] E.Mizutani and S.E.Dreyfus : Totally Model-Free Reinforcement Learning by Actor-Critic Elman Networks in Non-Markovian Domains, Proc. of IJCNN'98, pp. 2016-2021 (1998)

[4] T.Nishioka : Model-TD Reinforcement Learning for Action acquisition under Dynamic Environment, Master thesis of Dept. of Computational Intelligence and Sys. Sci. Tokyo Inst. of Tech. (1999)

[5] D. E. Rumelhart, G. E. Hinton, R.J. Wiliams : Learning Internal Representations by Error Propagation, Parallel Distributed Processing, The MIT Press, pp. 318-362 (1987)

[6] K.Shibata, M.Sugisaka and K. Ito : Fast and Stable Learning in Direct-Vision-Based Reinforcement Learning, Proc. AROB 6th'01, Vol1, pp.200-203 (2001)

[7] S.Maehara, M.Sugisaka and K.Shibata : Reinforcement Learning Using Gauss-Sigmoid Neural Network, Proc. AROB 6th'01, Vol2, pp.562-565 (2001)

Knowledge-Based Unmanned Automation and Control Systems for the SBR Wastewater Treatment Process

Hyeon Bae, Jae-Ryong Jung, Sungshin Kim, Man Hyung Lee[*] and Chang Won Kim[**]

School of Electrical and Computer Engineering, Pusan National University, Pusan, Korea
[*]School of Mechanical Engineering, Pusan National University, Pusan, Korea
[**]Department of Environmental Engineering, Pusan National University, Pusan. Korea

Pusan National University 30 Changjeon-dong, Keumjeong-ku, Pusan 609-735, Korea
E-mail: baehyeon@pusan.ac.kr, sskim007@pusan.ac.kr, mahlee@pusan.ac.kr, cwkim@pusan.ac.kr

Abstract –Wastewater treatment processes are usually located in rural seclusion. In this paper, unmanned and automated control system is designed for the SBR (Sequencing Batch Reactor) wastewater treatment pilot plant. The pilot plant is constructed in the countryside, which is little far from a main city. Networks and wireless modules are employed for the data transmission. A local controller is equipped in the SBR pilot plant as a client and a monitoring system is located in the other place as a server. The communication parts consist of ADSL (Asymmetric Digital Subscriber Line) network and CDMA (Code Division Multiple Access) module. Remote control and monitoring system are constructed at a laboratory in a metropolis. In this paper, a fuzzy inference system is applied which is constructed by the operator's knowledge and acquired sensor data, for determining the threshold and influent.

Keywords–SBR, remote monitoring, fuzzy inference system

I. INTRODUCTION

Wastewater treatment processes (WWTP) are usually located in the places, which are away from the haunts of men. Expert operators should deal these processes with continuous maintenance. These limited conditions make some problems for handling wastewater treatment plants. If the size of the plant is huge, many operators would be employed for the plant. On the other hand, if the size of the treatment plant is small, it is difficult to operate the process under the good condition for a long time. Because it is not easy to employ a good operator with a low salary.

In this paper, unmanned and automated control system is designed for the SBR (Sequencing Batch Reactor) wastewater treatment pilot plant. The pilot plant is constructed in the countryside, which is little far from a main city. Networks and wireless modules are employed for the data transmission. A local controller is equipped in the SBR pilot plant as a client and a monitoring system is located in the other place as a server. The communication parts consist of ADSL (Asymmetric Digital Subscriber Line) network and CDMA (Code Division Multiple Access) module. Remote control and monitoring system are constructed at a laboratory in a metropolis.

Sensing data from plant sensors are translated to the remote site using communication modules, and then the data could be displayed and analyzed by means of the remote monitoring and control systems. These remote control systems could play important major role in which the plant is in the abnormal conditions. If the plants have problems or faults, the remote system could diagnose the status of plants and then send warning messages to operators who are staying in the other place using CDMA wireless modules.

The SBR wastewater treatment plant is for piggery wastewater. This treatment method is little bit different with the method for normal wastewater treatment systems because contained ingredients are much different with wastewater. In this paper, the real time and time distribution control methods are applied. Control inputs are obtained from plant sensors which are for ORP (Oxidation Reduction Potential), pH, DO, NH4, etc. Among ORP, pH, DO (Dissolved Oxygen), and NH4 (ammonia), ORP data is used for the real time and time distribution controls. Because ORP could represent the feature of treatment process during the operation time. For example, an ORP curve is increasing in aerobic steps but which is decreasing in anoxic steps. These two steps are typical treatment processes for the normal wastewater treatment systems. The curves have usually two bends. One represents DO saturation point in which NH4 ingredients are almost removed. The other bending point shows that nitrogen materials are changed to nitrogen gas and evaporated to air. One treatment cycle is finished after these two stages. The operation cycle times for aerobic and anoxic steps will be optimized
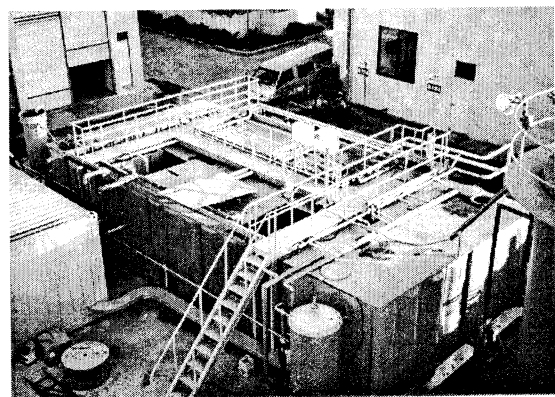
Fig. 1. Picture of the experimented SBR pilot plant.

because the operation time is very important to save energy and process states.

In the future experiment, integration control systems will be employed for several scattered wastewater treatment plants. And also the fuzzy inference system will be applied for the diagnosis of the overall process. The ORP curve, sensor signals, and pump states could be the input variables of the reasoning system. The fuzzy diagnosis system will analyze how the plant is operated and make a decision how it is controlled under the good condition.

II. SBR WWTP

Batch operated activated sludge processes, now commonly known as SBRs are often proposed as alternatives to conventional continuously operated plants. The recent availability of reliable process control systems has overcome many of the operational problems previously reported. The characteristics and performance of SBR systems have been extensively reviewed by Irvine and Ketchum. Several full-scale SBR installations exist in the USA and some of these plants serve quite large populations [1], [2], [3]. The operating principles of a batch activated sludge plant are illustrated in Fig. 2.

The performance of the WWTP in Fig. 2 is depends on a time sequence of operations, because the reaction is occurred in the same process tank. The various stages in the sequence are as follow.

- Fill: the aeration tank is filled with the influent wastewater. In batch-operated activated sludge plants the micro-organisms should experience a progressively decreasing substrate concentration as treatment proceeds in order to ensure the production of sludge with good settling properties. Influent sewage should therefore be admitted into the tank in a rapid, controlled manner to maintain a suitable substrate concentration gradient. Anoxic conditions will result if no aeration occurs during the fill period.
- React: during this period the micro-organisms utilize BOD and ammonia nitrogen. The length of the aeration period determines the degree of treatment. The oxygen demand decreases uniformly as treatment proceeds.
- Settle: aeration is stopped and the sludge settles leaving clear, treated effluent above the sludge blanket.
- Decant: effluent is removed from the tank without disturbing the settled sludge.
- Idle: aeration is restarted and proceeds until it is time to commence the cycle again with the filling stage.
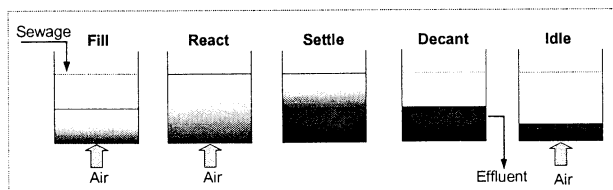
III. FUZZY INFERENCE SYSTEM

Fuzzy logic can be used for controlling a process that is nonlinear or ill-understood to apply the conventional controller design methods. And it enables control engineers to easily implement control strategies used by human operators. Fuzzy logic is a technology for developing intelligent control and information systems. Fuzzy logic achieves machine intelligence by offering a way for representing and reasoning that human knowledge that is imprecise by nature. Even though fuzzy logic is not the only technique for developing AI systems, it is unique in its approach for explicit representation of the impreciseness in human knowledge and problem solving techniques [4].

IV. SCHEME OF TOTAL PLANT CONTROL

As a tested in this paper, overall system is constructed with two wastewater treatment plants. One is real pilot plant that is constructed in the countryside. The system is controlled using PLC for the aerobic and anoxic stages. When the plant is operated, several sensor signals are detected and stored in database. This database values are transferred to a remote monitoring system by ADSL or CDMA module. These are combined with wire and wireless communication approach. CDMA module is for data backup in the abnormal network condition. FTP demon is used for wire communication system. It is good for the stable transmission and security.

On-line monitoring of ORP has been proved to be a practical and useful technique for process control of wastewater treatment systems. In this paper, the remote monitoring system is constructed that could display the transmitted data and diagnose the system status with analyzing the data.

The remote diagnosis system is applied for analyzing

Fig. 2. Process of the SBR plant.
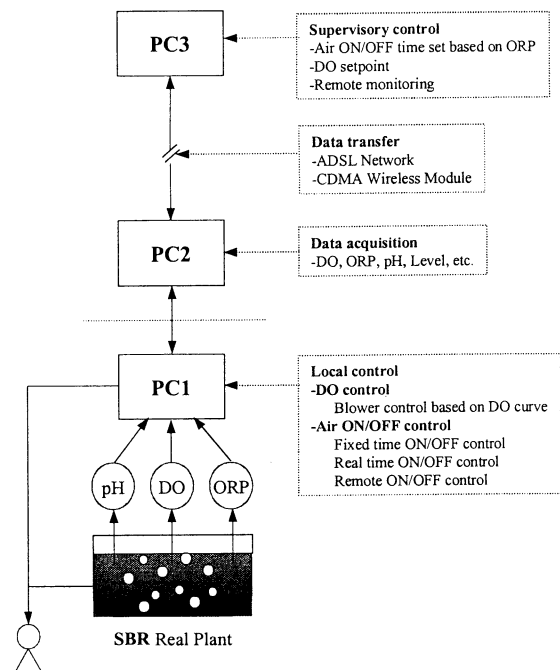
Fig. 3. Scheme of the total plant control.

ORP and DO curve. ORP and DO concentrations are very important values to operate the SBR wastewater treatment system. When the aerobic process is just finished, the ORP and DO values are increased steeply. Therefore, this condition could indicate that the aerobic process is all done. And this state gives us clues that the time to turn off the blower. In this paper, the fuzzy logic is applied for the diagnosis of the system status. For instance, DO curve is analyzed for determining the on/off threshold and influent volume.

A. Data Acquisition from Plant

In this paper, 4 sensors are assembled for obtaining the plant data. These data are transferred from the plant to PC1 shown in Fig. 3, which is for SCADA (Supervisory Control and Data Acquisition). There are the other kinds of data, which show the states of motors and pumps. And PLCs (Programmable Logic Controllers) are equipped for the local control of motors, pumps, and other devices.

In this paper, the SBR plant is controlled using ORP and DO signals. When blowers are turned on, the trend of ORP and DO concentrations could be changed. This state indicates that the aeration process is going on. After blowing enough for the nitrification, the ORP and DO curve is rapidly raised. This is the feature of the ending point of the removal ammonia process. Therefore, this situation points out that the blower should be turned off in this point.

B. Data Transfer from PC1 to PC3

After obtaining the sensor data, these data could be transmitted to PC2 using an internal network, which is easy way to be applied. OS supports this function.

Two applications are implemented for data transfer from PC2 to PC3 in this system. One is the way to use ADSL network line. This is a general application for the data transfer. The other is a technique to apply CDMA wireless modules. This is for the system emergence case. If the ADSL network line is in the trouble, the CDMA module has to be run to transfer the data. Wireless modules are very useful equipments for data communication. Especially, the plant is located in the area that has no network line. And a text message could also be sent and received using these modules. For
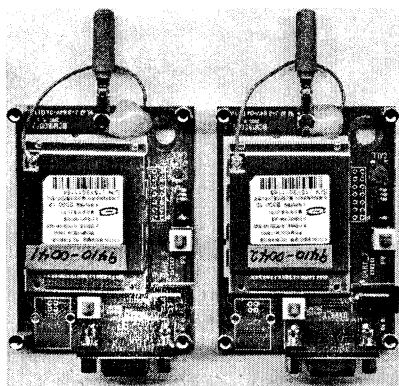
Fig. 4. CDMA wireless modules.
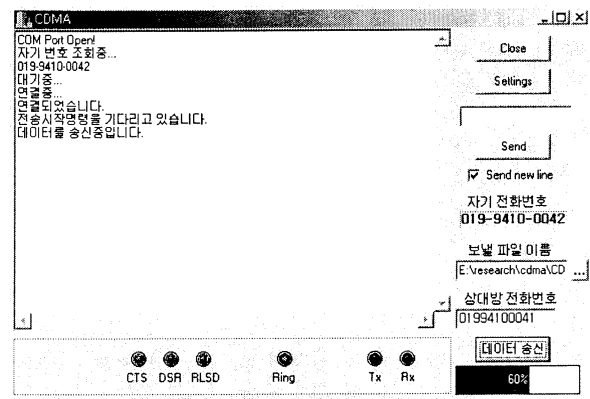
Fig. 5. Control window of the CDMA communication module.
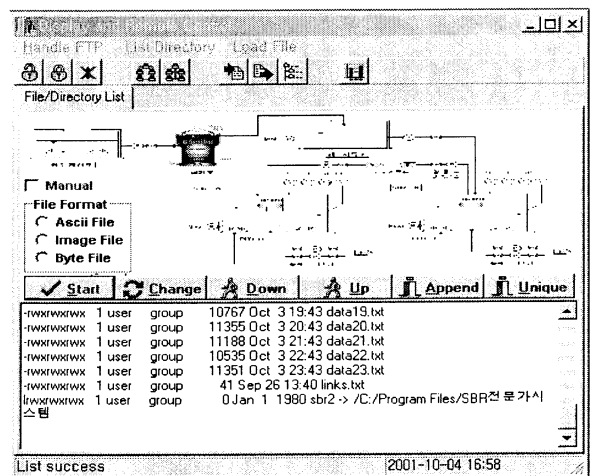
Fig. 6. Main window for the FTP control.

example, when the plant is in the abnormal condition but there is no operator, the SCADA system has to check the plant problem and sends an emergence message to the operators. The modules are shown in Fig. 4 and Fig. 5 is a setting window of the CDMA data transfer.

But the wireless communication technique needs the high operate cost. So it could just be operated for the special cases. In normal conditions, data are usually transferred using the FTP server. This is a stable method for the file transfer. In this application, the sensor signals are stored in PC2 with a text format. Firstly, data are stored into database and then these DB data are reformatted to the text format for the data transfer.

C. Data Display and Analysis in PC3

PC3 is the total plant control system. In this system, transferred data are displayed in the monitoring system. In this paper, ORP and DO concentration are displayed and analyized. The states of equipments are also represented using illustrations and text boxes. During displaying the data, the ORP and DO concentrations are analyized for the on/off control of the blowers and influent pump.

When the derivative value of the ORP curve is applied for making a decision of the blower-off time, the blower will be turned off at the value about under 0.5~0.75
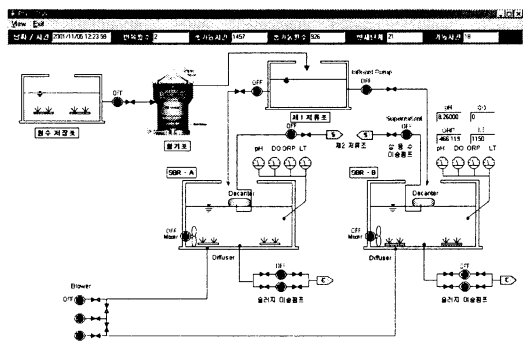
Fig. 7. Display window for the state and data of the SBR plant.
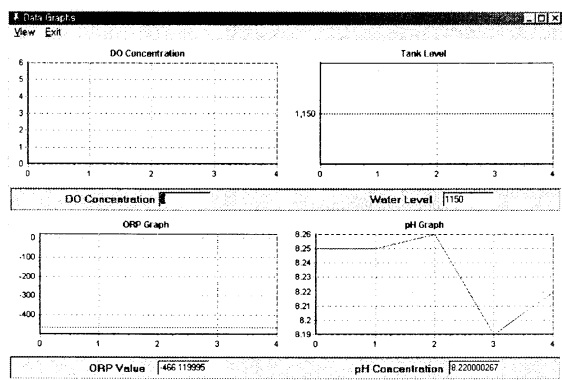
Fig. 8. Graphs window of the plant sensor Data.

dORP/dt. Untill this time, the blower is turned on. This stage is called as the aerobic process for the nitrification. In this stage, the amonia components could be changed to nitrate and nitrite (NO_X) components. At the point of motor turn off, an influent pump is turned on. This stage is called as the anoxic process for the denitrification. The process time is one hour which is decied by the skilled operator. So it could be changed owing to the operating condition. After this process, the NO_X components will be changed to nitrogen gases and then removed to the air. These two processes are the main processes of the SBR wastewater treatment plant.

Fig. 7 is the display window, which shows the states of pumps and sensor data. Fig. 8 is the graph window, which plots the graphs of the ORP, DO, influent level, and pH concentration.

D. Fuzzy Inference System

In this paper, the fuzzy inference system is applied for determining threshold values and influent volumes in PC3. The blower-off time is determined by the derivative value of the ORP or DO. When the ORP curve is applied for this work, the blower is turned off at the around 0.5~0.75 dORP/dt (0.1~0.2 dDO/dt). And the influent wastewater is determined at 0.2~0.4m³. These values are called as the threshold value and influent volume in this paper. And these are dependent on the load of the influent wastewater. This influent wastewater is filled at the point that the blower is turned off. From this point, the anoxic process is started. The amount of the influent wastewater, which should be filled, is important thing for
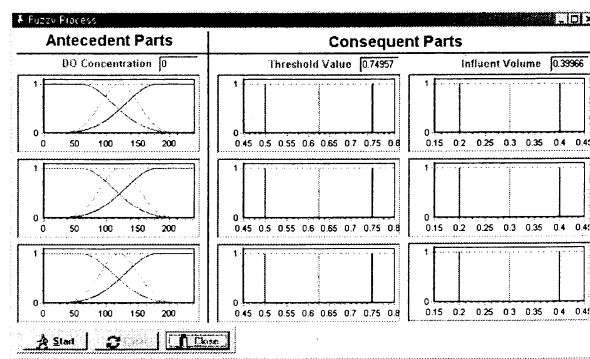
Fig. 9. Fuzzy inference system for the threshold and influent.

operating the SBR plant. Therefore, in this paper, the threshold value and influent volume are determined by the fuzzy inference system that is constructed based on the skilled operator's knowledge. Inputs of the fuzzy inference system are the time that the DO curve is fixed under $2mg/l$. If the influent is a high load, the time should be longer, and vice versa. The consequent values are the threshold and influent volume. The high load has 0.75 of the threshold value and 0.4 of the influent volume, while the low load has 0.5 and 0.2, respectively.

V. CONCLUSION

In this experiment, the on-line monitoring and diagnosis system could be applied. And the sensor data are displayed and analyzed in the remote site. The plants, which are located in the several places, could be remotely monitored and controlled at the same time. This technology is very useful for industrial fields. And the diagnosis information could be inferred by the fuzzy logic based upon the expert knowledge. In next generation, the diagnosis of equipments should play a large role in reducing costs and increasing the productivity. Using like this system, the plant could be operated and maintained economically.

REFERENCE

[1]G. Bortone, S. Gemelli, A.Rambaldi, and A. Tilch, "Nitrification, Denitrification and Biological Phosphate Removal in Sequencing Batch Reactors Treating Piggery Wastewater," *Wat. Sci. Tech.*, Vol. 26, No. 5-6, pp. 977-985, 1992.

[2]B. Chambers, "Batch Operated Activated Sludge Plant for Production of High Effluent Quality at Small Works," *Wat. Sci. Tech.*, Vol. 28, No. 10, pp. 251-258, 1993.

[3]N. H. Johansen, J.S Andersen, and J. la Cour Jansen, "Optimum Operation of a Small Sequencing Batch Reactor for BOD and Nitrogen Removal Based on On-Line OUR-Calculation," *Wat. Sci. Tech.*, Vol. 35, No. 6, pp. 29-36, 1997.

[4]John Yen and Reza Langari, *Fuzzy Logic: Intelligence, Control, and Information.* Prentice Hall, New Jersey, pp. 10-17, 1999.

Frequency variability of neural rhythm in a small network of pacemaker neurons

K. Sugimoto, Y. Nii, S. Doi, and S. Kumagai
Department of Electrical Engineering
Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

Abstract

Using the famous Hodgkin-Huxley equations, it is shown that a single neuron and a population of electrically coupled neurons can modulate their rhythm over an extremely wide range of frequency. The generation mechanism of such neural rhythm is analyzed from a viewpoint of bifurcation theory and the importance of existence of multiple time-scales in neuronal dynamics is stressed.
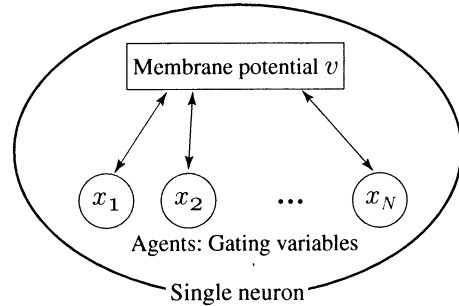
Figure 1: HH-type equations.

1 Introduction

There are variety of rhythms with various time-scales ranging from milliseconds to a year in biological phenomena. Such rhythmic phenomena play a central role in a life or we can say that the rhythmicity is a life itself. This paper examines how such variability of rhythm is produced and controlled using a neuronal caricature of Hodgkin-Huxley (HH) equations.

The HH-type equations are the most successful model in biology and can quantitatively predict electrical behavior of various neurons. The HH-type equations are generally denoted as follows:

$$\frac{dv}{dt} = G(v, \{x_i\}) + I_{\text{ext}} \tag{1a}$$

$$\frac{dx_i}{dt} = \frac{1}{\tau_{x_i}(v)}(x_i^\infty(v) - x_i), \; i = 1, 2, ..., N \tag{1b}$$

where the variable v denotes a membrane potential and $G(\cdot)$ is a total ionic current which depends both on v and on gating variables x_i. I_{ext} denotes an externally applied current to a cell. The number N of gating variables and the functional forms of $x_i^\infty(v)$ and $\tau_{x_i}(v)$ differ cell by cell and are determined by physiological experiments.

Figure 1 is a schematic diagram of a single neuronal dynamics of the HH-type equations. The gating variables $\{x_i\}$ interact *globally* with each other through the membrane potential. Usually, the time constants

of the dynamics of gating variables differ each other. Thus, we investigate how a neuronal rhythm of a membrane potential is affected by the interaction of gating variables with different time-scales and examine the relation between the time-scale of the macroscopic rhythm of membrane potential and the microscopic rhythms of gating variables.

2 Hodgkin-Huxley neuronal model

As a prototypical model among diverse HH-type equations family, we consider the following *slightly* modified Hodgkin-Huxley (HH) equations[1, 2]:

$$C\frac{dv}{dt} = G(v, m, n, h) + I_{\text{ext}} \tag{2a}$$

$$\frac{dm}{dt} = \frac{1}{\bar{\tau}_m \tau_m(v)}(m^\infty(v) - m) \tag{2b}$$

$$\frac{dn}{dt} = \frac{1}{\bar{\tau}_n \tau_n(v)}(n^\infty(v) - n) \tag{2c}$$

$$\frac{dh}{dt} = \frac{1}{\bar{\tau}_h \tau_h(v)}(h^\infty(v) - h) \tag{2d}$$

Only modification from the original HH equations which is a model of a squid giant axon, is the introduction of the 'time constants' $\bar{\tau}_m$, $\bar{\tau}_n$, and $\bar{\tau}_h$; in the case of $\bar{\tau}_m = \bar{\tau}_n = \bar{\tau}_h = 1$, these equations are equal to the
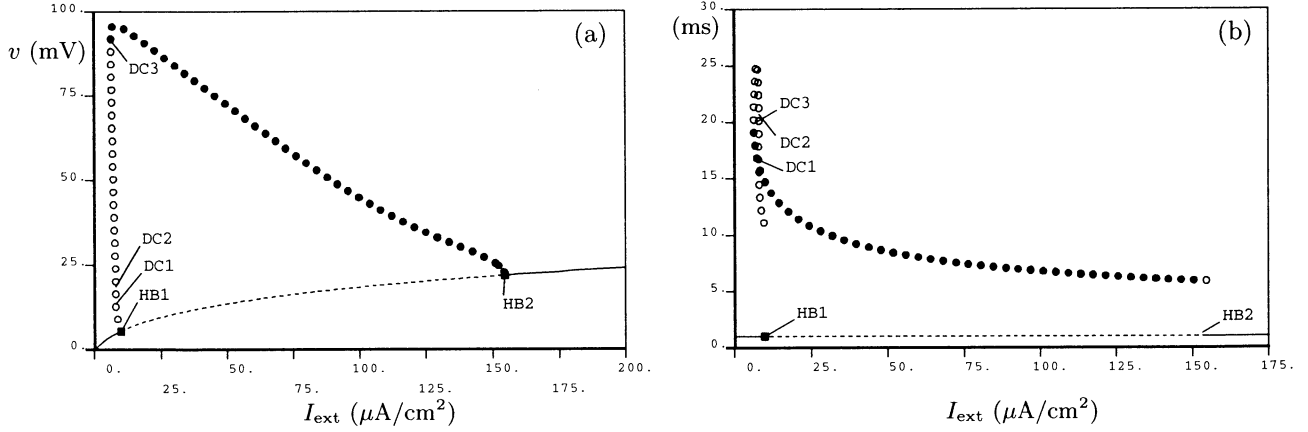
Figure 2: (a) One-parameter bifurcation diagram of the HH equations. The membrane potential v vs. I_{ext}. (b) The period of periodic solution vs. I_{ext}.

original HH equations. The advantage of the usage of this modified HH equations is that the time constants do not change the structure (the number or the position) of the equilibrium points of the equations; They only affects the stability of the *unique* equilibrium point of the HH equations and thus they are useful in order to study the effect of the time scales of gating variables on neuronal dynamics.

When $I_{\text{ext}} = 0$, the HH equations (2) have a unique stable equilibrium point (resting state) and produce an (transient) action potential which is a rapid increase and decrease of a membrane voltage in response to external stimuli and return to the resting state. If the value of external current I_{ext} is sufficiently large, the HH equations fire (produce an action potential) repeatedly.

Figure 2 is the bifurcation diagram of the HH equations (2) when $\bar{\tau}_m = \bar{\tau}_n = \bar{\tau}_h = 1$ (the case of original HH equations). Part (a) shows the dependence of the solution of the HH equations on the parameter I_{ext}; the v values of the stationary solution of the HH equations are plotted for various values of I_{ext} where the maximum value of v is plotted for a periodic (oscillatory) solution. Solid and dotted curves denote stable and unstable equilibria, respectively. The filled (open) circles denote stable (unstable, resp.) periodic solutions. At the point HB2 of part (a), a stable periodic solution bifurcates from an equilibrium point by the (supercritical or stable) Hopf bifurcation. An unstable periodic solution is bifurcated by the (subcritical or unstable) Hopf bifurcation at the point HB1. In the range of $6.3 < I_{\text{ext}} < 9.8$, the multi-stability of an equilibrium and a periodic solution occurs.

Part (b) shows the period of the periodic solutions shown in (a). The period of stable periodic solutions

(closed circle) varies in the range from several milliseconds to 20 milliseconds. The period does not change much totally although the variation is comparatively large in the small I_{ext} range.

3 Extraordinarily slow oscillations

In this section we examine the period or the inter-spike interval of periodic or non-periodic firings. We can easily expect that the period will be prolonged if the time constants $\bar{\tau}_n$ and/or $\bar{\tau}_h$ are increased. In the following, however, we will show the 'unexpected' slowing down of the period or inter-spike intervals.

Figure 3(a) shows the inter-spike intervals (ISI's) of the modified HH equations (2) as a function of the value of I_{ext} when n is slightly slow ($\bar{\tau}_n = 10$). As the value of I_{ext} decreases, the intervals (periods of periodic firings in this case) increases from several hundreds to more than two thousands millisecond. Since the period of the original HH equations ($\bar{\tau}_m = \bar{\tau}_n = \bar{\tau}_h = 1$) is at most a few tens millisecond (Fig.2(b)) and the time constant $\bar{\tau}_n$ is ten, this increase of periods is an unanticipated one. Same phenomena can be observed when $\bar{\tau}_h$ is 'slightly' slow.

Figure 3(b) is the similar one to part (a) when h is slow ($\bar{\tau}_h = 2$). This figure is more complicated since there are various chaotic firings with different ISI's. The total ISI's, however, increase drastically as I_{ext} decreases. It should be stressed again that the time constant is only twice of the original one.

We note that the time constants $(\bar{\tau}_m, \bar{\tau}_n, \bar{\tau}_h)$ do not change the structure of equilibrium of the HH equations but change only the stability of equilibrium. Even in the case of large time constants, the unique
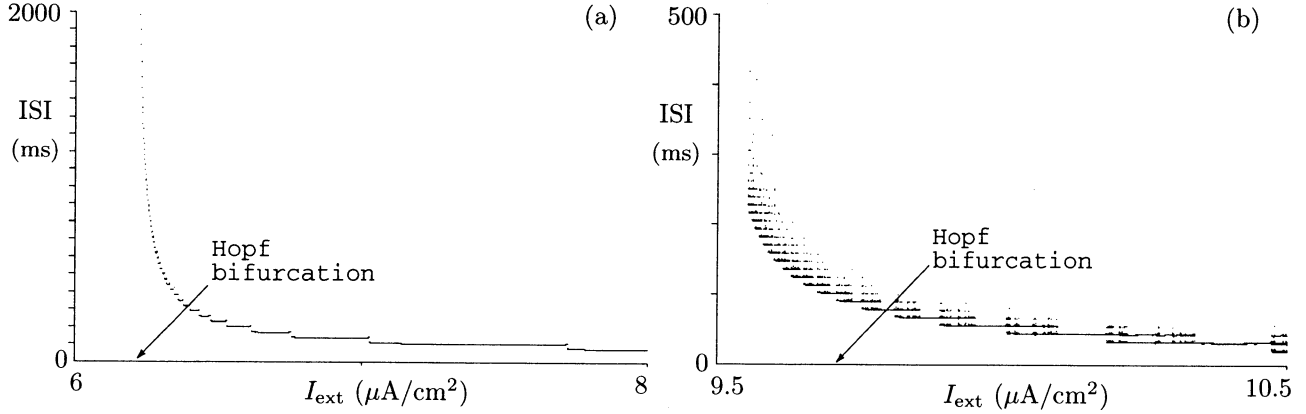
Figure 3: Inter-spike intervals vs. I_{ext}. (a) n-slow case ($\bar{\tau}_n=10$). (b) h-slow case ($\bar{\tau}_h=2$).

equilibrium point of the HH equation is destabilized through the Hopf bifurcation. The period of periodic orbits bifurcated from the Hopf bifurcation cannot be so large. Thus the existence of very slow oscillations suggests the existence of global bifurcations of periodic orbit such as homoclinic bifurcations.

4 Oscillation slowdown and death induced by electrical coupling

The generation of very slow oscillation presented above is not a special case. In fact, we have observed such oscillations in various situations. One example is the slow oscillations induced by electrical coupling of neurons.

In this section, we consider the effects of *slightly* slow gating variable on the electrically coupled neurons. For this purpose, we connect two neurons electrically, each of which is denoted by the HH equations (2), and we set the time constant of the gating variable $n^{(1)}$ of the first neuron as $\bar{\tau}_n^{(1)} = 4$ (other time constants are unchanged). We also set as $I_{ext}^{(1)} = I_{ext}^{(2)} = 10$. In this case both neurons fire repeatedly and the period of the first (slow) neuron is about twice longer than that of the second neuron.

Figure 4 is the ISI bifurcation diagram where the ISI's of both neurons are simultaneously plotted in one figure. When uncoupled (see the plot near '$D = 0$') or coupling is very weak, two pacemaker neurons fire almost freely with twice different periods. As the coupling coefficient D increases, several patterns of phase lockings such as 1:2, 2:2, and 1:1 phase lockings appear in this order in the region $D < 0.4$.

These patterns of phase lockings are not so different from phase lockings of usual oscillators. What is par-
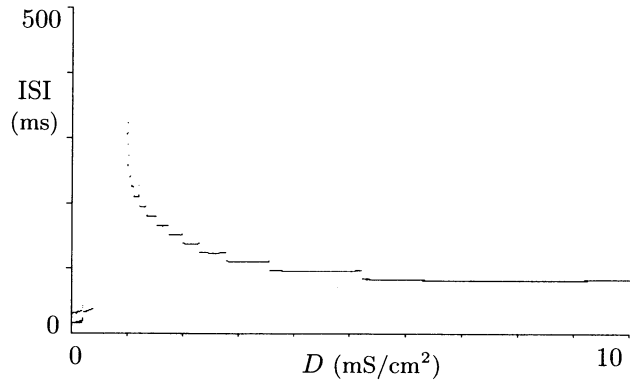
Figure 4: Interspike interval (ISI) bifurcation diagram of the two coupled HH pacemaker neurons (2) for the coupling coefficient D. $\bar{\tau}_n^{(1)} = 4$. $I_{ext}^{(1)} = I_{ext}^{(2)} = 10$.

ticular phenomena induced by a slow gating variable occur in the case of stronger coupling. In the range of $0.4 < D < 1$ of Fig.4, firings of both neurons are completely inhibited. After this inhibition, very slow oscillations appear suddenly (near $D = 1$). In this region, two pacemaker neurons are in 1:1 phase lockings and phase-locked patterns do not change by the increase of D value while the period of phase-locked oscillation decreases drastically and *stepwise*. These slow phase-locked oscillations are considered to be generated by the similar mechanism to the previous slow oscillations of a single neuron.

5 Rhythm in a population of globally coupled neurons

In the previous section, we have studied the behavior of two neurons directly coupled through an elec-
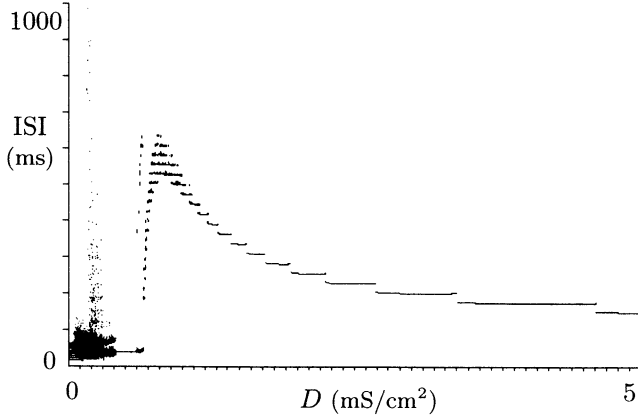
Figure 5: Interspike interval (ISI) bifurcation diagram of a population of ten neurons coupled through a common buffer (3). $\bar{\tau}_n^{(j)} = j$, $I_{\text{ext}}^{(j)} \equiv 10$ ($j = 1, ..., 10$).

trical synapse. In many biological organs such as a sinoatrial node of a heart, many cells can be considered to be coupled globally though a common buffer[3]. Thus, we consider the following population of M cells of the HH neuron (1):

$$\frac{dv^{(j)}}{dt} = G(v^{(j)}, \{x_i^{(j)}\}) + I_{\text{ext}}^{(j)} + D(z - v^{(j)}) \quad (3a)$$

$$\frac{dx_i^{(j)}}{dt} = \frac{1}{\tau_{x_i^{(j)}}(v)}(x_i^{(j)\infty}(v) - x_i^{(j)}), \quad (3b)$$

$$\frac{dz}{dt} = \frac{D'}{M}\sum_{j=1}^{M}(v^{(j)} - z), \quad i = 1...N, j = 1...M. \quad (3c)$$

where the variable $v^{(j)}$ denotes the membrane potential of jth neuron and other variables are similarly defined.

Figure 5 shows the similar diagram to Fig.4. ISI's of each neuron are plotted as a function of D ($\equiv D'$). The number M of cells is ten and the time constants are $\bar{\tau}_n^{(j)} = j$ ($j = 1, ..., 10$). In this case, inherent periods of each *uncoupled* oscillator are between ten milliseconds to several milliseconds (see lower-left corner of Fig.5). When a coupling is weak ($D < 0.4$), all neurons are not phase-locked. Figure 6 shows such an example, where waveforms of membrane potentials $v^{(j)}(t)$ (ascending order of j from bottom) of all neurons and $z(t)$ (uppermost trace) are drawn. A striking fact is that the fastest neuron (lowest trace) shows very long ISI (see also Fig.5). When a coupling is stronger ($D > 0.4$), all neurons settle in a 1:1 phase-locked state. But, as the coupling increases, extraordinarily slow oscillations appear again ($D > 0.6$).
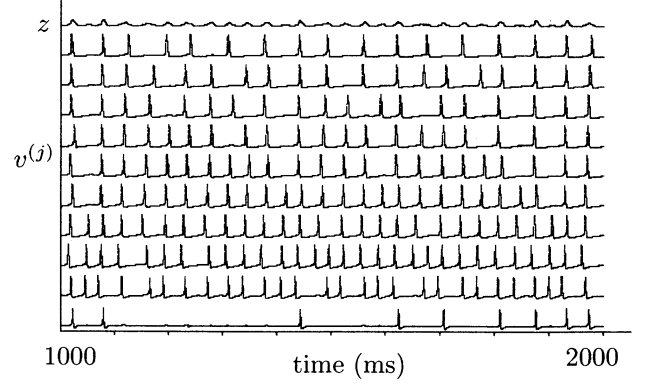
Figure 6: Waveforms of membrane potentials $v^{(j)}$, ($j = 1, ...10$) and z. $D = 0.25$.

6 Discussion

Using the Hodgkin-Huxley model of a neuronal pacemaker, it is demonstrated that a single neuron or a population of neurons can modulate their rhythm over a very wide range of frequency. An essential factor of this phenomenon is the heterogeneity of time-scales of microscopic process underling neuronal dynamics. More rigorous analysis can be made from a viewpoint of geometric singular perturbation theory[2].

References

[1] S.Doi, S.Nabetani, and S.Kumagai, "Complex nonlinear dynamics of the Hodgkin-Huxley equations induced by time scale changes," Biol. Cybern. 85, 51-64, 2001.

[2] S.Doi and S.Kumagai, "Nonlinear dynamics of small-scale biophysical neural networks," In: R.Poznanski (ed.), *Biophysical Neural Networks: Foundations of integrative neuroscience*, Chapter 10, pp.261-301, Mary Ann Liebert, Inc., 2001.

[3] S. Doi and S. Sato, "Regulation of differentiation in a population of cells interacting through a common pool," J. Math. Biol. 26, pp.435-454 (1988).

[4] S. Doi, J. Inoue, S. Sato, and C. E. Smith, "Bifurcation analysis of neuronal excitability and oscillations," In: Poznanski R (ed.) *Modeling in the neurosciences: from ionic channels to neural networks*, Harwood Academic Publishers, Chapter 16, pp 443-473, 1999.

Stability and Bifurcation Analysis of Robust Circadian Oscillators

Tetsuya Kobayashi[1+], Luonan Chen[2*] and Kazuyuki Aihara[1,3#]

[1] *Department of Mathematical Engineering and Information Physics,*
The University of Tokyo, Tokyo 113-8656, Japan.

[2] *Electrical Engineering and Electronics, Osaka Sangyo University, Osaka 574-8530, Japan.*

[3] *CREST, Japan Science and Technology Corporation (JST), Kawaguchi, Saitama 332-0012, Japan.*

[+] *tetsuya@sat.t.u-tokyo.ac.jp,* [*] *chen@elec.osaka-sandai.ca.jp,* [#] *aihara@sat.t.u-tokyo.ac.jp*

Abstract

It is shown, using stability and bifurcation analysis, that adding a positive feedback loop and a time delay to a gene network model with a negative feedback loop lead to strengthen the capability of the model to oscillate. Thus, this model can be a candidate of a robust artificial gene oscillator.

Key words: artificial gene network, positive feedback, time delay.

1 Introduction

Recently, a lot of mathematical models of circadian oscillators have been proposed based on experimental data. They include a simple feedback loop model involving only the period(per)[1], ones involving per and the timeless(tim)[2, 3], an interlocked model involving per, tim, and the *Drosphila* clock($dClk$)[4, 5] and one involving *Neurospora* frequency(frq)[4]. Most of these models were analyzed focusing on their ability to reproduce original biological behaviors, and the robustness of the oscillators was judged based on simple numerical simulations. However, the mechanisms of robust circadian oscillators are not yet fully understood. In particular, the suggestion that positive feedback strengthens robustness of the oscillation[6] is yet to be validated. On the other hand, a simple negative feedback loop oscillator constructed artificially in *E.coli* was found not to be resistant to noise, and it is required to explore how reliable oscillator can be constructed [7]. In this paper, motivated by the recent research advances in circadian oscillators and artificial gene oscillators, we use stability and bifurcation analysis to show that a time delay and a positive feedback loop strengthen the ability of a gene network with a negative feedback loop to oscillate.

2 Time delay and simple negative feedback model

A simple negative feedback loop model proposed a decade ago as a basic model for circadian oscillation[1] is now considered a necessary element of circadian oscillation[8]. Actually, negative feedback loops have been found in some circadian oscillators[9], and it has been proven that a gene network model without negative feedback loop does not show stable oscillatory behavior[10]. On the other hand, a negative feedback loop is not a sufficient condition for a self-sustained oscillation. In fact, it is easy to see by using Bendixon's criterion that a one-gene negative feedback loop model never shows oscillatory behavior[11]:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = f(p) - d_m m \tag{1}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = s(m) - d_p p, \tag{2}$$

Here, f and s are non negative monotonously decreasing and increasing functions, respectively, describing synthesis of the mRNA and of the protein coded by the gene, and d_m and d_p are the degradation rates of the mRNA and the protein, respectively.

However, the one-gene negative feedback loop model can show oscillatory behavior if it incorporates explicit time delays due to transcription and translation processes and the total time delay is large enough[12] or if it incorporates other genes in its negative feedback loop[11]. The incorporation of another gene into the one-gene negative feedback loop model can be in a sense considered to be introducing a time delay into the feedback loop because the product of the second gene acts as a buffer, delaying feedback of the change in the product of the first gene. Thus, a time delay should be long enough for a negative feedback loop model to show a sustained oscillation. It is suggested that the longer the time delay, the more

capable the negative feedback is of producing oscillations. This kind of time delays can also be incorporated into the model not only by adding a cascade of gene expressions but also by adding cascade of multiphosphorlations of gene products. In fact, a cascade of phosphorlations of *PER* in *Drosophila* is a possible source of time delay[1].

3 Coupling model of negative and positive feedback loops

As shown above, a negative feedback loop with a time delay is sufficient for sustained oscillation. However, recent experiments on an artificial gene oscillator with a negative feedback loop consisting of three genes showed that the oscillator is subject to stochastic fluctuations of its components and functions less reliable than real circadian oscillators[7]. Instead of the cascade of genes, circadian oscillators use both positive and negative feedback loops[9].

To investigate the effects of incorporating a positive feedback loop, we consider a simplified two-gene network model, which is a candidate of an artificial gene oscillator:

$$\frac{dp_1}{dt} = f'_1(p_1, p_2) - p_1 \qquad (3)$$

$$\frac{dp_2}{dt} = \kappa \left[f'_2(p_1, p_2) - p_2 \right], \qquad (4)$$

where p_1 and p_2 are the concentrations of the products of the two genes. f_1 and f_2 are nonnegative monotone functions describing the synthesis of the products, and κ is the ratio of the degradation rates of the products. κ also represents the difference of the time scales of expression between the two genes. The transcription processes are abbreviated here because mRNA turnover is rapid relative to the turnover of proteins[13]. Assume that gene 1 activates both its own expression and that of gene 2, and gene 2 represses the expression of gene 1. That is, $\frac{df'_1}{p_1}, \frac{df'_2}{p_1} \geq 0$ and $\frac{df'_1}{p_2} \leq 0$. It can then be proven that the model has at least one stable periodic solution if the following three conditions are satisfied (Fig. 1): (1) the nullclines of Eq. (3) and 4 intersect only once; (2) the intersection point is in the region where $df'_1/dp_1 - 1 > 0$, and (3) κ is small enough, although Eq. (3) and Eq. (4) never show oscillation if there is no positive feedback loop[11]. These results indicate that the incorporation of a positive feedback loop can enhance the ability of a negative feedback loop to show oscillatory behaviors and that the positive feedback loop should feedback

fast enough relative to negative feedback for emergence of the oscillation.

Graphical analysis shows that the bistable shape of the nullcline of Eq. (3) and the difference of time scales are essential for enhancing the ability. It should be noted that the phosphorlation of *per* acts a positive feedback loop, and the same bistable structure of Eqs. (3) and (4) is found in the negative feedback loop of *per* and *tim* in *Drosophila*[3]. Thus, the model can describe the essential features of the oscillator with coupling of positive and negative feedback loops.
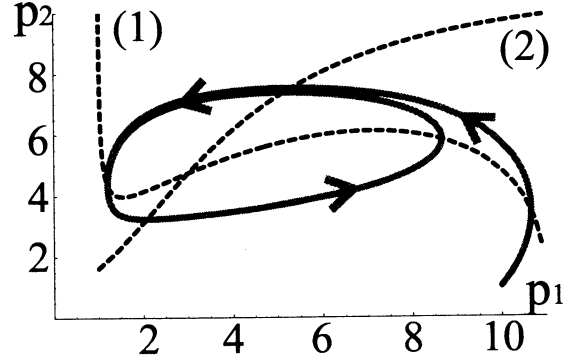
Figure 1: Dotted lines (1) and (2) are the nullclines of Eqs. (3) and (4), respectively. The solid line shows the orbit of Eqs. (3) and (4). A bimodal shape of nullcline (1) is obtained if there is an interval of $\mathbb{R}^+$ on which $df'_1/dp_1 - 1 > 0$ is satisfied. The orbit and nullclines are calculated using the following parameters and functions: $f'_1(p_1, p_2) = s_1 f_1(p_1, p_2)/d_{m1}$, $f'_2(p_1) = s_2 f_2(p_1)/d_{m2}$, $f_1(p_1, p_2) = r_1 + V_1 p_1^n / [(1 + (p_2/K_{12})^m) K_1 + p_1^n]$, $f_2 = r_2 + V_2 p_1^k / [K_{22} + p_1^k]$, $K_1 = 1, K_{12} = 1, K_{22} = 15, r_1 = r_2 = 1, V_1 = V_2 = 10, n = m = 3, k = 2, s_1 = 0.3, s_2 = \kappa s_1, d_{p1} = 0.06, d_{p2} = \kappa d_{p1}, d_{m1} = d_{m2} = 5$, and $\kappa = 0.2$.

Now then, we investigate the robustness of the coupled model to changes in the parameters by bifurcation analysis. First, we consider the case in which the third condition breaks down. It can be easily shown that an oscillation appears (disappears) via Hopf bifurcation as κ increases (decreases), and that the higher becomes the nonlinearity of f_1 in Eq. (3) at the intersection point, at the lower value of κ, an oscillation appears or disappears[11].

Next, we calculate two two-parameter bifurcation diagrams of the model (Figs. 2 and 3) using AUTO. One parameter is κ, and the other parameter is V_2 or K_1. Changes in the prameter values lead to a breakdown in the first and second conditions, respectively.

The diagrams show that the oscillation appears (disappears) via Hopf bifurcation when K_1 is changed (Fig. 3), while it can appear (disappear) via homoclinic bifurcation when V_2 is changed (Fig. 2). Because of the feature of homoclinic bifurcation, the oscillation is disturbed only a little just immediately before its disappearance via homoclinic bifurcation. In addition, the oscillation can be driven by random fluctuations of the variable because the ruin of the oscillation still remains after its disappearance. On the other hand, when κ is small enough, the Hopf bifurcation caused by changes in K_1 looks like a singular bifurcation. This type of Hopf bifurcation is called "canard-type bifurcation"[14] and generated because of the bimodal shape of the nullcline of Eq. (3). Thus, for the second case, the oscillation remains undisturbed immediately before its disappearance if κ is small and can also be driven by random fluctuations of the variable. Therefore, if κ is small enough, the oscillation is robust against changes in the two parameters in the sense that it remains steady immediately before its disappearance and can be driven by noise even after its disappearance, in contrast to the case of normal Hopf bifurcation. Furthermore, the condition that κ is small is robust to changes in parameters because κ is not an absolute value but simply the ratio of the time scales. In addition, this result generally holds even if we choose other bifurcation parameters as long as their changes lead to a break down in the two conditions.

Finally, we examine the effects of introducing an implicit time delay into the coupled model. As shown in the previous subsection, explicit consideration of transcription processes introduces a time delay into the model. Consider a coupling model of negative and positive feedback loops in which the transcription processes are not abbreviated:

$$
\begin{aligned}
\frac{m_1}{dt} &= (f_1 - d_{m1}m_1)/d_{p1} \\
\frac{m_2}{dt} &= (f_2 - d_{m2}m_2)/d_{p1} \\
\frac{p_1}{dt} &= (s_1 m_1 - d_{p1}p_1)/d_{p1} \quad\quad (5) \\
\frac{p_2}{dt} &= (s_2 m_2 - d_{p2}p_2)/d_{p2}.
\end{aligned}
$$

where d_{m1}, d_{p1}, d_{m2} and d_{p2} describe the degradation rates of the mRNAs and of the proteins of genes 1 and 2, respectively, and s_1 and s_2 are the synthesis rates of the mRNAs of genes 1 and 2. f_1 and f_2 are given by $f_1 = d_{m1}d_{p1}f'_1/s_1$ and $f_2 = d_{m2}d_{p2}f'_2/s_2$, and $d_{p2} = \kappa p_{p1}$.

The two-parameter bifurcation diagrams for $\kappa{:}K_1$ and $\kappa{:}V_2$ are shown in Figs. 2 and 3. They show that explicit incorporation of transcription processes

expands the parameter region for an oscillation and thus strengthens the capability of the model to oscillate. They also show that the expansion of the parameter region for an oscillation occurs on the boundary where normal Hopf bifurcation occurs. This suggests that an implicit time delay can strengthen the oscillations appearing via Hopf bifurcation. Thus, this effect of the time delay on the robustness of oscillation is complementary to the effects of a positive and negative feedback loop model with the time delay. The positive feedback loop model shows more robust oscillation than ones without one or both loops.
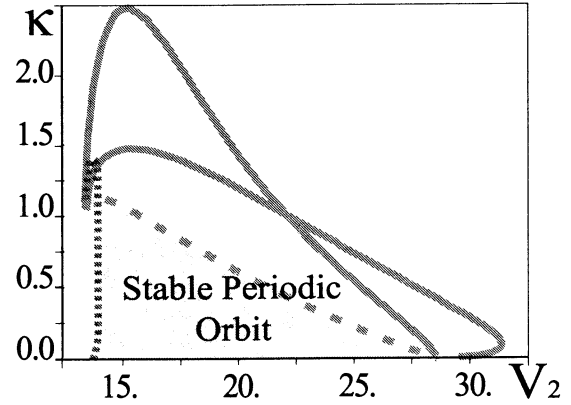
Figure 2: A two-parameter bifurcation diagram of Eqs. (3), (4), and (5). The dotted line shows the set of Hopf bifurcation points of Eqs. (3) and (4), and the solid lines show those of Eq. (5). The dashed lines are the sets of homoclinic bifurcation points of Eqs. (3), (4), and (5). This diagram is calculated using the same functions and parameters as for Fig. 1, except $K_{22} = 10$, $r_2 = 0.1$, and $n = 4$.

4 Conclusion

Using stability and bifurcation analysis, we have shown that adding a time delay and a positive feedback loop to a simple negative feedback loop model enhances its ability to oscillate and makes the model more robust. Although the model we described is not an exact reconstruction of experimental data, it can still be used to extract the essence of the robustness of oscillators, and its structure makes it well suited for implementation as an artificial gene network. Thus, it can be a candidate of a robust artificial gene oscillator.

On the other hand, the direct relationship between this simple model and actual circadian oscillators is
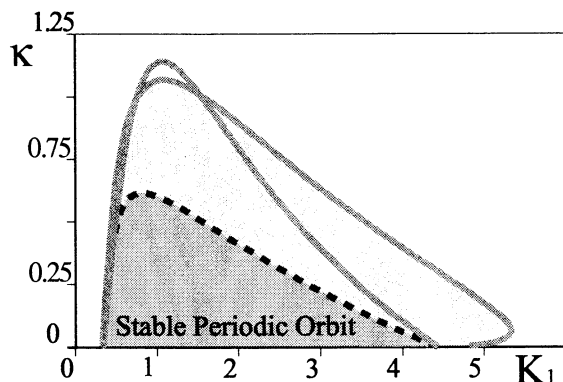
Figure 3: A two-parameter bifurcation diagram of Eqs. (3), (4), and (5). The lines show the same as those in Fig. 2 and the functions and parameters used for the calculation are the same as those used for Fig. 1.

not clear, and more detailed examination is still required. Moreover, this model does not posses such structural features of circadian oscillators as the coupling of *per* and *tim* and the interlocking of negative feedback loops[5]. The role of these structures in making oscillator robust to the changes in parameters and their relationship to other features of circadian oscillators, such as temperature compensation and robustness to noise, are left to further study.

Acknowledgements

References

[1] A. Goldbeter, "A Model for Circadian Oscillations in the *Drosophila* Period Protein (PER)," *Proc. R. Soc. Lond. B*, Vol. 261, pp. 319-324, 1995.

[2] A. Goldbeter and *et al*, "From Simple to Complex Oscillatory Behavior in Metabolic and Genetic Control Networks," *Chaos*, Vol. 11, pp. 247-260, 2001.

[3] J. J. Tyson and *et al*, " A Simple Model of Circadian Rhythms Based on Dimerization and Pro-

teolysis of PER and TIM," *Biophysical J.*,Vol. 77, pp. 2411-2417, 1999.

[4] P. Smolen, D. A. Baxter and J. H. Byrne, "Modeling Circadian Oscillations with Interlocking Positive and Negative Feedback Loops," *J.Neurosci*,Vol. 21, pp. 6644-6656, 2001.

[5] H. R. Ueda and *et al*, "Robust Oscillations within the Interlocked Feedback Model of *Drosophila* Circadian Rhythm," *J. theor. Biol*,Vol. 210, pp. 401-406, 2001.

[6] P. Cheng, Y. Yang and Y. Liu, "Interlocked Feedback Loops Contribute to the Robustness of the *Neurospora* Circadian Clock," *PANS*,Vol. 98, pp. 7408-7413, 2001.

[7] M. B. Elowitz and S. Leibler, "A Synthetic Oscillatory Network of Transcriptional Regulators," *Nature*,Vol. 403, pp. 335-338, 2000.

[8] J. C. Dunlap, "Molecular Bases for Circadian Clocks," *Cell*,Vol. 96, pp. 271-290, 1999.

[9] M. Merrow and T. Roenneberg, "The Circadian Cycle: is the Whole Greater Than the Sum of its Parts," *TRENDS in Genetics*,Vol. 17, pp. 4-7, 2001.

[10] T. Kobayashi and *et al*, "Analysis and Design of Gene Switches Composed of Positive Feedback Loops," *Proceedings of ICSB*(in printing), 2001.

[11] T. Kobayashi and K. Aihara "Stability and Bifurcation Analysis of Simple Gene Networks," *METR, Univ. Tokyo*, No. 02, 2001.

[12] L. Chen and K. Aihara, "Stability of Genetic Regulatory networks with Time Delay," *IEEE Trans. on Circuits and Systems, Part-I* (in submission).

[13] T. S. Gardner, "Design and Construction of Synthetic Gene Regulatory Networks," *Doctor thesis of Princeton Univ.*, 2000.

[14] P. Glendinning, *Stability instability and chaos*, Cambridge, 1994.

Pulse-type Hardware Chaotic Neuron Model Constituting from CMOS Processing

Jun Matsuoka[†], Yoshifumi Sekine[†], Katsutoshi Saeki[†] and Kazuyuki Aihara[††]

†. Department of Electronic Engineering, College of Science & Technology, Nihon University
7-24-1, Narashinodai, Funabashi-shi, Chiba, 274-8501, Japan
Phone: +81-47-469-5452, Fax: +81-47-467-9683
E-mail: ysekine@ecs.cst.nihon-u.ac.jp
††. Graduate School of Engineering, The University of Tokyo, and CREST, JST, Japan

Abstract

A number of studies have recently been published concerning chaotic neuron models and asynchronous neural networks having chaotic neuron models. In the case of large-scale neural networks having chaotic neuron models, the neural network should be constructed using analog hardware, rather than by computer simulation via software, due to the high speed and high integration of analog circuits. In the present study, we discuss the circuit structure of a pulse-type hardware chaotic neuron model constituting from CMOS processing. We show that the pulse-type hardware chaotic neuron model can be composed of a synaptic section and a cell-body section using CMOS. In addition, we show the bifurcation structure of our composed model, and discuss the bifurcation routes and return maps thereof.

Keywords — Pulse-type Hardware Neuron Model, Chaotic Neuron Model, Bifurcation Phenomena, Return Map, CMOS

1. Introduction

Brain subsystems have a high degree of information processing ability, namely recognition and learning. However, the information processing functions have not yet been clarified. Various neuron models and artificial neural networks (ANNs) have been investigated in order to clarify the information processing functions of biological neural networks subsequently and apply neuron models and ANNs to engineering problems. In traditional ANNs, for simplicity, digital-type and analog-type neuron models have been used. For these models, the user must determine the method by which the neuron communicates information. For example, input signals must be digitized and the pulse density must be made to correspond to the analog quantity.

In contrast, the actual biological neuron transmits nerve impulses to other neurons [1]. We have attempted to produce hardware in the belief that the development of a new hardware neuron model is one of the most important problems in the study of neural networks. If we regard the biological neuron as an element that processes several pulses that are received asynchronously, a single neuron model should be constructed of a pulse-type hardware neuron model. In addition, the biological neuron exhibits chaotic phenomena easily due to its nonlinear dynamics. For this reason, nonlinear characteristics including chaos, have been investigated in several neuron models and neural networks [2]-[5]. Recently, Osana et al. and Watanabe et al. have discussed learning and associative memory based on the chaotic neural network [6], [7].

We previously investigated the possibility of asynchronous neural networks having pulse-type hardware neuron models [8]-[10].

In the present study, we discuss the circuit structure of a pulse-type hardware chaotic neuron model (hereafter referred to as "P-HCNM") which consists of a synaptic section and a cell-body section, constituting from CMOS processing. In addition, we investigate the response potential of our composed model, where the periodic pulse train stimulation is applied to the cell-body section in order to discuss the bifurcation routes and return maps.

2. P-HCNM Constituting from CMOS Processing

In this chapter, we discuss the circuit structure of the P-HCNM constituting from CMOS processing. The P-HCNM can be composed of a synaptic section and a cell-body section having the following three properties of the mathematical model of an asynchronous chaotic neuron:

spatio-temporal summation of inputs, relative refractoriness, and graded responses of output pulse amplitude.

2.1 Synaptic Section Model
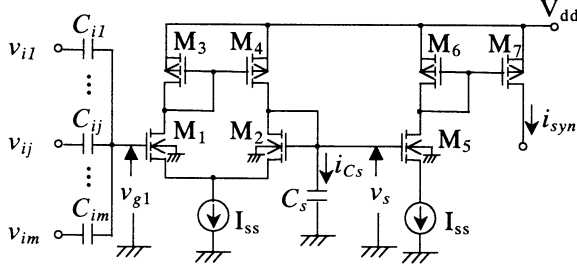
Figure 1: A basic circuit of synaptic section model.

Figure 1 shows a basic circuit of synaptic section model. The model can be constructed using a differential pair and capacitors, and added current miller circuit in order to make gate potential v_s of M_2, output current i_{syn}. Here, v_{ij} is applied to the connection weight on C_{ij}, and added to the other inputs. Therefore, the equation of gate potential v_{g1} of M_1, is as follows:

$$v_{g1} = (C_{i1}v_{i1} + \cdots + C_{ij}v_{ij} + \cdots + C_{im}v_{im}) \Big/ \sum_{j=1}^{m} C_{ij} . \quad (1)$$

Accordingly, the equation of synaptic weight w_{ij} is as follows:

$$w_{ij} = C_{ij} \Big/ \sum_{j=1}^{m} C_{ij} . \quad (2)$$

In considering the small signal equivalent circuit of Fig. 1, the current flowing into C_s, i_{Cs} is as follows:

$$i_{C_s}(s) = \alpha \left(v_{g1} / (s + R_s C_s) \right) \quad (3)$$

where, α and R_s are constants.

Therefore, v_s changes exponentially with time due to Eq. (3), while i_{syn} changes as well as v_s do. The synaptic section has a property of spatio-temporal summation of inputs.

2.2 Cell-body Section Model

Figure 2 shows a basic circuit of cell-body section model. The model can be composed of the negative resistance circuit including equivalent inductance, which consists of M_8-M_{10} and C_G, membrane capacitor C_M, and membrane leak resistor using M_{11}.
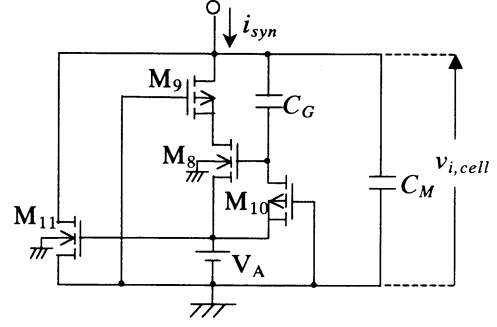
Figure 2: A basic circuit of cell-body section model.

The cell-body section has two properties of relative refractoriness and graded responses of output pulse amplitude [11]. The model of Fig. 2 is an effective model in constructing the cell-body section of the P-HCNM for CMOS processing, due to the simple circuit configuration and circuit construction using enhancement-mode MOSFETs.

3. Characteristics

We examine the response characteristics of the P-HCNM, where the periodic pulse train stimulation for i_{syn} is applied to the cell body section model of Fig. 2, in order to clarify the bifurcation phenomena of the P-HCNM.
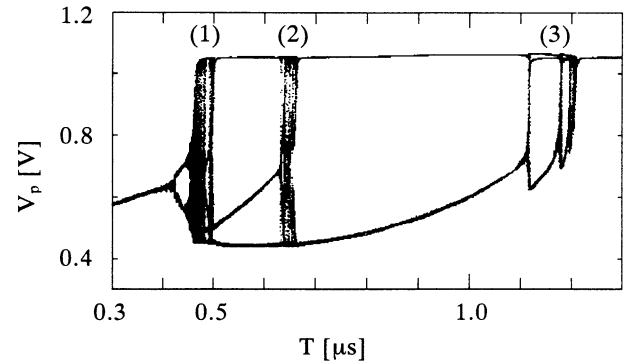
Figure 3: Bifurcation diagram.

Figure 3 shows an example of the bifurcation diagram for a different value of the pulse period T in the periodic pulse train stimulation. The response potential V_p is the maximum value of an output pulse voltage for an-input pulse. The response characteristics of the P-HCNM having pulse period T as a bifurcation parameter were analyzed by PSpice, using circuit parameters, the ratio $(W/L)_8$-$(W/L)_{11}$ of each enhancement-mode MOSFETs, M_8-M_{11}, are 10, 10, 0.1, and 0.3, respectively, where W is the gate width and L is the gate length, C_G=1.0[pF], C_M=1.0[pF], V_A=1.82[V],

pulse width is 0.1[μs], and pulse amplitude is 10[μA]. This figure shows that chaotic phenomena are observed between each periodic response.

Next, we show the magnifications of the bifurcation phenomena in Fig. 3 in order to clarify the bifurcation route.
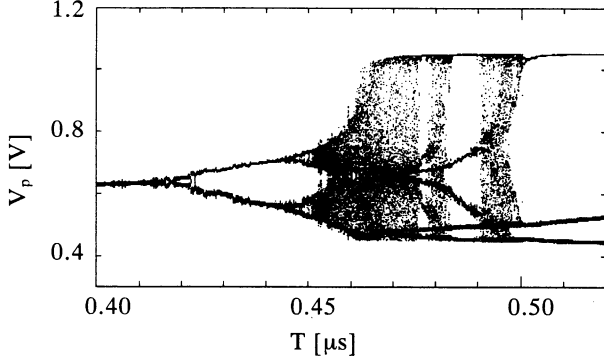
Figure 4: Magnification of area (1) in Fig. 3.
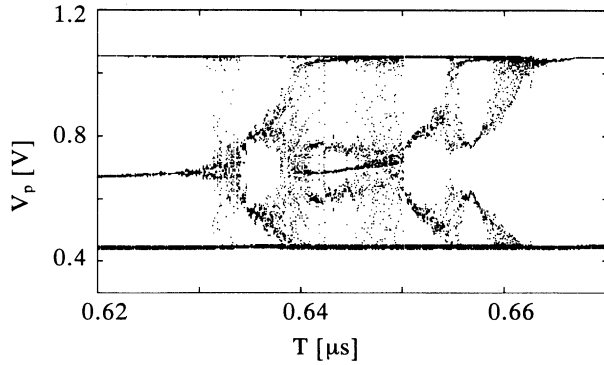
Figure 5: Magnification of area (2) in Fig. 3.
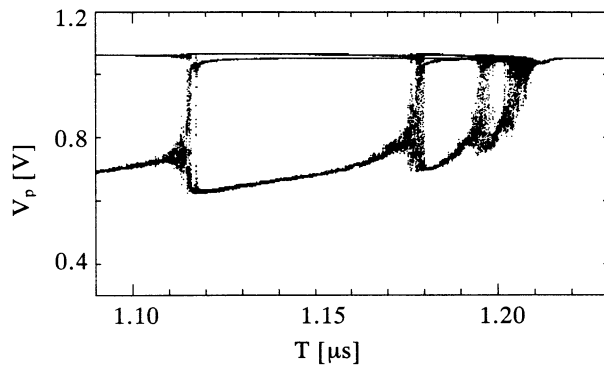
Figure 6: Magnification of area (3) in Fig. 3.

A magnification of area (1) in Fig. 3, where T is varied from 0.40[μs] to 0.52[μs], is shown in Fig. 4. As T becomes longer, a period-doubling bifurcation of period-2^k responses (k=0,1,2, ⋯) occurs from the period-one response under non-firing response in succession. Thus, the bifurcation of this area is a period-doubling bifurcation, and the periodic response bifurcates into the chaotic response through the successive period- doubling bifurcation route.

A magnification of area (2) in Fig. 3, where T is varied from 0.62[μs] to 0.67[μs], is shown in Fig. 5. Various periodic and chaotic responses were observed in this area.

A magnification of area (3) in Fig. 3, where T is varied from 1.09[μs] to 1.23[μs], is shown in Fig. 6. As T becomes longer, the periodic response bifurcates from the period-two response into the period-one response through the period-m responses (m=3,4,⋯), and chaotic response exists between each periodic response.

4. Return Maps

We observed the return maps of each chaotic area using PSpice in order to estimate the chaotic responses.
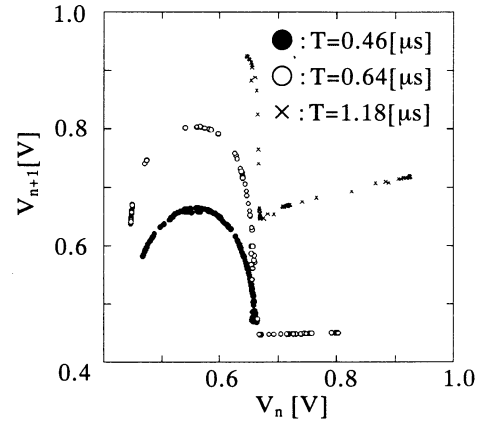
Figure 7: Return maps.

Figure 7 shows three examples of return maps, where V_n is the n-th discrete value of $v_{i,cell}$ obtained at the downward mode of the n-th pulse in the periodic pulse train stimulation, V_{n+1} is the next value of V_n. Solid circles (●) indicate the return map observed at the value of T=0.46[μs], which is similar to a logistic map having a single peak. Open circles (○) indicate the return map observed at the value of T=0.64[μs], which is similar to a logistic map, but having an additional flat area, and resembles the return map obtained by Hodgkin-Huxley equations taking into account the periodic pulses reported by Takabe et al. [12]. Crosses (×) indicate the return map observed at the value of T=1.18[μs], having a flat area of steep slope near the minimum point. Furthermore, if we identify the Lyapunov numbers from the return maps described by the solid circles, open circles, and crosses, the value of the Lyapunov numbers are approximately 1.39, 1.25, and 0.54, respectively. Therefore, these results prove the existence of the chaotic response.

5. Conclusions

We have discussed the features of the P-HCNM constituting from CMOS processing and have shown that the P-HCNM which consists of synaptic section and cell-body section could be constructed using CMOS. In addition, the bifurcation route in which the periodic response became unstable and bifurcated into the chaotic response occurred when the periodic pulse train stimulation was applied to the P-HCNM. Finally, the return maps obtained in the chaotic responses were shown to be approximately one-dimensional.

To experimentally study nonlinear dynamics of the chaotic neural network implemented by the P-HCNM is a problem for the future.

References

[1] W. J. Freeman, "Tutorial on neurobiology: from single neurons to brain chaos", Int. J. Bifurcation and Chaos, vol. 2, pp. 451-482, 1992.

[2] K. Aihara, T. Tanabe, and M. Toyoda, "CHAOTIC NEURAL NETWORKS", Physics Letters A, vol.144, No6,7, pp.333-340, 1990.

[3] K. Aihara and R. Katayama, "Chaos Engineering in Japan", Communications of the ACM, vol.38, no.11, pp.103-107, 1995.

[4] K. Aihara and N. Ichinose, "Modeling and complexity in neural networks", Artif. Life and Robotics, 3, pp.148-154, 1999.

[5] N. Ichinose and K. Aihara, "An Analysis on Pulse Propagation Dynamics in Asynchronous Chaotic Neural Networks", Trans. IEICE, J78-A, No.3, pp. 373-380, 1995.

[6] Y. Osana, M. Hagiwara, "Successive Learning in Chaotic Neural Network", Trans. IEICE (D-II), J82, No.1, pp. 83-90, 1999.

[7] M. Watanabe, K. Aihara, and S. Kondo, "Automatic Learning in Chaotic Neural Network", Trans. IEICE, J78-A, No.6, pp. 686-691, 1995.

[8] Y. Sekine, "A Pulse-type Hardware Neuron Model", Computer Today, No.90, pp.27-33, Science-sya, 1999.

[9] Y. Sekine, T. Kanai, T. Amamori, S. Takahashi and C. Takeuchi, "Two kind of pulse-type hardware neuron models and its application to a recognition circuit", Systems and Computers in Japan, 23(14), pp. 84-98, New York: John Wiley, 1991.

[10] Y. Sekine, A. Yamazaki, H. Kurosawa and N. Sato, "Universal Type Hardware Neuron Model Using Lambda-Shaped Transistor", Trans. IEICE (D-II), J78, No.1, pp. 131-139, 1995.

[11] J. Matsuoka, Y. Sekine, K. Saeki and K. Aihara, "Analog Hardware Implementation of a Mathematical Model of an Asynchronous Chaotic Neuron", to be submitted to IEICE Trans..

[12] T. Takabe, K. Aihara, and G. Matsumoto, "Response characteristics of the Hodgkin-Huxley equations to pulse-train stimulation", Trans. IEICE, J71-A, No.3, pp. 744-750, 1988.

A Design Method of Model Gene Networks

N. Ichinose
Human Genome Center, IMS
The University of Tokyo
Tokyo, 103-8639

K. Aihara
Graduate School of Frontier Science
The University of Tokyo
Tokyo, 113-8656

Abstract

Gene networks play an important role of biological functions in human and other creatures. We propose a method of designing model gene networks from gene expression data. We use a continuous-time switching network as the model, and show that the network can be constructed from binary switching data corresponding to gene expressions in continuous time. We also show that the construction can be done even if there is a known partial structure in the networks.

1 Introduction

1.1 Gene Network

When we consider gene expression and its function, an important concept is the "central dogma" proposed by F.H.C. Crick. As shown in Fig. 1, the central
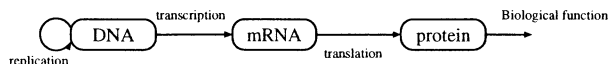
Figure 1: Central Dogma

dogma in genome science is the translational pathway from the genetic information to the protein, which is the functional factor in the body. The arrow in the figure shows the direction in which the genetic information is carried. Namely, after a partial region of the DNA sequence which codes the genetic information (called as coding region or "gene") has been transcripted to the messenger RNA (mRNA), the protein is generated from amino acids by using the mRNA as the template.

In the central dogma, a knowledge of a single gene seems to correspond to a single biological function. However, the actual situation is more complicated. Namely, the gene expression is controlled by transcriptional factors, which are also proteins translated from

the other genes. Figure 2 shows a sketch of transcrip-
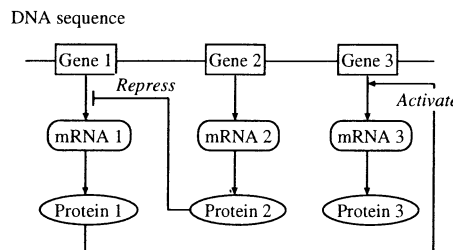
Figure 2: Sketch of transcriptional control

tional control of genes. The transcriptional factor is binded on the control region for the gene (usually its upstream region), then the transcription to the mRNA is activated or repressed. Because the transcriptional control is done by highly connected network, it is necessary to analyze the network structure and its dynamics in order to understand the biological functions of the genes.

1.2 Difference between Gene and Neural Network

The neural network is a well-known biological system and well studied. In this section we will show the difference between the gene and neural networks to understand the characteristics of the gene network.

An important point in the difference is their interactions between elements. In the neural network the neuron physically connects to the other neurons by the axons. On the other hand, in the gene network the gene chemically connects to the other genes by the proteins. Since the interaction between proteins is very rich, for example enzyme reactions, the gene interaction becomes higher order, while the neuron interaction is one-to-one order. Namely we can consider the interaction in which a gene is activated if two proteins exist simultaneously: these proteins can work as

the transcriptional factor if they are binded (see Fig. 3).
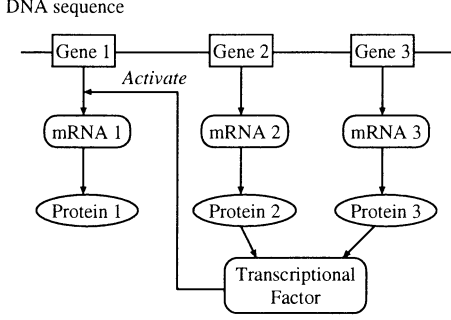
Figure 3: Higher-order interaction

As a theoretical extension of model neural networks, the higher-order neural networks have been proposed[1]. Because there may be no direct interaction between axons except for the special case (ex. synaptic-synaptic connection), the higher-order neuron is abstract. The gene network, however, generally has higher-order characteristics in the actually biological situation, therefore it may be an interesting object for model researchers.

1.3 Our Work

One of purposes in genome science is the gene therapy. Because the gene network should be directly controlled in the gene therapy, understanding of characteristics of gene network is important. Furthermore, we think that it is necessary that the gene network is controlled by itself, because the object of the therapy is tiny cells. Our purpose in this work, therefore, is to develop the method of designing model gene networks from gene expression data, and to understand a mechanism to express the desired genes in the known gene network by adding a partial gene network.

2 Continuous-time Switching Networks

2.1 Model

A model of continuous-time switching networks has been proposed for representing gene expression dynamics[2, 3, 4]. The model is represented as follows:

$$\frac{dy_i}{dt} = -y_i + F_i(x_1, x_2, \cdots, x_N), \qquad (1)$$

$$x_i = \begin{cases} 0 & y_i < 0 \\ 1 & y_i \geq 0 \end{cases}, \qquad (2)$$

where y_i is the variable corresponding to the expression concentration of the ith gene ($y_i \in R$), x_i is the binary variable representing expression or nonexpression of the ith gene ($x_i \in \{0, 1\}$), N is the number of genes in the network, F_i is the interaction function from all genes to the ith gene ($F_i : \{0, 1\}^N \to R$).

As mentioned in the previous section, the gene network has the higher-order interaction. Thus the interaction function F_i is modelled as follows:

$$\begin{aligned} F_i(x_1, x_2, \cdots, x_N) &= a_i + \sum_{j=1}^{N} \omega_{ij} x_j \qquad (3) \\ &+ \sum_{i \neq j \neq k}^{N} \omega_{ijk} x_j x_k + \cdots \\ &+ \omega_{i12\cdots N} x_1 x_2 \cdots x_N, \end{aligned}$$

where a_i is the basic expression concentration of the ith gene, and w_{ij}, w_{ijk} and $w_{i12\cdots N}$ are the weights of the interactions.

Although the model interaction we use is performed by the binary variable x_i, the actual interaction of genes is not binary. However it is known that the actual gene expression is sensitive to a transcriptional factor[5], therefore this model interaction is qualitatively acceptable. Since our purpose is the design of the gene network, then the simple model structure is an advantage. This is because the dynamics of gene expression should be represented by continuous time and continuous value of variable, but it is suitable for the design to adopt the qualitative binary expressions, as we will show it later.

2.2 Dynamics

The model of continuous-time switching networks is a piecewise linear system. In the system each linear region corresponds to a vector value of the binary variables $(x_1, x_2, \cdots, x_N)$. This implies that the region corresponds to the expression pattern. In a linear region the dynamics are simple: there exists a globally unique and stable equilibrium. The equilibrium, however, is not in the corresponding region always.

Figure 4 shows an example of toggle switch gene network and its dynamics, which is used in an experiment of genetic recombination[5]. In the network the parameters are fixed as: $N = 2$, $a_1 = a_2 = 1$ and $w_{12} = w_{21} = -2$. e_{00}, e_{01}, e_{10} and e_{11} are the equilibriums, namely:

$$e_{x_1 x_2} = (y_1^*, y_2^*) = (F_1(x_1, x_2), F_2(x_1, x_2)), \qquad (4)$$
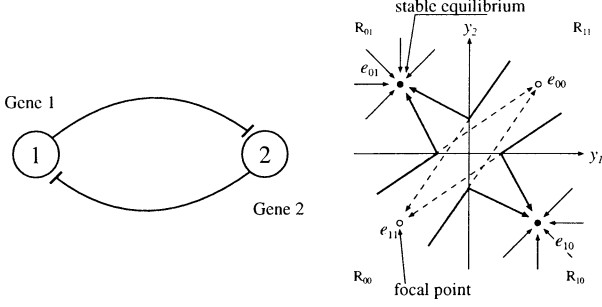
Figure 4: An example of toggle switch gene network and its dynamics.

where y_1^* and y_2^* are the values of the first and second genes at the equilibrium, respectively. R_{00}, R_{01}, R_{10} and R_{11} are the linear regions corresponding to the expression patterns $(x_1, x_2) = (0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$, respectively.

The equilibriums e_{00} and e_{11} are not in the corresponding regions: we call such equilibrium as a "focal point". Thus the network has the two stable equilibriums and the two focal points. The orbit starting from the region R_{00} or R_{11} goes to the corresponding focal point, but when the state reaches at the interface between the regions, it changes the direction and goes to the alternative stable equilibrium e_{01} or e_{10}. Consequently, we can say that a stable equilibrium and a focal point corresponding to each linear region are important for characterizing the dynamics in the model of continuous-time switching networks.

3 Design Method of Gene Networks

3.1 Method

The idea of the design is simple, namely we use the characteristics in which an orbit goes to the focal point in an expression pattern. For example, if we implement the pattern change $(0,0) \to (0,1)$, we set the focal point e_{00} as $e_{00} \in R_{01}$.

When a pattern change $(x_1, x_2, \cdots, x_N) \to (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_N)$ is given, for all i the following inequalities should be satisfied because of the condition of the focal point $(e_{x_1 x_2 \cdots x_N} \in R_{\hat{x}_1 \hat{x}_2 \cdots \hat{x}_N})$:

$$y_i^* = F_i(x_1, x_2, \cdots, x_N) \leq -\epsilon \qquad \text{if } \hat{x}_i = 0, \qquad (5)$$

$$y_i^* = F_i(x_1, x_2, \cdots, x_N) \geq \epsilon \qquad \text{if } \hat{x}_i = 1, \qquad (6)$$

where ϵ is a positive parameter (we fix it as $\epsilon = 1$). If the multiple pattern changes are given, we obtain a

set of the inequalities. Then a range of the parameters for the gene interactions is determined.

In our method we suppose that the pattern change occurs only on a single gene at the same time. If not, the orbit passes through multiple interfaces till the focal point, then the desired pattern change is not guaranteed.

In order to determine the parameter values in the range, we use the following evaluation function:

$$E = \sum_{i=j=1}^{N} |w_{ij}| + \sum_{i \neq j \neq k}^{N} |w_{ijk}| + \cdots + \sum_{i=1}^{N} |w_{i12\cdots N}| + \sum_{i=1}^{N} |a_i|,$$
$$(7)$$

namely this is the L_1-norm of the parameters. We determine the parameter values by minimizing the evaluation function E subject to the set of the inequalities. We adopt this L_1-norm evaluation function because the optimal solution becomes sparse[6], then we can obtain the sparsely interactive gene network. As the algorithm to minimize E, we adopt the linear programming method.

3.2 Example: oscillation network

We give the following oscillation patterns:

$$(0,0,0) \to (1,0,0) \to (1,1,0) \to (0,1,0) \to (0,1,1)$$

$$\to (1,1,1) \to (1,0,1) \to (0,0,1) \to (0,0,0) \qquad (8)$$

Figure 5 and 6 show the result of the design method
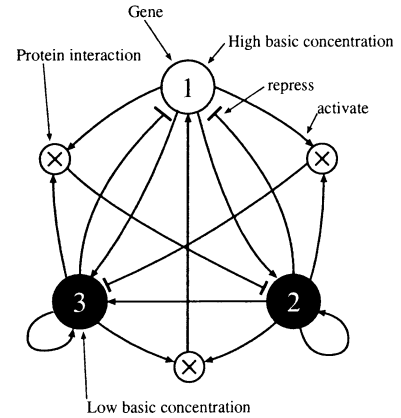
Figure 5: Result of the design method: oscillation network

by the given pattern and its dynamics, respectively. we can obtain the network which has the dynamics with the oscillation patterns we gave by our method.
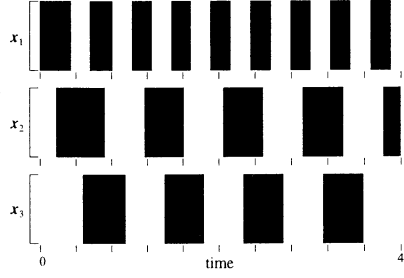
Figure 6: Dynamics of the oscillation network

3.3 Example: Control of known gene network

If there is a fixed network structure, we consider to control it such that each gene is expressed with desired patterns. Since the cell has many subnetworks corresponding to the biological functions, we consider the situation in which the functional subnetwork is controlled by the artificial gene network for the gene therapy.
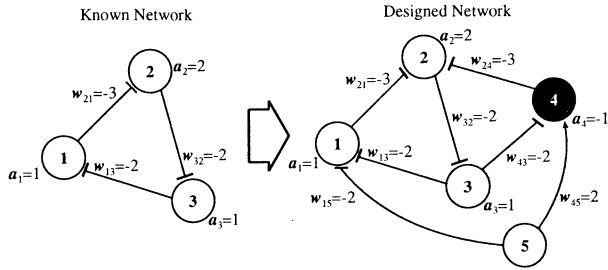
Figure 7: Control of known gene network

Figure 7 shows an example of the control of a known gene network. The left-hand side network is the known one, thus we fix those parameters on it. The right-hand size network is the designed network by the following two sets of the given patterns:

$$(1,0,0,0,0) \rightarrow (1,0,1,0,0) \rightarrow (0,0,1,0,0) \rightarrow (0,1,1,0,0)$$

$$\rightarrow (0,1,0,0,0) \rightarrow (1,1,0,0,0) \rightarrow (1,0,0,0,0), \quad (9)$$

$$(0,1,0,0,1) \rightarrow (0,1,0,1,1) \rightarrow (0,0,0,1,1) \rightarrow (0,0,1,1,1)$$

$$(0,0,1,0,1) \rightarrow (0,1,1,0,1) \rightarrow (0,1,0,0,1). \quad (10)$$

The first set is equivalent to the original expression patterns of the known network. we made the second set such that the oscillation is switched to it, if the 5th gene is expressed. Figure 8 is the dynamics of the
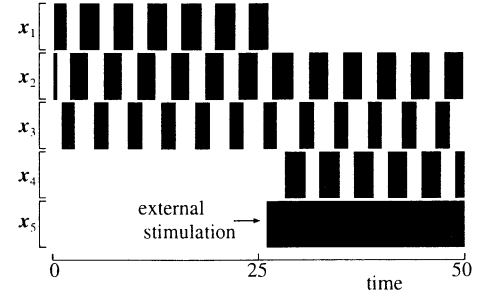
Figure 8: Dynamics of the designed network

designed network. We can see that the network state is changed by the 5th gene expression which we set as the control gene.

4 Summary

We have proposed the design method of gene networks and shown that we can obtain the network with the desired gene expression pattern.

As the future work, we will develop the method to design not only the pattern change but also the expression duration.

References

[1] S. Amari, Dualistic Geometry of the Manifold of Higher-order Neurons, *Neural Networks*, Vol. 4, No. 4, pp. 443-451, 1991.

[2] L. Glass, Classification of Biological Networks by their Qualitative Dynamics, *J. Theor. Biol.*, Vol. 54, pp. 85–107, 1975

[3] T. Mestl, C. Lemay and L. Glass, Chaos in High-dimensional Neural and Gene Networks, *Physica D*, Vol. 98, pp. 33-52, 1996

[4] R.Edwards, Analysis of Continuous-time Switching Networks, *Physica D*, Vol. 146, pp165–199

[5] T. S. Gardner, C. R. Cantor and J. J. Collins, Construction of a Genetic Toggle Switch in Escherichia Coli, *Nature*, Vol. 403, pp. 339-342, 2000

[6] K. Matsuura and Y. Okabe, A Robust Reconstruction of Sparse Biomagnetic Sources, *IEEE Trans. Biomedical Engineearing*, Vol. 44, No. 8, pp. 720–726, 1997

A Genetic Network Model on Selective Expressions

Yoshihiro Morishita[1] and Kazuyuki Aihara[1,2]

[1]*Department of Complexity Science and Engineering, The University of Tokyo, Hongo, Tokyo 113-8656, Japan*

[2]*CREST, Japan Science and Technology Corporation (JST), Kawaguchi, Saitama 332-0012, Japan*

Abstract

This paper describes a genetic network model for selective expressions of genes. We develop a simple unit model based on competition of activating and repressing pathways, that responds selectively to an input. We also discuss possible applications of the model.

Figure 1: DNA sequence

Figure 2: Regulations of transcriptions

1 Introduction

Recently progress in molecular biology has enabled exploring the mechanism of genetic information processing in organisms. So far, researches on the genome have focused mostly on the static analysis, such as to determine what part of the DNA sequence which encodes proteins and to clarify structure and functions of the proteins. To understand dynamic processes, such as metabolism and organism development, where the expression of genes changes with time, the mechanism of its regulation must be clarified. A genetic network is a mass of genes interacting with one another through expression and can be a fundamental model for analysis of these dynamic processes [1]. In this paper, we focus on selective expressions of genes, which means that certain genes are expressed under certain conditions both outside and inside a cell. A mathematical model for the selective expressions of genes was recently reported by Smolen et al [2]. We generalize their model and apply it to a genetic network. Based on the results, we propose a unit model for the selective expressions of genes and show its possible applications.

2 Genetic Network

The DNA is a sequence of four nucleotide bases, A, T, G and C (Fig. 1). A part of the sequence that encodes a protein is called a gene. In the synthesis process of a protein from a gene, a part of the sequence that includes the gene is transcribed into the mRNA, which is then translated into a protein. The

expression of a gene is mainly regulated at the transcription phase, and substances related to control of transcription are called transcriptional factors (TFs). A transcriptional factor binds to a transcriptional factor responsive element (TF-RE), in the upper stream of the gene whose transcription it regulates. Transcriptional factors are classified mainly into activators, which activate transcription, and repressors, which repress it (Fig. 2). Because a transcriptional factor is a protein, when a gene encodes a transcriptional factor that regulates the transcription of other genes, the genes interact through the expression. The genes in this interaction form a genetic network (Fig. 3).
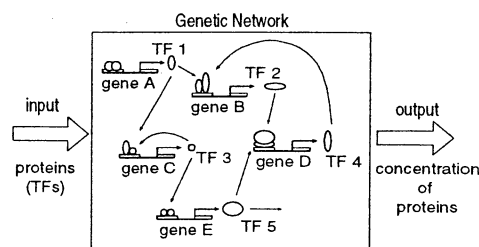
Figure 3: Genetic Network

3 Selective Expression

A cell must accurately recognize the conditions both inside and outside itself and responds accordingly. For example, for stimuli from outside the cell,

particular signaling pathways can be activated depending on the type and strength of the stimuli, which enables selective cellular response.

In this paper, we focus on the selective expressions, which are defined as the phenomenon where particular genes are expressed (i.e. coded-proteins are synthesized) for only the inputs of particular strength (Fig. 3)

4 A Genetic Network Model on Selective Expressions

A mathematical model for the selective expression of genes was recently reported by Smolen et al [2]. The model is a simple genetic regulatory one based on the competition between activators and repressors, that gives the maximum transcription for a stimulus of an optimal strength.

We generalize the model and develop a model that selectively responds to an input of a particular value (Fig. 4). First, in this model there are two pathways, the activating and repressing pathways stimulated by a common input. The activated pathways then compete on an output receptor. When the following conditions (and some additional conditions) are satisfied, the model can selectively respond to a particular input.

- The activating pathway is more sensitive to the input.

- The repressing pathway sends signals to an output receptor more easily.
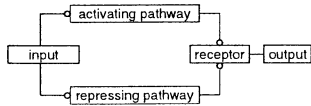
Figure 4: Generalized model

We apply the generalized model to genetic networks and propose an unit model for the selective expressions of genes, which we call the A-R unit. In the following, we describe this unit model in detail.

4.1 The A-R Unit

The A-R unit consists of the following four processes (Fig. 5).

1. Multimerization of the input transcriptional factors.
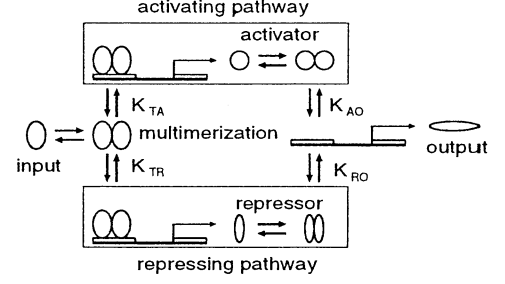
Figure 5: the A(Activator)-R(Repressor) unit

2. Combination of multimers and TF-REs of the genes coding the activator and the repressor with dissociation constants K_{TA} and K_{TR}, respectively.

3. Synthesis of the activator and the repressor, and multimerization of them.

4. Competition between multimers for the binding with TF-REs with dissociation constants K_{AO} and K_{RO}.

The variation of concentrations of the activator, the repressor, and the output protein are given by the following differential equations.

$$\frac{d[A]}{dt} = k_A \frac{[TF]^n}{K_{TA} + [TF]^n} - r_A[A], \tag{1}$$

$$\frac{d[R]}{dt} = k_R \frac{[TF]^n}{K_{TR} + [TF]^n} - r_R[R], \tag{2}$$

$$\frac{d[P]}{dt} = k_P \frac{D_t[A]^n}{[A]^n + K_{AO}(1 + \frac{[R]^n}{K_{RO}})} - r_P[P] \tag{3}$$

where [TF], [A], [R], and [P] are the concentrations of, respectively, the input transcriptional factor, the activator, the repressor, and the output protein. We assume here that the amount of TF-REs is dealt with as a concentration and that the concentration of each protein is proportional to that of each mRNA.

Terms of $[TF]^n/(K_{TA}+[TF]^n)$ and $[TF]^n/(K_{TR}+[TF]^n)$ in eqs. (1) and (2) are the concentrations of the complex of TF-REs and multimers of the input transcriptional factor in a steady state, where the reaction between the multimers and TF-REs is considered as an enzyme reaction with dissociation constants K_{TA} and K_{TR} [3]. The second terms in eqs. (1) and (2) are the rates of resolution of the activator and the repressor, which are proportional to the concentration of [A] and [R], respectively.

The first term in eq. (3) describes the concentration of the complex of TF-REs and multimers of the

activator in a steady state, and the second term is the rate of resolution of the output protein.

This unit can selectively respond to an input with a particular value, when the dissociation constants (K_{TA}, K_{TR}, K_{AO} and K_{RO}) are set as follows.

- K_{TA} is larger than K_{TR}, which means that the activating pathway is more sensitive to the input.

- K_{RO} is larger than K_{AO}, which means that the repressing pathway has higher affinity to the TF-REs.

Figure 6 shows the input-output curve of this unit model. We can see from Fig. 6 that this model responds selectively to the input.
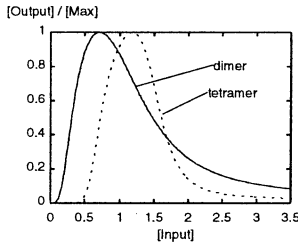
Figure 6: Input-Output curve

4.2 Basic Forms of Interaction and A-R units

Basic Forms of Interaction

There are two main forms of gene interaction; a series cascade and a convergence cascade.

Series Cascade
When the product of the first transcription is a transcriptional factor that causes a second transcription (Fig. 7 (left)), the genes interact in the form of a series cascade.

Convergence Cascade
When the products of the first two transcriptions are needed for a second transcription to occur (Fig. 7 (right)), the genes interact in the form of a convergence cascade.

(1) An A-R Unit in Series Cascade

Figure 8 shows what happens when the A-R unit is followed by a series cascade. As can see from Fig. 8,
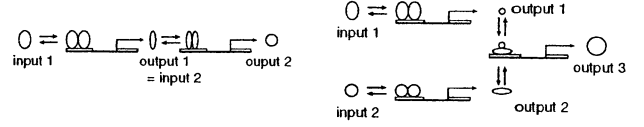
Figure 7: Series(left) and Convergence(right) cascades

the selectivity of an input is strengthened and it becomes more remarkable when the cascade is repeated. This means that this system can accurately recognize the analog value of an input.
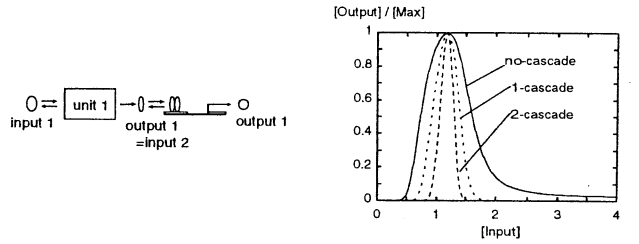
Figure 8: the model and input-output curve (1)

(2) A-R Units in Convergence Cascade

Figure 9 shows what happens when two A-R units are followed by a convergence cascade. In this system, because the coincidence of productions from the first two units is detected by the second transcription, the final output product is synthesized only for a particular set of input values $(input1, input2) = ([a_1, b_1], [a_2, b_2])$.
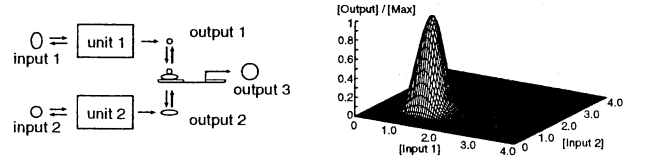
Figure 9: the model and input-output curve (2)

(3) Multiple A-R Units for Single Input

Figure 10 describes a model consisting of multiple A-R units and a single input. In this system, each unit has a different set of dissociation constants (K_{TA_i}, K_{TR_i}, K_{AO_i} and K_{RO_i}) and by setting them appropriately, each unit can be made to respond to selectively different values of the input. Here, the system is set as there is no interaction between the units which synthesize different proteins, therefore, it can transmit different information according to the value of the input.

(4) Multiple A-R Units for Multiple Inputs in
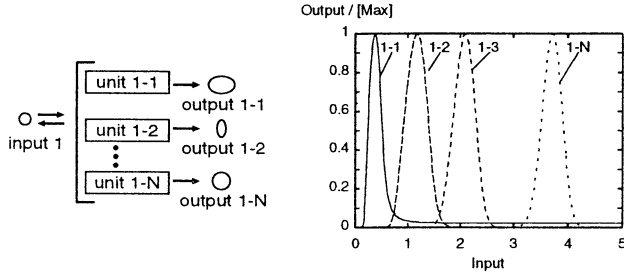
Figure 10: the model and input-output curve (3)

Convergence Cascade

Let's look at a model consisting of multiple A-R units, multiple inputs, and a convergence cascade. When the dissociation constants are set as in the former model and there is a parallel cascade, this system can recognize the values of multiple inputs $(input1, \cdots, inputN) = ([a_1, b_1], \cdots, [a_N, b_N])$ and synthesize different proteins for each set. Figures 11 and 12 shows how the model recognizes a set of two arbitrary input values $(([a_i, b_i], [a_j, b_j])(i, j = 1, \cdots, N))$.
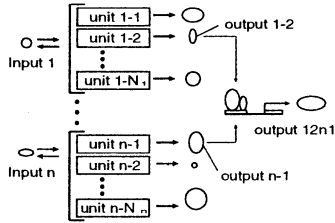
Figure 11: the model (4)
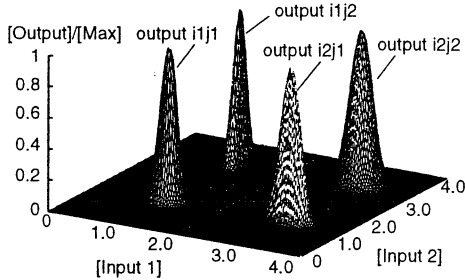
Figure 12: the input-output curve (4)

5 Conclusion

We have described a genetic network model for the selective expressions of genes by the A-R unit model, and have shown several phenomena which occur when the A-R units are combined in the basic form of interaction between genes.

The core of the selectivity of the A-R units is by appropriately choosing the dissociation constants of both the activating and repressing pathways, to control the sensitivity to an input and the affinity to an output receptor of both pathways. Therefore, by preparing sets of TF-REs and transcriptional factors with different dissociation constants, we may synthesize this unit model artificially. Although it remains unclear such a model does exist in real organisms, this model will be useful in designing an artificial genetic network.

Acknowledgements

References

[1] Paul Smolen, Douglas A. Baxter, and John H. Byrne (2000) "Mathematical Modeling of Gene Networks" *Neuron*, Vol. 26, pp. 567-580

[2] Paul Smolen, Douglas A. Baxter, and John H. Byrne (1998) "Frequency selectivity, multistability, and oscillations emerge from models of genetic regulatory systems" *American Journal of Physiology*, Vol. 274, pp. C531-C542

[3] Donald Voet, Judith G. Voet, *BIOCHEMISTRY*, John Wiley & Sons, Inc., 1995

A Study on the Predictability of High-frequency Financial Data

Mieko Tanaka-Yamawaki

Department of Computer Science and Systems Engineering
Miyazaki University, Miyazaki, 889-2192 JAPAN
mieko@cs.miyazaki-u.ac.jp

ABSTRACT

We study high-frequency data of USD/JPY exchange rate for over six years that contains ten million data points of 'ask' position and attempt to make a short-term prediction. We first show that the conditional probabilities of up/down motion at each TICK level for memory size up to four are remarkably stable for the entire time scale considered here. Although the stability is gradually lost as the memory length increases, the basic compositions remain almost the same for the entire range. We have also done the similar research on an individual stock price of 3.19 million data points and obtained qualitatively the same result although the level of the stability is not as high as the USD/JPY data. Based on these results, we show that it is possible to predict the next motion with a very high rate. Our result shows that the hit-rate can be larger than 75 percent in up/down motion and the error is less than a quarter of the error for the dummy-shift prediction, that is made by shifting the original data by one tick time.

1. Introduction

Predicting financial time series is a big challenge to all the theorists. It is widely believed that the financial market is efficient and it is not possible to make a profit simply by buying and selling one currency to another. Recently, high-frequency financial data (tick data) became available for academic use and many scientists began analyzing those data in order to study more quantitatively on various hypotheses in economics including the question of efficiency of the market. We have studied various kinds of tick data including foreign currency exchange rates, stock prices, stock index future prices and have observed that foreign currency data exhibits high level of regularities compared to the others. We discuss in this paper on the remarkable level of flatness of the conditional probabilities of up/down motion for a long interval of time scale that we have recently observed in studying the tick data of USD/JPY exchange rate for the past six years.

It is a wide belief in financial engineering that the price determined in a free competition is efficient and very close to a perfect gamble, since a clear chance of making a profit is immediately wiped away by prompt actions of investors. Recently, Ohira et. al. have shown that this naive myth is to be reconsidered [3]. Motivated by this observation, we investigated much wider sets of data and found

that the stability persists for a very long term compared to the average decision time of most investors. Also we have roughly identified the upper limit of the memory length of foreign currency market and some stock prices and make a comparison between them.

The process of our analysis is as follows. We first separate the 'ask' position data from the USD/JPY from January 2, 1995 to April 12, 2001, which has 10127289 data points. Then split them into fifty sets of data, having 200,000 data points each. For each memory depth, m (m=1-7), there are 2^m conditional probabilities, which are computed for 50 data sets and written on the 2^m automatically generated files. Figure 1 (a), (b), (c), (d) show the conditional probabilities plotted as a function of the time order of the 50 data sets. We use {0,1} to represent {down, up} motion of the tick level price change and neglect data if there is no change from the previous one. There are four conditional probabilities, $P(00|1)$, $P(01|1)$, $P(10|1)$, $P(11|1)$ for m=2, which are plotted in one graph of Figure 1 (a) in ascending order of the conditions. Similarly m=4 case is plotted in (b). Here only the first eight lines out of total 16 are shown due to the limitation of the space. The figure shown as (c) contains the first eight probabilities out if total 32 for the case of m=5. Finally the figure (d) shows the first eight out of 128 for the case of m=7. We can see that the level of stability is remarkably high for m=2 to m=4 throughout the entire time scale, and turns rougher for m=5, and further for m=7.
One naturally expects from the result shown here that the prediction is possible by utilizing the constant transition probabilities between 2^m

Markovian states for the memory length m. In this paper we will show our preliminary result of such scenario. The procedure of this program for predicting the next direction of motion based on the binary memory length m is as follows. In the first program, we compute a set of 2^m conditional probabilities and generate output files. Then in the second program we compute predicted values of each tick value by simply adding the previous data to the step width of the previous move times the probability told by the conditional probability computed in the first program. The rates of correct hits in terns of direction are 75% for all the data sets, and the absolute errors compared to that of the dummy-shift prediction are less than 1/2.

Acknowledgment

References

[1]Toru Ohira, et.al.``Predictability of Currency Market Exchange", cond-mat0107074.

Figure 1 Stability of conditional probabilities for the case of memory depth up to seven. The figures show that the level of the stability deteriorates for the depth larger than 4. (a) Case of memory depth 2, where P00 indicates the probability of P(--|+), P01 indicates the probability of P(-+|+), P10 indicates the probability of P(+-|+), P11 indicates the probability of P(++|+). P(-+|+) indicates the probability that the price increases after downside motion followed by the upside motion, etc.
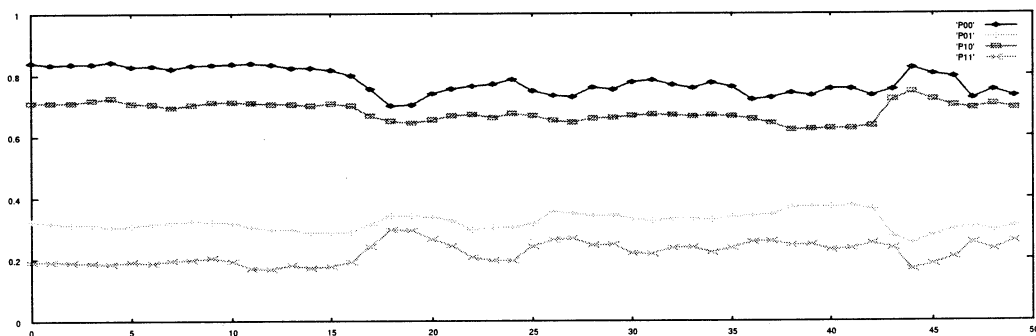
(b) Case of memory depth 4. Eight lines indicate P(0xyz|1) for x, y, and z are binary numbers.
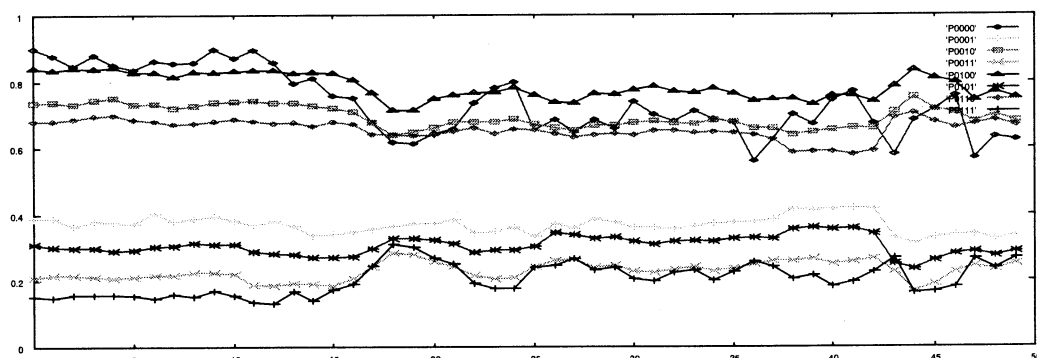
(c) Case of memory depth 5. Eight lines indicate P(00xyz|1) for x, y, and z are binary numbers.
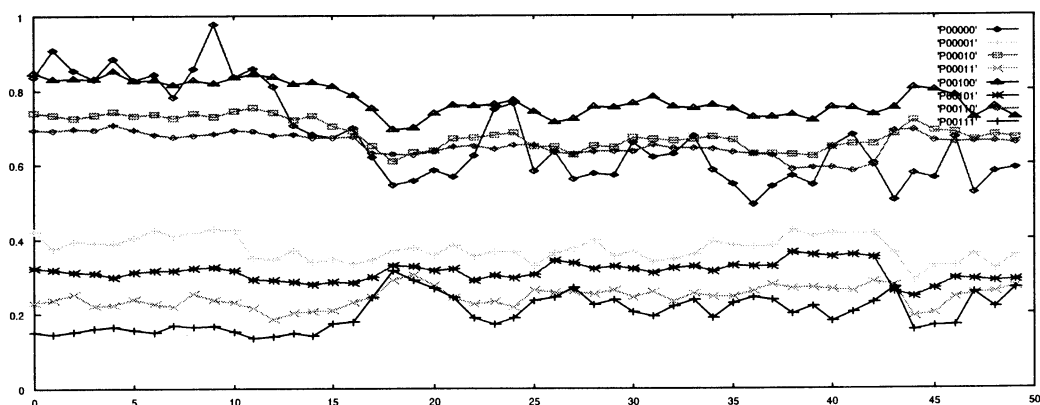
(d) Case of memory depth 7. Eight lines indicate P(0000xyz|1) for x, y, and z are binary numbers.
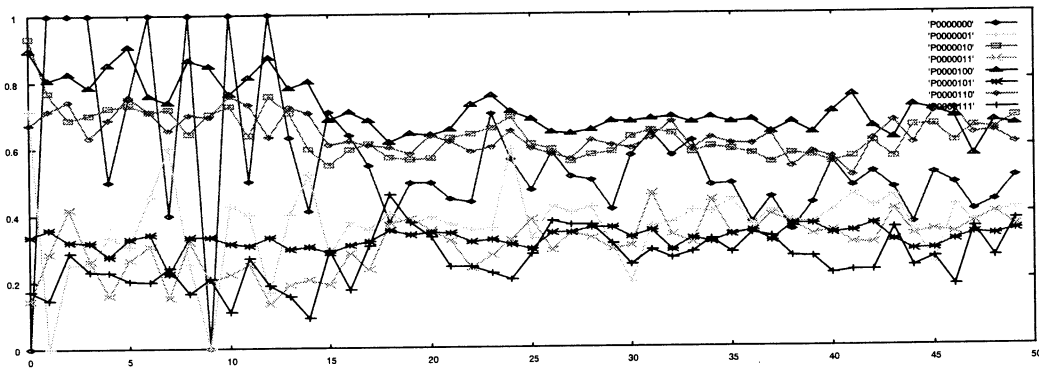
Figure 2   One tick prediction on the direction of motion for m=2. Solid line shows the tick points in the middle of the USD/JPY currency exchange data from 1995 to 2001. Crosses indicate the tick-wise prediction. The rate of hitting the correct direction throughout the data points is 75.1%, and the error is 0.000682 which is less than a quarter compared to the error of the dummy-shift prediction, 0.003130.Hit rate = 0.751321 (13937 / 18550). Error = 0.000682, less than a quarter of the error of the dummy-shift prediction = 0.003130. (data:USDJPY.a4 (a4dat.22), conditional probability: prd2a4.21)
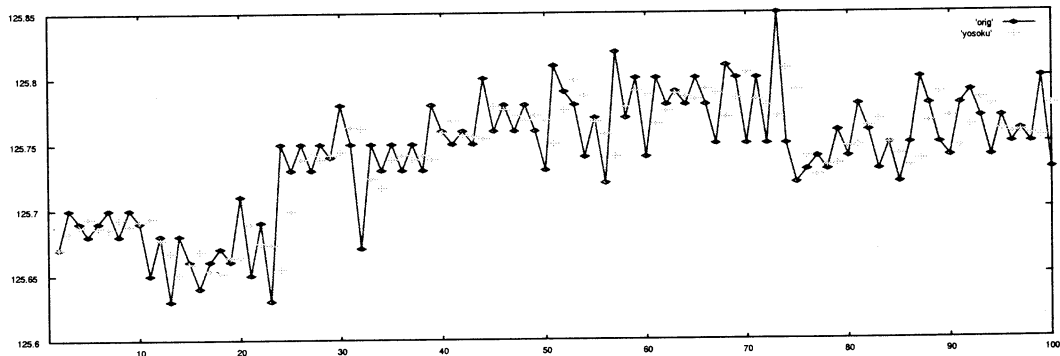
(b) Case of memory depth m=7 for the same prediction as in (a). The rate of hitting the correct direction throughout the data points is 75.0836%, and the error is 0.000694 which is less than a quarter compared to the error of the dummy-shift prediction, 0.003130. (data:USDJPY.a4 (a4dat.22), conditional probability: prd7a4.21)
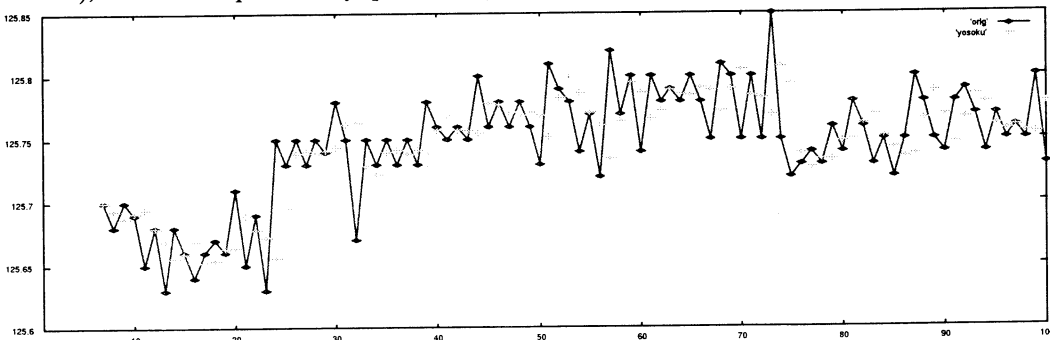
Does information flow in high-frequency financial data as energy does in turbulence?

Mieko Tanaka-Yamawaki, Tsuyoshi Itabashi and Shinya Komaki

Department of Computer Science and Systems Engineering

Miyazaki University, Miyazaki 889-2192 JAPAN

mieko@cs.miyazaki-u.ac.jp

Abstract

We study various high-frequency financial data in order to see how far the analogy between the scaling law in financial time series and the well-known scaling law in turbulence. The issue was first pointed out by Ghashghaie et. al. in 1996 by comparing the probability distribution of the price increment and the corresponding quantity in fully-developed, homogeneous, isotropic turbulence in three dimension. They asserted that there is a net flow of information from long to short time scales in financial market in the same way as flow of energy cascades in turbulence from large to small size eddies in turbulence. We study this hypothesis in more quantitative details by using tick data that became available recently.

1. Introduction

Scaling property is universally observed in many seemingly unrelated phenomena. Among them, the observation reported by Ghashgaie, et.al. [1] that the Kolmogorov's scaling law [2] for turbulence is approximately satisfied in foreign currency exchange market was one of the most exciting example. However, it has been rather difficult to access the high frequency data (tick data) until recently. We have studied various kinds of tick data including various foreign currency exchange rates, stock prices, stock index future prices, etc. that became available to us very recently. Considering the fact that the Kolmogorov's law

$$\langle (\Delta v)^n \rangle \propto (\Delta r)^{n/3} \quad ...(1)$$

is valid in a limited sense for turbulence itself, we take the hypothesis as a guiding rule to find a possible law that the financial market follows if there is any.

2. Currency exchange rates, stock price, and others

We compare the results of probability distributions and the moments for the following six categories: (a) the exchange rate of US dollar versus Japanese yen for six years (10127289 data points); (b) the first one million data points of (a); (c) the exchange rate of US dollar versus German mark (3851066 data points); (d) Euro versus Japanese yen (2242565 data points); (e) stock (IBM in 1998, first 200000 data points) and (f) model output for a parameter set of a=0.06, b=0.08 (80000 data points).

In drawing the histograms of the price increments, we noticed that the choice of the partitions (i.e. the inverse of the width between two points of horizontal axis) is rather crucial. We chose the three best candidates for the number of partitions as 50, 100 and 150 and checked if the results are the same for all the three choices. In Figure 1, the ones in the left hand side are the probability densities and those in the right hand side are the moments of n=2,4,6 plotted in the log scales on both axes.

USD/JPY shows the highest regularity both in the probability distributions and the scaling law of the moments. As in ref.[1], not only the power law indicated by the straightness of log-log plot of the moments as a function of time resolution parameter Δt,

$$\left\langle \left(\Delta x\right)^n \right\rangle \propto \left(\Delta t\right)^{\xi_n} \quad ...(2)$$

but also the power of Δt being the same as in the case of turbulence:

These formulas are held very well in USD/JPY data in(a) and (b). Although USD/DEM data in (c) do not have a high level of regularity compared to other data, the situation is much better for EURO/JPY data in (d). On the other hand, USD/DEM do not behave as regular as other data The moments formula do not holds for a stock price and a model output

as shown in (e) and (f).

Conclusion

We have drawn pdf for six different price data and computed n-th moments. It is found that n-th moments for the data USD/JPY follows the scaling law parallel to the case of turbulence, while other data have different laws.

Acknowledgment

References

[1] S. Ghashghaie et. al., "Turbulent cascades in foreign exchange markets". Nature, 381, 767-770 / (1996)

[2] A.N.Kolmogorov, "The local structure of turbulence in incompressible viscous fluid for very large Reynolds number", Dokl. Acad. Nauk. SSSR, 30, 9-13 (1941)

Figures Probability density distributions (pdf, computed as normalized histograms) of price increments for various levels of time resolution Δt and n-th moments of the price increments ΔX = X(t)-X(t-Δt) computed from the pdf in the left hand sides.

Figures

(a) USD/JPY.ask_all　(10127289 data points)
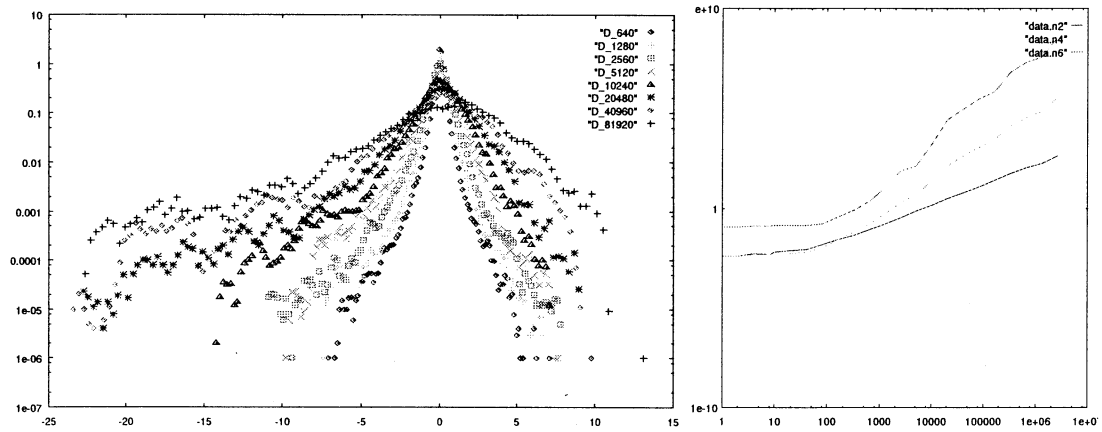
(b) USD/JPY.ask.1(1000000 data points)
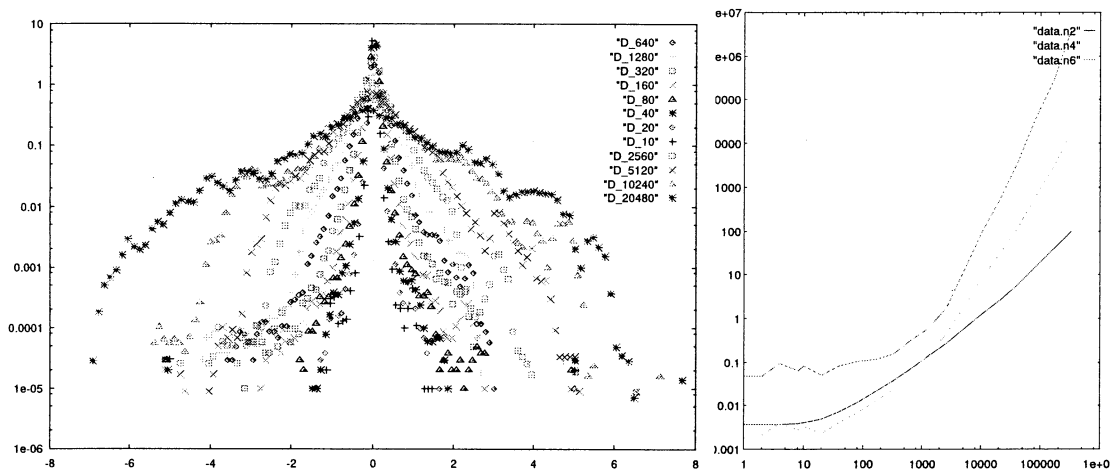
(a) USD/DEM ask (3851066 data points)
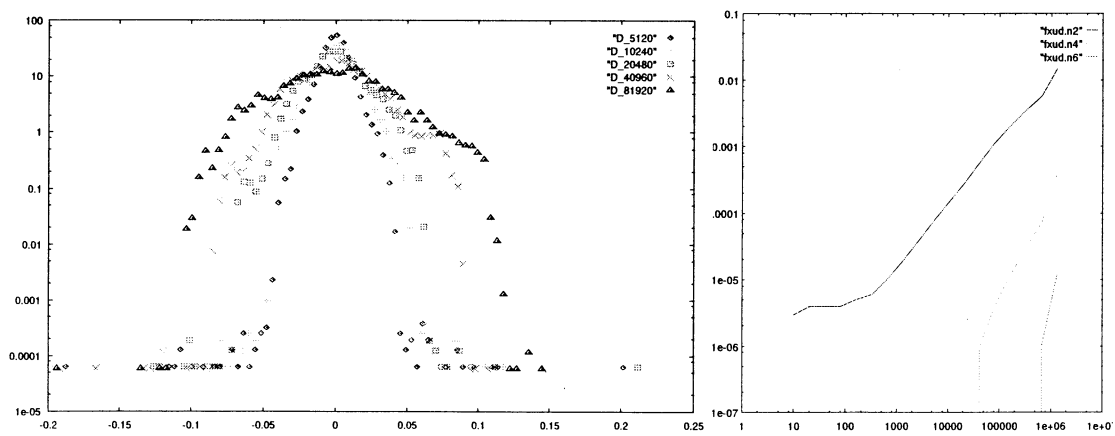
(b) eurjpy.a (2242565 data points)
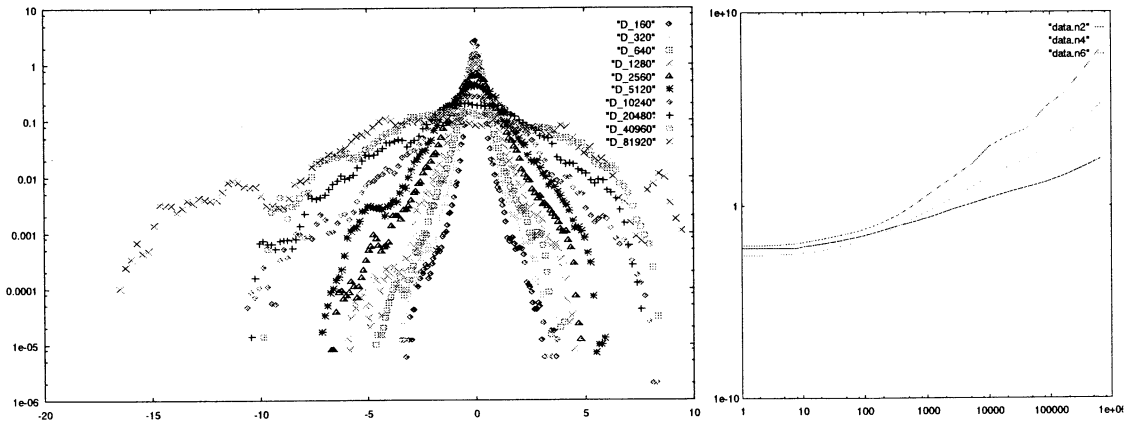
(e) Stock price time series (IBM) : 200000 data points
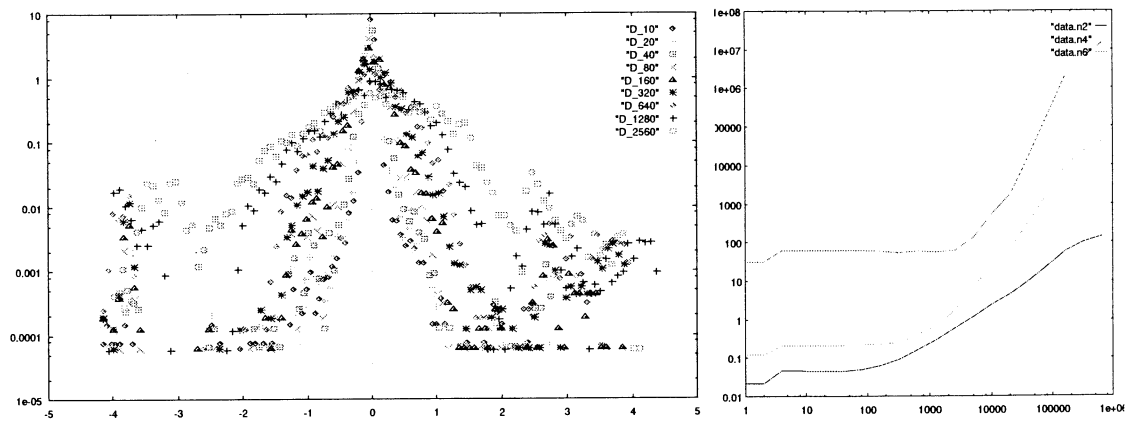
(f) Model output for a parameter set of a=0.06, b=0.08 (80000 data points)
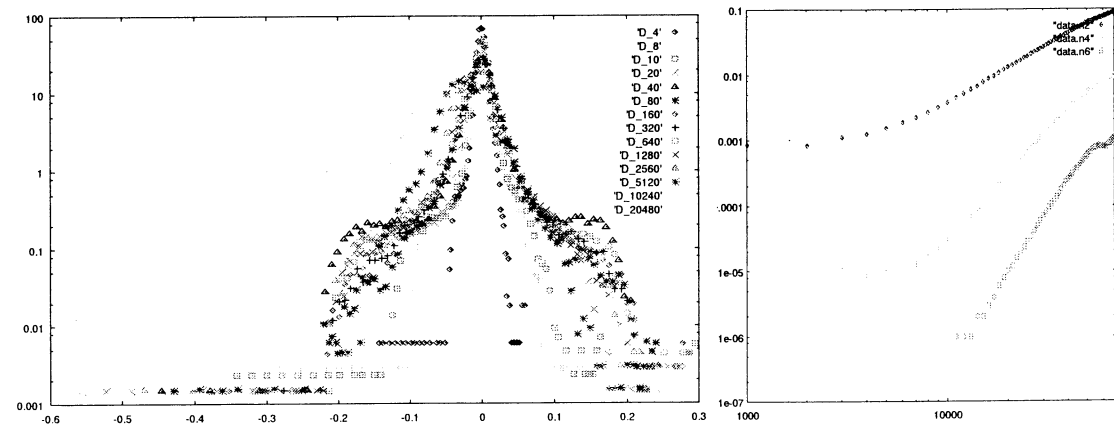
Complexity Science for the Design of Intelligent Geometry Compressors for Jet Aircraft Engines

Jeffrey Johnson, Anthony Lucas-Smith, and Paul Wiese

EPSRC Intelligent Geometry Compressor Project, Department of Design and Innovation,
Faculty of Technology, The Open University, Milton Keynes, MK7 6AA, UK.
j.h.johnson@open.ac.uk, a.j.lucas-smith@open.ac.uk, p.wiese@open.ac.uk

Abstract

The compressor in a jet aircraft plays a crucial role in overall performance. We consider compressors with moving rotor blades and stationary stator blades. The rotary motion of the former in the presence of the latter causes pressured airflow. The incidence angle of the stator blades affects performance, depending on aircraft conditions such as take-off and cruising. Concepts from complexity science are applied to this system, to see if the stator blades can *self-organise*, by changing their blade angles, to produce optimum performance, as an *Intelligent Geometry Compressor*. A simple simulation supports the tentative conclusion that this is possible.

1. Introduction

A jet aircraft engine works on the principle that fuel is burned under the great pressure created by a compressor at the front of the engine (Fig. 1). The compressor has two types of blade, the stationary *stator* blades, and the *rotor blades* (Fig 2, Fig. 3).

The behaviour and performance of the engine are dependent, among other things, on the shapes and angles of the blades. In particular, the optimum stator blade angle for aircraft take-off may be different to the optimum angle for cruising [1]. Some engines have actuators that allow the stator blades to be set at one angle for take-off, and another for cruising .

In this paper we investigate the possibility of the blade angles being continuously reset

during the operation of the engine to achieve optimal overall performance.
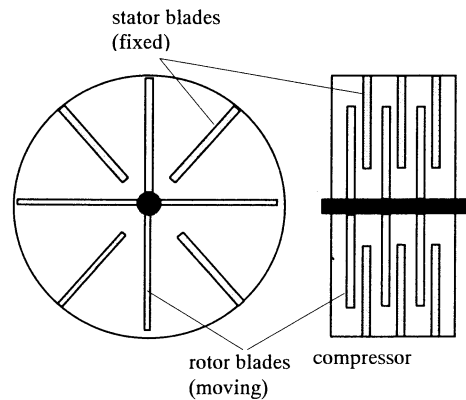
Figure 2.   A simplified compressor
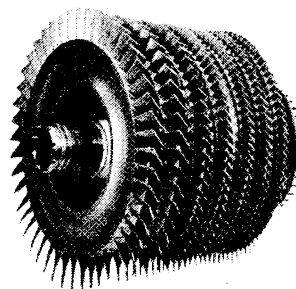
Figure 3. An axial flow compressor (stators omitted for clarity). High pressure compressor from a General Electric F404 engine.
(Source:.http://www.soton.ac.uk/~genesis)
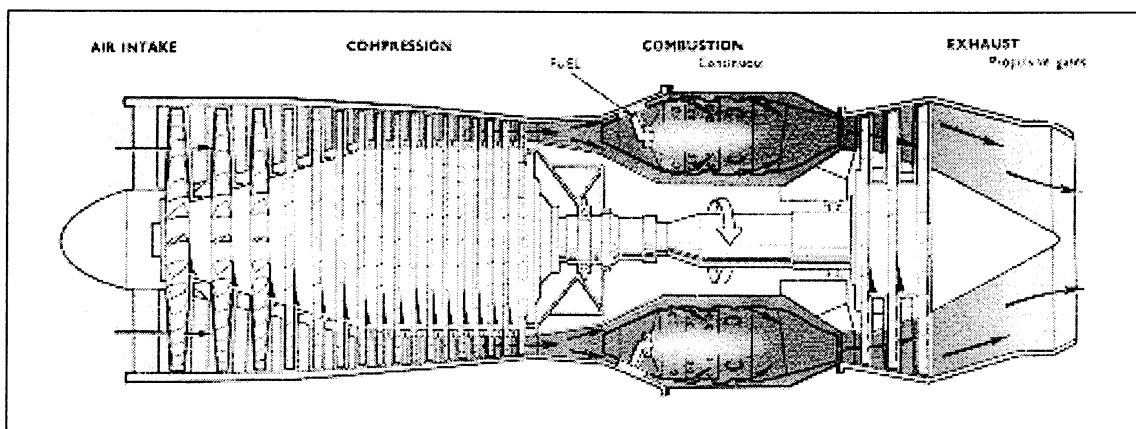
Figure 1. A schematic of a gas-turbine engine (Source:. http://www.soton.ac.uk/~genesis/Level2/Tech)

This leads to a complex control problem. We have investigated this from the perspective of what we call *complexity science*. The central idea is a shift from top-down control to bottom-up control in which the blades are considered to be *autonomous intelligent agents*. Based on *local* information, the blades themselves decide and implement their own optimal angle. The behaviour of the whole compressor then *emerges* from all these autonomous decisions.

2. Complexity Science

In physical systems, complexity science can be seen as an extension of the familiar sciences of physics, chemistry, and biology, and it often uses the same analytic mathematics. The extensions come from ideas and observations that have evolved over the last fifty years in response to problems for which conventional science has provided no solutions.

2.1 Representation

Generally, complex systems have to be modelled and analysed within computers. *Representation* is one of the central problems in Artificial Intelligence: how can things which seem so obvious to human beings be put inside computers? Physics has developed the mathematics of space-time as an incredibly powerful way of representing objects and their dynamic interactions. But this representation has its limitations.

In complexity science, systems usually have to be *represented* at different levels of detail and aggregation, corresponding to methods of *observation*.

For examples, a coiled steel spring could be represented at the molecular level, as billions of atoms of iron, carbon, and other elements. In principle, the behaviour of the spring could be calculated on the basis of the interacting atomic forces inside the spring, and at its boundary. In practice, this would be a difficult thing to do. The number of variables and equations for such a representation would be very large.

An alternative is to represent the spring at a higher level at which one can measure aggregate phenomena such as length and mass. Then one finds that, within certain limits, $\Delta\,length = k \times \Delta\,mass$, *i.e.* Hooke's Law. The point here is that both the concepts of length and mass refer to highly aggregate objects, and they are *directly observable*.
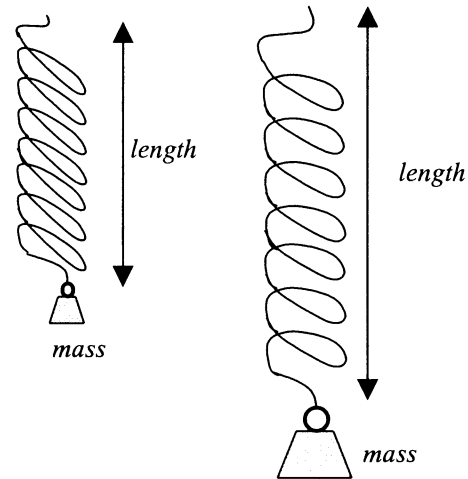
Figure 4. Representing an extended spring at a highly aggregate level.

The spring could be modelled at an intermediate level, between the atoms and the macroscopic object using an analytic description of its shape and internal forces, and using the computational technique of Finite Element Analysis.

In principle this could give insights into where the spring might fail. However, assuming the spring was designed to have regular properties along its length, failure would be due to random manufacturing variances which are, by definition, unpredictable. Then one falls back on the macroscopic principle, that the spring will not fail when the mass applied is within given operational limits.

2.2 Chaos and Computational Irreducibility

The ingredients of chaos include sensitivity to initial conditions and boundedness. In practice this means that very slight changes in initial conditions may result in wide divergence in system trajectories (Fig. 5), but the system does not go to infinity.
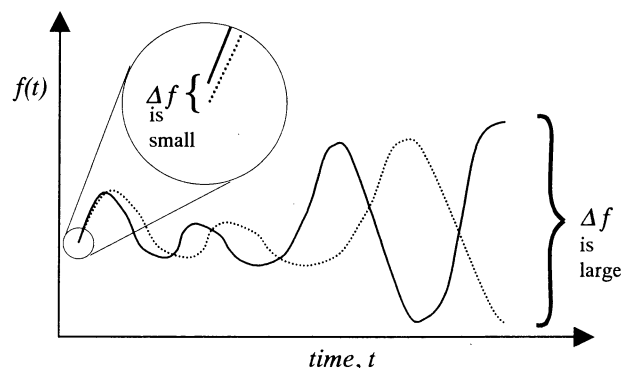
Figure 5. Sensitivity to initial conditions: small differences in initial conditions may result in large differences in system trajectory

Technically, chaos requires a little more than this, but sensitivity to initial conditions is important from a control perspective.

When systems are chaotic it is not possible to make long-term predictions of their behaviour. Even if the system is deterministic, the impossibility of replicating initial conditions exactly may mean that a prediction based one initial position may be very different from another initial condition that cannot be discriminated by the measurement technique. How can such systems be controlled reliably?

Many systems are operated 'on the edge of chaos'. In other words, the state space may not be everywhere chaotic, but the system performance improves as chaotic regions are approached. Can these systems be controlled within the chaotic region itself? To some extent, closed-loop control systems do exactly this. By continuously monitoring system variables and implementing control actions, such systems are kept to the desired trajectory.

A large class of systems is *computationally irreducible*. This means that to find the state of the system at time $t+k$, it is necessary to compute all the intermediate states, $t+1$, $t+2$, etc. Predicting the future states of computationally irreducible systems involves computing all the intermediate states. Thus, predicting the future states of computationally irreducible systems requires *computer simulation*.

Simulation is widely used in engineering. Finite Element Analysis, the basis for many simulations, can be thought of a computational extension of physics. Generally the finite element method is used where the shapes of objects is so irregular or complex, that finding simple formulae for the boundaries is intractable. With the representation based on regular polyhedra, complex shapes become computationally tractable because the physics can be broken down over the polyhedra, and re-assembled over their faces in coherent ways.

Given a control strategy, finite element methods allow its consequence to be investigated. The question still remains as how one might control the compressor blades to optimise system performance.

2.3 Self-organisation versus top-down control

Figure 6 shows schematically the possible positions for the sensors and actuator on a turbine. Typically, there are dozens of blades

and therefore many measurements, and many actuators to be controlled.
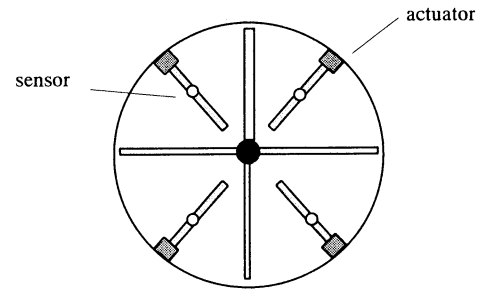
Figure 6. Possible positions for the sensors and the actuators on the stator blades.

One approach is to attempt to formulate a model of this systems as the basis of a controller. This assumes that one knows how the system responds to control inputs. In the case of a compressor, the system is quite complex, and this may not be known.

In complexity science it is common to have many autonomous intelligent agents, *e.g.* the vehicles in road systems self-organise, with emergent features such as shock waves and tailbacks. Other examples include birds and bees that self-organise into flocks and swarms. As far as we know, there is no chief bird or chief bee directing operations; system-behaviour emerges from the actions and interactions of individuals.

In many complex systems, no agent knows everything, and each agent decides what to do on the basis of local knowledge. So, it will be supposed that the blades will self-organise on the basis of physical measurements in their vicinity such as the pressure and their angles.

In this context, we have investigated the capability of the stator blades in aircraft compressors to self-organise to produce the most appropriate geometry for any given conditions.

Suppose that the state of the system at time t can be represented as s_t. Then the self-organising approach requires that some kind of transition mechanism, f, is formulated such that $s_{t+1} = f(s_t)$. In particular, let θ_t be the angle of a stator blade at t. Then we need a function f: $(\theta_t, p_{1t}, p_{2t}, p_{3t}, \ldots) \rightarrow \theta_{t+1}$, where $p_{1t}, p_{2t}, p_{3t}, \ldots$, are 'relevant' variables in a neighbourhood of the blade. We also need a function, g: $(\theta_t, p_{1t}, p_{2t}, p_{3t}, \ldots) \rightarrow p_{i,t+1}$, for $i = 1, 2, 3, \ldots$

The general idea here is that the angle of the stator blades will *co-evolve* with the adjacent pressures. The function g is provided by the physics of the compressor. The function f depends on the design of the controller.

2.3 Simulation Science

We formulated a computer simulation, in the context of what we call *simulation science*. This reflects the fact that simulation has its own theory, and this plays an important role in what is knowable.

When systems are computationally irreducible, as discussed above, any attempt to predict their behaviour is, effectively, a simulation.

Simulation science is a counterpart, or an extension, of mathematics and computer science. In the same way that the Halting Problem constrains what is computable, and that Gödel's theorem constrains what is knowable, there are limitations to simulation that place new constraints on what can be known or predicted.

Some of these constraints are due to the nature of parallel-distributed computation. Of more relevance here, is the 'can you trust it?' problem of simulation science [3].

A computer simulation involves a mapping of a real system onto explicit objects or structures within computers. The computer executes transition rules that compute new system states from old. By running the simulation iteratively through time, system states can be computed into the future.

The problem lies in the appropriateness of the mappings onto the representation, and the 'correctness' of the transition rules. There are many examples of simulations in which the representation is clearly inadequate, and the transition rules do not adequately reflect reality. Would you trust such a simulation?

At the moment there is no simple test that can be applied to a simulation to see if it is 'correct'. Of course, there are many considerations that can be invoked to give degrees of confidence in simulations. Perhaps the simplest is the 'Popper Test': if the simulation gives results that are inconsistent with observation, then it is not correct.

Unfortunately, for practical reasons, the issues are more complicated than this. When systems are chaotic, any single observation or simulation may mean nothing. In this case one needs to make many observations, and work with the statistics of distributions of outcomes. Then the Popper Test requires that simulations give distributions consistent with those that can be observed.

3. The Computer Simulation

In our research we are trying to show that it is possible to have turbines with self-organising blades, that can co-evolve dynamically with their surroundings to give desirable outcomes.

In particular, we propose to move away from a top-down analysis of the system to a bottom-up approach in which parts of the system have their own 'intelligence' and autonomy, allowing them to decide their own actions according to 'local' conditions. This assumes that parts of the compressor will have autonomy in what they do

3.1 Representation in the simulation

Our approach involves a discrete way of representing the jet engine. For this, we consider a division of the two-dimensional cross section of the compressor divided into pie-shaped *segments* (Figure 7).

The idea is that in time Δt, a 'tick of the clock', the rotor blade will move from a segment boundary to the next. Then, instead of variables such as pressure being continuous with the segments, there is a discrete value for such quantities.
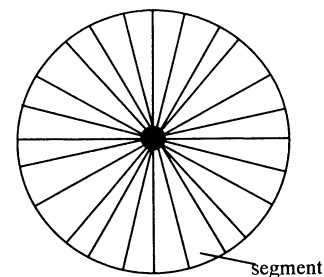
Figure 7. Cross section of the compressor divided into segments.

3.2 The transition rules

We hypothesise that the dynamics obey *transition rules* that map the pressure and blade angle for a segment at time t to that at time $t+\Delta t$.

A fundamental idea is that the domain of this mapping is *local* to the segment. This comes

from the general principle that in complex systems, no-one knows everything.

The aerodynamics of compressors are very complex, and a dynamic three-dimensional simulation was not appropriate for this research.

So, we addressed head-on the "can you trust it?" problem. Let us say at the outset of a simulation "No, of course you can't trust it!". that's not the point. The point is that the exercise may inform the research.

In our case we ask, can the principle of self-organisation work in a simplified system? If it can't, then why should it work in a real system?

Thus we assume that each segment, s_i has a pressure measurement, $p(s_i)$, and that each blade, b_i, has an angular measurement, $\theta(b_i)$. $p(s_i)$, will be abbreviated to p_i, and $\theta(b_i)$ will be abbreviated to θ_i (Fig. 8).
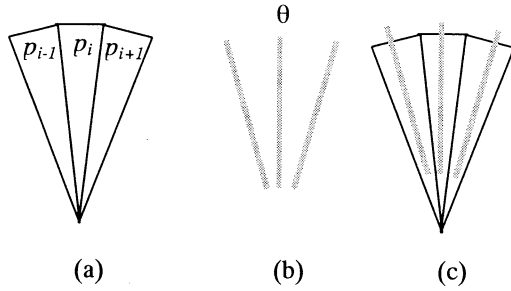
Figure 8. Segments, mapped to pressures; and stator blades, mapped to angles: (a) segments, (b) stator blades, (c) stator blades & segments.

For the simulation we considered a deviation from a desired steady state in which the initial pressures varied.

Thus we considered variables $\Delta\theta(b_i)$ and $\Delta p(s_i)$, for each blade and segment. For simplicity we assume there is one segment defined for each blade. $\Delta p(s_i)$, will be abbreviated to Δp_i. $\Delta\theta(b_i)$ will be abbreviated to $\Delta\theta_i$

We then defined the following rules

If Δp_i 50 then $\Delta\theta_i = -10$
else
If Δp_i 50 then $\Delta\theta_i = 10$
else
If Δp_{i-1} small and Δp_{i-1} small and $\Delta p_{i-1} > 0$
then $\Delta\theta_i = -1$
else

If Δp_{i-1} small and Δp_{i-1} small and $\Delta p_i < 0$
then $\Delta\theta_i = 1$
else
If Δp_{i-1} small and Δp_{i-1} small and $\Delta p_i = 0$
then $\Delta\theta_i = 0$
else
If $\Delta p_{i-1} < 0$ and $\Delta p_{i-1} < 0$ *and* $\Delta p_i < 0$
then $\Delta\theta_i = - (\Delta p_{i-1} + \Delta p_i + \Delta p_{i+1})/3$
else
If $\Delta p_{i-1} > 0$ and $\Delta p_{i-1} > 0$ *and* $\Delta p_i > 0$
then $\Delta\theta_i = - (\Delta p_{i-1} + \Delta p_i + \Delta p_{i+1})/3$
else
If $\Delta p_i < 0$
$\Delta\theta_i = $ | $\min\{ \Delta p_{i-1} + \Delta p_i + \Delta p_{i+1}\}$ |
else
$\Delta\theta_i = $ - | $\min\{ \Delta p_{i-1} + \Delta p_i + \Delta p_{i+1}\}$ |

The idea behind these rules is that the change in angle, $\Delta\theta_i$, of stator blade b_i will be determined by the deviation of the pressures measured in the segments s_{i-1}; s_i, and s_{i+1} from some specified pressure.

In real systems, the physics comes for free. In simulations, the physics has to modelled, and in our case, we used the rule

$$\Delta p_{i,t+1} = (\Delta p_{i-1,t} + \Delta p_{i,t} + \Delta p_{i+1,t} + \Delta\theta_t) / 3$$

meaning that the deviation in pressure for cell s_i at time $t+1$ is the sum of the deviations at t plus the change in angle, and divided by 3. In other words the pressures would be averaging, and changed by the blade angle change. The validity of this is discussed in the next section.

3.3 Results

This simulation was run many times, with the initial pressures set at random. These random configurations corresponded to severe disturbances in the pressures, much in excess of those that might be experienced in real aircraft engines.

None the less, every time the simulation was run, the blades self-organised so that the system rapidly converged to the desired steady state.

3.4 Can you trust it?

When we ran this program for the first time, we did not know what to expect. The transition rules had been defined in such an arbitrary way, that we were surprised at the outcome. In other words, we had no *a priori* reason to trust that the simulation would do anything interesting.

In fact the simulation did do something interesting, in as much as the blades self-organised, and in the absence of any top-down control, the system did co-evolve to the desired steady state.

One thing is certain: this simulation cannot be trusted to be a representation of any real system. The rule that gives the pressure at time $t+1$, based on the pressures at time t and the change in blade angle, is not based on any physical principle, and to that extent is arbitrary. So this simulation is a simulation of an artificial system that may have nothing to do with jet aircraft engine compressors. Does the simulation have any value?

We found that the simulation did inform our research. First, if we could not make this artificial system self-organise to achieve a desired outcome, why should we expect to make a real compressor blade system self organise?

Also, this simulation is a first attempt. As we reflect on it, we can critically decide why we do not trust it. We have already focused on a major problem in our representation of the physics of the gas transitions. But now we know have to look for something more realistic in this area.

There are other questions too. Is a single pressure value sufficient to represent the air pressures of a segment? How do the pressures vary over the segment? How many values are required; it has to be finite, because we can only deploy a finite number of sensors).

In this case the simulation has given us useful insights into the control problem we are investigating. This is often an important outcome in simulation science. Sometimes it is useful not to confuse the simulation model with learning from the simulation model.

5. Emergent control of stall cells?

There is a well-known phenomenon in which the blades 'stall' in the passing airflow. Then areas of low pressure, *stall cells*, move round the axis, at half the speed of the in the opposite direction to the blades [1].

This situation is highly dynamic, making conventional mathematics difficult to apply. In this case, conventional control approaches may not work because the dynamics are too complex.

We have identified this as a situation to which emergent control from self-organising stator blades might be applied. It will the subject of further analysis and simulation.
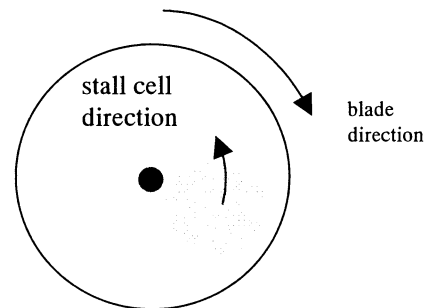
Figure 8. A stall cell moves around the axis at half the engine speed in the opposite direction.

6. Conclusions

In the context of complexity science, we have investigated the possibility of using intelligent self-organising blade agents to controlling the blade angles of turbines on jet engines. We have built a simulator that suggests that self-organisation is a possible control strategy. We have criticised this simulation from the perspective of the 'can you trust it?' problem. The results suggest that self-organising control strategies are possible, but much more realistic simulations are necessary for this tentative conclusion to be more definitively established.

References

[1] Paduano, J.D., 'Active control of rotating stall in a low speed axial compressor', *Trans ASME*, **115**, 48-56, Jan. 1993.

[2] Rolls Royce plc, *The Jet Engine*, 5th Edition, 1996, Technical Publications Department, Rolls Royce plc, (Derby), ISBN 0 902121 2 35

[3] Johnson, J.H., 'The 'can you trust it?' problem of simulation science', *Complexity*, *Complexity*, Vol. **6**, No 2, 34-40, 2001.

[4] Longley, J. P., 'A review of nonsteady flow models for compressor stability', *Trans ASME*, **166**, 202-215, April 1994.

3D Position Analysis and Real Time Transportation
of an Object Using Image process

Dong Youp Lee
Dept. of Interdisciplinary Program in Mechatronics
Pusan National Univ.
Pusan 609-735, Korea

Jong Il Bae
Dept. of Electrical Engineering
Pukyung National Univ.
Pusan , Korea

Sung Kwun Kim
Samsung Electronic Co.Ltd.,
Korea

Man Hyung Lee
School of Mechanical Engineering
Pusan National Univ.
Pusan 609-735, Korea

Abstract

The purpose of this paper is to obtain 3D-information on an object by a low cost USB compatible PC camera. This system can be implemented to calculate the height of an object by using triangle method to a relation between reference points of circumstance and edge points of an object. In this paper, it is possible to make and set up the system by Java program in all computer operating systems. This study has two advantages : one is easy to make and set the system by Java program, The other is very compatible in all computer operating systems. Moreover, this camera system can be easily used with no capture board. The JMF and Applet of Java can be also used in the Internet. In addition, a Real-Time JPEG/RTP network has been developed by UDP/IP on the Ethernet.

1. Introduction

From old times many researches to make measurement devices like the five human senses have been developed. Now the better sensors than the senses of human are used in the actual industrial field. But when compared with the researches of other kinds of sensors, many visual sensors have not been developed until these days. Moreover, the monitoring systems using a camera[2] have been numerously studied. A watcher always keeps an image by a camera under surveillance because the monitoring systems used recently show an image. He cannot recognize a movement of a particular object and not obtain any information of an object. The most monitoring cameras are mostly fixed on a particular area and the one made specially is composed as a system moving up and down. The camera used in this system is to be fixed and to process information of an image gotten.

The characteristic of this system is to be able to compute the absolute position of an object in the area, where a camera is installed, by using edges and characteristic points of poles with the limited information of an image. In this paper we make it possible easily to implement the system by using a USB[5] port camera, simply to compose the monitoring system by making use of the network devices constructed in a building and the computer used in an office currently, and not only to show the system applied a image processing technique [1][3][7] but also to inform the information of a image. By using JAVA[4] to implement all above, we make it possible to install the system regardless of any operation system, which we is used wherever internet[6] is available.

2. Trigonometric relation

The trigonometric relation equation is used to identify the height of a particular object.

2.1 Circumstances of the place where a camera installed

The below situation is installed because the position of a monitoring camera used is set in the intersect point of an entering road or a corridor.
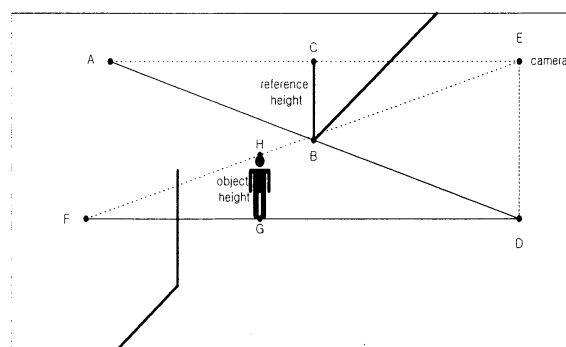
Fig. 1 Height of an object and reference height

A simple trigonometric equation is induced from Fig. 1 in order to measure the height of an object, where the reference height is meaning the corner of a pillar

standing erectly on the bottom. And this is used as the reference to measure the height of an object.

2.2. Inducement of trigonometric relation equation

The height H of an object is assumed to be perpendicular to the surface in the Fig. 1. And then there are triangle $\triangle EDF$ and $\triangle EDA$ which have common height DE of the height of a camera, where each has the line FD and the line AD as a base line respectively. The following equations are obtained with a resemble ratio to each triangle when seeing the relation of two figure by using the fact that two triangle are perpendicular to a corridor.

$$\frac{\overline{FG}}{\overline{FD}} = \frac{\overline{GH}}{\overline{DE}} \tag{1}$$

$$\frac{\overline{AB}}{\overline{AD}} = \frac{\overline{BC}}{\overline{DE}} \tag{2}$$

The equation (3) is obtained when arranging about common denominator DE of each equation (1), (2).

$$\overline{GH} = \frac{\overline{FG}\,\overline{AD}}{\overline{FD}\,\overline{AB}}\overline{BC} \tag{3}$$

Therefore we can see that the height GH of an object is the relative height of the reference BC. The more this reference object is perpendicular the more correct the height of the object measured is. Therefore the height of an object is measured by using the corner of a wall as the reference height.

2.3. Absolute Coordinate system

A trigonometric fundamental equation (3) is induced in order to calculate the height of an object above. An absolute coordinate system of an actual corridor for each point is needed to apply the equation to an actual image. This is possible to be accomplished that by a coordinate transform function which present a relation between a position of an input image and a position of an actual corridor.

3. Image-Model transformation

A plane projective transformation, a general plane-to-plane mapping, is introduced to transform the position of an input image into the position of an actual corridor.

3.1 Plane projective transformation

Fig.2 shows that points on arbitrary plane S, which exist on the 3-D space, are projected on an image plane s by a camera projection. This mapping is generally called

a plane-to-plane mapping. The plane-to-plane mapping relation by a camera projection is that a plane projection is accomplished in the viewpoint of a projective geometry. This plane projection is defined by at least 4 corresponding points between a plane S(A, B, C, D) and a plane s(a, b, c, d). The transformation defined in this way is called the plane projective transformation.
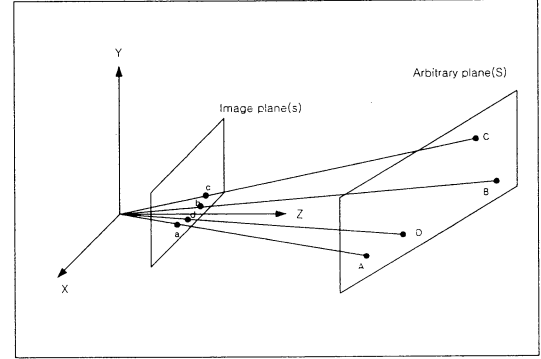
Fig. 2 Image of a 3-D arbitrary plane according to Camera projection

A plane projective transformation is presented as a linear equation using a matrix by using a homogeneous coordinate system. Then, this plane projective transformation is presented as a 3*3 square matrix, so the relation between a plane S and a plane s is denoted as the following linear matrix equation

$$X_2 = TX_1 \quad \text{or,} \quad \text{(where, } \lambda \text{ is scale)}$$

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{4}$$

Where, $X_2 = (x_2, y_2, 1)$ the position of a given point in a plane S coordinate system, $X_1 = (x_1, y_1, 1)$, the position of a given point in a plane s coordinate system. A 3*3 square matrix T representing the relation between this two coordinates is called a plane projective coordinate system. The transformation T depends on 9 freedoms. Given a scale term, it actually depends on 8 freedoms below.

$$\lambda \begin{bmatrix} x_{2i} \\ y_{2i} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{1i} \\ y_{1i} \\ 1 \end{bmatrix} \tag{5}$$

where, x_{1i} and x_{2i} represent more 4 correspondences between two planes. The a_{33} is 1 at the equation (5), which is the form scale that term disappears. The following equations are obtained from equation (5).

$$\lambda x_{2i} = a_{11}x_{1i} + a_{12}y_{1i} + a_{13} \tag{6}$$

$$\lambda y_{2i} = a_{21}x_{1i} + a_{22}y_{1i} + a_{23} \tag{7}$$

$$\lambda = a_{31}x_{1i} + a_{32}y_{1i} + 1 \qquad (8)$$

The following 2 independent equations are obtained from above equations

$$x_{2i} = \frac{a_{11}x_{1i} + a_{12}y_{1i} + a_{13}}{a_{31}x_{1i} + a_{32}y_{1i} + 1} \qquad (9)$$

$$y_{2i} = \frac{a_{21}x_{1i} + a_{22}y_{1i} + a_{23}}{a_{31}x_{1i} + a_{32}y_{1i} + 1} \qquad (10)$$

Therefore 2 independent equations are obtained from one correspondent point. So a plant projective transform T(to have 8 freedoms) is obtained if only 4 correspondent points are given. Let the equation (9), (10) be arranged

$$x_{1i}a_{11} + y_{1i}a_{12} + a_{13} - x_{1i}x_{2i}x_{31} - y_{1i}x_{2i}a_{32} = x_{2i} \quad (11)$$

$$x_{1i}a_{21} + y_{1i}a_{22} + a_{23} - x_{1i}x_{2i}x_{31} - y_{1i}x_{2i}a_{32} = y_{2i} \quad (12)$$

Above two equations are represented as the form of matrix for 4 correspondent points (N=4).

$$Aq = z \quad \text{or,}$$

$$
\begin{bmatrix}
x_{11} & y_{11} & 1 & 0 & 0 & 0 & -x_{11}x_{21} & -y_{11}x_{21} \\
0 & 0 & 0 & x_{11} & y_{11} & 1 & -x_{11}y_{21} & -y_{11}y_{21} \\
x_{12} & y_{12} & 1 & 0 & 0 & 0 & -x_{12}x_{22} & -y_{12}x_{22} \\
0 & 0 & 0 & x_{12} & y_{12} & 1 & -x_{12}y_{22} & -y_{12}y_{22} \\
x_{13} & y_{13} & 1 & 0 & 0 & 0 & -x_{13}x_{23} & -y_{13}x_{23} \\
0 & 0 & 0 & x_{13} & y_{13} & 1 & -x_{13}y_{23} & -y_{13}y_{23} \\
x_{14} & y_{14} & 1 & 0 & 0 & 0 & -x_{14}x_{24} & -y_{14}x_{24} \\
0 & 0 & 0 & x_{14} & y_{14} & 1 & -x_{14}y_{24} & -y_{14}y_{24}
\end{bmatrix}
\begin{bmatrix}
a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32}
\end{bmatrix}
=
\begin{bmatrix}
x_{21} \\ y_{21} \\ x_{22} \\ y_{22} \\ x_{23} \\ y_{23} \\ x_{24} \\ y_{24}
\end{bmatrix}
\quad (13)
$$

Then the vector q is obtained like below

$$q = A^{-1}z$$

A transformation T is obtained from each term of the vector q. Where, in order that the inverse of matrix A may exist, each row of A is independent. For this, at least 3 points of 4 points used as corresponding points must be non-collinear. Otherwise, the inverse of A is not exist because a dependency among rows of matrix A happens due to a dependency among 4 points. By using the driven T, the positions of corresponding points on an arbitrary plane though equation (4) are obtained for others points on one plane.

3.2 In case N is 4 over

In case the number of corresponding points is 4 over. A transformation T is obtained by a matrix equation like below.

$$Aq = z \quad \text{or,}$$

$$
\begin{bmatrix}
x_{11} & y_{11} & 1 & 0 & 0 & 0 & -x_{11}x_{21} & -y_{11}x_{21} \\
0 & 0 & 0 & x_{11} & y_{11} & 1 & -x_{11}y_{21} & -y_{11}y_{21} \\
x_{12} & y_{12} & 1 & 0 & 0 & 0 & -x_{12}x_{22} & -y_{12}x_{22} \\
0 & 0 & 0 & x_{12} & y_{12} & 1 & -x_{12}y_{22} & -y_{12}y_{22} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
x_{1N} & y_{1N} & 1 & 0 & 0 & 0 & -x_{1N}x_{2N} & -y_{1N}x_{2N} \\
0 & 0 & 0 & x_{1N} & y_{1N} & 1 & -x_{1N}y_{2N} & -y_{1N}y_{2N}
\end{bmatrix}
\begin{bmatrix}
a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32}
\end{bmatrix}
=
\begin{bmatrix}
x_{21} \\ y_{21} \\ x_{22} \\ y_{22} \\ \cdot \\ \cdot \\ x_{2N} \\ y_{2N}
\end{bmatrix}
\quad (15)
$$

The vector q is obtained like the following equation by the pseudo inverse.

$$q = (A^{T}A)^{-1}A^{T}z \qquad (16)$$

At least 3 points of N points used in here must be non-collinear.

3.3 The Model of a Corridor and the Coordinate of an Input Image

An actual corridor model requires to see where of an actual corridor the given position of an input image is.
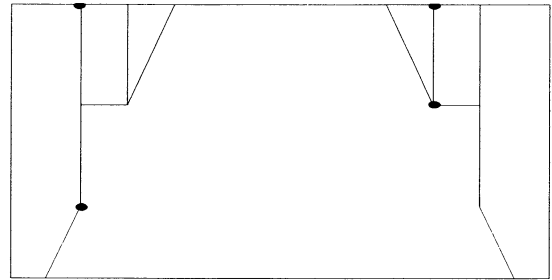
Fig. 3 Model of a corridor

Fig. 3 shows the model made by measuring the speculation of the corridor where a camera is installed actually. On the basis of this model an image coordination is transformed into the reference coordination of model. Particular points in the corridor are marked in the Fig. 3. If the model of a corridor is put as a plane S , the image of an input is put as a plane s, the transformation T between the model and the input image can be obtained by using the matching point of a model shown in a input image and the 4 matching points between input images.

4. Experiment results

In this experiment the reference points are set by fixing a camera in similar region to the model of a corridor. And the height of an object from the reference point can be measured.

Fig. 4 Measurement of the distance from the camera and the height of an object

Fig.4 shows the measured values of the distance from the camera and the height of an object entering the corridor. On the other hand, RTP is used in the experiment of a data transfer.
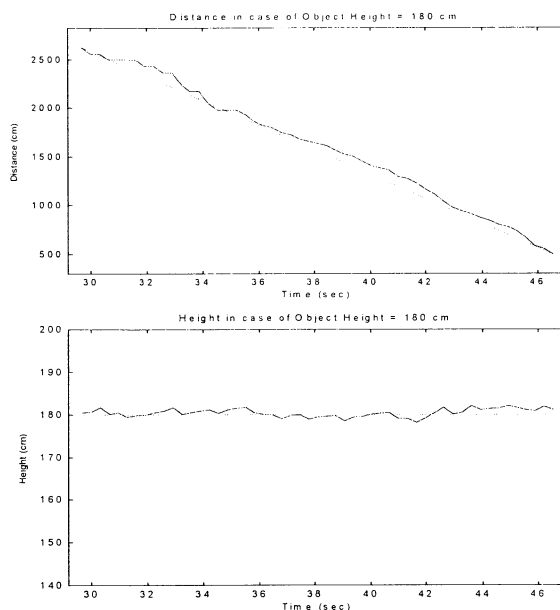
Fig.5 Measurement graphs of distance and height for the object of 180cm

Fig. 5 shows the Measurement graphs of distance and height for the object. Where, a dotted line is reference and a solid line is real data.
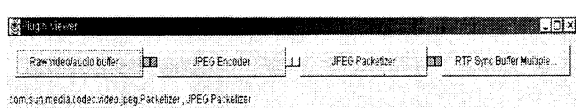
Fig. 6 RTP process

Fig. 6 shows RTP process. File format is set as raw video/audio format. After the format is encoded as JPEG format, data is packetlized. And then the data is transferred.
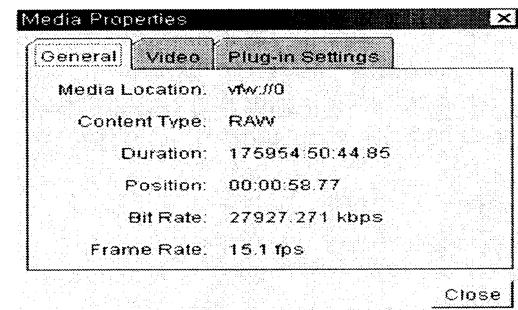
Fig.7 Data transfer rate

Fig. 7 shows the data transfer rate.

5. Conclusions

In this experiment the height of an object and the distance between a camera and an object are measured, but there are still a lot of problems. A sudden variation of illumination makes the measurement of a camera impossible. And a lot of delay problems occur during rtp communication.

In future researches in illumination are more needed. And the technology of compression to reduce a large quantity of data in order to solve the delay problems

Reference

[1] Johnson I Agbinya and Daviud Rees, "Multi-Object Tracking in Video," CSIRO Telecommunication and Industrial Physics, Image & Signal Processing Group, PO Box 76,Epping, NSW 2121, Australia,1999.
[2] Seo, Y., Choi, S., Kim, H. and Hong, K.-S., "Where are the ball and player? Soccer game analysis with colorbased tracking and image mosaic," Proc. ICIAP,1997.
[3] Cavallaro, R. "The Foxtrax hockey puck tracking system," IEEE Computer Graphics Applications 17 : 6 - 12 , 1997.
[4] JAVA Media Framework API Guide, Sun Microsystems, Inc. 1998-99.
[5] Universal Serial Bus Specification, Compaq, Intel, Microsoft, NEC, 1998.
[6] Th. Markousis, D. Tsirikos, M. Vazirgiannis, Y. Stavrakas, "WWW-enabled delivery of interactive multimedia documents," Computer Communcations 23 242-252, 2000.
[7] Christoph Bregler, "Learning and Recognizing Human Dynamics in Video Sequences," Proc. IEEE Conf. Comp. Vision and Pattern Recognition, San Juan, Puerto Rico, 1997

Might we train an artificial brain as a baby?

Yongguang ZHANG
Institute of Systems Science,
Academia Sinica, Beijing, China
E-mail: yzhang@staff.iss.ac.cn

Masanori SUGISAKA
Department of Electronic and Electrical
Engineering, Oita University, Japan
E-mail: msugi@cc.oita-u.ac.cn

Abstract

We have many discussing about how to develop intelligent robot with artificial brain by using computer? We aim to the future's prospective research. In this paper we describe a principle of design for Trainable Artificial Brain (TAB). Computer vision and hearing are equipped with our TAB and based on rough set theory knowledge base might be created, the TAB can obtain knowledge by training it as a baby. This TAB would be evolvable. According to the analysis it is possible now tto create a prototype of TAB.

Key words: Artificial Brain; Computer vision; computer hearing; Multi-modal information processing.

I. Introduction

In the last quarter of 20th century, the development of Automation and Robotics has an obvious feature that this is a more interdisciplinary field. Also, the development of Biology stimulates the traditional discipline to combine more new biological thinking. For example, in the research of robotics the humanoid and biped robots appeared at the end of the 20th century. The Honda's Asida and Sony's Aibo caused a huge interesting over all of the world. Unfortunately, although their shape looks like human being and they can walk by two legs, also they have CCD camera vision and speaker to say some words, but they has no BRAIN. They cannot understand the environment and learn knowledge autonomously; their vision and hearing did not connected with knowledge base. Of course, we are not asking for a biological brain here; it is an artificial brain that can be mounted on robots.

Note! the "**Artificial brain**" we mentioned here not serves biology or medical science. It is a new technology concern with artificial life, artificial intelligence, computer science, multi-modal sensing information synthesizing technology and so on.

Computer vision is not usual technology of image processing, it concerns image capture, image understanding, image memory and image retrieve. How to express the knowledge in image by the computer vision is a very important but very difficult research topic. As to computer hearing it has similar problem. Computer hearing not means Fourier decomposition, it implies the verbal speech understanding and learning. There still is the same difficult problem that is how to express the knowledge in natural language? The most of knowledge of human being is from real vision and hearing sensing. The ability obtaining knowledge is the base of human intelligence. In biological world animals including reptiles and mammals in different evolution layer have different intelligence level, although they all have vision and hearing sense. This means that intelligence are evolvable in bottom-up way with the evolution of animal species. This point of view explains the connection between **artificial life** and **artificial intelligence**.

The most basic intelligence of animals is to get knowledge from cognising environment by their sensing organs, to find food, to avoid enemy, to communicate each other, and so on. So, the level of intelligence depends on the ability to understand environment. Here we do not want to discuss more detail about cognitive science concepts that "what is knowledge" and how many ways to express the knowledge from cognitive science, but we try to discuss how to build up an **artificial brain**, or smart computer, that can deal with multi-modal sensing information and transfer them as knowledge.

In fact, to build up an artificial brain started several years ago. In ATR, Japan, there is a hardware named "CAM-brain", that is cellular automata-based artificial brain. The author of this paper, Professor Sugisaka proposed a neural network based artificial brain for robot, and also we proposed an artificial brain with KANSEI technology [1,2]. In MIT the Kismet is another artificial brain, which has emotive response while human face to it [3].

Our research here is aimed to propose

another principle to design a new type Trainable Artificial Brain (TAB).

II. The knowledge discovery based on rough set structure

From above we know the key technology for artificial brain, or smart computer, is to obtain knowledge. In order to avoid too wide discussion, we limit our discussion within to understand environment where the artificial brain stays.

A knowledge we discussed here can be expressed as $K = K(X_1, X_2,...,X_N)$, where Xi represents i-th **knowledge content (attribute) vector**. It could be quantitative data or descriptive qualitative data. For example, if we saw a teacup, its content attribute might be expressed as TC = TC [name(teacup), size(D=8, H=12), shape(round, with handle), colour(inside = white, outside = green), material(Portray), use(drink water and tea),...]. The knowledge is different from a record in database, the later usually has fixed length and number of fields. The other difference between knowledge base and database is that dynamical update of knowledge is in two directions: number of knowledge could be increased vertically and each knowledge could be described in more detail by extension of number of contents horizontally. In addition, knowledge is often not precise, even if the content is quantitative. It usually is fuzzy. We need a new tool for knowledge discovery, learning, and dynamical update of knowledge base. The one of alternatives is the rough set theory.

We now observe that a baby how to get environment knowledge. At early stage he get environment knowledge mainly from his mother. He looks at the object mother shown him and listens to his mother's speech, gradually he establish a connection (map) between the object and its name, a noun, taught by his mother's speech. Gradually, the baby can understand what is eatable food, what is chair, even if their shapes have a little difference. Mother helps him to classify all things she taught and baby learned concepts about food and chair. Also, mother teaches him some verb by her actions. This training gives the baby knowledge and he stores it in his brain as memory. The understanding to environment is really the basic knowledge for the baby; in fact many animals have this kind of intelligence from its mother, similar to human baby's intelligent in his early age. His knowledge accumulated up and became a knowledge base that can be retrieved from his memory. This mode to get knowledge and intelligence is bottom-up way. Why we do not try to build up an artificial brain with vision and hearing sensing and train it as a baby?

The theory of **Rough sets** was founded in 1970's by Polish professor Zdzislaw Pawlak as a result of a long-term program of fundamental research on logical properties of information systems, carried out by him and a group of logicians from Polish Academy of Sciences and the University of Warsaw, Poland. The methodology is concerned with the classificatory analysis of imprecise, uncertain or incomplete information or knowledge expressed in terms of data acquired from experience. The classification formally represents our knowledge about the domain, i.e. the knowledge is understood here as an ability to characterize all classes of the classification. The main specific problems addressed by the theory of rough sets are:
1) Representation of uncertain or imprecise knowledge;
2) Empirical learning and knowledge acquisition from experience;
3) Knowledge analysis;
4) Analysis of conflicts;
5) Evaluation of the quality of the available information with respect to its consistency and the presence or absence of repetitive data patterns;
6) Identification and evaluation of data dependency
7) Approximate pattern classification;
8) Reasoning with uncertainty;
9) Information-preserving data reduction;
and so on.

A number of practical applications of this approach have been developed in recent years in areas such as medicine, drug research, process control and others. The recent publication of a monograph on the theory and a handbook on applications facilitate the development of new applications [4,5,6]. One of the primary

applications of rough sets in AI is for the purpose of knowledge analysis and discovery in data [7].

III. To train the Artificial Brain by using of verbal language

The main function of artificial brain, which we proposed here, is that it is trainable by its sensing ability: computer vision and computer hearing. We will explain the principle to train it in the following sections.

3.1) The essence of leaning by verbal language

In early stage of baby, knowledge is leaned from mother's model actions and speech. Then they learn more themselves. Speech language has played a very important role for learning of knowledge. Language makes knowledge learning be fast and precise, it can express abstract thought. Originally, human's baby has the zero level knowledge that is almost the same as animal's baby when they were borne. However, human mother trains baby by both action and speech language, and human's baby obtained knowledge much fast then animal's baby. So, in the evolutionary history of human being the speech language is the result of evolution of intelligence either, also it is a tool for the evolution of intelligence.

Training by speech will establish a map between the subjective image and voice, at beginning time baby learns some simple words, which corresponds to those objects he saw by vision. It will have a small vocabulary that includes mainly noun, few verb and some adverb those give the space relation and geometrical description. From these the basic knowledge could be expressed and fill out into knowledge base. Of course, the artificial brain has to equipped some sensor and software to receive verbal language training and understand the meaning.

3.2) Related problem

Language processing usually has two aspects, one is language recognition, the other is language understanding. Speech recognition usually possess the following structure:
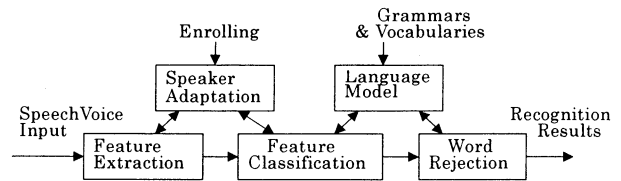
Fig. 1 The structure of speech recognition

It has already got useful progress [8-12] and there has already been some commercial software to do this job. The later is more difficult target and it is a key problem in our implementation. Fortunately, the achievement that has got already is enough to implement a prototype of our trainable artificial brain.

VI. To train the Artificial Brain by using of computer vision

Computer vision usually use CCD camera as image receiver. However, computer vision has different target from image processing. The vision is one of the tools to get knowledge, it is a very complicated process, including receive image on retina of eyes, attention and focus on specified sub-object in viewing field, extraction of object feature, short term memory and long term memory, and so on. Computer vision should simulate these functions as much as possible. The CCD with different resolution in one image has already implemented. The research on retrieve of object feature achieved certain progress. The new standard MPEG7 might be an adequate choice. All these build up important base for constructing artificial brain. What we has to do is that to get knowledge from computer vision [13].

To do this computer vision should have a function of "selecting attention" to its viewing field. While we show some object in front of CCD camera and speak training language, it could focus on it in a very narrow viewing angle, it implies that the sub-image of the object is very clear (with high resolution) but the other parts of image is vague (with low resolution). This function is similar to cut off the sub-image from original image. The sub-image will save as icon and the complementary will save as a high ratio

compressed image. TAB can synthesise both as reconstruction. This is the key problem for obtaining knowledge from computer vision. In this case, we may do the same as a mother teaching a baby. Tell the TAB what is the name, what is the colour, what is the shape, and so on. Then all this knowledge should input knowledge base we designed. That means to establish a map between the sub-image and the speech teaching. Thus, TAB has learned this knowledge. Later if TAB gets more detail attributes about that object, the knowledge might be expanded in knowledge base.
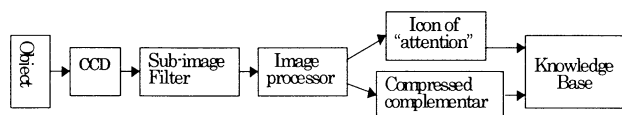
Fig. 2 Get knowledge from vision

Dynamic vision-"impression", "connection" and "memory". Static learning basically is available, but how to learn process knowledge from computer vision? In fact, this is more general situation. Usually human being vision is continuous image on retina of eyes, it cannot be separated as many frames, but the computer vision has got succeeded image sequence with 25 frames in a second, which likes movie. If we want to memory a dynamic process by computer image processing technology, it needs a huge memory space that is impossible almost (even if we use the most advanced image compression technology). So far we do not know the exact process happened in human brain, why it can memory so many seeing and recall them? Here we try to establish a reasonable and practical hypothesis to give a principle to design an artificial brain with dynamic memory function. We try to use a notion: **Impression.**

*Impression: this word is often used to describe some piecewise memory in human brain. Indeed it is a basic way to memory something. Here we say that an "impression" is just a few frames of process image in CCD camera, which are in successive order. Two successive "impressions" have to be interrupted by amount of frames, but there are some relations in the content of images

between them. The relations can be expressed by language explanation. We call this language explanation as "connection" between two impressions. Thus, we could define a "memory" in our TAB as a following sequence:

Impression—connection—impression—conn ection-- ......---impression (or connection).

If a memory is very clear, that means the "impressions" are dense and "connections" between each impression is very strong. On the contrary, if the "impressions" are sparse and "connections" is weak, the memory will be vague.

As to recall of memory to human being, so far we do not know the exact mechanism. Some recall is quite clear but some is very vague and fuzzy. Even if the memory is very clear you still cannot *reconstruct* the whole image exactly. This fact told us the memory in human being is not image storing process and recall is not reconstructing process. Here you can feel again the difference between image processing and computer vision. How to obtain recall from our artificial brain? There are two ways. One is to inverse the memory process above changing storing as retrieving; the other is to combine impressions and connections by "flash" technique as a short video clip.

V. Parallel processing of computer vision and hearing

Analysis of human being vision and hearing sensing tells us that information processing in human brain deals with multi-modal information from sensing organs, and synthesis in certain zone of human brain. For our artificial brain it has to follow this manner.

There are several ways to do parallel processing, one possible way is the following structure.
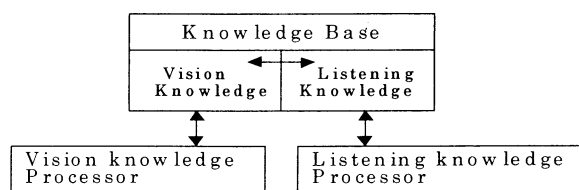
Fig. 3 The parallel structure of TAB

VI. Potential applications

6.1) Intelligent robot

The matter of fact, Intelligent robot needs an artificial brain. Serving this target the computer vision, computer hearing and other computer sensing technology have already got important achievements. The hardware and software are good enough for making a prototype of artificial brain. This means if our artificial brain were created the intelligent robot will be born.

6.2) Intelligent mobile telephone

Using our artificial brain design we may create a new mobile phone, which can be dialled by language instead of pressing button. Thus people only talks mobile phone the number he want to dial.

6.3) Intelligent database

In some job workers have to fill out table with writing or typing, for example the gas counter, they have to count the number of decimal digits. If the workers just read and by using of verbal language to tell TAB in his hands, our smart TAB will records all this verbal information to update database. This TAB will have very wide practical use like barcode reading machine in supermarkets.

6.4) Intelligent appliance

Now some advanced electrical appliances installed fuzzy control, the makers advertise that it is intelligent control. In fact we could make it even more clever than fuzzy controller. We may put our artificial brain into the remote controller and people just say some words to it instead of pressing button, then starting appliances.

6.5) Intelligent toy

If a doll can have conversation with girls, definitely it will be more attractive then SONY's Aibo.

6.6) Intelligent Music box

Music box only play a short section of famous melody, they cannot change songs and cannot compose some new melody. If a music box can ask you to hum a tune or tell it that what kind of melody do you like, then after a few seconds, it can compose a fresh music, this music box will be preferred by a lot of peoples.

VII. Conclusion

Above we have showed the principle of design of **Trainable Artificial Brain (TAB)**, it is easy to see that this is a bottom-up intelligent similar to human being's one. By this way we could find an approach to evolve the artificial brain. "Trainable" implies that it can learn itself, so the level of intelligent would be increased during a period of training. Also, our artificial brain showed here will be evolvable. Each classification in knowledge base implies knowledge to become more general. Based on rough set theory the knowledge base might be constructed and implemented by software. Using verbal language training we need solve a problem that to fill out a knowledge base by verbal language. This is one of the key technologies. Another one is to cut off a part of image taken by computer vision and to extract its attribute automatically. The third one is to making a rough set based knowledge base to deal with computer vision and computer hearing parallel. We think that it is the time to build up a prototype of Trainable Artificial Brain (TAB). However, so far, we do not know how to make our TAB to generate abstract mode of thinking.

REFERENCES
[1] Zhang, Y.G. and Sugisaka, M., "Humanized Robot (Hubot) with K- Artificial Brain", **The Proceedings of International Symposium on Artificial Life and Robotics (AROB'01)**, Jan. 2001, pp.288-291, Tokyo, Japan,

[2] Zhang, Y.G., Nagashima, T, and Shimohara, K. "Humanization of computer with KANSEI technology",
The Journal of Three Dimensional Image, Vol.14, No.4, pp.76-80, Dec. 2000

[3] Kismet video in the home page of AI Lab of MIT.   See
http://www.ai.mit.edu/projects/socialble/video.himl

[4] Slowinski, R. and Stefanowski J. (eds.) **"Foundations of Computing and Decision Sciences"**, Vol. 18. No. 3-4, Fall 1993.

[5] Pawlak, Z., Rough Sets: **"Theoretical Aspects of Reasoning About Data"**. Kluwer Academic Publishers, Dordrecht, 1991.

[6] Slowinski, R. (ed.), **"Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory"**, Kluwer Academic Publishers, Dordrecht, 1992.

[7] Ziarko, W. "The Discovery, Analysis and Representation of Data Dependencies in Databases In sky", Shapiro, G. and Frawley, W.J. (eds.), **Knowledge Discovery in Databases**, AAAI Press/MIT Press, 1991, pp. 177-195.

[8] Fallside, F. and Woods, W.A. editors, **"Computer Speech processing"**, Englewood Cliffs: Prentice-Hall, New York, 1985

[9] Lee, K.F., **"Automatic speech recognition: the development of the SPHINX system"**. Kluwer Academic Publishers, Boston, 1989

[10] Ince, A.N., editor. **"Digital speech processing: speech coding, synthesis, and recognition"**, Kluwer Academic Publishers, Boston, 1992

[11] Rabiner, R. and Juang, B.H., **"Fundamentals of Speech recognition"**, Prentice-Hall, New Jersey, 1993

[12] Picone, J., Ebel, W.J. and Deshmukh,N., "Automated speech understanding: The next generation", **Digital Signal processing Technology**, CR57:101-114, 1995

[13] The journal of Computer vision and understanding, Academic Press, San Diago.

Application of Uncertain Variables for a Class of Intelligent Knowledge-Based Assembly Systems*

Z. Bubnicki

Institute of Control and Systems Engineering
Wroclaw University of Technology
Wyb. Wyspianskiego 27, 50-370 Wroclaw, POLAND
E-mail: bubnicki@ists.pwr.wroc.pl

Keywords: assembly systems, uncertain variables, inteligent systems, uncertain systems

Abstract

A decision problem consisting in the determination of a sequence of assembly operations for a class of uncertain assembly systems described by a relational knowledge representation is considered. Unknown parameters in the knowledge representation are assumed to be values of *uncertain variables* described by certainty distributions given by an expert. The algorithms of the decision making (of the control) are given and illustrated by a simple example. The structure of the "intelligent" closed-loop control system is presented.

1. Introduction

For the analysis and decision making in a class of uncertain knowledge-based systems with unknown parameters an approach based on so called uncertain variables has been elaborated [1, 2, 3]. The purpose of this paper is to show how this approach may be applied to a class of assembly systems.

In many practical situations there exist uncertainties in the description of an assembly process: the sequence of assembly operations is not *a priori* determined and the relationships between the successive operations, states and features describing the assembly plant are nondeterministic [see e.g. 4, 5, 6, 7]. This is a frequent situation in small batch production processes with changes in the relationships describing the assembly plant in different cycles. In such cases it is reasonable to apply artificial intelligence tools and methods of decision making in uncertain systems to the planning and control of the assembly process.

The paper deals with a class of knowledge-based assembly systems described by a relational knowledge representation consisting of relations between the operations, states and features, i.e. variables characterizing the current effect of the assembly process (e.g. dimensions or sizes evaluating the precision, accuracy or tolerance in the placement and fastening of

elements). The problem of choosing assembly operations from the given sets of operations on each stage is considered as a specific multistage decision process for a relational plant with unknown parameters. The unknown parameters in the description of the assembly process are assumed to be values of uncertain variables described by a certainty distribution given by an expert. The certainty distribution expresses the expert's knowledge concerning the different approximate values of the unknown parameter.

2. Knowledge representation and decision problem

Let us consider an assembly process as a sequence of assembly operations $O_n \in \{O_{n1}, O_{n2}, ..., O_{nl_n}\}$ executed on the successive stages n. On each stage the assembly plant is characterized by a state $s_n \in \{S_{n1}, S_{n2}, ..., S_{nm_n}\}$. The state s_{n+1} depends on the state s_n and the operation O_n (Fig.1). To formulate the description of the assembly process let us introduce the following notation:
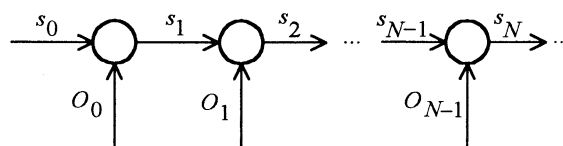
Fig.1.

i_n – index of the operation on n-th stage,

$i_n \in \{1, 2, ..., l_n\} \triangleq L_n$ – set of the operations, $n \in \overline{1,N}$,

j_n – index of the state, e.g. if $j_n = 2$ then $s_n = S_{n2}$,

$j_n \in \{1, 2, ..., m_n\} \triangleq M_n$ – set of the states,

$y_n \in Y_n$ – real number vector of features describing the

assembly plant on n-th stage. The evaluation of the result of n-th operation and the quality of the process concern the components of y_n, e.g. $a \leq y_n^{(v)} \leq b$ may denote the requirement concerning the size $y_n^{(v)}$ (a component of y_n).

The knowledge representation consists of relations between the variables i_n, j_n, j_{n+1} and y_n, and may be divided into two parts:

I. A relation between i_n, j_n and j_{n+1}

$$R_{\text{I}n}(i_n, j_n, j_{n+1}) \subset L_n \times M_n \times M_{n+1} .$$

II. A relation between j_{n+1} and y_{n+1}

$$R_{\text{II}n}(j_{n+1}, y_{n+1}) \subset M_{n+1} \times Y_{n+1} .$$

The relations R_{I} and R_{II} may be presented in the form of families of sets:

I. $$D_{j,n+1}(i_n, j_n) \subset M_{n+1} \qquad (1)$$

for all pairs $(i_n, j_n) \in L_n \times M_n$,

II. $$D_{y,n+1}(j_{n+1}) \subset Y_{n+1} \qquad (2)$$

for all $j_{n+1} \in M_{n+1}$.

Consequently, the relational knowledge representation consists of $l_n \cdot m_n$ sets (1) and m_{n+1} sets (2). The decision making (control) in the assembly process should be based on the knowledge representation (1), (2) given by an expert. On each stage the decision consists in the proper choosing of the assembly operation O_n (i.e. the index i_n) from the given set of operations (i.e. from the set L_n), satisfying the requirement concerning the features y_{n+1} presented in the form $y_{n+1} \in D_{y,n+1}$ where the set $D_{y,n+1} \subset Y_{n+1}$ is given by a user. For making the decision a knowledge of the state j_n in the form $j_n \in D_{j,n} \subset M_n$ is used.

The decision problem on n-th stage may be formulated as follows: For the given knowledge representation, the set $D_{j,n}$ (the knowledge of the state j_n, determined on the stage $n-1$) and the set $D_{y,n+1}$ (a user's requirement), one should find the set of all operations i_n satisfying the requirement, i.e. the largest set $D_{i,n} \subset L_n$ such that the implication

$$i_n \in D_{i,n} \rightarrow y_{n+1} \in D_{y,n+1}$$

is satisfied. Consequently, the assembly operation should be chosen from the set $D_{i,n}$. The decision problem may be decomposed into two parts.

A. For the given sets (2) and $D_{y,n+1}$ find the largest set $D_{j,n+1} \subset M_{n+1}$ such that the implication $j_{n+1} \in D_{j,n+1} \rightarrow y_{n+1} \in D_{y,n+1}$ is satisfied.

B. For the given sets (1) and the sets $D_{j,n}$, $D_{j,n+1}$ find the largest set $D_{i,n}$ such that the implication

$i_n \in D_{i,n} \rightarrow j_{n+1} \in D_{j,n+1}$ is satisfied for each $j_n \in D_{j,n}$.

For the problem solving a general solution of the decision problem based on relational knowledge representation may be used. It is easy to note that:

A.
$$D_{j,n+1} = \{ j_{n+1} \in M_{n+1} : D_{y,n+1}(j_{n+1}) \subseteq D_{y,n+1} \},$$
$$\qquad (3)$$

B.
$$D_{i,n} = \{ i_n \in L_n : \bigwedge_{j_n \in D_{j,n}} D_{j,n+1}(i_n, j_n) \subseteq D_{j,n+1} \}. \qquad (4)$$

Using (3) and (4) for $n = 0, 1, ..., N-1$, we can determine an *assembly plan* in the form of a sequence $D_{i,1}, D_{i,2}, ..., D_{i,N-1}$. The set of initial states $D_{i,0}$ must be known and on n-th stage we use $D_{i,n}$ determined on the former stage. The decision process is then performed in an open-loop control system: on each stage the assembly operation i_n may be chosen randomly from the set $D_{i,n}$ according to the assembly plan.

3. Uncertain variables

The uncertain variable $\bar{x}$ is defined by the set of values X, the function $h_x(x) = v(\bar{x} \cong x)$ (i.e. the certainty index that $\bar{x}$ is *approximately* equal to x, given by an expert) and the following definitions:

$$v(\bar{x} \; \widetilde{\in} \; D_x) = \max_{x \in D_x} h_x(x),$$

$$v(\bar{x} \; \widetilde{\notin} \; D_x) = 1 - v(\bar{x} \; \widetilde{\in} \; D_x),$$

$$v(\bar{x} \; \widetilde{\in} \; D_1 \vee \bar{x} \; \widetilde{\in} \; D_2) = \max\{v(\bar{x} \; \widetilde{\in} \; D_1), v(\bar{x} \; \widetilde{\in} \; D_2)\},$$

$$v(\bar{x} \; \widetilde{\in} \; D_1 \wedge \bar{x} \; \widetilde{\in} \; D_2) = \min\{v(\bar{x} \; \widetilde{\in} \; D_1), v(\bar{x} \; \widetilde{\in} \; D_2)\}$$

where $D_x, D_1, D_2 \subset X$ and $\bar{x} \; \widetilde{\in} \; D_x$ denotes a soft property: "$\bar{x}$ approximately belongs to D_x". The function $h_x(x)$ is called a *certainty distribution* [1, 2]. It may be useful to apply another version of the uncertain variable called C-uncertain variable, for which the certainty index of the property "$\bar{x} \; \widetilde{\in} \; D_x$" is defined as follows

$$v_c(\bar{x} \; \widetilde{\in} \; D_x) = \frac{1}{2}[v(\bar{x} \; \widetilde{\in} \; D_x) + 1 + v(\bar{x} \; \widetilde{\in} \; \overline{D}_x)]$$

where $\overline{D}_x = X - D_x$.

Let us consider a plant with the input vector $u \in U$ and the output vector $y \in Y$, described by a relation $R(u, y; x) \subset U \times Y$ where x is an unknown vector parameter which is assumed to be a value of a random variable $\bar{x}$. For the requirement $y \in D_y$ given by a user we can formulate and solve the following decision problem: find the decision u^* maximizing the certainty index

$$v[u \; \widetilde{\in} \; D_u(\bar{x})] = v[\bar{x} \; \widetilde{\in} \; D_x(u)]$$

where $D_u(\bar{x}) \subset U$ is the largest set such that $u \in D_u(x) \to y \in D_y$, i.e.

$$D_u(x) = \{u \in U : D_y(u; x) \subset D_y\},$$
$$D_y(u; x) = \{y \in Y : (u, y) \in R(u, y; x)\},$$
$$D_x(u) = \{x \in X : u \in D_u(x)\}.$$

4. Assembly process with uncertain parameters

Let us consider the assembly process described by the knowledge representation with constant unknown parameters $x \in X$ and $c \in C$ in the first and the second part, respectively. Now the sets $D_{j,n+1}(i_n, j_n; x)$ in (1) depend on x and the sets $D_{y,n+1}(j_{n+1}; c)$ in (2) depend on c. Consequently, the set of the decisions $D_{i,n}(x, c)$ in (4) depends on (x, c). The values (x, c) are assumed to be values of uncertain variables $(\bar{x}, \bar{c})$ with the certainty distribution

$$h(x, c) = v[(\bar{x} \cong x) \wedge (\bar{c} \cong c)]$$

given by an expert. The decision problem consists in the determination of the optimal decisions i_n^* maximizing the certainty index that i_n approximately belongs to $D_{i,n}(\bar{x}, \bar{c})$ (precisely speaking: that i_n belongs to $D_{i,n}(x, c)$ with the approximate values x and c), i.e.

$$i_n^* = \arg \max_{i_n \in I_n} v(i_n)$$

where

$$v(i_n) = v[i_n \; \widetilde{\in} \; D_{i,n}(\bar{x}, \bar{c})] \,.$$

It is easy to note that

$$v[i_n \; \widetilde{\in} \; D_{i,n}(\bar{x}, \bar{c})] = v[(\bar{x}, \bar{c}) \; \widetilde{\in} \; D_{xc}(i_n)]$$

where

$$D_{xc}(i_n) = \{(x, c) : i_n \in D_{i,n}(x, c)\} \,.$$

Then

$$v(i_n) = \max_{(x,c) \in D_{xc}(i_n)} h(x, c) \,.$$

As a result we obtain an *assembly plan* in the form of a sequence of the assembly operations i_n^*. If $\bar{x}$ is considered as a C-uncertain variable then

$$v_c(i_n) = \frac{1}{2} \{v[i_n \; \widetilde{\in} \; D_{i,n}(\bar{x}, \bar{c})] + 1 - v[i_n \; \widetilde{\in} \; \overline{D}_{i,n}(\bar{x}, \bar{c})]\}$$

and $\overline{D}_{i,n} = I_n - D_{i,n}$.

If it is possible to recognize the current states j_n (i.e. to determine the set $\overline{D}_{j,n}$ such that $j_n \in \overline{D}_{j,n}$) then the decision process may be performed in a closed loop control system (Fig.2) and $\overline{D}_{j,n}$ may be used in (4) instead of $D_{j,n}$.

In particular, the relation R_{II} describing the second part of the knowledge representation, may be a function $j_{n+1} = F_n(j_n, i_n)$, i.e. a matrix with the members

j_{n+1} in the j_n-th line and i_n-th column. Then we can determine $D_{j,n+1}(c)$ (see part A in the procedure described in Sec. 2) and

$$j_{n+1}^* = \arg \max_{j_{n+1}} v[j_{n+1} \; \widetilde{\in} \; D_{j,n+1}(\bar{c})] \,. \tag{5}$$
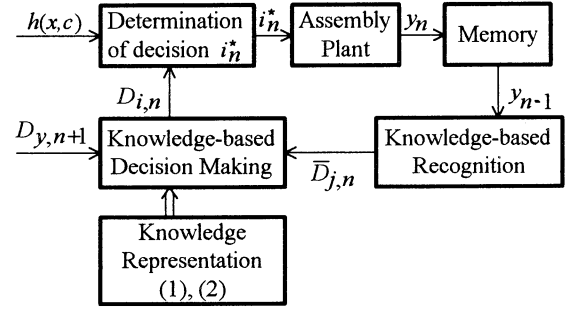
Fig.2.

Consequently, the set of the decisions $D_{i,n}$ for the given j_{n+1}^* and j_n^* obtained on the former stage ($j_0^* = j_0$ is given) may be obtained from the matrix F_n:

$$D_{i,n} = \{i_n \in L_n : F_n(j_n^*, i_n) = j_{n+1}^*\} \,. \tag{6}$$

The procedure of the determination of the assembly operation i_n^* is then the following:

1. Determination of the set $D_{j,n+1}(c)$ according to (3)

$$D_{j,n+1}(c) = \{j_{n+1} : D_{y,n+1}(j_{n+1}; c) \subseteq D_{y,n+1}\} \,. \tag{7}$$

2. Determination of the set

$$D_{c,n+1}(j_{n+1}) = \{c \in C : j_{n+1} \in D_{j,n+1}(c)\} \,. \tag{8}$$

3. Choosing the best index j_{n+1}^* from the set M_{n+1} according to (5)

$$j_{n+1}^* = \arg \max_{j_n} \max_{c \in D_{c,n+1}(j_{n+1})} h_c(c) \,. \tag{9}$$

4. Determination of $D_{i,n}$ according to (6).

5. Random choosing of i_n from $D_{i,n}$.

The decisions i_n are executed in the open-loop control system (Fig. 3) where G denotes the generator of random numbers for the random choosing of the assembly operation i_n from $D_{i,n}$.
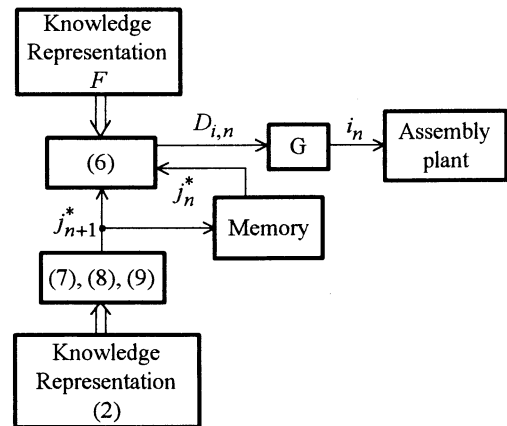
Fig.3.

5. Example

Let the sets $D_{y,n}(j_n; c)$ be described by inequalities

$$ca(j_n) \le y_n^{\mathrm{T}} y_n \le 2ca(j_n), \quad c > 0, \quad a(j_n) > 0, \quad (10)$$

given by an expert. It is then known that the value $y_n^{\mathrm{T}} y_n$ (where y_n is a vector of the features characterizing the assembly plant) satisfies the inequality (10) and the bounds are the coefficients a depending on the state, i.e. for the different assembly operations i_n and consequently – the different states j_n we have the different bounds for the value $y_n^{\mathrm{T}} y_n$ which denotes a quality index. The requirement concerning the quality (i.e. the set $D_{y,n}$) is the following

$$\beta_n \le y_n^{\mathrm{T}} y_n \le \alpha_n, \quad \beta_n > 0, \quad \alpha_n \ge 2\beta_n.$$

Then, according to (3)

$$D_{j,n}(c) = \{ j_n \in M_n : \beta_n \le ca(j_n) \le \frac{\alpha_n}{2} \}.$$

Consequently

$$j_n^* = \arg\max_{j_n} \max_{c \in D_{c,n}(j_n)} h_c(c) \overset{\Delta}{=} \arg\max_{j_n} v(j_n) \quad (11)$$

where $h_c(c)$ is the certainty distribution for $\bar{c}$ and

$$D_{c,n}(j_n) = \{ c \in C : j_n \in D_{j,n}(c) \}.$$

Having j_n^* and j_{n-1}^* one may obtain the set of possible assembly operations (in particular, one operation i_n^*) from the matrix F. For example, assume that $h_c(c)$ has the parabolic form: $h_c(c) = -(c - d)^2 + 1$ for $d - 1 \le c \le d + 1$ and $h_c(c) = 0$ otherwise. Assume also that $j_n \in \{1, 2, 3\}$, $a(1) = 1$, $a(2) = 0.6$, $a(3) = 0.9$ and $\alpha_n = 6.4$, $\beta_n = 2.8$. Then for $j = 1$

$$\frac{\beta_n}{a(1)} \overset{\Delta}{=} g_1 = 2.8, \quad \frac{\alpha_n}{2a(1)} \overset{\Delta}{=} g_2 = 3.2$$

and according to (11) $v(1) = h_c(3.2) = 0.36$. For $j = 2$ we obtain $g_1 \approx 4.7$, $g_2 \approx 5.3$, $v(2) = h_c(4.7) = 0.51$. For $j = 3$ we obtain $g_1 \approx 3.1$, $g_2 \approx 3.5$, $v(3) = h_c(3.5) = 0.75$. Consequently $j_n^* = 3$.

6. Conclusions and related problems

The uncertain variables are proved to be a convenient tool for the design of a class of uncertain assembly processes described by the relational knowledge representation. The assembly process may be considered as a specific decision plant with unknown parameters described by the certainty distributions given by an expert. The following related problems may be indicated: 1) The approach based on the uncertain variables may be combined with the application of the learning process consisting in *step by step* knowledge updating [8, 9]. 2) The presented approach may be applied to the control of a complex assembly process, similar to the control of a complex of parallel manufacturing operations [3, 9]. 3) For the complex assembly process described by a distributed knowledge representation, a decomposition of the decision problem may be applied [10, 11].

References

[1] Bubnicki Z (2001), Uncertain variables and their applications for a class of uncertain systems. *International Journal of Systems Science*, **32**, no. 6.

[2] Bubnicki Z (2001), Uncertain variables and their application to decision making. *IEEE Trans. on SMC, Part A: Systems and Humans*, **31**, no. 6 (in press).

[3] Bubnicki Z (2001), Application of uncertain variables to control for a class of production operations with parametric uncertainties. *Proc. of IFAC Workshop on Manufacturing, Modelling, Management and Control*, Prague, pp. 29-34.

[4] Simmons JEL, Ritchie JM, Dewar RG and Carpenter ID (1999), The elicitation of expert knowledge for assembly and other manual tasks using immersive virtual reality. *Proc. of the II World Manufacturing Congress*, Durham, UK, pp. 204-212.

[5] Neshkov T, Yordanova S and Videnov L (2000), A fuzzy model for automated precision robot assembly of parts. *Proc. of IFAC Symposium on Manufacturing, Modeling, Management and Control*, Patras, Rio, Greece, 12-14 July, pp. 493-497.

[6] Bozma HI and Koditschek DE (2001), Assembly as a noncooperative game of its pieces: analysis of 1D sphere assemblies. *Robotica*, **19**, 93-108.

[7] Huang YF and Lee CSG (1991), A framework of knowledge-based assembly planning. *Proc. of IEEE International Conference on Robotics and Automation*, Sacramento, California, USA.

[8] Bubnicki Z (2000), Learning processes in a class of knowledge-based systems. *Kybernetes*, **29**, no. 7/8, 1016-1028.

[9] Bubnicki Z (2002), Learning process in a class of computer integrated manufacturing systems with parametric uncertainties. *Int. Journal of Intelligent Manufacturing. Special Issue on CIM Workflow*, (in press).

[10] Bubnicki Z (2000), Knowledge validation and updating in a class of uncertain distributed knowledge systems. *Proc. of 16th IFIP World Computer Congress. Intelligent Information Processing*, (Beijing: Publishing House of Electronics Industry), pp. 516-523.

[11] Bubnicki Z (2001), Application of uncertain variables and logics to complex intelligent systems. *Proc. of the 6th Int. Symposium on Artificial Life and Robotics*, Tokyo, Vol.1, pp. 220-223.

Grazing Bifurcation and Mode-Locking in Reconstructing Chaotic Dynamics with Leaky Integrate-and-Fire Model

Y. Ono[1]*, H. Suzuki[2], J. Murakami[3], T. Shimozawa[3], and K. Aihara[1,4]

[1] Department of Complexity Science and Engineering, The University of Tokyo, Hongo, Tokyo 113-8656, Japan

[2] Graduate School of Information Science and Technology, The University of Tokyo, Hongo, Tokyo 113-8656, Japan

[3] Research Institute for Electronic Science, Hokkaido University, Kita-ku, Sapporo 060-0812, Japan

[4] CREST, Japan Science and Technology Corporation (JST), Kawaguchi, Saitama 332-0012, Japan

Abstract

We examine the firing patterns of a chaotically forced leaky integrate-and-fire (LIF) model and the validity of reconstructing the input chaotic dynamics from the spike sequence. The input to the model is generated from the Rössler system. The bifurcation parameter of the system is set at various values so that the input becomes either periodic or chaotic. The *generalized rotation number* is proposed as an extension of the conventional *rotation number* to quantify the mode-locked behavior of our model. Similar behaviors as periodically forced LIF are observed: (i) emergence of qualitatively distinct behaviors across a border of the parameter space through *grazing* bifurcation at the border and (ii) mode-locked regions when the leak of the LIF is greater than the border. Our results suggest that *grazing* bifurcation is related to the reconstruction of chaotic dynamics with the LIF.

Keywords: leaky integrate-and-fire, chaos, interspike interval, reconstruction, grazing, mode-locking

1 Introduction

Delay embedding is a widely used technique for reconstructing a dynamical system from observed time series data [1]. This has recently been extended to point processes, i.e., a series of event timing [2][3]. The application to neural spike sequences, in particular, has been attracting much attention as a new approach to explore the information processing in neural systems [2][3]. Predictability of a reconstructed dynamics and topological features of the reconstructed attractor have been examined in both numerical simulations and experiments [4][5][6].

Previous research [5][6] used leaky integrate-and-fire (LIF) model to address this issue. LIF is the

widely used neuron model. In numerical simulations, when the leak of the LIF goes beyond a critical value, topological features of the reconstructed attractor become entirely different and predictability of the reconstructed dynamics change sharply [5][6]. It has been suggested that the reconstructed attractor is closely related to the set of state variables at the time a spike occurring and this phenomena may be due to a qualitative change of the LIF forced by chaotic input [6]. Therefore, investigating bifurcation structure of chaotically forced LIF is an important step in understanding what dynamical system is reconstructed using delay embedding with the LIF.

Bifurcation structure of chaotically forced LIF, however, has not been investigated in details. On the other hand, a great deal of research has been carried out in *periodically* forced LIF, where the results of circle maps can be applied [7][8][9][10][11]. Here, we have considered a chaotic input to be an extension of a periodic input and have examined whether some results for a periodic input may hold for a chaotic input. For that purpose, the Rössler system is used to generate periodic or chaotic time series data according to the bifurcation parameter of the system.

2 Theory

The leaky integrate-and-fire (LIF) model is represented as

$$\frac{du}{dt} = -ku + S(t) \quad , \qquad (1)$$

$$u(t^+) = 0 \quad \text{if} \quad u(t) = \Theta, \qquad (2)$$

where u, k, and $S(t)$ denotes the internal state, the leak, and the input respectively. When u reaches the threshold Θ, the model *fires*: a *spike* is generated and u is immediately reset to 0. T_i denotes the time at which the ith spike occurs. The interspike interval (ISI) is

*e-mail: ono@sat.t.u-tokyo.ac.jp

defined as $t_i = T_i - T_{i-1}$. The input $S(t)$ is assumed to be continuous and positive for all values of t.

In this section, we focus on the case that the input $S(t)$ has a period of Ω. Keener et. al. [7] were the first to have investigated the model when forced by a sinusoidal input. The firing map, the map of the times between successive spikes, can be defined as $F : T_i \mapsto T_{i+1}$. Because it satisfies $F(T+\Omega) = F(T)+\Omega$, it can be viewed as a circle map

$$\phi_{i+1} = F(\phi_i), \qquad (3)$$

where $\phi_i \equiv T_i \bmod \Omega$ is the phase of the ith spike within $[0, \Omega)$. Recently, the following classification of the parameter space has been shown to hold for *any* periodic input [11] rather than sinusoidal input [7]:

region 1 , $k\Theta < \min_t S(t)$, the circle map F is continuous and increasing;

region 2 , $\min_t S(t) < k\Theta$ and $\Theta < \max_t\{\int_{-\infty}^{0} e^{ks}S(t + s)ds\}$, the circle map F is discontinuous and increasing;

region 3 , $\max_t\{\int_{-\infty}^{0} e^{ks}S(t + s)ds\} < \Theta$, firing terminates. the circle map F is not defined.

The *rotation number* is defined as

$$\rho = \frac{\langle ISI \rangle}{\Omega}, \qquad (4)$$

where $\langle ISI \rangle = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} t_i$ is the mean of ISIs. This is well-defined and is independent of the initial conditions for a non-decreasing circlem map [12]. ρ is rational *if* and *only if* firing is periodic.

We consider the increase of the rotation number ρ with increasing the leak k. Note that the effect of increasing the leak is equivalent to that of increasing both the intensity and the frequency of the input (see the Appendix). Increasing the leak k induces a family of monotonically increasing circle maps $\{F_k\}$. Because the ISIs increase with increasing the leak, namely $F_{k_0}(x) < F_{k_1}(x)$ for $k_0 < k_1$ and all x, it can be shown from the definition [eq.(4)] that ρ is a non-decreasing function of k. Moreover, as the leak k increases, ρ increases continuously in *region 1* and increases like a *Devil's Staircase* in *region 2* [7][9][13].

Two kinds of bifurcation occur [10] in periodically forced LIF: tangent bifurcation and *grazing* bifurcation. *Grazing* bifurcation has been studied as a form of bifurcation in non-smooth dynamical system [14]. In our model, it occurs when the trajectory of the internal state u is tangential to the threshold level Θ; there exists T_i such that $-k\Theta + S(T_i) = 0$. It should be noted that *grazing* bifurcation occurs when the leak k goes beyond the border between *region 1* and *region*

2,

$$k = \min_t \frac{S(t)}{\Theta}, \qquad (5)$$

assuming that firing occurs at every phase of the input on the border.

The stability of mode-locked solutions may be quantified by the Lyapunov exponent. For integate-and-fire systems, Coombes [9] has shown that this exponent is written as

$$\lambda = -k + \lim_{n \to \infty} \frac{1}{T_n - T_0} \sum_{j=1}^{n} \ln \left| \frac{S(T_i)}{-k\Theta + S(T_j)} \right|. \qquad (6)$$

This is undefined for solutions that cause *grazing*. $\lambda < 0$ when the solution is stable and $\lambda = 0$ at tangent bifurcation points.

3   Method

The Rössler system, $(\dot{x}, \dot{y}, \dot{z}) = (-y+z, x+ay, bx - cz + xz)$, was used to generate the input to the LIF: $S(t) = x/20 + 2$. The threshould Θ was set to 1. The system has a chaotic attractor with a funnel-shape as shown in Fig.1a. A bifurcation diagram of the Rössler system is shown in Fig.1b. Fig.1b shows that we can choose the values of the parameter c with $(a, b) = (0.36, 0.4)$ so that the input $S(t)$ becomes either periodic or chaotic.

a   b
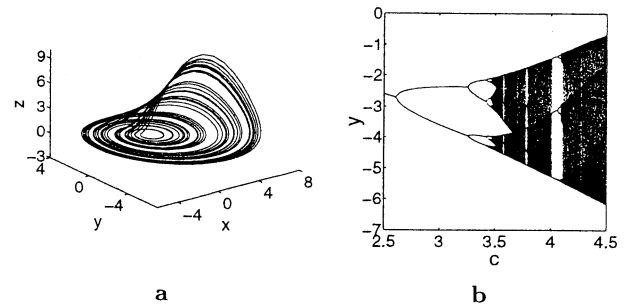
Figure 1:   The Rössler system.   a.The attractor at $(a, b, c) = (0.36, 0.4, 4.5)$. b.The bifurcation diagram; the value of y on the Poincaré section $x = 0 \cap y < 0$ with $(a, b) = (0.36, 0.4)$

The asymptotic firing pattern of the LIF was calculated in numerical simulations. The parameter c was set at various values such that the input was either periodic or chaotic. For each input, the leak k was also set at various values. Delay embedding was applied to

the interspike intervals of the output spike sequence. The reconstruction is represented by the map

$$x_i \mapsto (t_i, t_{i-1}, t_{i-2}, ..., t_{i-m+1}), \qquad (7)$$

where x_i is the state variable of the Rössler system at the time T_i, m is called an embedding dimension. When $k = 0$, the reconstruction map [eq.7] is one-to-one under certain genericity conditions [3]. We set the embedding dimension $m = 2$ for simplicity.

We propose a *generalized rotation number* to quantify mode-locked behavior of the spike pattern. This is defined as

$$\omega = \frac{\langle ISI \rangle}{\tilde{\Omega}}, \qquad (8)$$

where $\tilde{\Omega}$ is the mean period over which each trajectory of the Rössler system rotates around the origin in the xy plane. When the input is periodic, ω is rational *if* and *only if* firing is periodic because it is an integer multiple of the *rotation number*. The inverse of ω denotes the mean number of times the model fires during a period $\tilde{\Omega}$.

4   Results and Discussion

Figure 2 shows the variable (x, y) of the Rössler system at the firing times. The input is periodic when $c = 2.5, 3.4$ and chaotic when $c = 4.0, 4.5$. As can be seen, qualitatively distinct behaviors are oberved across the border [eq.(5)] between *region 1* and *region 2*. The border was caluclated numerically as $k = 1.891, 1.826, 1.785, 1.753$ at $c = 2.5, 3.4, 4.0, 4.5$ respectively. In *region 1* where the leak k is less than the border, spikes seem to occur at every point on the state-space attractor. Although for some inputs, we also observed some intervals of the leak parameter in which this is not the case, these lengths were small(*not shown*). This suggests that *grazing* bifurcation occur at the border for the chaotic inputs as well as for the periodic inputs. On the other hand, in *region 2* where the leak k is greater than the border, spikes seem to occur in a very limited region of the state space attractor. Mode-locked behaviors are observed for the periodic inputs ($c = 2.5, 3.4$). This is consistent with the theory of periodically forced LIF [7][11][9][13]. Notice that even for the chaotic inputs ($c = 4.0, 4.5$), the firing patterns seem like mode-locking. This issue is discussed later using the notion of the *generalized rotation number* and the Lyapunov exponent.

The reconstructed attractors using delay embedding from the interspike intervals are shown in Fig.3.

Qualitatively distinct behaviors are again observed across the border [eq.(5)] in the same way as the state space. It seems that the topological features are preserved through the reconstruction in *region 1*. At $(c, k) = (4.5, 2.05)$ in *region 2*, however, it seems that the topological features are broken up and the reconstruction map [eq.7] has become discontinuous. The embedding theorem of integrate-and-fire dynamics (Nishikawa T, Sauer T,"Embedding theorem for integrate-and-fire dynamics," *preprint*) holds in *region 1*, where the internal state u is always increasing. Our result is consistent with the embedding theorem. Notice that even in *region 2*, the reconstruction map seems to be continuous and to preserve the topological features in the state-space when the leak k is near the border. This *partial* reconstruction was actually observed in the experiment with cricket cercal neurons [6].

Figure 4 shows the interspike intervals (ISI's), the *generalized rotation number*, and the Lyapunov exponent. For a periodic input ($c = 2.5$), ISI seems to be periodic in *region 2* ($k > 1.891$) and to be quasiperiodic in *region 1* ($k < 1.891$) except some small intervals, around $k = 1.87$ for instance [Fig.4a]. For a chaotic input ($c = 4.5$), on the other hand, ISI seems not to be periodic in both *region 1*($k < 1.785$) and *region 2* ($k > 1.785$) [Fig.4d]. For the periodic input, the *generalized rotation number*, equivalent to the *rotatoin number* in this case, was constant over some intervals of k in *region 2* [Fig.4b]. At the right-hand boundary of each mode-locked region, the Lyapunov exponent λ has a sharp peak [Fig.4c]. It seems that λ is close to 0 to the right-hand boundary of each region while λ is certainly negative at the left-hand one. This suggests that tangent bifurcation occur at the right-hand boundary and *grazing* occurs at the left-hand one. These results agree with the previous result for a sinusoidal inputs [10]. Notice that even for the chaotic input, the *generalized rotation number* was remained constant over some intervals of k [Fig.4e]. At the right-hand boundary of a mode-locked region, the Lyapunov exponent λ has a sharp peak as for the periodic input although those peaks are less clear [Fig.4f]. Similar phenomena as tangent bifurcation may occur at the boundary of a mode-locked region for the chaotic inputs.
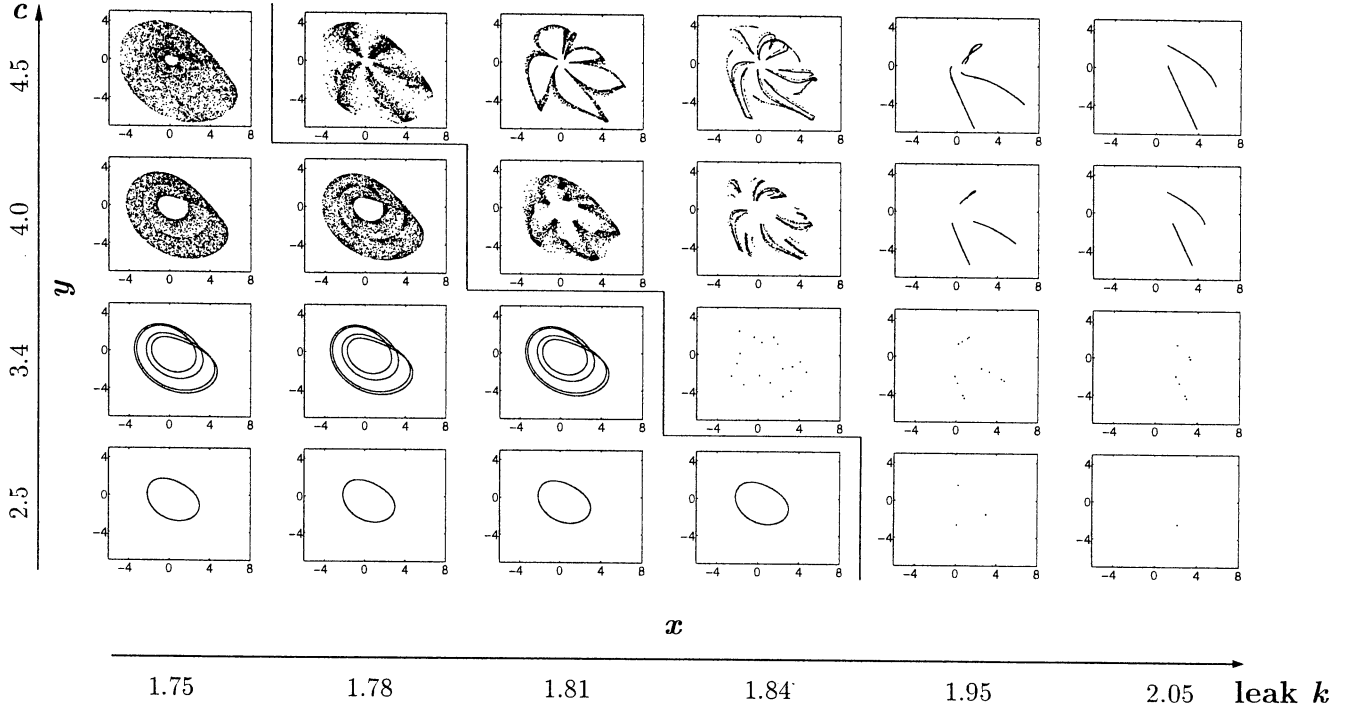
Figure 2: The state variable (x, y) of the Rössler system at firing times. c is the bifurcation parameter of the Rössler system and k is the leak parameter of the leaky integrate-and-fire model. The respective figures are at $c = 4.5, 4.0, 3.4, 2.5$ (column) and $k = 1.75, 1.78, 1.81, 1.84, 1.95, 2.05$ (row). The solid line denotes the border [eq.(5)] between *region 1* and *region 2*. 4001 spikes after an initial transient are used in each plot.
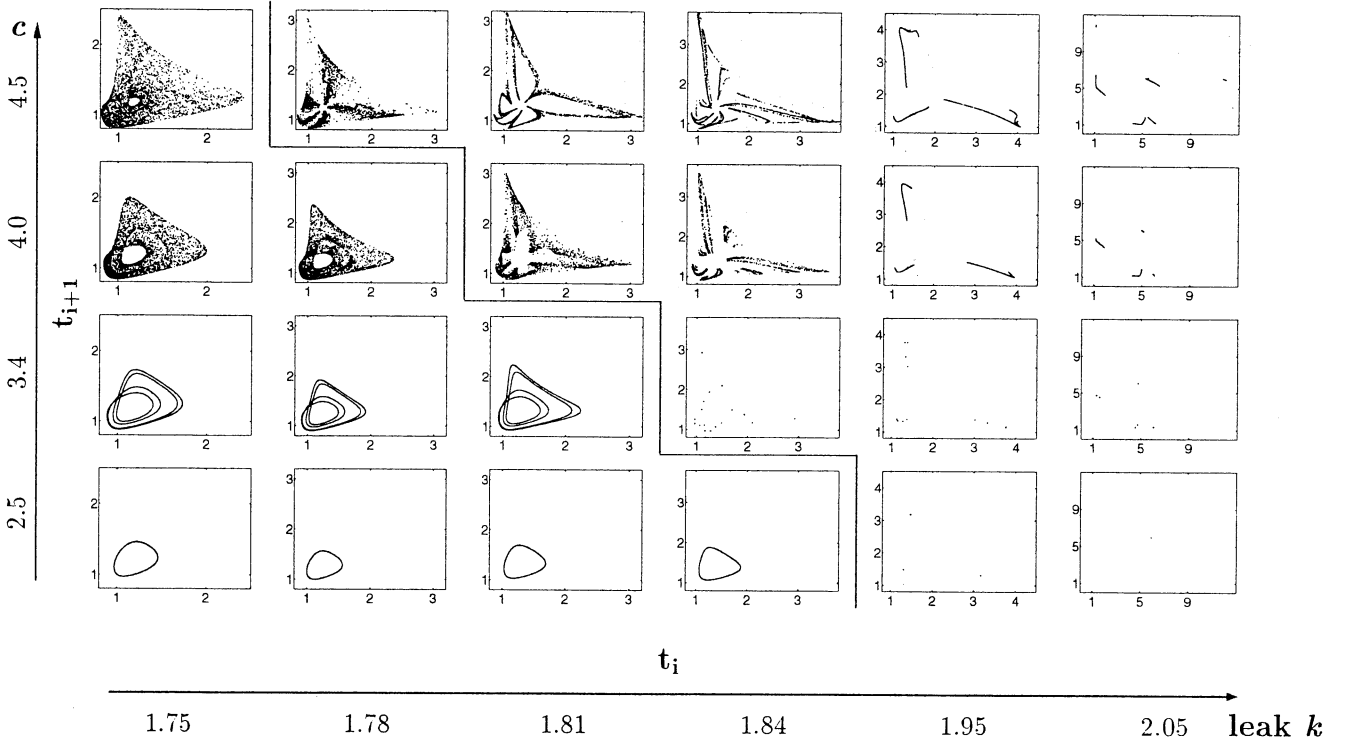
Figure 3: The reconstructed attractors from the interspike intervals. The respective figures are at $c = 2.5, 3.4, 4.0, 4.5$ (column) and $k = 1.75, 1.78, 1.81, 1.84, 1.95, 2.05$ (row). The solid line denotes the border [eq.(5)] between *region 1* and *region 2*. 4000 interspike intervals after an initial transient are used in each plot.
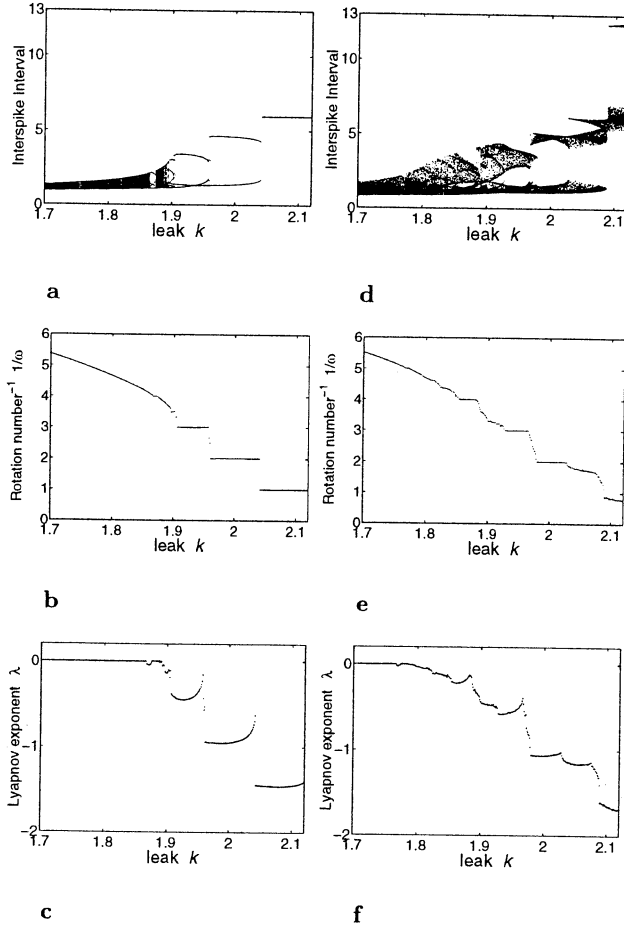
Figure 4: The interspike intervals (**ad**), the *generalized rotation number* (**be**), and the Lyapunov exponent (**cf**) with increasing leak k. The input is either periodic ($c = 2.5$, **abc**) or chaotic ($c = 4.0$, **def**). 500 or 3000 spikes are used to calculate the *generalized rotation number* and the Lyapunov exponent for the periodic or chaotic input respectively.

5   Conclusion

We have examined the firing patterns of a chaotically forced leaky integrate-and-fire (LIF) model and the validity of reconstructing the input chaotic dynamics from the spike sequence. Similar behaviors as periodically forced LIF are observed: (i) emergence of qualitatively distinct behaviors across a border of the parameter space through *grazing* bifurcation at the border and (ii) mode-locked regions when the leak of the LIF is greater than the border. Our results suggest that *grazing* bifurcation is related to the reconstruction of chaotic dynamics with the LIF. We indicate that *grazing* bifurcation may explain the previous result [5][6] that the reconstructed dynamics has changed at a value of the leak parameter. Moreover, it may be of much relevance in the reconstruction of a dynamical system with other excitable systems such as the FitzHugh-Nagumo neuron model. The mode-locked behaviors may be related to phase synchronization of chaotic oscillators [15].

Acknowledgement

Appendix

The formula of the leaky integrate-and-fire model (1) is rewritten as

$$\frac{d\tilde{u}}{ds} = -\frac{k}{\mu}\tilde{u} + \frac{1}{\mu}S(\mu s),$$

where $s = t/\mu$ and $\tilde{u}(s) = u(t)$.

References

[1] Takens F, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence*, Spinger-Verlag, Berlin, vol. 898, pp. 366-381, 1981.

[2] Sauer T, "Reconstruction of dynamical systems from interspike intervals," *Phys. Rev. Lett.*, vol. 72, pp. 3811-3814, 1994.

[3] Sauer T, "Reconstruction of integrate-and-fire dynamics," in *Nonlinear dynamics and time series*, American Mathematical Society, Providence, RI, pp. 63-75, 1997.

[4] Richardson KA, Imhoff TT, Grigg P, et. al. , "Encoding chaos in neural spike trains," *Phys. Rev. Lett.*, Vol. 80, pp. 2485-2488, 1988.

[5] Racicot DM, Longtin A, "Interspike interval attractors from chaotically driven neuron models," *Physica D*, vol. 104, pp. 184-204, 1997.

[6] Suzuki H, Aihara K, Murakami J, et. al. , "Analysis of neural spike trains with interspike interval reconstruction," *Biol. Cybern.*, vol. 82, pp. 305-311, 2000.

[7] Keener JP, Hoppensteadt FC, Rinzel J, "Integrate and fire models of nerve membrane response to oscillatory input," *SIAM J. Appl. Math.*, vol. 41, pp. 503-517, 1981.

[8] Coombes S, "Liapunov exponents and mode-locked solutions for integrate-and-fire dynamical systems," *Phys. Lett. A*, vol. 255, pp. 49-57, 1999.

[9] Coombes S, "Mode locking and Arnold tongues in integrate-and-fire neural oscillators," *Phys. Rev. E*, vol. 60 pp. 2086-2096, 1999. Erratum: *Phys. Rev. E*, vol. 6305, no. 059901, 2001.

[10] Coombes S, Owen MR, Smith GD, "Mode locking in a periodically forced integrate-and-fire-or-burst neuron model," *Phys. Rev. E*, vol. 6404, no. 041914, 2001

[11] Pakdaman K, "Periodically forced leaky integrate-and-fire model," *Phys. Rev. E*, vol. 6304, no. 041907, 2001.

[12] Rhodes F, Thompson CL, "Rotation numbers for monotone functions on the circle," *J. London Math. Soc.*, vol. 34, pp. 360-368, 1986.

[13] Rhodes F, Thompson CL, "Topologies and rotation numbers for families of monotone functions on the circle," *J. London Math. Soc.*, vol. 43, pp. 156-170, 1991.

[14] Budd CJ, "Non-smooth dynamical systems and the grazing bifurcation," in *Nonlinear mathematics and its applications*, Cambridge University, pp. 219-235, 1996.

[15] Rosenblum MG, Pikovsky AS, Kurths J, "Phase synchronization of chaotic oscillators," *Phys. Rev. Lett.*, vol. 76, pp. 1804-1807, 1996.

Dimension analysis of the Hodgikin-Huxley equations with noise:

Effect of random noise on chaos

Hiroaki Tanaka[2] and Kazuyuki Aihara[1,2]

1: Department of Mathematical Engineering and Information Physics
The Univ. of Tokyo Bunkyo-ku, 113-8656, Japan
2:CREST, Japan Science and Technology Co. (JST)
4-1-8 Hon-cho, Kawaguchi, 332, Japan

Abstract

The chaotic behavior of Hodgikin-Huxley equations modulated by noise [1] is studied with the dimension and the information entropy at various bin sizes.

It is found that the local dimension increases when the box or bin size is smaller than about 30 times the noise amplitude. The dimension reaches the degree of freedom of the system (=4), where the bin size is the same with the noise amplitude.

Where the box size is about 1/30~1/3 of the phase space size, the information entropy is remarkably less than where there is no noise. Although, it increases at smaller than 1/30 of the space size or 30 times the noise amplitude.

These suggest that (1) it takes about 30 times larger scale to converge the noise fluctuation on the strange attractor plane and (2) the noise induced order (NIO) emerges at rather macroscopic scale.

Key words: Hodgikin-Huxley equations, noise, dimension, information entropy, noise-induced order

Introduction

The squid giant axon shows chaotic behavior under certain sinusoidal current stimulation [2,3,4]. The chaos in the neuron was well simulated by the numerical calculation of Hodkin-Huxley equations [5]. We found previously that the chaotic behavior of the Hodgikin-Huxley equations was modulated by small white noise [1]. We suggested that the noise-amplitude dependence on the chaotic behavior was caused by two different mechanisms [6].

In this paper, the chaotic behavior of Hodgikin-Huxley equations modulated by noise is studied with the box and correlation dimensions and the information entropy at various bin sizes. It is found that (1) it takes about 30 times larger scale to converge the noise fluctuation and (2) the noise induces order where the box size is 1/30~1/3 of the phase space size.

Simulation

Detail methods and parametric conditions for the numerical calculation are based on Usami et.al. [5]. Gaussian noise is superimposed to all variables of the Hodgikin-Huxley equation, V, m, h and n, with certain normalization, respectively. The noise amplitude of 1.0 is adjusted to that the deviation of the noise is same as the phase space size.

Time series of Poincare section data at 240 degree of sinusoidal current phase are calculated. We have studied the following points with various box or bin size: (1) local dimension from the slope of the box counting versus box size, (2) information entropy from the distribution probability of the data points and (3) local dimension from the slope of the bin counting versus bin size.

Results

The box counting method is applied on the Poincare section data of 200,000 points, which is practically available data number for our computer system. Fig. 1 shows the box count versus box size in the logarithmic scale. When the box size is decreased, the box count increases until the limitation of data number. When the noise is applied, the box count becomes larger than the case of no noise, where the box size is smaller than some certain size.

The slope of the logarithmic relation gives the local box dimension, as is shown in Fig. 2. It is found that the calculated dimension for the case of no noise is around 1.3. When the noise is applied, the dimension increases where the box size is smaller than some certain value. But the maximum value is only around 2.3, which should be 4 at the area comparable to the noise amplitude. This is related to the fact that the data number is insufficient for the accurate calculation.

Contrary to this, the information entropy shows a clear increase with decreasing box size in direct correlation to the noise amplitude (Fig. 3). Fig. 4 shows the point of box size where entropy starts to increase in the logarithmic scale. Those are about -0.5, -1.0, -1.5, -2.0 and -2.5 for the noise amplitude (logarithmic scale) of -2.0, -2.5, -3.0, -3.5 and -4.0, respectively, showing that the point is about 30 times larger than the noise amplitude. In the case of noise amplitude -6.0, this relation is not conserved, which is probably related to the insufficient data number.

In addition, there is a remarkable decrease of the information entropy, when the box size is about -1.5 to -0.5 in logarithmic scale. This may relate to the noise induced order (NIO), which is reported by Matsumoto and Tsuda [7].

The correlation integration method is applied on the Poincare section data of 10,000 points, which is rather easy data number for our computer system [8,9]. Fig. 5 shows the bin count versus bin size in the logarithmic scale. When the bin size is decreased, the bin count decreases. When the noise is applied, the bin count decreases steeply from some certain bin size, which has close correlation to the noise amplitude.

The slope of the relation gives the local correlation dimension, as is shown in Fig. 6. It is found that the calculated dimension for the case of no noise is around 0.5 for larger bin size, which may reflect the macroscopic

structure of the chaos, and is increased to 2.0, where bin size is decreased.

When the noise is applied, the dimension increases at smaller bin size region. The point of bin size where local dimension starts to increase in the logarithmic scale is about -0.5, -1.0, -1.5, -2.0 and -2.5 for the noise amplitude (logarithmic scale) of -2.0, -2.5, -3.0, -3.5 and -4.0, respectively. These show that the point is about 30 times larger than the noise amplitude.

The maximum value of the local dimension is about 4.0, which is same to the noise dimension. This also implies that the data number is sufficient for the correlation integration method and that the method is superior for the smaller data number [8,9].

Discussion

From the local correlation dimension data, the dimension is almost same to the noise's dimension where the bin size is smaller than the noise amplitude. This larger dimension is compressed to the intrinsic system dimension when the bin size is increased, however it takes about 30 times larger.

The noise fluctuates the strange attractor orbits to the outside of the attractor super plane. Thus the size for the convergence may reflect the structure of the phase space around the plane.

NIO, the information entropy decrease, is observed where the box size is 1/30~1/3 of the phase space size, which shows no dependency on the noise amplitude. Although, the extent of NIO is remarkable where noise amplitude is -3.0 or -3.5. This should be also noted that NIO observable box sizes are significantly larger than the noise amplitude. It indicates that during the compression of the noise fluctuation the orbit slips off from the extending area in the phase space and converges to give the NIO.

Conclusion

In this study it is found that (1) it takes about 30 times larger scale to converge the noise fluctuation on the strange attractor plane and (2) the noise induces order when the box size is 1/30~1/3 of the phase space size.
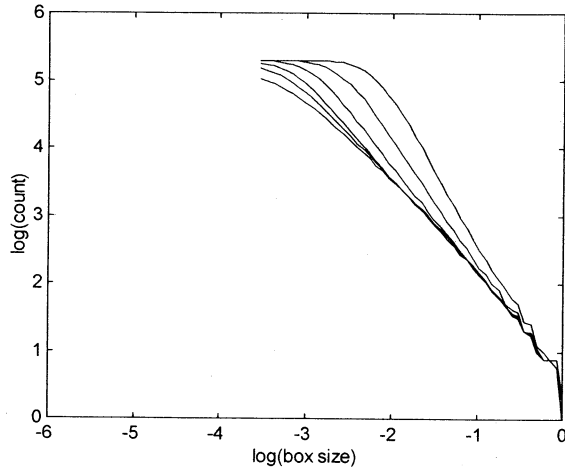
Fig. 1. The relation between box count versus box size. The box count and the box size are shown in the logarithmic scale. Poincare section data of 200,000 points were used for the box dimension analysis. The curves from right to the left are the results of noise applied in amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4} and 0, respectively.
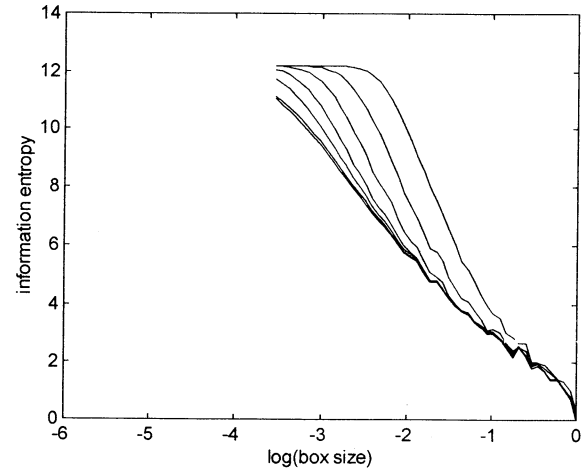
Fig. 3. The relation between information entropy versus box size. The box size is shown in the logarithmic scale. Poincare section data of 200,000 points were used for the calculation of information entropy. The curves from right to the left are the results of noise applied in amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4}, 10^{-6} and 0, respectively.
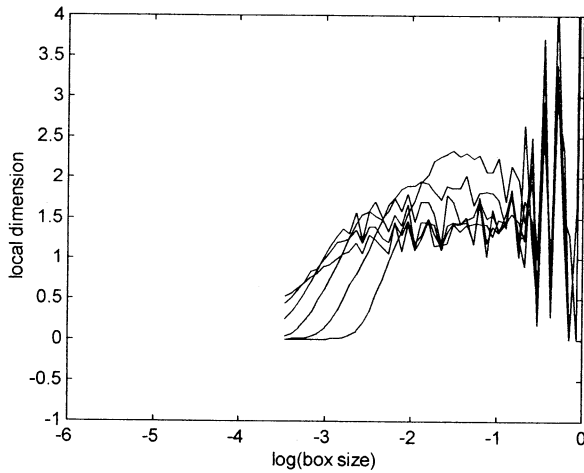
Fig. 2. The local box dimension versus the box size. The slope of the logarithmic relation between the box count versus the box size gives the local box dimension at the certain box size. The curves from right to the left are the results of noise applied in amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4} and 0, respectively.
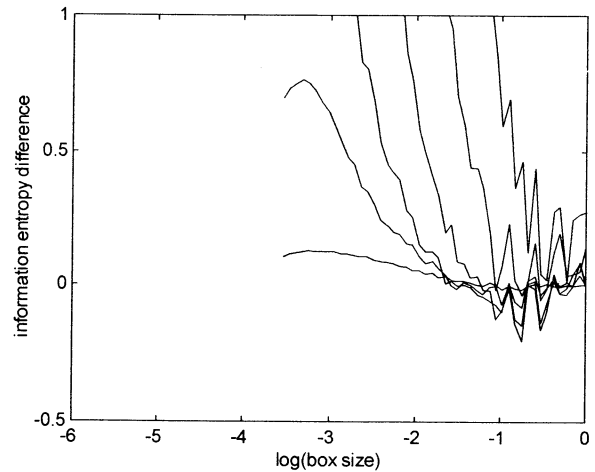
Fig. 4. The difference in information entropy between larger amplitudes of noise and no noise. The curves from right to the left are the results of noise applied in amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4} and 10^{-6}, respectively,
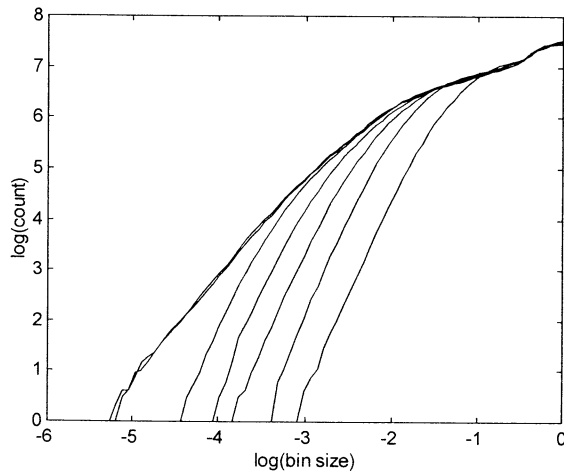
Fig. 5. The relation between bin count versus bin size. The bin count and the bin size is shown in the logarithmic scale. Poincare section data of 10,000 points were used for the box dimension analysis. The curves from right to the left are the results of noise applied in amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4}, 10^{-6} and 0, respectively.
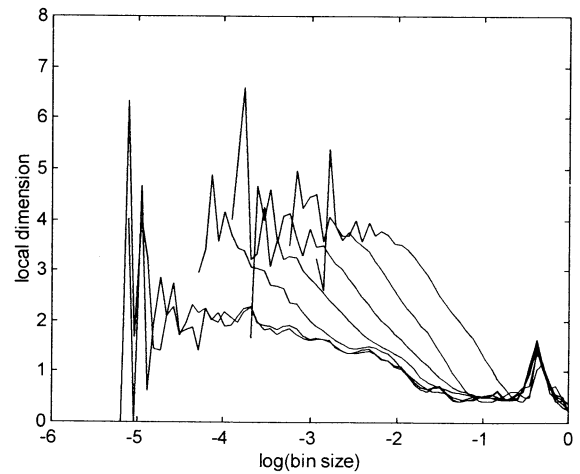
Fig. 6. The local box dimension versus the box size. The slope of the logarithmic relation between the bin count versus the bin size gives the local correlation dimension at the certain box size. The curves from right to the left are the results of noise applied in amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4}, 10^{-6} and 0, respectively.

References

[1] Tanaka H, Aihara K (1997), Chaotic behavior of the Hodgikin-Huxley equations under small random noise. Complexity and Diversity. Springer-Verlag, Tokyo, pp.172-174

[2] Aihara K, Matsumoto G, Ikegaya Y (1984), Periodic and non-periodic responses of a periodically forced Hodgkin-Huxley oscillator. J. theor. Biol. 109:249-269

[3] Aihara K, Matsumoto G, Ichikawa M (1985), An alternating periodic-chaotic sequence observed in neural oscillators. Phys. Lett. A 111:251-255

[4] Aihara K, Numajiri T, Matsumoto G, Kotani M (1986), Structures of attractors in periodically forced neural oscillators. Phys. Lett. A 116:313-317

[5] Usami T, Yamada T, Ichinose N, Aihara K(1995), Hodgikin-Huxley equation and its response to the periodic stimulation. J.SICE. 34:769-774

[6] Tanaka H, Aihara K (2000), Effect of small random noise on the chaotic behavior of the Hodgikin-Huxley equations. Proc. of The Fifth Int. Symp. on Artificial Life and Robotics. pp.165-168

[7] Matsumoto K, Tsuda I (1983), Noise-induced order. J. Stat. Phys. 31:87-106

[8] Grassberger P, Procaccia I (1983), Measuring the strangeness of strange attractors. Physica. 9D:189-208

[9] Grassberger P, Procaccia I (1983), Characterization of strange attractors. Physical Review Letters. 50:346-349

Statistical Analysis on BVP Equations Stimulated by White, Colored and Chaotic Noises

R.Hosaka[1]   Y.Sakai[2]   T.Ikeguchi[1,2]   and   S.Yoshizawa[1,2]

[1]Graduate School of Science and Engineering

[2]Faculty of Engineering

Saitama University

255 Shimo-Ohkubo, Saitama, 338-8570, JAPAN

Abstract

In this paper, we analyze responses of the Bonhöfer-van der Pol (BVP) equations stimulated by white, colored and chaotic noise as externally applied inputs. We evaluate them through observing interspike intervals in the case of changing parameters of input signals. As a result, under physiologically reasonable conditions, when white noise inputs are applied, we observe that random spike sequences are generated. When colored noise inputs are applied, irregular and uncorrelated spike sequences are also observed. As a source of chaotic noise, we utilize the logistic map with the parameter $a = 4$. Then, we find that BVP equation stimulated by the chaotic noise produces the same statistical features as white noise.

1 Introduction

Neuron is a basic component in cerebral cortex. Its fundamental function is to receive input spike sequences, then to apply information processing, and finally to produce output spike sequences. If we observe them from real biological systems, they exhibit very complex, irregular and noisy behavior.

It is very important to consider how can we describe a spiking mechanism in a simple way. As one of the basic models, the leaky integrate-and-fire (LIF) model has been utilized in order to describe the spiking mechanism. In order to solve the above issue, Sakai et al. analyzed the LIF model and discussed whether or not the LIF model can reproduce statistical features of the real biological spiking data [1], [2]. In Refs. [1], [2], the biological spiking data obtained from the delay response task experiment were examined. The experiments were carried out by Funahashi et al. [4] and Golodman-Rakic et al. [5]. Under the assumption that inputs can be described by a stationary random noise, response properties of the LIF model stimulated by several noises are analyzed [1], [2]. As a result, it is shown that colored input with a long temporal corre-

lation is necessary to reproduce statistical properties of real biological spiking data.

From the view point of neural spiking mechanisms, there are many neuron models. For example, the Hodkin-Huxley equation [3] is one of the good models which can represent spiking mechanisms in nerve axons. However, since the Hodgkin-Huxley equation is described by the 4-order ordinary different equation, and has a very complicated form, it might be better to use more simple models. Another good candidate is the Bonhöfer-van der Pol (BVP) equation [6] which has a simpler structure than the Hodgkin-Huxley equation. It is well known that BVP equation is a widely used model to analyze neural dynamics, chemical vibrations, circadian rhythms, and so on [7].

In this paper, we use the BVP equation in order to analyze output responses through interspike intervals (ISIs), when input parameters are changed. We use the same framework by Sakai et al [1], [2]. First, output responses by white noise are analyzed by three statistical coefficients and they are compared with those of the real biological data obtained from the experiments [4] [5]. Second, we also analyze that the distributions of statistical coefficients when colored noise and chaotic noise are applied to the BVP equation.

2 BVP equation

The BVP equation is described by the following Equation:

$$\begin{cases} \dot{u} = C(u + w - u^3/3 + I(t)), \\ \dot{w} = (A - Bw - u)/C, \end{cases} \quad (1)$$

where the variable u is the membrane potential of the neuron, w represents the refractoriness, and $I(t)$ is a current stimulus applied externally. For the equations to mimic the behavior of a real neuron, the parameters A, B, and C are set as follows [6]:

$$1 - \frac{2}{3}B < A < 1, 0 < B < 1, B < C^3. \quad (2)$$

Spike widths in the BVP equation are about 5, but real biological spike widths are about 2[ms]. Namely, there are 2/5 scale differences between the temporal unit of the BVP equation and real neural systems. Then, the BVP equation is transformed so that spikes width become 2, then the following equations are obtained by introducing the time constant τ:

$$\begin{cases} \tau \dot{u} = -w + u - u^3/3 + I(t), \\ \tau \dot{w} = (-A + Bw - u)/D, \end{cases} \quad (3)$$

where $\tau = \dfrac{2}{5C}$ and $D = \dfrac{1}{C^2}$. In this paper, we use the modified BVP equation (Eq.(3)) in the following analyses. For solving the BVP equation, we use the Runge-Kutta method, and numerically integrate the equation with constant time step 0.02.

For defining spike times of neurons, we use the variable $u(t)$. Namely, the ith time t_i is defined when the variable $u(t)$ equals to zero and the derivative of $u(t)$ is positive.

With the above definition of t_i, ISI sequences $T_i = t_{i+1} - t_i$ are obtained. In the Section 6, ISI sequences are analyzed for calculating statistical coefficients.
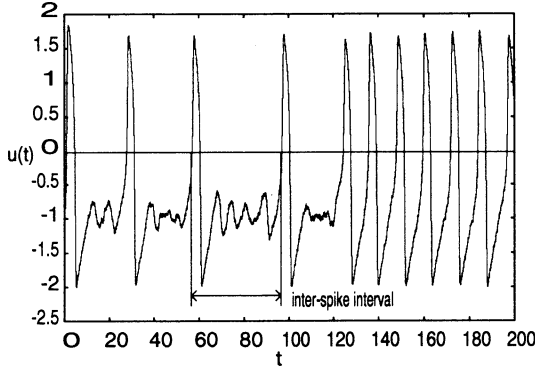
Figure 1: An example of a time wave form of $u(t)$ of the BVP equation.

3 Externally applied inputs

In this paper, the following three noises are used for externally applied stimuli to the BVP equation.

3.1 White Noise

Membrane potentials of neurons are changed by receiving signals from other neurons. This potential variation has a sharp peak, and usually returns to a near-resting level within the order of milliseconds. Sizes of potential variations are almost constant, and they takes either positive and negative values. In the case that output signals from neurons are temporally uncorrelated, if the individual impulses are small in

magnitudes compared with the height of the threshold potential, the temporally uncorrelated inputs can be approximated as Gaussian white noise [8].

In this paper, the following inputs are used as an approximation of uncorrelated inputs.

$$I_w(t) = \mu + \sigma \xi(t) \quad (4)$$

where $\xi(t)$ represents Gaussian noise with zero mean and unit variance. Parameter μ and σ decide mean and variance of $I_w(t)$.

3.2 Colored Noise

Here we consider the situation that several thousand of neurons oscillate not independently but have correlation. In this paper, the following first order delay input is used as a simple description of correlated inputs.

$$\begin{cases} I_c(t) = \mu + \sigma \eta(t), \\ s\dot{\eta}(t) = -\eta(t) + \xi(t), \end{cases} \quad (5)$$

where $\xi(t)$ represents Gaussian noise with zero mean and unit variance. Then, $I_c(t)$ has a temporal correlation represented by exponential functions which are decreasing with time constant s. When the time constant $s \to 0$, η converges to white noise $I_w(t)$ and input has no correlation. In Figs.2(a) and (b), $I_c(t)$ with $s=0$ and $s=100$ are shown.

(a) $s=0$ (b) $s=100$
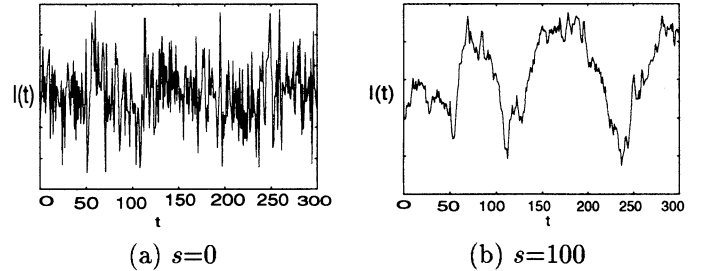
Figure 2: Two examples of $I_c(t)$.

3.3 Chaotic Noise

It has been indicated that even though the temporal signals exhibit very random behavior, there may exist underlying deterministic dynamical structures. Recently, it is called deterministic chaos. Chaos is complex behavior produced from nonlinear deterministic dynamics with a few degrees of freedom. One of the good example of deterministic chaos is the logistic map [9]. We use the logistic in this paper.

The logistic map is described by Eq.(6).

$$x(t+1) = ax(t)(1 - x(t)) \quad (6)$$

where a is a parameter. It is known that when the parameter a is 4, the logistic map becomes chaotic but is equivalent to a stochastic process, for example, random coin-tossing.

Time series of the logistic map is shown in Fig.3. Then, this time series $x(t)$ is used for producing chaotic noise input,

$$I_l(t) = \mu + \sigma x(t). \tag{7}$$
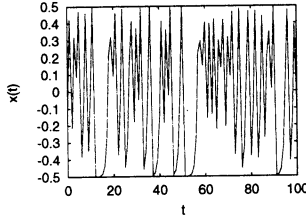
Figure 3: Responses of the logistic map

4 Statistical evaluation of outputs

For evaluating ISI series statistically, the following three coefficients are used; CV(coefficient of variation) is a measure of the spiking irregularity; SK(skewness coefficient) is a measure of asymmetry of ISI distributions; COR(correlation coefficient) is a measure of correlation of successing two ISIs. They are defined as follows.

$$CV = \frac{\overline{(T_i - \overline{T_i})^2}^{\frac{1}{2}}}{\overline{T_i}}, \tag{8}$$

$$SK = \frac{\overline{(T_i - \overline{T_i})^3}}{\overline{(T_i - \overline{T_i})^2}^{\frac{3}{2}}}, \tag{9}$$

$$COR = \frac{\overline{(T_j - \overline{T_i})(T_{j+1} - \overline{T_i})}}{\overline{(T_i - \overline{T_i})^2}}. \tag{10}$$

In Fig.4, we show the results form the biological spiking data of a monkey [1], [2], [4], [5]. In the figure, dots represent the estimated (CV,SK), (COR,SK) values of 666 spike sequences of the biological data.

5 Physiological view

From the physiological point of view, it may be desirable to remove several data from the simulated results from simulations. If neurons represent information by spikes, there must be enough temporal intervals between two successive spikes. Spike of neurons do not exhibit a perfect delta function system, but it
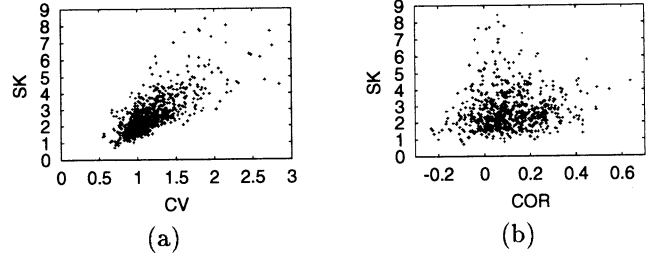
Figure 4: Distributions of statistical coefficients of the biological spiking data.

takes a time width about 2 [ms]. Moreover, it is reported that mean ISIs of spike sequences observed in experiments are at least larger than 20 [ms], and take generally almost 100 [ms]. Namely it is also important to consider that the time of about 10-times of spike widths is plausible for mean ISIs.

Since spike widths of the modified BVP equation are almost 2, mean ISI must be larger than 20. From the above consideration, if the mean ISIs are less than 20, the ISI sequences are removed from the results obtained from simulation. If the mean ISI are larger than 200, they are also removed. Namely, we use the following physiological condition for observing ISI sequences:

$$50[ms] < \overline{T_i} < 200[ms]. \tag{11}$$

6 Results

In the following, parameters of Eq(3) are set to $A = 0.7, B = 0.8, C = 3$ with which the response of the BVP equation is an equilibrium point.

6.1 White noise

The modified BVP equation (Eq.(3)) is stimulated by the white noise input. Output sequences which satisfy Eq.(11) among 1000 spike interval series are evaluated by the statistical coefficients CV, SK and COR. These values are projected on the planes shown in Fig.5.

As for CV, SK, and COR, if we compare the results with Fig.4, we see that they are distributed in very small areas. It turns out that the BVP equations which are stimulated by the white noise generate more random spike sequences compared with the real biological spiking data as shown in Fig.4.

6.2 Colored noise

Next, we analyze the output responses of the BVP equation which is stimulated by the colored noise input. Time constant s in Eq.(5) is changed with 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. The results show that the increase of s leads to the movements
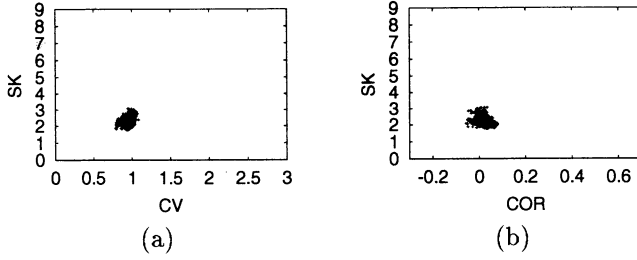
Figure 5: Distributions of statistical coefficients of output ISIs of the BVP neurons stimulated by white noise.

of (CV,SK) with keeping the relation of $SK = 2CV$ (Fig.6). However, as for COR, a significant change cannot be seen, even if s is increased.
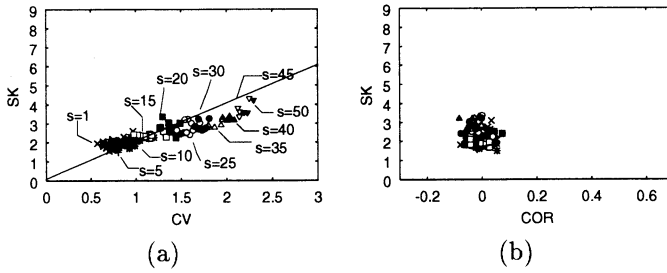
Figure 6: Distributions of statistical coefficients of output ISIs of the BVP equation stimulated by colored noise.

6.3 chaotic noise

When the chaotic noise inputs stimulate the BVP equation, the same statistical coefficients are obtained as the case of white noise inputs (7). It is because the results shown in Fig.7 corresponds to the case that parameter value $a = 4$, which is equivalent to random sequences. It is very important future problem to use another parameter values of the logistic map.
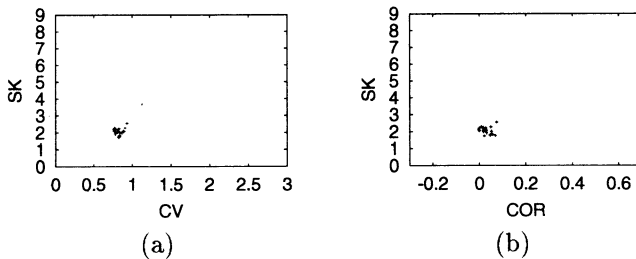
Figure 7: Distribution of statistical coefficients of output ISIs of the BVP equation stimulated by chaotic noise.

7 Conclusion

In this paper, we stimulate the BVP equation by white noise, colored noise and chaotic noise. The results show that the statistical characters of the real biological spiking data cannot be reproduced. However, different from the case of white noise and chaotic noise, the distributions of the statistical coefficients in the case of colored noise stimuli look like the distribution of the real biological spiking data, it may be possible to account for the real biological data by the BVP neurons with colored inputs.

References

[1] Y. Sakai, S. Funahashi and S. Shinomoto, "Temporally correlated inputs to leaky integrate-and-fire model can reproduce spiking statistics of cortical neurons," Neural Networks, Vol.12, No.8, pp.1181-1190, 1999.

[2] S. Shinomoto, Y. Sakai and S. Funahashi, "The Ornstein-Uhlenbeck Process Does Not Reproduce Spiking Statistics of Neurons in Prefrontal Cortex," Neural Computation, Vol.11, pp.935-951, 1999.

[3] A. L. Hodgkin & A. L. Huxley " A quantitative description of membrane current and its application to conduction and excitation in nerve," J.Physiology, Vol.117, pp.500-544, 1952.

[4] S, Funahashi, C. J. Bruce, and P. S. Goldman-Rakic, "Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex," J.Neurophysiology, Vol.61, pp.331-349, 1989.

[5] P. S. Goldman-Rakic, C. J. Bruce, and S. Funahashi, "Neocortical memory circuits," in " Cold Spring Harbor Symposia on Quantitative Biology,"Vol. LV. 'pp.1025-1038 Cold Spring Harbor Laboratory Press, 1990.

[6] J.Nagumo, S. Arimoto, and S.Yoshizawa, "An Active Pulse Transmission Line Simulating Nerve Axon," Proceeding.of the Inst. Radio Engineers, Vol.50, No.10, pp.2061-2070, October, 1962.

[7] S.Doi, and S.Sato, "The Global Bifurcation Structure of the BVP Neuronal Model Driven by Periodic Pulse Trains," Mathematical Biosciences,Vol.125, pp.229-250, 1995.

[8] H. C. Tuckwell, Introduction to theoretical neurobiology, Cambridge University Pres, Cambridge, 1988.

[9] R. May, "Simple mathematical models with very complicated dynamics,"Nature, Vol.261, pp.459-467, 1976.

Symbolic Dynamics of Bimodal Maps and Homomorphism

Koji Fukuda[1] and Kazuyuki Aihara[12]

[1] *Department of Mathematical Engineering and Information Physics,*
The University of Tokyo, Hongo, Tokyo 113-8656, Japan

[2] *CREST, Japan Science and Technology Corporation (JST), Kawaguchi, Saitama 332-0012, Japan*

Abstract

Chaotic neurons change their internal state according to a bimodal map, and when they communicate with other neurons, their internal state is transformed into one of two separate outputs — firing or non-firing.

Symbolic dynamics is used to analyze the system. The topological entropy of the two-value output of a chaotic neuron is shown to form a devil's staircase-like property.

1 Introduction

A chaotic neuron changes its internal state according to a certain bimodal map [1]. When it communicates with other neurons, its internal state is transformed into one of two separate outputs because of waveform shaping with a threshold on the axons. That is, if the value of the internal state is over a certain threshold, the output of the neuron is an excited state with firing, otherwise it is a non-excited state without firing. (Fig. 1)
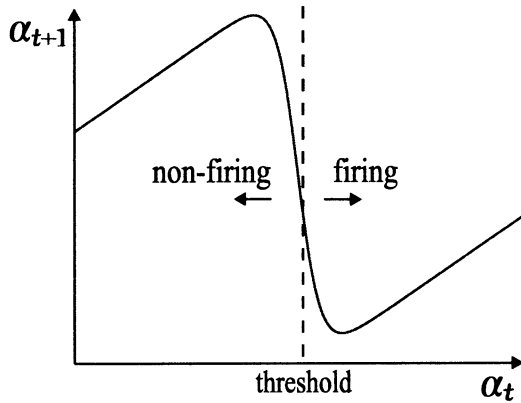
Figure 1: a chaotic neuron map

The purpose of our study is to understand what occurs in the two-value output of a chaotic neuron when the parameters of the neuron are changed.

2 Symbolic Dynamics of Bimodal Map

Consider the following discrete system with real-valued initial condition $\alpha_0 = \alpha$:

$$\alpha_{t+1} = f(\alpha_t), \qquad (1)$$

where f is a bimodal map with slope of $(+ - +)$ type. We use symbolic dynamics to investigate this system.

By dividing the real axis into three regions named L,M, and R separated by two real numbers, C and D, where the bimodal map f is at a local minimum and local maximum respectively, we define a map A from $\mathbb{R}$ to set $\Pi = \{L, D, M, C, R\}$, such that $A(\alpha)$ is the name of the region in which α is located. We define *the itinerary of f with the initial condition $\alpha_0 = \alpha$* as

$$I(\alpha) = A(\alpha_0)A(\alpha_1)A(\alpha_2)\ldots . \qquad (2)$$

For example, for the case shown in fig. 2,

$$I(\alpha) = \text{MRLMML}\ldots . \qquad (3)$$

The itinerary of f with an initial condition of either
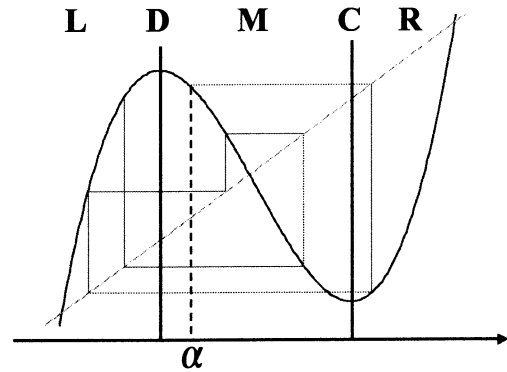
Figure 2: symbolic dynamics of a bimodal map

$f(C)$ or $f(D)$ is called a *kneading sequence* and written as,

$$\begin{cases} K_C & = & I\left(f(C)\right) \\ K_D & = & I\left(f(D)\right) \end{cases}. \qquad (4)$$

The interval $[f(C)\ f(D)]$ is the invariant set of f, i.e. whatever the initial value of α_0, there is a certain n_0 such that α_n are included in $[f(C)\ f(D)]$ for all $n \ge n_0$. Therefore, α_0 is often restricted in $[f(C)\ f(D)]$.

2.1 Admissibility Condition

Π^* denotes the set of symbol sequences of Π with a finite length. We introduce order relations $\prec$ and $\preceq$ into Π^* [2].

Definition 2.1 *Define order relation $\prec$ between symbol sequences x_1 and x_2 as follows.*

- *When both x_1 and x_2 consist of one symbol,*

$$\mathsf{L} \prec \mathsf{D} \prec \mathsf{M} \prec \mathsf{C} \prec \mathsf{R}. \qquad (5)$$

- *When x_1 and x_2 have common prefix y and different symbols A_1 and A_2 s.t. $A_1 \prec A_2$,*

$$\begin{cases} x_1 & = & yA_1z_1 \\ x_2 & = & yA_2z_2 \end{cases} \qquad (6)$$

where z_1 and z_2 are symbol sequences, then

$$\begin{cases} x_1 \prec x_2 & \text{if the number of } \mathsf{M} \text{ in } y \text{ is even,} \\ x_2 \succ x_2 & \text{if the number of } \mathsf{M} \text{ in } y \text{ is odd.} \end{cases} \qquad (7)$$

Define order relation $\preceq$ as follows.

$$x_1 \preceq x_2 \qquad \text{unless } x_1 \succ x_2. \qquad (8)$$

It is easy to show that $\prec$ and $\preceq$ are order relations.

Kneading sequences K_D and K_C can be used to determine which symbol sequences are *admissible* for a given bimodal map.

Theorem 2.2 *(Admissibility Condition)*
Given bimodal map f, there is an itinerary of f that has $x \in \Pi^*$ as the prefix if and only if

$$\begin{cases} & & \mathcal{L}(x) & \preceq & K_D, \\ K_C & \preceq & \mathcal{M}(x) & \preceq & K_D, \\ K_C & \preceq & \mathcal{R}(x) \end{cases} \qquad (9)$$

where K_C and K_D are the kneading sequences of f, and $\mathcal{L}(x)$, $\mathcal{M}(x)$, and $\mathcal{R}(x)$ denote the set of the symbol sequences that follow $\mathsf{L}, \mathsf{M}, \mathsf{R}$ in x, respectively. And $\mathcal{L}(x) \preceq K_D$ means all sequences y in $\mathcal{L}(x)$ satisfy $y \preceq K_D$.

Restricting the inidition condition to the invariant set, Eq. (9) can be simplified as follows:

$$K_C \preceq \sigma^m(x) \preceq K_D \quad \text{for } 1 \le {}^\forall m < |x|, \qquad (10)$$

where σ is the left shift map, and $|x|$ is the length of x.

The set of all symbol sequences admissible for a given f is called *the language of f* and is written as $\mathfrak{L}(f)$ or $\mathfrak{L}(K_C, K_D)$. When f and (K_C, K_D) are clear from the context, it is abbreviated to $\mathfrak{L}$ in the following.

For the sake of simplicity, we use the restricted version, Eq. (10), as the admissibility condition. Moreover, we consider only the sequences that contain neither C nor D. This means that we exclude real numbers α that satisfy $f^n(\alpha) = C$ or $f^n(\alpha) = D$ at a certain integer n. The measure of the whole of all such real numbers equals zero. Hereafter we assume that the language of f contains neither C nor D. we denotes the set $\{\mathsf{L}, \mathsf{M}, \mathsf{R}\}$ as Σ.

The admissibility condition also gives the condition for a pair of symbol sequences being kneading sequences.

Theorem 2.3 *Given a pair of sequences K_C, $K_D \in \Pi^*$, there is a bimodal map that has K_C and K_D as the kneading sequences if and only if*

$$\begin{cases} & & \mathcal{L}(K_C, K_D) & \preceq & K_D, \\ K_C & \preceq & \mathcal{M}(K_C, K_D) & \preceq & K_D, \\ K_C & \preceq & \mathcal{R}(K_C, K_D). \end{cases} \qquad (11)$$

It should be noted that kneading sequences are elements of Π^*, so they may contain C or D.

2.2 Minimal Forbidden Blocks

we can obtain $\mathfrak{L}(f)$ from Σ^* by excluding some sequences. These excluded sequences are called *forbidden blocks*. Forbidden blocks that do not contain other forbidden blocks as subsequences are called *minimal forbidden blocks* (MFBs). Generally, the MFBs of language L on symbol set Λ are characterized as follows [4]:

$$\mathrm{MFB}(L) = \Lambda L \cap L\Lambda \cap (\Lambda^* \setminus L). \qquad (12)$$

The condition for a symbol sequence to be an MFB of $\mathfrak{L}(f)$ is shown below [2].

Theorem 2.4 *K_C and K_D denote the kneading sequences of f. Assuming $x \in \Sigma^*$ is a symbol sequence of length 2 or more, x is MFB of $\mathfrak{L}(f)$ if and only if the following two conditions are satisfied:*

(i) x' is a prefix of K_C, and $x \prec K_C$
         or
      x' is a prefix of K_D, and $x \succ K_D$,
      where x' denotes x with the last symbol deleted.

(ii) x has no proper prefix that satisfies condition(i).

Corollary 2.5 *If both K_C and K_D contain either C or D, the number of MFBs of $\mathfrak{L}(f)$ is finite.*

3 Symbolic Dynamics of Homomorphism

We applied homomorphism ϕ from the language of a bimodal map f onto a language with the symbols $\{0,1\}$:

$$L \to 0, \qquad M \to 0, \qquad R \to 1. \qquad (13)$$

If we regard 0 as the non-firing state and 1 as the firing state, the language on $\{0,1\}$ obtained by this homomorphism corresponds to the two-value output of a chaotic neuron whose threshold is exactly at the local minimum, C.
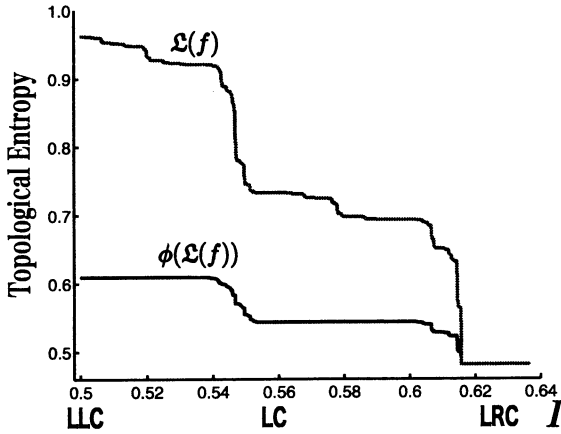
Figure 3: Change in entropy fixing with K_D = RRD. $f(\alpha) = k\alpha - 1/(1 + \exp(-\alpha/eps)) + I$ is used while changing the value of parameter I. The parameter k and *eps* are determined to keep K_D = RRD. Symbol sequences shown below the x-axis are K_C at the corresponding value of a.

Figure 3 shows the change in the topological entropy of $\mathfrak{L}$ and $\phi(\mathfrak{L})$ with K_D = RRD. The curve does not increase, because according to admissibility condition of Eq. (10),

$$\mathfrak{L}(K_C, K_D) \supset \mathfrak{L}(K'_C, K_D) \quad \text{if } K_C \prec K'_C \qquad (14)$$
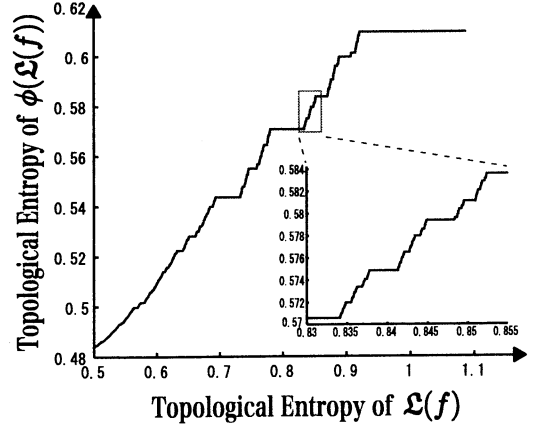
Figure 4: Entropy of $\mathfrak{L}(f)$ vs. Entropy of $\phi(\mathfrak{L}(f))$ with K_D = RRD. The same function as Fig. 3 is used. The lower right graph shows expansion of the curve inside the rectangle.

holds when K_D is fixed.

The topological entropy of $\mathfrak{L}$ versus the parameter I forms *devil's staircase-like* structure [3]. The topological entropy of $\phi(\mathfrak{L})$ seems to form devil's staircase-like structure more clearly. This is because even if $\mathfrak{L}$ changes, $\phi(\mathfrak{L})$ dose not necessarily change. As shown in Fig. 4, the topological entropy of $\mathfrak{L}$ versus that of $\phi(\mathfrak{L})$. forms a devil's staircase-like shape again.

For example, when $(K_C, K_D) = $ (LLC, RRD), the MFB of $\mathfrak{L}$ is {LLL, LLM, RRR, RRM}. If we apply homomorphism ϕ, the MFB of $\phi(\mathfrak{L})$ turns into {111}, because RRR, which is the only element of $\mathfrak{L}$ mapped onto 111, is forbidden. If the kneading sequences change to $(K_C, K_D) = $ (LLRC, RRD), although the MFB of $\mathfrak{L}$ increases to {LLL, LLM, LLRL, LLRM, RRR, RRM}, the MFB of $\phi(\mathfrak{L})$ is still {111}. This is because while both the new forbidden words, LLRL and LLRM, are mapped onto 0010, there are other words admissible in $\mathfrak{L}$ that can be mapped to 0010.

Here, we describe how the MFBs of $\phi(\mathfrak{L})$ are calculated. The theory of *finite-state automata* is helpful in doing this. First, Thorem 2.4 gives the MFBs of $\mathfrak{L}$. The finite-state automaton accepting $\mathfrak{L}$ is built from the MFB according to [4]. Next, relabeling its edges according to Eq. (13), we obtain the finite-state automaton accepting $\phi(\mathfrak{L})$. Finally, we calcurate the MFBs of $\phi(\mathfrak{L})$ from the finite-state automaton accepting $\phi(\mathfrak{L})$ by using Eq. (12).

Table 1 shows the outline of the changes in the MFBs of $\phi(\mathfrak{L})$ with K_D = RRD as K_C is increased.

If K_C is lower than LL(RRL)$^\infty$, the MFB of $\phi(\mathfrak{L})$

Table 1: Change in MFBs of $\phi(\mathfrak{L})$ with $K_D = $ RRD

K_C	MFB $\{0,1\}$
LLRRLRRLRC	111
LM(RRL)$^\infty$RC	111, 1100(110)$^\infty$11
LMRRLRRLRC	111, 110011011011
LMRRLRRLC	111, 11001101101
LMRRLRC	111, 110011011
LMRRLC	111, 11001101
LMRC	111, 110011
$\vdots$	$\vdots$
LMRMLRRLRRLRC	111, 110011
LMRMMRRLRRLRC	111, 110011, 110010011011011
LMRMMRRLRC	111, 110011, 110010011011
LMRMMRC	111, 110011, 110010011
$\vdots$	$\vdots$
LMRMMRMMRC	111, 110011, 110010011, 110010010011
LMRMMRMMRMMRC	111, 110011, 110010011, 110010010011, 110010010010011
$\vdots$	$\vdots$
LM(RMM)$^\infty$RC	1(100)*11
LMRLRRLRRLRRLRC	1(100)*11, 1100(100)*1011011011011
LMRLRRLRRLRC	1(100)*11, 1100(100)*1011011011
LMRLM(MLM)$^\infty$C	111, 11001
LMC	111, 11001

is $\{111\}$ because RRR is forbidden in $\mathfrak{L}$. When K_C has increased to LL(RRL)$^\infty$RC, it next jumps to LM(RRL)$^\infty$RC. This change in K_C makes LL the new forbidden sequence of $\mathfrak{L}$, and causes 1100(110)$^\infty$11 to become forbidden in $\phi(\mathfrak{L})$. This kind of change in which a new forbidden sequence is made for $\phi(\mathfrak{L})$ is called an *increase change*. A *reduction change* follows increase change. For example, if K_C is changed from LMRRLRRLRC to LMRRLRRLC, the MFB of $\phi(\mathfrak{L})$ changes from $\{111, 110011011011\}$ to $\{111, 11001101101\}$. One of the MFBs of $\phi(\mathfrak{L})$ becomes shorter. More detailed analysis shows that this kind of change consists of many smaller changes. To ·explain this, the change in K_C from LMRC to LMC is shown in Table 1.

If K_C is LMRC, the MFB of $\phi(\mathfrak{L})$ is $\{111, 110011\}$. When K_C reaches LMRMMRRLRRLRC, an increasing change occurs, and 110010011011011 becomes the new forbidden sequence of $\phi(\mathfrak{L})$, as it does when

K_C is LMRRLRRLRRLRC. It is followed by a succession of reduction changes from 110010011011011 to 110010011. The same process repeats also when $K_C = $ LM(RMM)^{k}RC with an arbitrary integer k. Finally, if K_C increases to LM(RMM)$^\infty$RC, the MFB of $\phi(\mathfrak{L})$ becomes 1(100)*11, which denotes $\{111, 110011, 110010011, \ldots\}$. A language like this with infinitely many MFBs is called *strictly sofic*. After a series of these increase and reduction changes, K_C becomes LMRLM(MLM)$^\infty$C. By this process, when $K_C = $ LMC, the MFB of $\phi(\mathfrak{L})$ is $\{111, 11001\}$. This is an outline of the reduction change from $K_C = $ LMRC to $K_C = $ LMC that changes the MFB of $\phi(\mathfrak{L})$ from $\{111, 110011\}$ to $\{111, 11001\}$.

This example illustrates why the devil's staircase-like shape in Fig. 4 forms. That is, each reduction change can be decomposed into a succession of increase and reduction changes that again consist of smaller changes.

4 Conclusion

We have used symbolic dynamics to analyze chaotic neurons and shown the topological entropy of the two-value output of a chaotic neuron forms a devil's staircase-like shape. This shape occurs because each reduction change can be decomposed into a succession of increase and reduction changes that again consist of smaller changes.

Acknowledgements

References

[1] Aihara K, Takabe T, Toyoda M, "Chaotic Neural Networks" *Phys. Lett. A*, 144, pp. 333–340, 1990.

[2] Bai-Lin Hao, Wei-Mou Zheng, em Applied symbolic dynamics and chaos, World Scientific Publishing, 1998.

[3] Shou-Li Peng, Xu-Sheng Zhang, "The Generalized Milnor-Thurston Conjecture and Equal Topological Entropy Class in Symbolic Dynamics of Order Topological Space of Three Letters", *Commun. Math. Phys.*, 213, pp. 381–411, 2000.

[4] M. Crochemore, F. Mignosi, A.Restivo, "Automata and forbidden words", *Information Processing Letters*, 67, pp. 111–117, 1998.

Tabu Search for Traveling Salesman Problems
and Its Extension to Chaotic Neurodynamical Search

M. Hasegawa

Wireless Commun. Div.

Communications Research Laboratory

Yokosuka, 239-0847 Japan

T. Ikeguchi

Dept. Infom. and Math. Sci.

Saitama University

Saitama, 338-8570 Japan

K. Aihara

Dept. Compl. Sci. and Eng.

University of Tokyo

Tokyo, 113-8656 Japan

Abstract

The k-opt method is introduced as a base of chaotic neurodynamical search, for solving the Traveling Salesman Problems. First, the tabu search based on the k-opt method is realized with very simple tabu list. Then, it is implemented on a neural network with refractory effects. The proposed method using chaotic neurodynamics is further extension of the tabu search neural network to chaotic neural network version. In order to obtain high performance, we also introduce an adaptive k-opt. Then, solutions within about 1% from the known best could be solved in average, even on large TSPs, such as 85900-city problems.

1 Introduction

We often have to solve optimization problems on many situations in industries and engineering. Although many effective methods have been proposed for solving such problems, it takes huge amount of computational time to obtain the exact optimum solutions in the case of large problems. Furthermore, it is almost impossible especially on the large NP-hard problems. Therefore, it is strongly desired to develop effective algorithms which solve very good near optimum solutions in reasonable time.

The chaotic dynamics also have been applied to searching solutions of the combinatorial optimization problems [1]–[4]. The first approach using chaotic neurodynamics [5] was based on the Hopfield-Tank neural network [6]. The effectiveness of this search was shown by comparing its solvable performances with those of stochastic searches, such as the Boltzmann machines [1, 2], on the small Traveling Salesman Problems (TSPs). However, this first chaotic approach does not have so high performances as the other conventional heuristic searching methods, because the basic part of the conventional chaotic approach is the Hopfield-Tank neural network which is a kind of mutual connection neural networks. Namely, the computational complexity is huge for large problems, and even searching of feasible solutions is difficult for this approach.

In order to apply efficient chaotic search to much larger and difficult combinatorial optimization problems, another chaotic neurodynamical approach, which is not based on the Hopfield-Tank neural network but on the heuristic algorithms, was also proposed [7]. The tabu search is one of the effective heuristic methods for combinatorial optimization. From the viewpoint of the tabu effect of the tabu search which avoids occurrences of the previous moves, the chaotic neurons [5] utilized in the conventional chaotic approaches has similar characteristics to the tabu search. The chaotic neuron includes the refractory effect in each neuron that is characteristics real biological neurons; neurons become hard to fire after previous firings. Then, it is possible to implement the tabu search on the chaotic neural networks by realizing the tabu effects by refractory effects of each neuron. Since this search includes both effects of tabu search and chaotic search, it has very high performance [8, 9].

In this paper, k-opt methods are introduced as a base of chaotic neurodynamical searches, for solving the TSPs. First, the tabu search based on the k-opt is realized with a very simple tabu list. Then it is extended to the chaotic neural network version. Since the computational amount of the k-opt with large k is very large, we also propose an algorithm for adaptively changing k. The proposed methods are applied various problems up to the 11,849-city problem, and performances of the proposed methods are evaluated.

2 Tabu search for a base of chaotic search

The k-opt method is introduced for the base of the chaotic search. The algorithm of the k-opt method is shown in Fig. 1. k crossovers are applied simultaneously to the same tour at the same time. In order to make it possible to apply the method to very large TSPs, we use very simple tabu list, which memorizes only one-dimensional elements. This tabu list consists of cities which have appeared in the k-opt updating.

The advantage of using the one-dimensional list is that the chaotic search based on this tabu list requires only small size of memories [9].

Here, we use very simple tabu search based on k-opt as follows, in order to make it possible to apply the method to very large TSPs: first, a city, which provide the largest gain, is selected as the city j from non-tabu moves. If $k \geq 3$, the cities $j_l (k \geq l \geq 3)$ also have to be selected at each step increasing l of l-opt. The moves, which most shorten the total tour length, are selected at each step. Then, the largest gain move in all $i(i = 1 \ldots n)$ is selected and the tour is updated by the corresponding move. Here, since the gain value of the largest gain move is not always positive, escaping from the local optimum becomes possible.
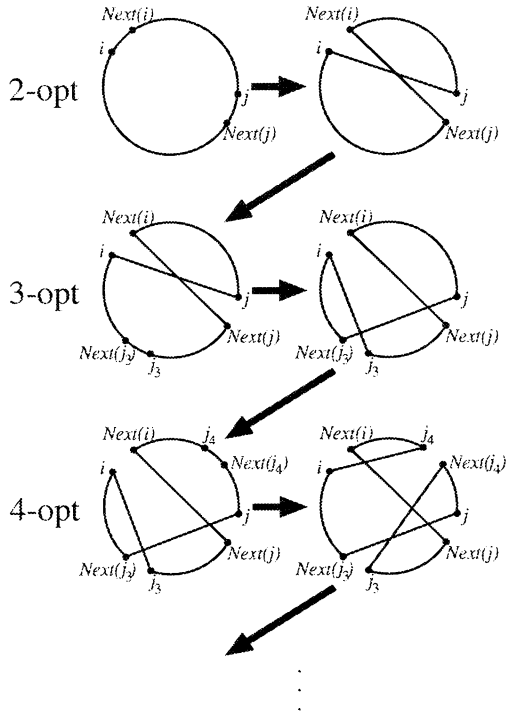
Figure 1: The k-opt method. The above examples are 2-opt, 3-opt and 4-opt. k-opt with larger k is realized by selecting other cities than already selected ones in similar procedure.

3 Tabu search neural network

In order to implement the above tabu search on a neural network, n neurons are prepared for an n-city TSP. The i th neuron corresponds to the city i. For a k-opt move selected for an update, tabu effects of the ith and the jth neuron of the corresponding k-opt move has to small. Furthermore, the move should offer largest gain for being selected as a real update.

Then, we construct the neural network as follows: the input to the i th neuron, $\xi_i(t)$, includes both the tabu effect of the jth neuron and the gain effect which is offered by the corresponding k-opt move. Here, this input, $\xi_i(t)$, is only from the best one for connecting with the city i, namely, the smaller tabu effect $\zeta_j(t)$ and the larger gain $\Delta_{ij}(t)$, where $\Delta_{ij}(t)$ is the total gain effect offered by the k-opt. In order to select such a best j, the city corresponding to the maximum of $\zeta_j(t) + \Delta_{ij}(t)$ is chosen. Then, this maximum value is applied to the i th neuron as the external input $\xi_i(t)$. In this neural network, if the i th neuron fires, the corresponding k-opt update is executed.

Then, the tabu search neural network based on cities can be realized by the following equations:

$$\xi_i(t+1) = \max_j \{\zeta_j(t+1) + \beta \Delta_{ij}(t)\}, \qquad (1)$$

$$\zeta_i(t+1) = -\alpha \sum_{d=0}^{s-1} k_r^d x_i(t-d). \qquad (2)$$

If $\{\xi_i(t+1) + \zeta_i(t+1)\}$ is the largest in all neurons, the corresponding move is actually executed. Namely, the k-opt update corresponding to $\Delta_{ij}(t)$ selected in Eq. (1) is applied.

By this neural network, if the i th neuron have fired in previous s iterations, this firing is avoided by the internal state $\zeta_i(t)$. Even though the tabu effect of the jth neuron which is selected in Eq. (1) is very small, it does not fire when the i th neuron has a strong tabu effect.

The tabu search in Sec. 2 can be realized with setting $\alpha \to \infty$, $k_r = 1$ and the size of memory s as the length of the tabu list.

4 Chaotic search based on the k-opt

The tabu search neural network in the previous section is extended to the chaotic version. Since it is already clear that chaotic dynamics can be realized by refractory effects and analog sigmoidal function [5], the output function of the tabu search neural network is replaced by the sigmoidal function and connections for detecting the maximum output are implemented in dynamics of the internal state $\eta_i(t)$. Then the chaotic neural network based on the tabu search memorizing cities can be realized by the following equations:

$$\xi_i(t+1) = \max_j \{\zeta_j(t+1) + \beta \Delta_{ij}(t)\}, \qquad (3)$$

$$\eta_i(t+1) = -W \sum_{k=1}^{N} x_k(t) + W, \qquad (4)$$

Table 1: The results of the Boltzmann machine (BM), the conventional tabu search (TS), the exponential tabu search (EX-TS) and the novel chaotic search (CS). Results are expressed in percentages of gaps between the obtained solutions and the optimum solutions. Numerals in italic indicate the best results. These results are on the case, $k = 2$.

problem	BM	TS	EX-TS	CS
KroA100	0.183	2.629	0.331	*0.059*
Lin105	0.562	2.199	0.200	*0.000*
KroA200	1.158	3.985	1.570	*0.767*
Lin318	2.211	5.120	2.705	*1.599*
Pcb1173	3.785	6.629	4.494	*2.605*
Pr2392	3.446	7.830	5.602	*3.329*

$$\zeta_i(t+1) = -\alpha \sum_{d=0}^{s-1} k_r^d x_i(t-d) + \theta, \qquad (5)$$

$$x_i(t+1) = f\{\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\}. \quad (6)$$

If $x_i(t+1) > \frac{1}{2}$, the k-opt update corresponding to the maximum of $\{\zeta_j(t+1) + \beta\Delta_{ij}(t)\}$ in Eq. (3), using the k-opt exchange.

For numerical calculation, Eq. (5) can be reduced to the following equations: if $t < s$,

$$\zeta_i(t+1) = k_r\zeta_i(t) - \alpha x_i(t) + R, \qquad (7)$$

otherwise,

$$\zeta_i(t+1) = k_r\zeta_i(t) - \alpha x_i(t) + R + \alpha k_r^s x_i(t-s), \quad (8)$$

where $R = \theta(1 - k_r)$.

In the original chaotic neural network [5], $s - 1 = t$, then Eq. (7) becomes the same as the original chaotic neuron model, and the chaotic dynamics can be realized by the above neural network. Since this chaotic searching method requires only n chaotic neurons for an n-city TSP, it is applicable to large TSPs.

5 Results

First, results of the proposed chaotic search (CS) is compared with stochastic dynamics which was realized by replacing each neuron with Boltzmann Machine neurons(BM), the conventional tabu search (TS) and the exponential tabu search (EX-TS) which reduces tabu effect exponentially [8]. Here, we introduce the 2-opt as a basic heuristic search. From the Table 1, it is clear that the chaotic neural network based method is the most effective for the TSPs. It should be noted that the solution of the BM is the second best but it requires huge computational time because many moves occurs in a single iteration by stochastic firings.
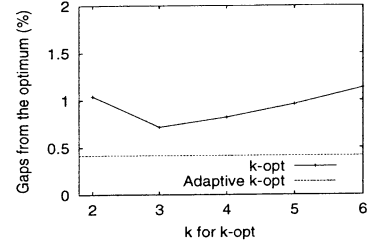
Figure 2: Results of the k-opt methods and the adaptive k-opt method, on a 200-city problem, KroA200 [10].
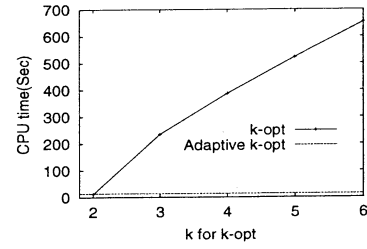
Figure 3: CPU time spent on the k-opt methods and the adaptive k-opt method, on a 200-city problem, KroA200 [10].

On the other hand, the computational amounts of the TS and the EX-TS are almost the same but EX-TS exhibits better than the conventional TS. The chaotic search is further extension of the EX-TS to chaotic version and it exhibits the best in these methods.

Then, the performances of the k-opt based chaotic searches are compared with changing the value of k. Fig. 2 shows gaps of average solutions from the known exact optimum of the problem. The problem introduced here is KroA200, which is a 200-city TSP in TSPLIB [10]. From Fig. 2, the performance of the 3-opt based chaotic search is better than the 2-opt based one. Here, k is completely fixed, so sometimes a 3-opt move much degrades the tour length that may be working efficiently for diversification. On the other hand, when we use further larger k, larger than 3, the performances get worse. This is because it becomes difficult to search small area close to some minimum in the searching space. Fig. 3 shows CPU time of each methods. From the Fig. 3, k-opt with larger k requires much larger computational time. This is because selections of the j_k are required many times, when we use larger k, even though we use very simple tabu search as a base of chaotic search.

As another method, we also propose an adaptive k-opt based chaotic search. The base of this method which we call adaptive k-opt is almost the same as the Lin-Kernighan method [11], which is one of the

Table 2: Results of the 2-opt based chaotic search, that with the DB and the adaptive k-opt based chaotic search with the DB. 5000 iterations are used for each method. Results are expressed by percentages of gaps between average obtained solutions and the best known solutions.

problem	size	2-opt	2-opt + DB	k-opt + DB
KroA100	100	0.2866	0.0699	0.0454
KroA200	200	0.5989	0.4847	0.4152
Pcb442	442	1.0336	0.9817	0.8246
Pcb1173	1173	1.6923	1.7478	1.5692
Pr2392	2392	1.9518	1.9995	1.8390
Rl5915	5915	2.3950	2.2733	1.7418
Rl11849	11849	2.2286	1.7295	1.1855

most effective heuristic algorithm for the TSP. In this method, k is not fixed, but increased while the positive gain crossovers exist. Although the computational amount of the k-opt with large k is too large, this adaptive k-opt will use large k only when the tour length can be reduced, so the computational time becomes much smaller.

In Figs. 2 and 3, results of the chaotic search with the adaptive k-opt are also shown. The adaptive k-opt applies many crossovers only when the tour lengths could be reduced. From the Fig. 2, the result of the adaptive k-opt is better than all k-opts with various k. Results in Fig. 3 show that CPU time is not large, because the k-opt with large k is required only when the tour length is reduced. From these results, we consider this method as the best chaotic search in this paper, and next further experiments are applied.

For easier applications to large TSPs, computational amount is reduced by only taking into account closer cities, on selecting j and j_l [9]. The parameter tuning method proposed in Ref. [8] is also applied, because it is very difficult to find the parameter values for realizing the best performances. Moreover, the chaotic annealing [8] is also applied for making convergence to better solutions. Furthermore, in order to make further improvement of the chaotic search, a kind of diversification is also applied. This is introduced in the chained Lin-Kernighan method [12], and is a kind of special 4-opt called double-bridge(DB). The DB is applied when the better solutions could not be found for more than 100 iterations.

Table 2 shows the results of the 2-opt based chaotic search which includes control and annealing [8], that with the DB [12] and the adaptive k-opt based chaotic search with the DB. In this experiment, we have applied only 5000 iterations for each calculation. The results in Table 2 are shown by percentage of gaps be-

tween average obtained solutions and the best known solutions. From the Table 2, it is clear that the results are much improved by using the adaptive k-opt and the DB. The gaps from the known optimum are almost 1% even for very large TSP such as 11849-city problem. By using longer iterations, solutions may be further improved. Moreover, it is possible to apply the proposed method to further large TSPs, since the computational costs are not large.

6 Conclusion

In this paper, we have applied the k-opt methods for the base of the chaotic search which have been shown to be effective for the 2-opt based methods [9]. First, the k-opt based tabu search is realized, and then it is extended to the chaotic neurodynamical search. Since the k-opt with larger k requires large computational amount, the adaptive k-opt is also applied that is almost the same algorithm as the Lin-Kernighan method which is one of the most effective algorithm for the TSP. By using the adaptive k-opt as a base of chaotic search, performance is improved, even for very large TSPs.

References

[1] H. Nozawa: Chaos, 2, 3, 377–386, 1992.

[2] T. Yamada and K. Aihara: Journal of Intelligent and Fuzzy Systems, 5, 1, 53–68, 1997.

[3] L. Chen and K. Aihara: Neural Networks, 8, 6, 915–930, 1995.

[4] M. Hasegawa et al.: Proc. of International Conference on Neural Networks, 3140–3145, 1995.

[5] K. Aihara et al.: Physics Letters A, 144, 333–340, 1990; K. Aihara: in *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems*, ed. H. Kawakami, 143–161, World Scientific, 1990.

[6] J. J. Hopfield and D. W. Tank: Biological Cybernetics, 52, 144–152, 1985.

[7] M. Hasegawa et al.: Physical Review Letters, 79, 12, 2344–2347, 1997.

[8] M. Hasegawa et al.: Control and Cybernetics, 29, 3, 773–788, 2000.

[9] M. Hasegawa et al.: Proc. of International Symposium on Nonlinear Theory and its Applications, 711, 1998.

[10] G. Reinelt, TSPLIB, available via Internet to http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

[11] S. Lin and B. Kernighan, Operations Research, 21, 498–516, 1973.

[12] D. Applegate et al.; Chained Lin-Kernighan for large traveling salesman problems, available from http://www.caam.rice.edu/keck/reports/chained_lk.ps,1999.

Cascade Process in the Transient Behavior of the "Game of Life"

S. Ninagawa
Division of Information and Computer Science
Kanazawa Institute of Technology
Nonoichi, Ishikawa, 921-8501

Abstract

We have investigated the relationship between the array size and the transient time steps from random initial configurations in the Game of Life. The simulations show that the average transient time steps $\langle T \rangle$ from random initial configurations increase logarithmically with square array size $N \times N$, $\langle T \rangle \sim \ln N$ in null and periodic boundary conditions. We measure the frequency distribution of cluster size in the evolution and we delete deliberately the smaller cluster in the middle of the evolution. As a result, we found that the larger clusters divide into to the smaller ones in the evolutions. This phenomenon is reminiscent of a cascade process of turbulence in viscous fluid which is the transfer of the kinetic energy of a flow from the large scale eddies to the smaller ones.

1 Introduction

The Game of Life [1] is one of the two-dimensional cellular automata introduced by J. H. Conway. Although the rule for the evolution is very simple, it generates complicated patterns such as glider which propagates infinitely until it is annihilated when it collides with another object on the array. Furthermore it is supposed that a universal computer can be constructed on the Game of Life by considering a glider as a pulse in a digital circuit. From the viewpoint of dynamical systems, the Game of Life is characterized by self-organized criticality [2] and 1/f fluctuation [3].

These properties of the Game of Life are closely related with initial configurations. The implementation of a universal computer needs carefully adjusted initial configurations. Self-organized criticality emerges at stable configurations and 1/f fluctuation is observed in the evolution from random initial configurations.

The array size needed for the universal computation is supposed to be enormous and it has not been implemented yet. The connection between array size and transient behavior caused by perturbing stable configurations has been made clear by the study of self-organized criticality[4]. But the dependence of the transient behavior from random initial configuration upon array size has not been investigated yet. The purpose of this paper is to elucidate the relationship between transient time steps and array size in the hope of clarifying the transient behavior of the Game of Life.

2 Game of Life

The Game of Life is a two-dimensional lattice system which evolves in discrete time steps. Each site takes on state 0 or state 1 at any one time step and is updated synchronously according to a rule.

Let $s_{x,y}(t)$ denote the state of the site at position (x, y) at time step t. The state of the site (x, y) evolves by the rule F,

$$s_{x,y}(t+1) = F(s_{x,y}(t), n_{x,y}(t)), \qquad (1)$$

where $n_{x,y}(t)$ denotes the sum of the states of the eight nearest neighboring sites around the site (x, y) at time step t. This kind of rule that the state of a site depends on the sum of the states of sites in its neighborhood and on the state of the site itself at the previous time step is called outer totalistic [5].

The rule of the Game of Life is defined by

$$F(0, 3) = 1,$$
$$F(1, 2) = 1, \quad F(1, 3) = 1,$$
$$otherwise \quad F = 0. \qquad (2)$$

3 Array size dependence

In the evolutions from random initial configurations in the Game of Life, the structures which remain after long time steps are almost always ones with time period two and three. So the detection of reaching into stationary state can be done by comparing each configuration with the configuration six time steps earlier.

The structures with higher time period are uncommonly remained. In that case the detection is done by using higher time steps.

The array consists of $N \times N$ cells and null (absorbing) boundary conditions under which sites beyond each end are modified to maintain state 0 and periodic boundary conditions are used. Each array is started from random initial configurations in which each site takes state 0 or state 1 randomly with independent equal probabilities. N is varied from 100 to 4000 with increments of 100 (with increments of 500 when N is more than 2000) and for each value of N, the evolutions from 150 distinct random initial configurations are sampled. The relationship between the average transient time steps $\langle T \rangle$ and the size of the side in square array N is plotted in Fig. 1.

The method of least-square by logarithmic function

$$\langle T \rangle = a + b \ln N, \tag{3}$$

is fitter than by power law

$$\langle T \rangle = cN^d, \tag{4}$$

where $a = -16894.4 \pm 551.7, b = 3711.4 \pm 78.8$ in null boundary conditions and $a = -14558.7 \pm 455.7$, $b = 3474.6 \pm 65.1$ in periodic boundary conditions.

4 Cluster size distribution

The simulations show that the average transient time steps increases logarithmically with the size of the side in square array. The most significant characteristic of the logarithmic function is expressed as follows.

$$\ln(MN) = \ln M + \ln N. \tag{5}$$

In this situation this equation means that the average transient time steps in array size $MN \times MN$ are the sum of the ones in array size $M \times M$ and in array size $N \times N$. So we can expect that the transient behavior can be divided into the one in the smaller arrays and the other in the larger arrays.

We measure the evolution of the frequency distribution of cluster size in square array to verify this expectation. Fig. 2 shows the frequency distribution of cluster size at time step $t = 10, 500, 4000$ on the evolution from a random initial configuration in a 1000×1000 array under periodic boundary conditions. The results show that the larger clusters disappear quickly and the smaller ones remain for a long time. Moreover we delete deliberately the smaller cluster in the middle of the evolution. Fig. 3 shows the frequency distribution
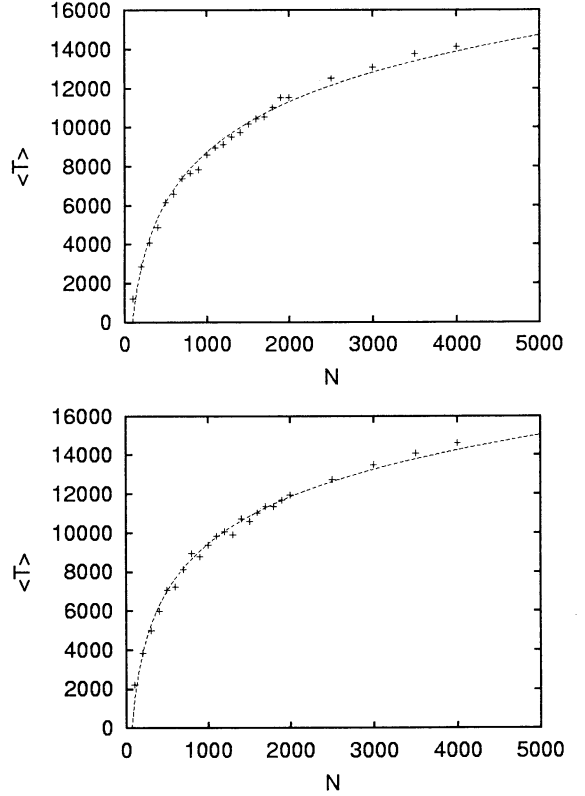
Figure 1: Average transient time steps $\langle T \rangle$ as a function of size of side N in square array in null boundary conditions (top) and periodic boundary conditions (bottom). The array size ranges from 100×100 to 4000×4000 with increments of 100 (with increments of 500 when N is more than 2000). For each array size, 150 distinct evolutions have been sampled. The dotted line represents the least-square fitting by logarithmic function $\langle T \rangle = a + b \ln N$ where $a = -16894.4 \pm 551.7$, $b = 3711.4 \pm 78.8$ in null boundary conditions and $a = -14558.7 \pm 455.7$, $b = 3474.6 \pm 65.1$ in periodic boundary conditions.

of cluster size at time step $t = 50$ when the clusters whose size is under 50 are deleted deliberately in the middle of the evolution and the frequency distribution of cluster size at next time step of the deletion. As a result, while the larger clusters disappear, the smaller ones are created at the next time step.

These results reveal that the larger clusters divide into to the smaller ones in the evolutions of the Game of Life. This phenomenon is reminiscent of a cascade process of turbulence in viscous fluid which is the transfer of the kinetic energy of a flow from the large scale eddies to the smaller ones.

5 Discussion

The Game of Life is classified into class IV cellular automata [3] in according to the phenomenological classification scheme for cellular automata proposed by Wolfram [6]. Wolfram's class IV cellular automata are thought to be capable of supporting universal computation and are characterized by their long transients. The relationship between transient length and array size in class IV cellular automata has been studied by some researchers. Langton [7] discovered that average transient length depends exponentially on array size in a class IV one-dimensional four-state five-neighbor cellular automaton. Li and Nordahl [8] reported that in the case of one-dimensional two-state three-neighbor cellular automaton rule 110 which has been thought to be classified into class 4, average transient length T increases with array size L, $T = L^{\alpha}$, $\alpha \approx 1.08$. Wolfram [9] pointed out that the Game of Life with $L \times L$ sites reaches steady state within L^2 time steps by Monte Carlo simulation On the contrary, our experiment shows that average transient time steps in the Game of Life depend on array size logarithmically. These results suggest that the transient time steps in class IV cellular automata are not identical but dependent on cellular automata themselves.

A different approach to the study of the dynamical behavior of the Game of Life is the investigation of self-organized criticality. Self-organized criticality was introduced by Bak et al. [10, 11] for the explanation of ubiquity of 1/f fluctuation and fractal structure in nature. Bak at al. [2] have shown that the Game of Life exhibits self-organized criticality for array sizes up to 150×150. After some debates [12, 13], it was pointed out [4] that the Game of Life is sub-critical which means criticality is observed only for small array size.
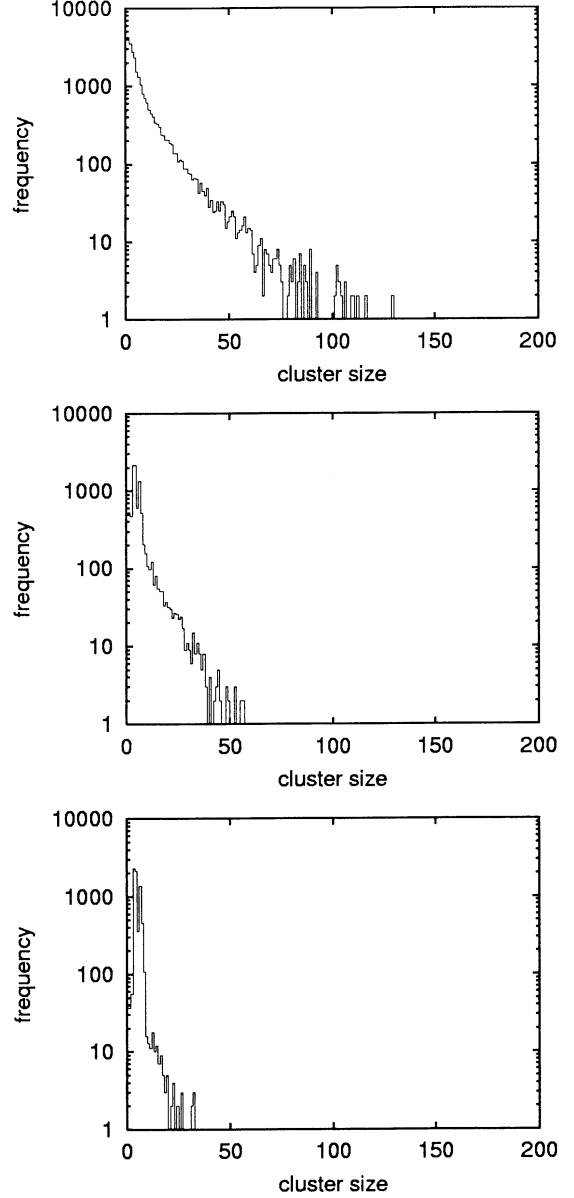
Figure 2: Semi-logarithmic plot of the frequency distribution of cluster size from a random initial configuration at time step t=10 (top), t=500 (middle), t=4000 (bottom). The array size is 1000×1000 and periodic boundary conditions are used. The evolution reaches into stationary state at t= 7412.
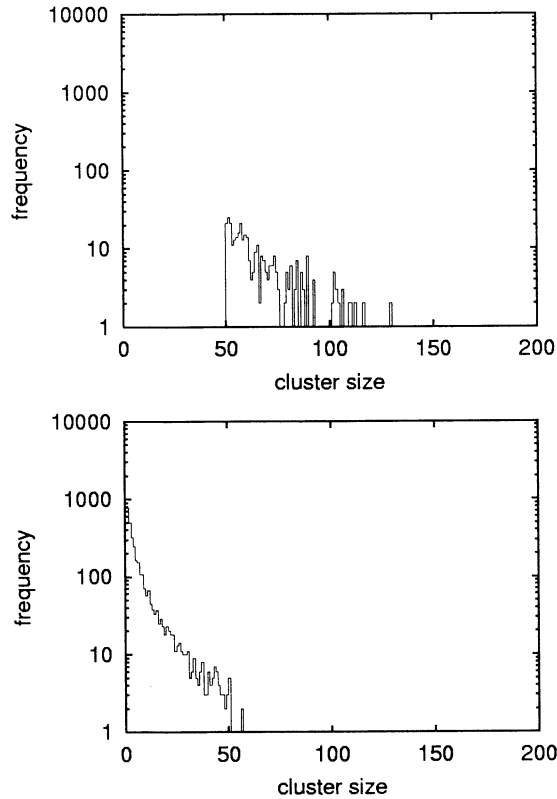
Figure 3: Semi-logarithmic plot of the frequency distribution of cluster size from the same random initial configuration as in fig.2 at time step t=10 when the clusters whose size is under 50 are deleted deliberately (top), and at next time step of the deletion (bottom). The array size is 1000×1000 and periodic boundary conditions are used.

Our result shows that the average transient time steps from random initial configurations increase logarithmically for array size up to 4000×4000. If the transient behavior from random initial configurations cause 1/f fluctuation in the Game of Life, we can suppose that the duration of 1/f fluctuation in the Game of Life will increase logarithmically according to array size.

References

[1] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning ways for your mathematical plays*, Vol.2 Academic Press, 1982.

[2] P. Bak, K. Chen, and M. Creutz, "Self-organized criticality in the 'Game of Life'," *Nature*, Vol. 342, pp.780-782, 1989.

[3] S. Ninagawa, M. Yoneda, and S. Hirose, "1/f fluctuation in the Game of Life," *Physica D*, Vol. 118, pp. 49-52, 1988.

[4] J. Hemmingsson, "Consistent results on 'Life'," *Physica D*, Vol. 80, pp.151-153, 1995.

[5] N. H. Packard, S. Wolfram, "Two-dimensional cellular automata," *J. Stat. Phys.*, Vol. 38, pp. 901-946, 1985.

[6] S. Wolfram, "Universality and complexity in cellular automata," *Physica D*, Vol. 10, pp. 1-35, 1984.

[7] C. G. Langton, "Computation at the edge of chaos: phase transitions and emergent computation," *Physica D*, Vol. 42, pp. 12-37, 1990.

[8] W. Li, M. G. Nordahl, "Transient behavior of cellular automaton rule 110," *Phys.Lett. A*, Vol. 166 pp. 335-339, 1992.

[9] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, Vol. 55, pp.601-644, 1983.

[10] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality: an explanation of 1/f noise," *Phys. Rev. Lett.*, Vol. 59, pp.381-384, 1987.

[11] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality," *Phys. Rev. A*, Vol. 38, pp.364-374, 1988.

[12] C. Bennet, M. S. Bourzutschky, "'Life' not critical?," *Nature*, Vol. 350, pp.468, 1991.

[13] P. Alstrøm, J. Leão, "Self-organized criticality in the "game of Life"," *Phys. Rev. E*, Vol. 49, pp.R2507-R2508, 1994.

Cellular Automata Modeling in Edge Recognition

Chen Yang
Dept. of Automation, Tsinghua Uinv.
Beijing, 100084, P .R. China
yangchen@proc.au.tsinghua.edu.cn

Hao Ye, G. Z. Wang
Dept. of Automation, Tsinghua Uinv.
Beijing, 100084, P .R. China
yehao@proc.au.tsinghua.edu.cn

Abstract

This paper focuses on parallel algorithms in image processing applications with cellular automata (CA) modeling. In the paper, we explore the possibilities of applying CA to edge recognition. Inspired by the satisfying result, we apply CA to detecting edge in a more complicated image. Simulation results are given.

Keywords: Cellular automata; image processing; edge detection; recognition

1. Introduction

As the demand for high speed of real time image processing is ever increasing, the need for parallel algorithms instead of sequential ones is becoming more and more important [1]. As an intrinsic parallel computational model, cellular automata (CA) can cater to this need. CA has already successfully used in many aspects such as fluid dynamics, biology, industrial application, population, traffic control, cryptography, environment modeling, finance, artificial vision and graphics, etc [2, 3, 9].

In the image processing, using Artificial Life (ALife), Kagawa [5] proposed a method for image segmentation based on autonomous agents, which is quite like region extraction in traditional image processing. Sahota [6] used a genetically controlled automaton model to tackle the general image processing problems including the edge detection by using the Genetic Algorithm, which could theoretically find out the precise cellular automaton function required to solve every given problem. But the doubt lies in that the needed computation would become enormously extensive when facing a complex situation. By the way, it cannot be proved whether only one function or a group of functions can complete the required image processing. On the other side, to recognize the follicle in ultrasonic images of women's ovaries, Viher [7] specified a transition rule of CA, which is so artificially given that it does not have generality.

In this paper, a new approach for edge recognition based on the combination of CA and traditional method of image processing is proposed, in which the concept of boundary operator [8] is used to represent the state of a cell and the local rule is defined based on priori knowledge. The method needs much less computation compared with [6] and has much more generality compared with [7]. Simulation results show its effectiveness.

2. Definition of CA

Let L be a regular lattice, elements of L cells, S a finite set of states, N a finite set (of size n=|N|) of neighborhood indices such that $\forall r \in L, \forall c \in N : r + c \in L$, and let $S^n \to S$ be a transition function. Then, we call the 4-tuple (L,S,N,F) a cellular automaton [7, 9].

A configuration $C_t : L \to S$ is a function that associates a state with each cell of the lattice. The effect of the transition function f is to change the configuration C_t into the new configuration C_{t+1} according to

$$C_{t+1}(r) = f(\{C_t(i)|i \in N(r)\}), \tag{1}$$

Where $N(r)$ denotes the set of neighbors of cell r,

$$N(r) = \{i \in L|r - i \in N\}, \tag{2}$$

3. Rectangle edge recognition based on cellular automata

3.1. Basic idea

In this section a new approach for edge recognition based on the combination of CA and the concept of boundary operator in traditional image processing is proposed. In the approach, we firstly translate an image into a CA. The basic state of each automaton is the color of the corresponding pixel. To represent the relationship between one automaton and its neighbors, the concept of the boundary operator is used to get a vector to represent the characteristic state of a cell. Defining the local rule by this vector, we can make the rectangle borders automatically recognized out of a binary image in which there are many similar objects.

Before we use CA to perform the recognition function, it is necessary to make several decisions in the construction of a specific CA and translate the problem into a CA model. The important decisions are the choices of lattice geometry, neighborhood size, boundary conditions, initial conditions, state set and transition rule.

It is natural to choose a 2-dimensional rectangle lattice of the same size as the image so that a one-to-one relationship between image pixels and CA cells could be established. In this paper, for neighborhood Moore's neighborhood is chosen. In addition, Reflective boundary conditions are used [9]. Naturally the pixel's color could be the cell's initial state. For a binary image, we use 0,1 to denote the black and the white respectively.

The transition rule is the most important aspect of a CA, which allows CA to perform the recognition tasks. The proposed method includes two phases as shown in Fig. 1, i.e. the characterizing phase and the evolving phase. In the characterizing phase, a vector is determined for each cell according to the initial state so that we can establish the transition rule more easily. In the second phase, the task of recognition is completed through the evolving of the cells according to the transition rule.
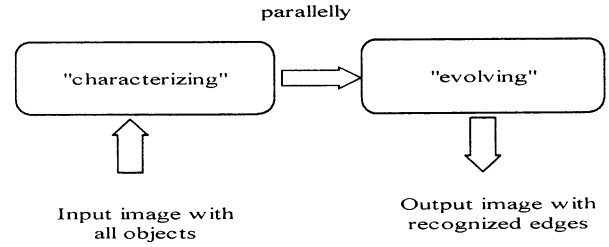
Fig. 1. Applying CA to edge recognition

3.2. Characterizing phase

After the initial input of the image, we begin to use the concept of boundary operator in traditional image processing to identify the characteristic vector of each cell. The procedure can be summarized as the following:

（1） For each cell, according to the template defined in Fig. 2, calculate the eight direction values corresponding to the eight location coordinates (1…8) shown in Fig. 3, using the traditional boundary operator algorithm [8];

（2） Find the maximum from the eight direction values of the cell, and use its corresponding coordinate to denote the characteristic vector. If the maximal direction value is less than 3, which means that the cell is not on the edge at all, we set the characteristic vector to be 16 to convenience the definition of the translation rule later.
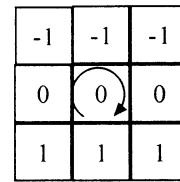
Fig. 2. Boundary operator template   Fig. 3. Eight location coordinates
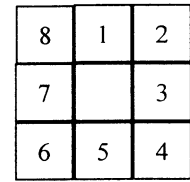
3.3. Evolving phase

To illustrate the procedure, let's observe the edge of an example rectangle in Fig. 4. It can be easily concluded that:

● If a cell is on the edge of an object, it must has exactly two other edge cells in its neighborhood;

● For an edge cell of a rectangle object, it must have

the following features: the difference between its characteristic vector and that of each of its two neighbors must be 1 or 7 (when the cell is at the corner because the angels of the corners of a rectangle are equal to 90 degree and the cell of the corner has a difference of 45 degree or 135 degree which can be translated into the difference of 1 or 7 in characteristic vector) or 0(when cells in the middle of a straight line of the edge).

1	1	1	1	1	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	1

4	5	5	6
3	16	16	7
3	16	16	7
2	1	1	8

Fig.4. An example rectangle and its characteristic vectors

Therefore, the transition rule can be given as the following:

For each cell:

(1) Find out all its neighbors, which satisfy that the difference between their characteristic vectors and that of the cell itself is 0,1or7. If the number of such neighbors is not equal to 2, then change its state into 16;

(2) Repeat (1) until the characteristic vectors do not change any more.

(3) Show the cells, which have characteristic vectors other than 16.

3.4. Simulation result and conclusions

Fig. 5 depicts an example of the recognition process based on CA. The first image is the initial state of the CA. The series of images following the first image show the evolving process of the CA. In the course of the experiment, every cell can change parallely. The local transition rule determines the final global structure. It can be seen that the edge of the two integrated rectangles are recognized out of the image.

4. Edge detection in a complicated image

4.1. Basic idea

It is natural for us to consider employing the idea in the above section on edge detection of a complicated
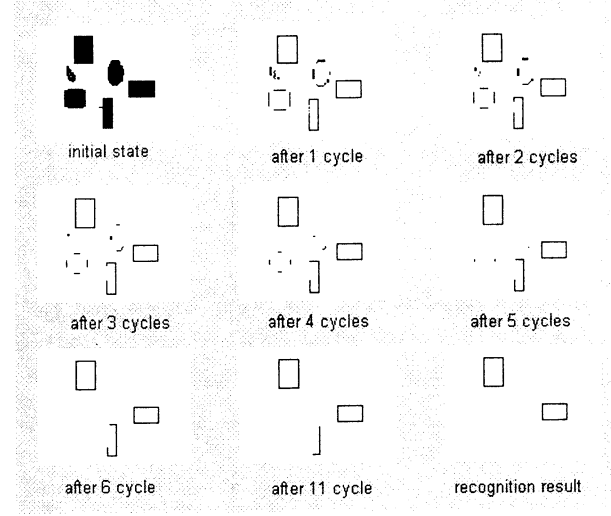
Fig. 5. The evolving process of the CA

image. In such an image, the color of a pixel has more possible values than 0 or 1. So we cannot precisely decide whether a cell is on the edge as in section 3. To solve the problem, when calculating the characteristic vector of each cell as introduced in section 3.2, we should record not only the coordinate corresponding to the maximum of the eight direction values, but also the maximal direction value itself. By the way, on a gradually color-changed edge, a cell may have more than two neighboring cells, which satisfy the conditions. But on the other side, we can think that if a cell has less than two neighboring edge cells, it must be not on a close edge and can be eliminated. By eliminating the cells not on edge, we can also get the edge.

4.2. Modifications in the two phases:

According the above idea, some modifications of the rules should be made in the two phases:

● Characterizing phase:
   After the initial input of a complicated image, we should also record the maximal direction value itself besides the characteristic vector;

● Evolving phase
   The transition rule can be given as the following:
   (1) If a cell does not have more than two cells in

its neighborhood which satisfy the following conditions:

$$|v(x_1) - v(x)| \le a_1 , \tag{3}$$

$$|v(x_1)| + |v(x)| \ge a_2 \tag{4}$$

$$|d(x_1) - d(x)| \le a_3 \tag{5}$$

Set the characteristic vector to be 16;

Where x denotes a cell, x_1 its neighbor, v(x) the maximal direction value of the cell x, d(x) the corresponding coordinate, a1,a2,a3 given threshold.

(2) Repeat (1) for enough cycles;

(3) Show the cells, which have states other than 16.

4.3. Simulation result

An example is used to compare the approach proposed in this paper with the traditional edge detection method [7]. The left and the right images in Fig. 6 show the results of edge detection of the image "cameraman" with the traditional method and the approach in this paper respectively. From Fig. 6, it can be seen that while keeping the edge generally integrate, the edge detection with CA can eliminate more noises and make the edge more precisely located.

Fig. 6. Edge detection
with traditional method and CA method

5. Conclusion

This paper demonstrates that CA is a potential tool for image processing. We have successfully applied it to edge recognition and edge detection in this paper. This method gives us a new point of view to describe and extract characteristics of images. The local transition

rules that guide CA during their execution can be easily adapted to the real situation. The parallel evolving process of CA makes it perfectly suited for execution on modern SIMD computer architectures. It can be put into a wide range of applications of image processing for recognition purposes if we could appropriately adapt the rules to the to-be-solved situations.

References

[1] Talia D, Sloot P, Cellular automata: promise and prospects in computational science, Future generation computer systems, Volume: 16, Issue: 2-3, December, 1999, pp. v-vii

[2] Succi S, Benzi R, Higuera F (1991), The lattice Boltzmann equation: A new tool for computational fluid dynamics, Physical D 47:219-230

[3] Burks A W (1970), Von Neumann's self-reproducing automata, in: Burks A W(ed), Essays on cellular Automata, University of Illinois Press, Champaign, IL, 1970,pp.3-74

[4] Gregorio S D, Serra R(1999), An empirical method for modeling and simulating some complex phenomena by cellular automata, Future Generation Computer System 16:259-271

[5] Kagawa H, Kinouchi M, Hagiwara M (1999), Image segmentation by artificial life approach using autonomous agents, International Joint Conference on Neural Networks, vol.6, pp 4413-4418

[6] Sahota P, Daemi M F, Elliman D G (1994), Training genetically evolving cellular automata for image processing, Proceedings of International Symposium on Speech, Image Processing and Neural Networks, Hong Kong, April 13-16,1994,vol.2, pp 753-756

[7] Viher B, Dobnikar A, Zazula D (1998), Cellular automata and follicle recognition problem and possibilities of using cellular automata for image recognition purposes, International Journal of Medical Informatics, Volume: 49, Issue: 2, April, 1998, pp. 231-241

[8] Zhang Yu Jin (2001), Tu Xiang Fen Ge, pp20-80

[9] Weimar J, Simulation with cellular automata, http://www.tu-bs.de/institute/WiR/weimar/ZAscript

A Performance Analysis of A Mobile Anti-virus System

Takeshi Okamoto   Yoshiteru Ishida

Department of Knowledge-based Information Engineering
Toyohashi University of Technology
1-1, Tempaku, Toyohashi, Aichi 441-8580, Japan
take4@sys.tutkie.tut.ac.jp   ishida@tutkie.tut.ac.jp

Abstract

We propose a mobile anti-virus system inspired by the immune system, and analyze the performance for detecting and removing viruses with the system by simulations. We show that the mobile anti-virus system depresses not only the virus population, but also the anti-virus population required for virus extinction in the network.

Keywords: computer virus, anti-virus, the immune system.

1 Introduction

The immune system can be considered as a security system for protecting our body from intruders such as pathogens. The immune cells, circulating around our body, detect and remove the pathogens. Inspired by the immune system, we propose a mobile anti-virus system. Similarly to the immune cells, each mobile anti-virus, which detects and removes viruses, circulates around a network. In this paper, a simulation is conducted to study the performance for detecting and removing viruses with the mobile anti-virus system, comparing it with the immobile anti-virus system.

There are some immune-based systems against computer viruses: a system learned from the self/non-self discrimination mechanism of the immune system [1], the digital immune system into which many features of the immune system are incorporated [2], and a distributed system to virus detection and neutralization similar to the antibody [3].

Circulating anti-viruses could give a whole network uniform resistance against viruses. It is expected that not only is the virus population depressed, but the anti-virus population required for virus extinction also decreases in the network. Further, the mobile anti-virus system does not need any central control systems, for each anti-virus is autonomous.

Section 2 describes a simulation model of the mobile anti-virus system. Section 3 presents the simulation results and discusses the performance. Section 4 summarizes the performance.

2 Simulation Model

We use an agent-based simulator. There are two kinds of agents: a virus agent and an anti-virus agent. Agents act autonomously in a virtual network. In simulations, each agent is sequentially processed within a unit time. The order of processing of the agents within a unit time is random.

2.1 Network Model

It may be reasonable to assume that an anti-virus only migrates to the computers whose owner permits the anti-virus to enter, whereas a virus infects computers without permission when the owner shares an infected media with someone. Also, the relations between users sharing media and permitting anti-virus migration are directional rather than non-directional.

Hence, we assume two networks for virus propagation and anti-virus migration. These two networks share the N nodes in common, but their arcs are different. The connection probability for a particular directional arc to be included in the network for virus propagation is δ_v. The connection probability of anti-virus migration is δ_{av} ($< \delta_v$). The type of graph is a directed random graph, due to its structural simplicity and the relative ease of analysis. The node represents a computer. Only one agent exists on a node (see Section 2.2 and 2.3 for detail).

Both the networks must be transitive closure[1] . (i.e. a simulation does not depend on the initial location of agents.) Our network is constructed by adding arcs to

---

[1] All nodes are reachable via one or more nodes [4].

a circle (node 1 → node 2 → ... → node N → node 1) to assume the transitive closure.

2.2 Virus Model

A computer virus is modeled as a virus agent on the agent-based simulator. There are three assumptions for the virus agent. First, the network for virus propagation is a directed random graph with the connection probability with δ_v. Secondly, the virus infects to all the computers directly connected by the network at the infection rate β per computer per unit time. But the virus does not infect the computers already infected, in fact that most of viruses do not infect at the same time. In short, only one virus exists on the computer. Thirdly, the virus is removed only when the anti-virus exists on the computer.

2.3 Anti-virus Model

An anti-virus which detects and removes viruses is modeled as an anti-virus agent on the agent-based simulator. There are three assumptions for a mobile anti-virus agent. First, the network for anti-virus migration is a directed random graph with the connection probability with δ_{av} $(< \delta_v)$. Secondly, the anti-virus detects and removes a virus in a computer, and prevents viruses from invading to the computer. Thirdly, the anti-virus migrates to a connected computer (randomly chosen among all the connected) with no anti-virus per unit time. Therefore, only one anti-virus or one virus exists on the computer.

We assume that the immobile anti-virus does not migrate. Hence, only computers with the immobile anti-virus are protected from viruses.

3 Simulations

We have conducted simulations to answer the following questions about the performance of the mobile anti-virus system; 1) how many viruses are decreased by the system; 2) how many anti-viruses are required to exterminate viruses; and 3) how many computers are infected.

Let the total infection rate at which a virus tries to infect computers per unit time be ν $(= \beta \times \delta_v \times N)$, and the fraction of computers with an anti-virus be σ. The maximum of total infection rate ν_{max} is equal to N.
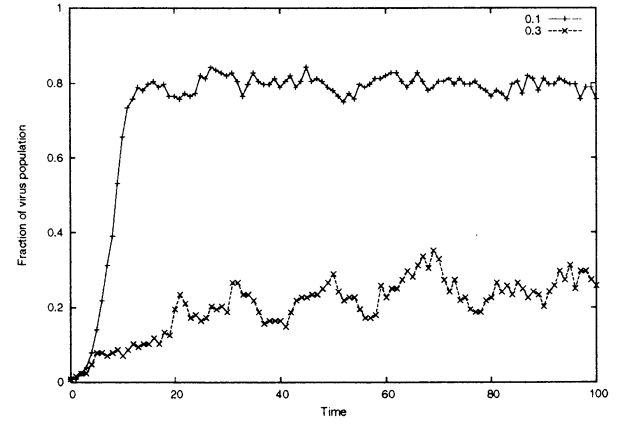
Figure 1: Comparison of fraction of virus population as a function of time. The simulation parameters are as follows: $\sigma = 0.1$ or 0.3, $\delta_v = 0.078125$, $\beta = 0.1$ (i.e. $\nu = 1.0$), the initial virus population is 1.

3.1 Evolution of Virus Population

We have performed simulations to investigate the fraction of the virus population in the total computers as a function of time. Figure 1 is a typical example of the simulation result for the following parameters: $N = 128$, $\sigma = 0.1$ or 0.3, $\delta_{av} = 0.04$, $\delta_v = 0.078125$, $\beta = 0.1$ (i.e. $\nu = 1.0$), the initial virus population is 1.

For $\sigma = 0.1$, the fraction of virus population increased exponentially at first and eventually converged on about 0.8. In equilibrium, the number of increased viruses is nearly the number of removed viruses. For $\sigma = 0.3$, the fraction of virus population seems not to converge. Many anti-viruses lead the virus population to decrease rapidly (sometimes reach 0), and hence, to increase the number of computers susceptible to a virus.

3.2 Equilibriums of Virus Population

We have investigated the equilibrium of virus population size varying σ. All other parameters are as given in Figure 1.

Figure 2 shows the average fraction of virus population in equilibrium for the mobile anti-virus system and the immobile anti-virus system. Each point represents an average over 500 simulation runs using the same parameters. As for the mobile anti-virus system, the average fraction decreased rapidly until it reached 0.0. As for the immobile anti-virus system, most of the computers without an anti-virus were infected for
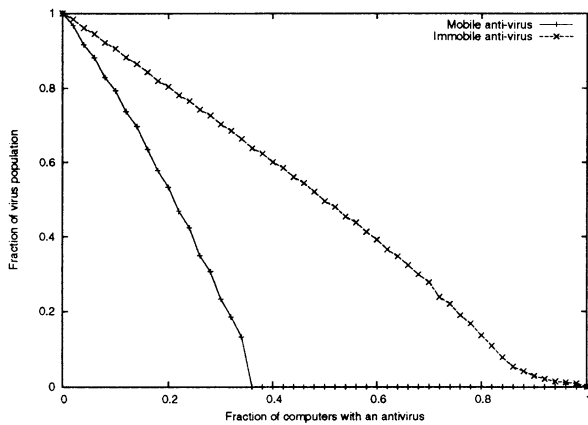
Figure 2: Average fraction of virus population in equilibrium vs. fraction of computers with an anti-virus. All parameters are the same as in Figure 1 except for σ. Each point represents an average over 500 simulation runs using the same parameters.

Figure 3: Average fraction of virus extinction vs. fraction of computers with an anti-virus. One simulation starts from only one computer infected, another starts from all the computers without an anti-virus infected. All other parameters are the same as in Figure 2.

$0.0 < \sigma < 0.6$. For $\sigma \geq 0.6$, the immobile anti-virus system protected computers without an anti-virus by segregating uninfected computers from infected computers.

We have also performed simulations for ν_{max}. The mobile anti-virus system exterminated viruses for $\sigma \geq 0.8$, but the computers without an anti-virus in the immobile anti-virus system were infected with viruses for $\sigma < 1.0$.

3.3 Virus Extinction

To investigate the mobile anti-virus population required for virus extinction, we have measured the fraction of virus extinction in 500 simulation runs using the same parameters. Figure 3 shows two simulation results: one simulation starts from only one computer infected; another starts from all the computers without an anti-virus infected. All other parameters are as given in Figure 2.

When only one computer was initially infected, the fraction of virus extinction increases with the increase of σ until viruses were exterminated at $\sigma = 0.38$. When all the computers without an anti-virus were initially infected, viruses did not go extinct for $\sigma < 0.3$, but viruses went extinct for $\sigma \geq 0.38$. This means that the fraction σ' of computers with an anti-virus required for virus extinction is independent of the initial virus population.

Next, we have investigated the dependence of σ'

upon the total infection rate by keeping $\beta = 1.0$ and varying δ_v and σ. All other parameters are the same as in Figure 3. The results are shown in Figure 4. With the increase of the total infection rate, σ' increased rapidly, but it converged on 0.8. In other words, any viruses certainly go extinct for $\sigma \geq 0.8$. This motivates us to improve the strategy of anti-virus migration in order to further decrease σ'.

3.4 Computers Infected More Than Once

To investigate how many computers are infected within a fixed time, we have performed simulation using the same parameters of Figure 2, and measured the fraction of computers infected more than once at the 500th unit time [2]. Figure 5 shows the average and maximum of the fraction of computers infected more than once over 500 simulation runs for the mobile anti-virus system and the immobile anti-virus system.

The average fraction with the mobile anti-virus system was lower than that with immobile anti-virus system for $0 < \sigma < 1$. The maximum fraction with the mobile anti-virus system was higher than that with immobile anti-virus system for $\sigma < 0.42$. In the worst case for the mobile anti-virus system, even if all viruses finally went extinct, all the computers were infected more than once.

We have also performed simulations for the maximum total infection rate ν_{max} for the mobile anti-

---

[2] The result did not change more than the 500th unit time.

-134-
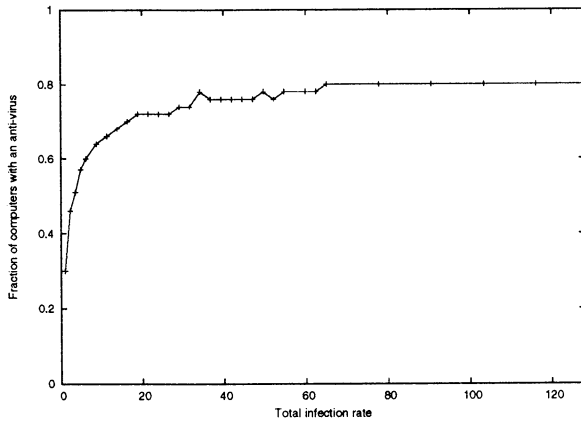
Figure 4: Fraction of computers with an anti-virus for virus extinction vs. total infection rate. The total infection rate varies with δ_v keeping $\beta = 1.0$. All other parameters are the same as in Figure 3.

Figure 5: Comparison between average fraction of computers more than once vs. fraction of computers with an anti-virus. All parameters is the same as in Figure 2.

virus system. Even if $\sigma = \sigma'$, all the computers were infected more than once. To prevent all the computers from being infected, the mobile anti-virus must be installed on all the computers.

4 Summary

We proposed the mobile anti-virus system inspired by the immune system, and conducted simulations to analyze the performance for detecting and removing viruses with the system. The performance are summarized as follows:

- The mobile anti-virus system depresses the virus population more than the immobile anti-virus system dose.

- Any viruses go extinct if the fraction of computers with the mobile anti-virus exceeds 80%.

- In the worst case for the mobile anti-virus system, however, all the computers are infected more than once, even if viruses eventually go extinct.

As for future problems, it is important to investigate various types of network for virus propagation, because the performance of the mobile anti-virus system is considered to be different from the type of network. Another interesting work is about a strategy of anti-virus migration. The improvement of the strategy may contribute to the decrease of the virus population.

Acknowledgements

References

[1] S. Forest, S. Hofmeyr, A. Somayaji, et el.(1994), Self–nonself discrimination in a computer, Proceedings of IEEE Symposium on Research in Security and Privacy, pp. 202–212.

[2] J. O. Kephart(1995), A biologically inspired immune system for computers, Proceedings of the 14th International Conference on Artificial Intelligence, pp. 20–25.

[3] T. Okamoto and Y. Ishida(2000), A distributed approach against computer viruses inspired by the immune system, IEICE Transactions on Communications, Vol. E83–B, No. 5, pp. 908–915.

[4] F. Cohen(1987), Computer viruses, theory and experiments, Proceedings of the 7th DOD/NBS Computer & Security Conference, pp. 22–35.

Development of Simulation System for Artificial Life

Norihito Tanaka
School of Science and Technology
MEIJI University
Tama-ku, Kawasaki, 214-8571, JAPAN

Hideaki Kanoh
School of Science and Technology
MEIJI University
Tama-ku, Kawasaki, 214- 8571, JAPAN

Abstract

This paper proposes an artificial life simulator of the cheap and light calculation load which researchers and students can lightheartedly use. In this system, the simulation is carried out using the concept of agent class and pool and as result drastically reduces the labor of the programming by the integrated operation function for groups.

1 Introduction

We propose a simulation system of artificial life which is optimum for the use of research and learning. It provides an original programming language which is easy to handle and has high-function.

Though the general-purpose system has very rich function, it requires the high spec and very expensive computer which students are not ready to use or the experiment.

We must say that there is the difficulty so that students may lightheartedly use it for experiments or learning etc.

As the general simulation systems are not made for the research of the artificial life field, the optimization of learning and programming is very time consuming task and the efficient development is difficult.

When we consider the genetic algorithm as an example, we must evaluate the arrangement which encodes the genetic information and next make ranking, processing of deletion and copying, then at last the operation of one time ends.

In this study, we develope a system with the following features:

1. It is a convenient descriptive language which can be used like the BASIC. The programming is object oriented one, it can treat complex number, matrix operation and class, furthermore communication between classes by which we easily construct multicellular system and system of the neural network. can be also carried out.

2. Of genetic algorithm and neural network, etc. The fundamental tool boxes for AI and AL field, such as genetic algorithm and neural network, etc., have standardly been built in. Since the function is offered in the above mentioned descriptive language, the user is possible to freely modify it.

3. It has a powerful graphics mechanism which can describe two, three-dimensional graphs and image display and also can edit graphics such as simple image processing, color tone conversions, paste of pictures. In addition, it is possible to easily carry out the animation processing.

4. As the design does not consume the resource of the computer, we can carry out experiment by a low spec personal computer.

5. By expansion kit of the plug-in style, the function can be easily expanded. The specification for the plug-in is opened to public, and it is possible that it can freely develop it even in whom and opens it again to public.

2 Concept of this system

The most part of the work in the genetic algorithm or in the neural network are design and management for data structure of large number of individuals and large number of nodes. In this system, the simulation is carried out using the concept of **agent class** and **pool** in order to omit such labor. For example, each individual in GA and each node in NN are the all agent classes. The agent indicates the individual having the autonomic judgment. Based on this fact, this system defines the agent class as an element with data and the function that manipulates the data. This is synonymous with the concept of the class in the object orientation. As it is possible that this agent class adds the dynamic change for the element of itself, for

example, it is also possible to dynamically change the evaluation function of the specific agent class in the midst of the simulation.

On the other hand, the pool is the environment where each agent class exists. All agent classes belong to the pool, and all operations for the agent class are carried out through the pool. Using the features of the data of the agent class, the pool can freely carry out extraction, sorting, change of the agent class and then can make integrated treatment for the **group** of the agent classes obtained, which is the feature of this system. We define the group as a set of agent classes and also define operations such as sum, difference, product, quotient, remainder, power, substitution.

As an example, we consider the of construction of GA by this system. It is necessary to manipulate ordering, crossover and mutation according to the evaluation function for the large number of agents (individuals). If we are going to program this problem using C++ language, we have to design the class expressing the agent and devise the sorting algorithm and spend the time for troublesome memory- management and image processing.

Concretely, this system takes the following procedure to express GA:
1. Design of the agent;
2. Design of the pool;
The programming is as follows:

```
class MyAgent          / Setting of agentclass
op variable Gene       / Data of agent
variable e             / Evaluation of criterion
function Evaluate      / Setting of criterion
end                    /
pool MyPool            /
MyAgent x 50 :         /
        MyAgents       / Num. of agents in Pool
function MyPool        / Starting of simulation
end                    /
function MyPool        / Def. external function
MyAgent x 50 :         /
        Temp           /
Temp = MyAgents        /
sort MyAgents.e        /
end
```

Using this example, we explain the concept of the operation of groups in detail. The agent class takes a linear list structure in each group and has the numerical value showing it's position in the internal variable 'Index' where the head position is the 0th and the number is kept in order. We define the sum of groups

as a sum of internal variables of agent classes with the same Index value and same name. In the above program, the qualifier 'op' is added to the variable X1. In this system, the variable declared with 'op' qualifier is only the object of the operation. Differences and products, substitutions etc. can be similarly defined. Note that in these operations, there is no necessity that the number of the member is same. As to substitution, there are two kinds of substitution. When two agent classes having element a,b,c with qualifier 'op' are described by A(a,b,c) and B(a,b,c), the usual substitution B=A means that the element of A is copied to the element of B having same name as A:

$$
\begin{aligned}
B-> a &= A-> a \\
B-> b &= A-> b \\
B-> c &= A-> c
\end{aligned}
$$

where the symbol $->$ indicates element.
To expaine another substitution using the notation of A(a,b,c), we express two sets of agent classes of m and n individuals that exist in some group X as follows:

$$\{A_0(a,b,c), A_1(a,b,c), \cdots, A_{m-1}(a,b,c)\}$$
$$\{B_0(a,b,c), B_1(a,b,c), \cdots, B_{n-1}(a,b,c)\}$$

Substitutin $A. = B$ means

$$
\begin{aligned}
B_0 -> a &= [A_0 -> a, A_1 -> a, A_2 -> a, ...A_{m-1} -> a] \\
B_0 -> b &= [A_0 -> b, A_1 -> b, A_2 -> b, ...A_{m-1} -> b] \\
B_0 -> c &= [A_0 -> c, A_1 -> c, A_2 -> c, ...A_{m-1} -> c] \\
B_1 -> a &= [A_0 -> a, A_1 -> a, A_2 -> a, ...A_{m-1} -> a] \\
B_1 -> b &= [A_0 -> b, A_1 -> b, A_2 -> b, ...A_{m-1} -> b] \\
B_1 -> c &= [A_0 -> c, A_1 -> c, A_2 -> c, ...A_{m-1} -> c] \\
&\vdots \\
B_{n-1} -> a &= [A_0 -> a, A_1 -> a, A_2 -> a, ...A_{m-1} -> a] \\
B_{n-1} -> b &= [A_0 -> b, A_1 -> b, A_2 -> b, ...A_{m-1} -> b] \\
B_{n-1} -> c &= [A_0 -> c, A_1 -> c, A_2 -> c, ...A_{m-1} -> c]
\end{aligned}
$$

As in this system all variables are treated as vectors, we can manage such substitution.

3 The application of the simulation system to real problems

3.1 Application to the neural network

As a simple application example, a perceptron of three layers shown in Fig.1 is considered.

Input layer Middle layer Output layer
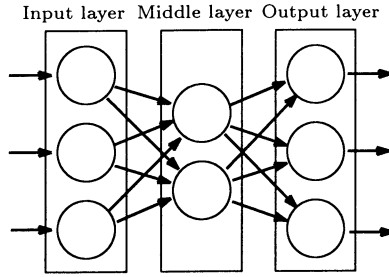
Fig.1 Simple perceptron

The first, second and third layers have three, two and three nodes respectively. By grouping the agent classes defined above, it is included in the pool. The programming is as follows:

```
class Node
    op variable Signal
    variable w1,w2,w3
    func Gate()
end
pool Par
    {Node x 3} : Layer1
    {Node x 2} : Layer2
    {Node x 3} : Layer3
    function Par()
end
function Par()
    variable i
    {Node x 3} : Input
    {Node x 3} : Teacher
    {Node x 3} : Output
    for i = 0 to 100 do
        Layer2 .= Layer1
        :
    next
end
```

where {} represents grouping and Node x 3 means that three instances of Node are formed.

For a realization of NN, it is necessary to link the node of each Layer. In this system, it can be easily done by the operation of the special substitution of group which is explained in chapter 2. It is possible that some layers can get the input from other layer only by the special substitution operation. By the integrated operation function for groups such as this function, the labor of the programming is drastically reduced, when this system is used.

3.2 The application to the genetic algorithm

There are 2 types of GA; one uses a binary value sequence as a gene, the other one uses an actual value vector as a gene. Here, we consider the application to Unimodal Normal Distribution Crossover (UNDX) that is a kind of the actual value GA. In this method, two individuals are selected by the adequate method from the population, and the the next generation is formed by the normal distribution rotational symmetry to the straight line connecting the two individuals. The individual retains a gene as a vector in the n-dimensional space. By the use of . operator, it is possible to access agent classes simultaneously which are included in the group elements having the same name. An example of programming for GA is given as follows:

```
class UNDXAgent
    op variable X
    op variable e
    function Evalute()
end
pool UNDXPool
    /UNDXAgents/
    UNDXAgent x 1000 :
    function UNDXPool()
    function Roulette()
    function Children()
end
function UNDXPool()
    UNDXAgent parent1
    UNDXAgent parent2
    UNDXAgent parent3
    rand UNDXAgents.X
    variable i
    for i = 0 to 1000 do
        UNDXAgents.Evalute()
        sort UNDXAgents.e
        parent1 = UNDXAgents[0]
        parent2 = Roulette(UNDXAgents)
        :
        UNDXAgents = Children(UNDXAgents)
    next
end
```

Roulette selection routine and other functions are provided as GA toolbox. This UNDX program is also offered as a toolbox.

4 Details of the function

4.1 Language division

The language used by this system has language structure resembling the BASIC. The common use mathematical functions such as log, ln, sin, cos, tan, sqrt, etc are provided. Main possible reserved words and functions used in this system are listed in the right field.

4.2 Agent analysis

For a realization of multiagent system, simultaneous analysis for time series of each agent are necessary. In the simultaneous analysis by the conventional system, the cue of the agent in loop structure is analyzed and has been evaluated in the order.

In this system, the node of the network is represented by a class in order to briefly express neural network and data correspondence between classes is also provided. Therefore, the signal can travel in the network architecture, if the programmer defines only node and network architecture.

This system is also correspondent again to the change of the dynamic class. In structure and class of C language, all necessary elements must be beforehand defined.

5 Conclusions

This paper proposes a simulation system of artificial life which is optimum for the use of research and learning. Introducing concepts of group, pool and integrated operation of groups, this system drastically reduce the labor of the programming by the integrated operation function for groups,

References

[1] J.D.Ulman,A.V.Aho,R.Sethi, "Compilers," Addison Wesley,1986.

[2] I.Ono,H.Satoh,S.Kobayashi,"A Real-Coded Genetic Algorithm for Function Optimization Using the Unimodal Normal Distribution Crossover,, Japanese Society for Artificial Intelligence, Vol.14,No.6,pp.1146-1155, 1999.

List of Commands

command	Function
inputfile	Reading from outside file
number	Declaration of variable at numerical value type (real number, imaginary number)
string	Declaration of variable at character string type
variable	Declaration of variable
global	Declaration of global variable
function	Declaration of function (The variable declared in a function is effective in the function)
entry	Special function showing the starting position of the program
end	End of the function
pool	Declaration of pool
class	Declaration of agent class
if equation elseif equation else endif	Conditional jump
switch case endsw	Case jump
goto label	Jump to label
gosub	Jump to subroutine
for next	Incremental repetition
while wend	Loop repetition
line(x1,y1,x2,y2)	Depiction of line
ellipse(x1,y1,x2, y2, start,end)	Depiction of ellipse
box(x1,y1,x2,y2)	Depiction of rectangle
keyin	Input from keyboard
lbutton	Detection of left mouse button

Emergent Function as rapid Changing the Foraging route formation by Ants Pheromone maker

Yasuo Yonezawa *and Hiroshi Matsuda
System Engineering Division, Graduate school of Science and Engineering
Ibaraki University
Nakanarusawa 4-12-1, Hitachi, 316-8511, Ibaraki, Japan
yonezawa@life01.dse.ibaraki.ac.jp

Abstract

In order to elucidate the group intelligence for the foraging behavior we are propose the reinforcement learning of Ants colony by pheromone maker. We are carried out that simulation of the foraging behavior of ants could be observed at macro-scale generated with micro-scale interacting among many respective ants. Results of this research have shown that Ants pheromone maker are produce the emergent function called reinforcement learning gave on rapid changing the foraging route formation in according to food collect efficiency.

Keywords — Collective intelligence, Ants colony ,Pheromone maker, Foraging route.

1 Introduction

Ants Colony Optimization discussed in this research investigate artificial system that take inspiration from behavior of real ant colony and which are used to solve discrete optimization problems. Recently, the ants colony optimization meta-heuristic has been defined at 1992(R.Becker et al[1]). Also, this new algorithm are already used various fields of some application technique(S.Goss et al[2], B.Holldobler[3]). But, the generation mechanism for emergent function of Ant colony optimization is not enough clear.

At previous our research (Y.Yonezawa and T.Tamiya[4],Y.Yonezawa and T.Kikuchi[5] and Y.Ka -sahara and Y.Yonezawa[6]), we are carried out the group intelligent models of Honeybee group by simulation in order to elucidate the generation of intelligence based on behavior rules for population control. These honeybee results indicated that the behavior rules of

*To whom correspondence should be address ; E-mail:yonezawa@life01.dse.ibaraki.ac.jp, FAX : +81-294-38-5208

individual honeybee into population in honeybee comb could be generated on group intelligence. And so, we are discovered the emergent properties generated with transmission behavior in honeybee comb transmission are described as bellows :

(A)formation of food position and amounts by honeybee comb transmission are generated the rapid decision of foraging action. One the other hand, Real ants are capable of finding shortest from a food source to nest without using visual cues. Also, they are capable of adapting to changes in the environment, for example finding a new shortest route once the old one is no longer feasible to new obstacle. It well known that main means used by ants to form and maintain the route is a pheromone trail. Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction high concentration in pheromone rather than lower one(R.Becker et al[1],S.Goss et al[2], B.Holldobler[3]).

This elementary behavior of real ants can be used to explain how they find the shortest route, which reconnects a broken line after the sudden appearance of an unexpected obstacle, has interrupted the initial route (M.Dorigo et al[7], M.Dorigo and L.Gambardella[8], M.Dorigo et al[9]). It is interesting to note that those ants which chooses, by changes, the shorter path around the obstacle will more rapidly reconstitute the interrupted pheromone trail compared to those which choose the longer route. Hence, the shorter route will received a high concentration pheromone in the time unit and this in turn causes a higher number of ants to choose the shorter route. Due to this positive feed back(auto catalytic)process, very soon all the ants will choose the shorter route. (A.Colorni et al[10],R.Schoonderwoerd et al[11], R.Schoonderwoerd et al[12]).

In order to elucidate the generation mechanism of the rapid changing the foraging route formation

as emergent, we are proposed the detail simulation of several environments at function under these ant pheromone makers as group communication.

2 Computational experiment of Ant system

In this section, we presented the preparation of ant system used by this research as described as bellows(Shown in Figure 1).
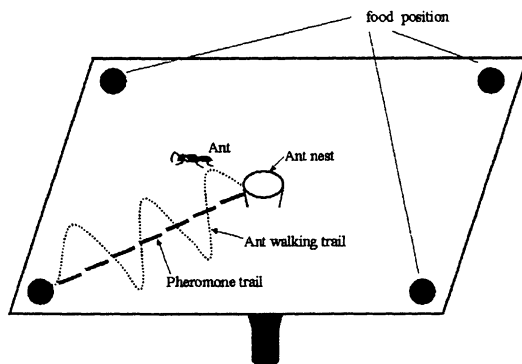
Figure 1: Outline of Ant-Pheromone system.

As shown in Figure 1, Ants working area of Ant system are developed with lattice Map. Detail of each action of ants are described as bellows(Section 2.2). This computational experiment is consisted with very simple action of each ant into ant populations. (Total Ant population is 5000 ants.)

2.1 Pheromone Maker of Ant systems

Our model, nest pheromone and inducement pheromone are used for maker communication. This maker communication has generally knowledge with Animals and Social Insect in biological Science. The inducement pheromone works as landmark toward bait sites. The ants are attached to strong inducement pheromone in the air space, and a trace trail of inducement pheromone secreted on the working ground. Then, strength of volatilizes pheromone in the airspace and strength of trail ground is depending times. The inducement pheromone secreted on the working ground by ant volatizes gradually(in according to time course), and diffuse widely(shown in Figure 2).

2.2 Primitive Behavior rules of Ants

In this ants system, the behavior as total Action of Ant is determined to increase the foraging efficiency. Ant behavior consisted with action rules are describe as fellows convents.

a) 1st Food search action

The Ant in Food search is carried out random walk.

And also, if it taken on misses at food position, it changes its action to the transport on the return back procedure, respectively.

b) Return back Action

In case of food harvest, Ant returns back Action with Pheromone secretions to nest.
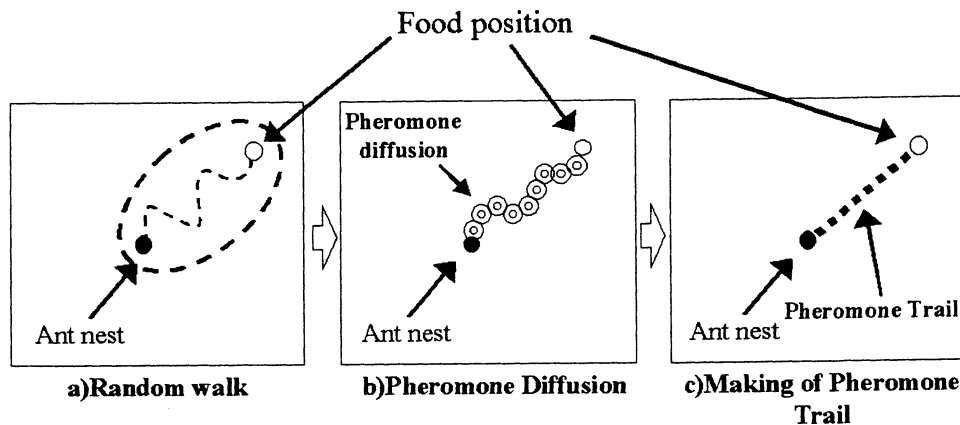
Figure 2: Pheromone maker procedure.

c) This action is started with first food collection at food position. After the food harvest, harvested ants are come back to nest with food transportation. In case of come back to nest for food transportation, Ants are secreted on the Pheromone at came back route toward the ants nest. If case of 2nd food harvest at some food position, food harvest ants are re-secreted on the pheromone at 1st trail route. By repeated the these action of pheromone secretion according to food transportation from same food position, Pheromone concentrations under walking trail are increasing in according to repeat numbers (Pheromone Maker=0.5/action time).

d) 2nd food search procedure.

The ants of 2nd food search procedure are moved to stronger pheromone secreted position. If it detected the pheromone trail, it done its action to the trace action, against Pheromone. The ants in the trace action in according to pheromone trail to the corresponding food position. But, if it has on misses food at the food position, it change at action to the transportation on the return back action, respectively. In case of the misses the pheromone trail, it change to 1st food search action.

e) Action rule depended food volume of food position

Food volume of same food position are decreased for the food harvest by foods transportation ants numbers in according to pheromone tail concentration. Finally, Food volume at same position is missing of foods. In the empty food position, Ants action is not able to return back to ant nest. In case of this condition, ants are exchange at 1st Food search action. By this case of ants action, the food transportation is not repeated at same food position. Therefore, the concentration of Pheromone tail are decreased with diffusion at air space (Diffusion rate =0.2/action time). Instead of these phenomena, not food harvested ants are search of new food positions by random walk and the sensing (Sensing ability=0.6) of another pheromone trail.

3 Results of Computational Experiment

Concentrated food harvest trail to a single most rich food position is organized (Shown as Figure 3).

These computational experiments are carried out used by cellar automaton under Lattice map. (Map Size: 500 * 500). In shown at Figure 3, the ants behavior are indicated the concentration in according to food volume of each food position. Namely, Ant system consisted with simple action rule is generated on the intelligence phenomena as the determination problem of optimal food harvest route. These phenomena are called as emergent properties generated with the movement at simultaneously of several simple rules.

These phenomena may be generated with the interaction of several ants actions. This is very important content for the intelligence generation of ant system. But, Generative procedures of these emergent phenomena are not yet clear as actions arrangements.
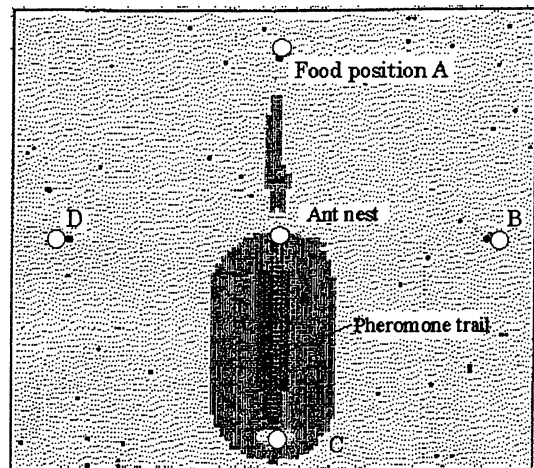
Figure 3: Concentrated food harvest trail. Food volume of food position A, B, C and D is described as following. A 50, B 30, C 90, D 30

4 Discussion and conclusion

In previous research, we well know Social Insectants, Bees, Termites and wasps-exhibit collective problem solving ability. In particular, several ant's species are capable of selecting the shortest route, among a set of alternative routes, from their nest to a food source.

Ants deploy a pheromone trail (or chemical trail) as they walk; this trail attracts other ants to take the route that has the most pheromone. This reinforcement process results in the selection of shortest route: the first ants coming back to the nest are those that shortest took shortest route twice, so that more is

present on the shortest route than on longer routes immediately after these ants have returned, stimulating nest mates to choose the shortest route. It is interesting to note that those ants' which choose, by chance, the shorten path around the obstacle will more rapidly reconstitute the interrupted pheromone will received a higher amount of pheromone in the time ants and tile will in turn curse a higher number of ants to choose the shorten path. In this paper, these contents gave with ant behavior in according to pheromone make sure with our research results. As future work, a future possibility is that exploration fanction in the system with a more random transition probability function.

Acknowledgements

References

[1] R.Becker., J.L.Deneubourg and S.Goss., Journal of Theoretical Biology,(1992) 157,pp579-581

[2] S.Goss., S.Aron., J.L.Deneubourg and J.M.Pasteels., Naturwissenschafen,(1989),76,pp579-581.

[3] B.Holldobler and E.O.Wilson, The ants,Springer-Verlag(Berlin), (1980)

[4] Y.Yonezawa and T.Tamiya(1996)Proceedings of Alife V.,pp163-168.

[5] Y.Yonezawa and T.Kikuchi(1996) Proc IEEE 7th International Symposium on Micro Machine and Human Science,pp247-256.

[6] Y.Kasahara and Y.Yonezawa.(1996),Proc of 6th Parallel computing workshop,P1-B-1-B-7.

[7] H M.Botee and E.Bonabeau(1998),Adv.Complex Systems,1,pp149-159.

[8] M.Dorigo., V.Maniezzo and A.Colorni(1996),IEEE Transactions on systems,Man and Cybernetics-Part B,26(1) pp29-41.

[9] M.Dorigo., L.Gambardella(1997),IEEE Transaction of Evolutionary Computation,1,pp55-66.

[10] M.Dorigo., Di Caro.G and L.Gambardella(1999), Alife,5,pp137-172.

[11] A.Colorni.,M.Dorigo and V.Maniezoo(1991), Proceedings of ECAL'91.,Elsevir,pp 134-142.

[12] R.Schoonderwoerd., O.Holland., J.Bruten and L.Rothkrantz(1997),Adaptive behavior,5(2),pp 169-207.

[13] R.Schoonderwoerd., O.Holland, and J.Bruten(19-97), Proceedings od Agent'97,(ACM:Marina del Rey,CA)pp 209-216.

A method to correct the distortion of the projector using an artificial life type of function discovery system

Kazuki YAMASHITA*, Seiichi SERIKAWA**, and Teruo SHIMOMURA***
Dept. of Electrical Eng., Kyushu Institute of Technology 1 -1, Sensui-cho, Tobata-ku,
Kitakyushu, Fukuoka 804 -8550 Japan,
kazu@elcs.kyutech.ac.jp *, serikawa@elcs.kyutech.ac.jp**
simomura@elcs.kyutech.ac.jp ***

Abstract In a projection system, the shape of a figure on the screen greatly depends on the position of the projector. If the projector is not set up at the correct position, the projected figure is distorted. Therefore, when the projector is not set up at the correct position, it is necessary to correct the shape of the figure. Thus a method for correcting the distorted figure is proposed. This method uses an artificial life type function discovery system. A conversion function, which shows the relationship between the figure on the display of the computer and that of the projected screen, is automatically calculated by the use of S -System. As the result, the shape of the figure on the screen can be displayed without distortion, r egardless of the projector-position and the direction.
Key words: artificial life, genetic programming, projector, figure distortion

1. Introduction

The chance to present with a liquid crystal projector has, recently, increased. T he shape of a figure on the screen greatly depends on the position of the projector. If the projector is not set up at the correct position, the projected figure is distorted. Therefore, when the projector is not set up at the correct position, it is necessary to correct the shape of the figure. However, it is difficult to correct it uniquely, as the shape of a figure differs greatly according to the position of the projector. Thus a method for correcting the distorted figure is proposed.

J. Koza[1] first proposed the fu nction discovery system based on Genetic Programming (GP). However, the system has some disadvantages. We thus proposed a bug type of artificial life based system to overcome these problems [2][3][4]. We called the system as S -System to indicate the incorporation of the concept of sexual and asexual reproduction in it. In this system, some input data x and output data y are given. The approximate function, which represents the relationship between them, is automatically found by the "S-system" using the process of evolution. We utilize the system. A conversion function, which shows the relationship between the figure on the display of the computer and that of the projected screen, is automatically calculated by the system. As the result, the shape of the figure on the screen can be displayed without distortion, regardless of the projector -position and the direction.

2. Method to correct figure distortion
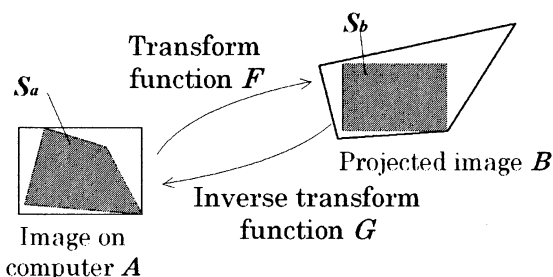
Fig.1 The relationship between the image on the display of computer A and the projected image to the screen B.

Figure 1 shows the relationship between the image on the display of computer A and the projected image to the screen B. In our investigation, we consider a system where a personal computer is connected with a projector, and the figure in the display of the computer is projected onto the screen (Fig. 2). Let the figure on the display of the computer be denoted by A and the projected figure be denoted by B. Regarding the transformation of the figure as a function conversion, we define the conversion function from A to B as F and from B to A as G. Considering an example of displaying a square S_B on the screen, if the function G is already known, the figure S_A on the display of the computer can be obtained by converting S_B to S_A using G. Therefore, by displaying S_A on the display of the computer, S_B can be correctly projected onto the screen. In a similar way, if the co nversion function G is decided, an arbitrary figure can be correctly displayed on the screen.
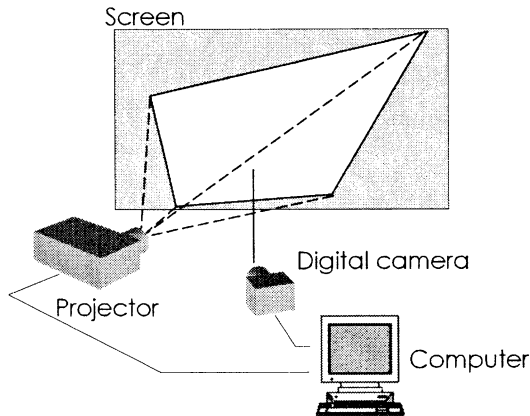
Fig. 2 The structure in this system.

Fig. 3 Projected reference points.

vertically and horizontally. The 16 points are projected on the screen (Fig. 3). We regard the points on the computer screen as output data y and the ones on the projected screen as input data x. A digital camera is used to acquire the points x on the screen. The function G that expresses the relationship between x and y is found by using the S-system.

Figure 4 shows the flow chart to correct the figure distortion. It is summarized as follows.

1. A standard image is projected to the screen. In the image, 16 points are arranged in equal intervals both vertically and horizontally. We regard the positions of the points as output data y.
2. The projected image is captured into the personal computer with a digital camera from the front position. One of the images is demonstrated in Fig. 3.
3. The 16 points are extracted from the captured image as shown in Fig. 3.
4. The each position is calculated. We regard the position as input data x.
5. The function f where represents the relationship between x and y is found by using the artificial life type of discovery system (S-System).
6. The shape of the image is transformed by the obtained function f.
7. The transformed image is displayed on the display of the personal computer. As the result, the figure without the distortion is projected onto the screen.

Fig. 4 The flow chart to correct the figure distortion.

3. System structure

For finding the conversion function G, we adopt the function discovery system called "S-system" that uses a bug type of artificial life . In this system, some input data x and output data y are given. The approximate function, which represents the relationship between them, is automatically found by the "S-system" using the process of evolution. To find G, we display 16 reference points on the display of the computer. The points are arranged in equal intervals both

4. Algorithm of function -discovery

The outline of the function -discovery system called S-System is mentioned here.

Fig. 5 Flowchart (left):

Start
(1) Generation of initial bugs
(2) Gene = 0
(3) $Energy_p = 0$ ($p=1,...,$Pop)
(4) Bug $p = 1$
(5) Movement of bug p
(6) Capture of data of bug p
(7) Find the function ? — Y → End
N
$p = p+1$
(4) $p=$Pop ? — N → Gene=Gene+1
Y
(8) $Gene=Gene_{MAX}$? — N → Descendant generation routine (9)
Y
End
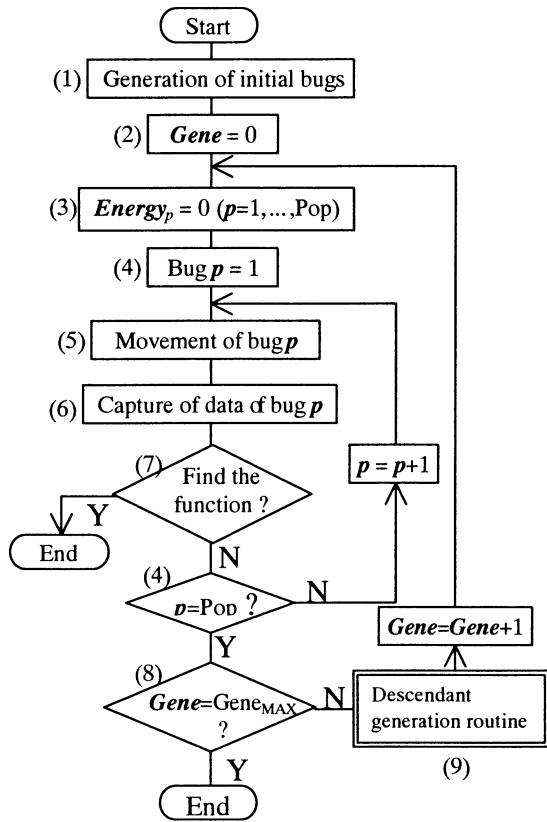
Fig. 5   The flowchart of the algorithm of function -discovery using the bug type of artificia l life.

Fig. 6 Flowchart (right):

SUB
(a) Generation gap Bug No. **Par**
(b) $p = 0$
(c) Selection of bug by tournament strategy
(d) Does the bug sexually reproduce ? — N → $p=p+2$
Y
(e) Sexual reproduction
(f) Asexual reproduction
(g) Mutation
(h) $p=(Pop-Par)$? — N
Y
RETURN

Fig. 6 The flowchart of descendant -generation routine.

4.1 Main routine

Figure 5 is the flowchart of the algorithm of function -discovery by the use of a bug type of A-life proposed by us[2]. The flow is summarized as follows.

(1) Numerous bugs with the arbitrary function are generated at random. The number **Pop** is selected from the numerous bugs in order of high fitness.

(2) The generation **Gene** of the bug is set to 0.

(3) The value of the internal energy, $energy_p$, of all the bugs is initialized to 0.

(4) The procedures from (5) to (7) are repeated for all the bugs; the bug number ranges from 1 to **Pop**.

(5) The bug p moves. This means that the values of constants K in the chromosome change slightly. That is to say, the values of K are replaced by $K + dK$, where dK is the small change of K , $\overline{K} = (K_1, K_2, \cdots, K_n)$, $\overline{dK} = (dK_1, dK_2, \cdots, dK_n)$ and n is the number of constants in the chromosome. This concept is based on Ref. (4). The details are given in

Ref. (2).

(6) The bug p catches the observation data (i.e., fitness fit_p of bug p is calculated from the observation data).

(7) In the case that fitness fit_p reaches the threshold fitness Fit_{TH}, this algorithm ends. This means a bug has discovered the function f.

(8) The algorithm ends when the current generation **Gene** reaches the maximum Generation $Gene_{max}$.

(9) After the descendant -generation-routine is called, **Gene** is added to 1 and the algorithm returns to procedure (3).

4.2 Descendant -generation -routine

The flowchart is displayed in Fig. 6, and is summarized as follows.

(a) Based on the generation -gap, **Par** bugs are selected and they are passed down to the next generation. The elite strategy is adopted for the generation -gap.

(b) The bug number p is set to be 0. By the repetition of the following procedures from (c) to (g), **Pop - Par** bugs are generated.

(c) A bug is selected by the tournament strategy.

(d) The selected bug is judged whether it has the
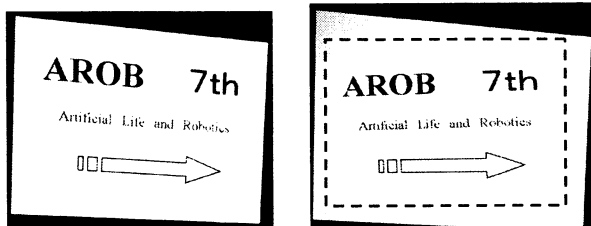
(a) before correction   (b) after correction

Fig. 7 An example of correct of image -distortion (I).

(a) before correction   (b) after correction

Fig. 8 An example of correct of image -distortion (II).

ability to sexually reproduce. In the case that the selected bug has the ability of sexual reproduction, procedure (e) is performed. In the other case, procedure (f) is carried out.

(e) The bug finds its partner, and they produce two children by crossover. Jump to procedure (g).

(f) Two children are produced by asexual reproduction.

(g) A part of the chromosome is changed by mutation at the rate of R_{mut}.

Thus, the descendants of the number of *Pop* are generated. For the details sexual/asexual reproduction and mutation , see Ref. (2).

In this study, we regard the points on the computer screen as output data *y* and the ones on the projected screen as input data *x*. The function *G* that expresses the relationship between *x* and *y* is estimated by using the S -system.

5. Application to correct image distortion

To confirm the validity of this method, we projected some figures to a screen from various directions, and the convert function *G* is calculated by S-System. By the use of *G*, the figures are corrected. The corrected figures are also projected to the screen. The examples are shown in Figs. 7 and 8. As understood form these figures, the distorted figures are successfully corrected in this method.

6. Conclusions

A method for correcting the distorted figure is proposed. This method utilizes an artificial life type function discovery system. As the result, the shape of the figure on the screen can be displayed without distortion, regardless of the projector-position and the direction.

References

(1) J. Koza, *Genetic Programming II*, *Auto Discovery of Reusable Subprograms*, MIT Press, p.109 (1994).

(2) S. Serikawa and T. Shimomura, "Proposal of a System of Function-Discovery Using a Bug Type of Artificial Life", *Trans. IEE*, Vol.118-C, 2, p.170, (1998).

(3) S. Serikawa and T. Shimomura,"Improvement of the Search Ability of S -System (A Function-Discovery System)", *Trans. IEE*, Vol.120-C, 8, *in Press*, (2000).

(4) T. Shimmomura, K.Yamashita and S.Serikawa "Function discovery system model using non-linear optimization method ", *Proc. Of 6th Symp. On Artificial Life and Robotics (AROB 6th '01)*, pp.321-324 (2001).

(5) Iba, H., Higuchi T., de Garis, H, and Sato, T., "Evolutionary Learning Strategy using Bug-Based Search", *Proc. of the 13 Int. Joint Conf. on Artificial Intelligence (IJCAI-93)*, vol.2, pp.960-966 (1993).

(6) H. Okumura, *Algorithm dictionary using C language*, Gijutsu-hyoron, p.260, 1991, in Japanese.

Correlation Functions in Nonlinear Neural Networks

Naohiro Ishii, Mayumi Nakamura and Mikiya Ohta

Department of Intelligence and Computer Science
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan
{ishii@ics.nitech.ac.jp}

Abstract

In the biological visual neural networks, one of the prominent features, is nonlinear functions, which play important roles in the visual system. First, we will analyze the structure and the function of the nonlinear asymmetric networks in the visual system. Second, sensory integration is realized in the nonlinear neural networks of which functions are based on the auto and cross cornelations between neural pathways. Finally, the integration of the information, is discussed as the optimization problem in the neural networks.

keywords nonlinear analysis, correlation functions, nonlinear neural network

1 Introduction

The visual information is inputted first in the retina of the visual pathway of the biological network. Reichardt [1] described that the auto correlation is a principle for the evaluation of sensory information in the central neural system. Retinal ganglion cells produce two types of response: linear and linear. The nonlinear responses are generated by a separate and independent nonlinear pathway. The nonlinear pathway is composed of a sandwich model in filters. The nonlinear characteristic shows excellent visual processing, to respond the moving objects. It is very useful and important to clarify the structure and the function of the linear and the nonlinear pathway in the biological neural network systems. In this paper, we show the auto and cross correlations are impotrant in the sensory functions of the neural networks. We discuss here the integration of the information as the optimization problem in the neural network.

2 Correlation Computations

2.1 Correlation Computations in Neural Network

In the biological neural networks, the structures of the network, are closely related to the functions of the network. The network suggests the biological function of the organism. Reichardt evaluated the sensory information by the correlations in the neural networks. In this paper, correlation computation are suggested in the catfish retinal neural networks in Fig.1. The functions of the asymmetric network are realized in Fig.1. The first and the second order kernels com-
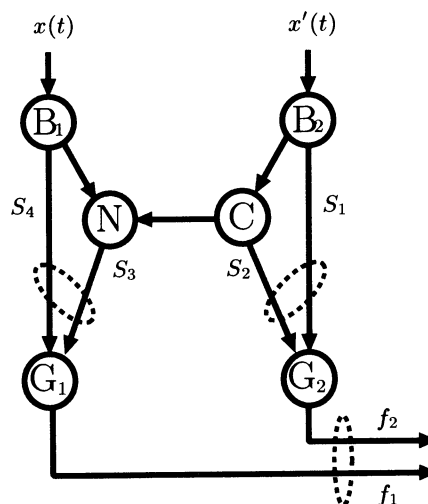
Fig 1: Correlation Computations in Biological Neural Network

putations, are carried out by computing correlations before and after cells **G** in Fig.1. We assume here temporal correlations with two steps. In the first step of the time the first order kernel on the linear pathway, is approximately computed among B_1, **N** and G_1 cells,

since these cells have linear functions, while **C** cell has a nonlinear function with quadratic property. The alphabets S_1, S_2, S_3 and S_4 in Fig.1 denote outputs of cells. From the correlation computations between S_1 and S_2 in Fig.1, the approximate values of the first order kernel $C_{11}(\lambda)$, auto-correlation and those of the second order kernel $C_{21}(\lambda_1, \lambda_2)$, cross-correlation are carried out. The dotted lines in Fig.1 indicate the correlation computations. Thus, the approximate values of the first order kernel, which are shown by $C_{11}(\lambda)$, are computed in the following,

$$C_{11}(\lambda) \quad \propto \quad h'_1(\lambda)$$

In the second step of the time, similary, the approximate values, shown by $C_{21}(\lambda_1, \lambda_2)$, are computed

$$C_{21}(\lambda_1, \lambda_2) \quad \propto \quad \alpha^2 h''_1(\lambda_1) h''_1(\lambda_2)$$

, where the second step of the time, means that two inputs $x(t - \tau_1)$ and $x(t - \tau_2)$ are assumed on the pathway from B_1 to G_1 comparing with the delay of the pathway from B_1 to G_1 via N. Similary, From the correlation computations between S_3 and S_4 in Fig.1, the approximate values of the first and the second kernels, $C_{12}(\lambda)$ and $C_{22}(\lambda_1, \lambda_2)$, are carried out. Then the approximate values of the first and the second order kernels, are shown in the network of B_2, **C** and G_2 cells,

$$C_{12}(\lambda) \quad \propto \quad 0$$

$$C_{22}(\lambda_1, \lambda_2) \quad \propto \quad h''_1(\lambda_1) h''_1(\lambda_2)$$

in the first and the second steps of the time. The approximate first order kernel $C_{12}(\lambda)$ shows the movement vector equation, which means the movement from the left to the right. The movement vector equation can be computed from the correlations of the outputs of G_1 and G_2 cells, which shows the approximate the first order kernel $C_{12}(\lambda)$. In Fig.1, only the motion ratio α is computed from the correlations. The movement vector equation can be computed in the upper network of Fig.1. The alphabets, f_1 and f_2, denote outputs of Ganglion cells G_1 and G_2, respectively. The output correlation, shown as $f_1 * f_2$, becomes

$$\begin{aligned} f_1 * f_2 \quad &\propto \quad (S_4 + S_3) * (S_2 + S_1) \\ &= \quad S_4 * S_2 + S_4 * S_1 + S_3 * (S_2 + S_1) \end{aligned}$$

Since S_3 has the nonlinear and the linear components, we set $S_3 = S'_3 + S'_4$. Thus, the above right equation of $f_1 * f_2$ becomes,

$$S_p * S_1 + S'_4 * S_1 + S'_3 * S_2$$

Since $S'_4 * S_1 \simeq S_4 * S_1$ and $S'_3 * S_2$ is the double-quadratic characters, ($\simeq$ correlations between S_1 and S_2), the following equation holds,

$$\begin{aligned} f_1 * f_2 \quad &- \quad \text{correlations between } S_1 \text{ and } S_2 \\ &\simeq \quad C_{12}(\lambda) \end{aligned}$$

, which shows that the movement vector equation can be computed in the upper network of the retina. Thus the network structure in Fig.1, may suggest and reflect the functions of the neural networks.

2.2 Stimulus Effects in the Structures

The cell **C** in Fig.1 plays an important role in nonlinear function as squaring of output of bipolar cell B_2. The mean luminance of the stimulus effects to the cell **C**, thus it changes the structure of the network. The change of mean luminance of the stimulus, effects on the nonlinear pathway, which is often observed in the experiment data. Then the quadrature nonlinear characteristics of the cell **C**, become $(x - b)^2$, which is assumed to be x^2. The structure of the network is changed as shown in Fig.2. The dotted line in Fig.2, shows the linear pathway from the cell B_2.
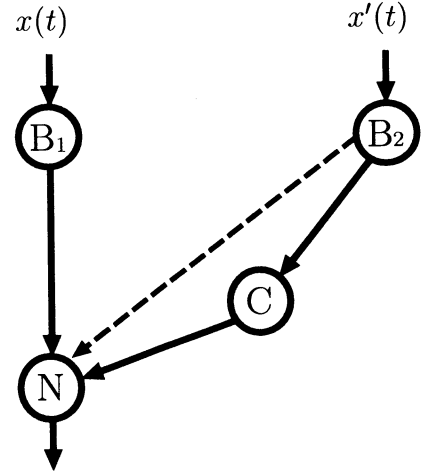
Fig 2: Asymmetric Network by the Mean Stimulus Shift

The nonlinear analysis in Fig.2, becomes

$$C_{11}(\lambda) = h'_1(\lambda) + b\alpha h''_1(\lambda)$$

$$C_{21}(\lambda_1, \lambda_2) = \alpha^2 h''_1(\lambda_1) h''_1(\lambda_2)$$

$$C_{12}(\lambda) = \frac{\alpha}{(\alpha^2 + k\beta^2)} h'_1(\lambda) + b h''_1(\lambda)$$

$$C_{22}(\lambda_1, \lambda_2) = h_1''(\lambda_1)h_1''(\lambda_2)$$

The α equation is computed from $C_{21}(\lambda_1, \lambda_2)$ and $C_{22}(\lambda_1, \lambda_2)$. The parameter b, which shows the degree of the dotted linear components, becomes

$$b = \frac{C_{12}(\lambda) - \frac{\alpha}{(\alpha^2+k\beta^2)}C_{11}(\lambda)}{\sqrt{C_{22}(\lambda_1,\lambda_1)} - \frac{\alpha}{(\alpha^2+k\beta^2)}\sqrt{C_{21}(\lambda_1,\lambda_1)}}$$

This parameter b, plays on important role in the learning of the movement, shown in the later section.

2.3 Cross Spatial Interaction Network

The cross spatial interaction network in Fig.3, is developed from the network in Fig.2. The cell $\mathbf{C'}$ has the function of the cross product, which is a kind of the nonlinear function. To clarify the ability of the movement detection and the learning, the following equations are derived.
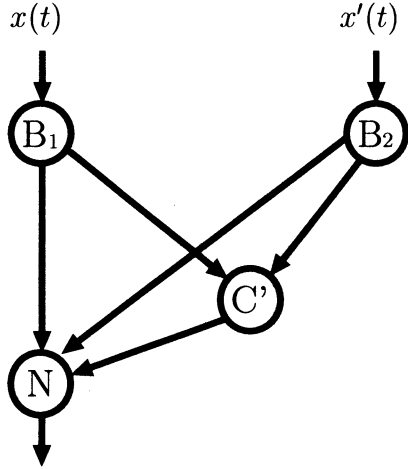
Fig 3: Asymmetric Neural Network for Cross Spatial Interaction

$$C_{11}(\lambda) = h_1'(\lambda) + \alpha h_1''(\lambda)$$

$$C_{21}(\lambda_1, \lambda_2) = \alpha h_1'(\lambda_1)h_1''(\lambda_2)$$

$$C_{12}(\lambda) = \frac{\alpha}{(\alpha^2 + k\beta^2)}h_1'(\lambda) + h_1''(\lambda)$$

$$C_{22}(\lambda_1, \lambda_2) = \frac{\alpha}{2(\alpha^2 + k\beta^2)}\{h_1'(\lambda_1)h_1''(\lambda_2)+h_1'(\lambda_2)h_1''(\lambda_1)\}$$

From the third and the fourth equations,

$$\alpha^2 + k(1 - \alpha)^2 = \frac{C_{21}(\lambda_1, \lambda_2) + C_{22}(\lambda_1, \lambda_2)}{2C_{22}(\lambda_1, \lambda_2)}$$

is obtained. The α-movement equation is derived from this equation. Thus, the movement vector equation is also derived from the first and the second equation. In the cross spatial interaction network in Fig.3, detection equations are not simple as those of the network.

The learning ability is shown in the network in Fig.3, since the network realizes the XOR problem. Here the movement $x(t) \longrightarrow x'(t)$ is given by the following stimulus:

$$+ \longrightarrow +, \quad + \longrightarrow -, \quad - \longrightarrow +, \quad \cdots.$$

The learning problem is shown whether stimulus movements $(+ \longrightarrow +)$ and $(+ \longrightarrow -)$ are perceived as the same class, while those $(- \longrightarrow -)$ and $(- \longrightarrow +)$ are perceived as another class, or not.

This learning problem is reduced to the XOR problem for the case of movements stimulus. Then, the network in Fig.3, has powerful ability for the movement stimulus. But the asymmetric network cannot classify the movements stimulus, since $\mathbf{C}$ cell on the nonlinear pathway has only squaring function. The squaring function cannot discrimate the stimulus $+$ and $-$ from $\mathbf{B_2}$ cell. Thus, the movement information $x'(t)$ may be needed as shown in the dotted line in Fig.2, for the learning. However, the XOR problem is not solved in the network of Fig.2.

Linear pathways on the left and the right, play an important role in the network. The inputs $(+)$, $(-)$ which are two valued, are changed to analogue values $(+1)$ and (-1) respectively, in the network. The input variables for the movements XOR problem, are transformed to the output variables of $\mathbf{G_1}$ and $\mathbf{G_2}$ cells, which are linearly separable. Thus the learning is realized by the upper cell using $\mathbf{G_1}$ and $\mathbf{G_2}$ outputs. From the cell $\mathbf{N}$ to the cell $\mathbf{G_1}$, an inhibitory weight transforms to a separable movements classes.

3 Optimization in the Neural Network

The conventional penalty method in the optimization problem using neural networks, is difficult in the with care constraints. In this study, a continuous gradient projection method is applied to the problem with linear constraints in the neural network. It is shown that the optimum solution of the constrained problem is obtained effectively by this method.

3.1 Continuous Gradient Projection Method

In this chapter, a proposed method is based on Tanabe's Continuous gradient projection method [4], in

the following.

$$\min \quad f(x)$$
$$\text{subject to} \quad g_j(x) = 0 \quad (j = 1, \cdots, m)$$

The optimization problem is described in the following.

$$L(x, \lambda) \equiv f(x) - \lambda^T g(x)$$

Lagrange function of this optimization problem is made to be $L(x, \lambda) \equiv f(x) - \lambda^T g(x)$. Here, $\lambda \equiv (\lambda_1, \lambda, \cdots, \lambda_m)^t \in R^m$ is the Lagrange multiplier. $\lambda^* \in R^m$ exists for the optimum solution x^*, and (x^*, λ^*) satisfies the Karus-Kuhn-Tucker condition, namely the simultaneous nonlinear equation on variable x, λ as shown in the following.

$$\begin{cases} \nabla_x L(x, \lambda) &= \nabla_x f(x) - J(x)^T \lambda &= 0 \\ \nabla_\lambda L(x, \lambda) &= g(x) &= 0 \end{cases}$$

$J(x)$ is a Jacobian matrix of $m \times n$ of $g(x)$ in the following, and the rank is m.

$$J(x) = \nabla g(x) = \begin{pmatrix} \frac{\partial g_1(x)}{\partial x_1} & \cdots & \frac{\partial g_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(x)}{\partial x_1} & \cdots & \frac{\partial g_m(x)}{\partial x_n} \end{pmatrix}$$

In short, it is necessary to solve this simultaneous equation in order to solve the optimization problem with constraints. The simultaneous equation is obtained, when the Neutonian method is applied in this simultaneous equation.

$$\begin{bmatrix} I_n & -J(x_k)^T \\ J(x_k) & 0 \end{bmatrix} \begin{bmatrix} dx_k \\ \lambda_{k+1} \end{bmatrix} = - \begin{bmatrix} \nabla_x f(x) \\ g(x) \end{bmatrix}$$

dx_k, λ_{k+1} will be computed as following.

$$dx_k = -(I_n - J_k^+ J(x_k)) \nabla f(x_k) - J_k^+ g(x_k)$$
$$\lambda_{k+1} = (J_k^+)^T \nabla f(x_k) - (J(x_k) J(x_k)^T)^{-1} g(x_k)$$

, where, $J_k^+ \equiv J(x_k)^T (J(x_k) J(x_k)^T)^{-1}$ is the Moore-Penrose pseudo-inverse of $J(x_k)$. It can be expected that x_k, λ_k is converged to the optimization values by equations.

3.2 Application to the Neural Network

In this chapter, the solution to the optimization problem with linear constraints applied to the Hopfield model on the basis of equation dx_k, is explained. The optimization problem with linear constraints is shown in the following.

$$\min \quad f(x)$$
$$\text{subject to} \quad Ax - b = 0$$
$$0 \le x_i \le 1 \quad (i = 1, \cdots, n)$$

,where x shows n-dimensional variable vector, A on $m \times n$ coefficient matrix, b with the m dimension constant vector. From dx_k and the above equation, the following equation is obtained.

$$dx = -(I_n - A^+ A) \nabla f(x) - A^+ (Ax - b)$$

$A^+ = A^T (AA^T)^{-1}$ is Moore-Penrose pseudo-inverse of matrix A. In this equation, a function P is introduced so that the value of variable x may not exceed region $[0, 1]^n$.

$$dx = P \left\{ -(I_n - A^+ A) \nabla f(x) - A^+ (Ax - b) \right\}$$

Here, matrix P is a diagonal matrix of the following equation.

$$P = \begin{pmatrix} x_1(1 - x_1) & & 0 \\ & \ddots & \\ 0 & & x_n(1 - x_n) \end{pmatrix}$$

4 Conclusion

Movement perception is a fundamental problem in the neural network. This study discusses the movement detection and learning in the biological neural networks. Correlations play an important role in the function of the neural networks which make an effects on the detection and the learning in the neural networks. Finally, we discussed the integration of the information as the optimization problem in the neural networks.

References

[1] Reichard, W., "Autocorrelation, A principle for the evaluation of sensory information by the central nervous system", Rosenblith Edition, Wiley, New York, 1961.

[2] Ishii, N. and Naka, K.-I., "Function of Biological Asymmetrical Neural Networks", Lecture Notes in Computer Science 1240, Springer, pp.1115-1125, 1997.

[3] Ohta, M., Ishii, N. et. al., "Optimization with Linear Constraints in the Neural Networks", Lecture Notes in Computer Science 2084, Springer, pp.561-569, 2001.

[4] Tanabe, K., "Geometric Method in Nonliner Programming", Journal of Optimization Theory and Applications, Vol. 30, No. 2, pp. 181-210, 1980.

Dynamics of a Recurrent Neural Network Acquired through the Learning of a Context-based Attention Task

Katsunari Shibata & Masanori Sugisaka

Dept. of Electrical & Electronics Engineering, Oita University, 700 Dannoharu, Oita 870-1192, Japan.

shibata@cc.oita-u.ac.jp

Abstract

A context-based attention task is employed in this paper. An Elman-type recurrent neural network is utilized to extract and keep the context information, and only the reinforcement signal that indicates whether the answer is correct or not is given. Through this learning, the function of an associative memory is observed in the Elman-type neural network. Adaptive formation of the basins are examined by varying the learning conditions.

Keywords: attention, associative memory, reinforcement learning, recurrent neural network, adaptive basin formation

1 Introduction

Selective Attention is one of the important research items for robots in the real world. There are huge pieces of information, and necessary part of them should be extracted. Some context information is often utilized to suggest which part is necessary. Further, the context information itself should be extracted from the past series of information and be kept until the time when the context information is utilized.

Zipser showed that associative memory, in other words, fixed-point convergence dynamics can be obtained through learning in an Elman-type recurrent network with one hidden unit[2]. The hidden unit becomes equal to one input signal when the other signal is activated, and otherwise the hidden unit keep its value like a flipflop.

The authors showed that when there are more than one input signals and more than one hidden units, the recurrent network forms some basins, and each of the basins corresponds to one of the categories required in the task[3]. The coding of the category is decided through the learning, while in the conventional associative memory, the coding has been decided by the designer.

In this paper, the network dynamics obtained through learning is observed in detail. Furthermore, the result is reported to examine how flexible the basins are formed.

2 Context-Based Attention Task

Fig. 1 shows the attention task and system employed in this paper. One arrow pattern whose direction is one of four corners is presented at first as the left half of Fig. 1. After some while, another pattern, which consists of 4 small sub-patterns, is presented on the same visual sensor as the right half of Fig. 1. Then, the system is required to classify the sub-pattern at the corner where the first presented arrow pattern pointed. The sub-pattern can be classified into one of three categories.

The arrow direction presented at first can be "upper right", "upper left", "lower right" or "lower left". The size of the original arrow image is $7 \times 7 = 49$. As a noise, one pixel value is inverted randomly with the ratio of one-half. The visual sensor consists of $5 \times 5 = 25$ visual cells, and one part of the original arrow image is cropped. So totally $3 \times 3 = 9$ patterns can be presented for each arrow direction if the noise is not added. Considering the noise, $9 \times (25 + 1) = 234$ patterns can be presented for each arrow direction. 5×5 image is put into Elman-type recurrent neural network. The input signal is -1.0 for the white pixel, and 1.0 for the black one. The time is set as $t = 0$.

At a randomly selected time from 5 to 14, which is denoted by T, a pattern that consists of 4 small sub-patterns is presented on the same visual sensor. The size of the sub-pattern is 3×3, and it can be "square", "cross", or "plus". Since the sensor size is 5×5, the sub-patterns are overlapped with each other at the middle row and column. In such areas, the sensor signal is the average value of the overlapped pixels.

There are three output units, each of which is corresponding to each sub-pattern. The answer of the system is decided according to the probability that is proportional to the sum of the output and 0.5. The output function of each unit in the network is sigmoid function whose value range is -0.5 to 0.5 except for the input layer. When the answer is correct, the system obtains a reward 1.0, otherwise it obtains a penalty -1.0 as a reinforcement signal r. The output corresponding to the answer is trained to be $0.4 \times r$. The other outputs are trained to be -0.4 when the answer is correct, and otherwise they are not trained. This
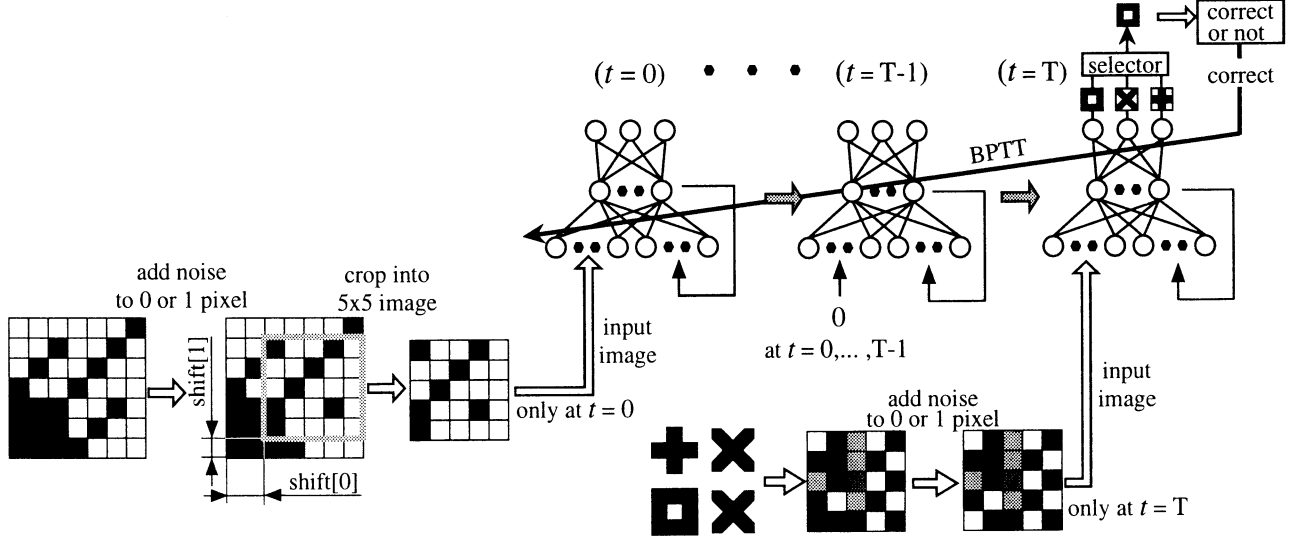
Figure 1: The flow of the context-based Attention task employed in this paper.

means that the system cannot know the correct answer directly when the answer is not correct.

At every time step except $t = 0$ or $t = T$, all the input signals are 0.0. The number of the hidden units is 20, and the values of the hidden units are 0.0 at $t = 0$. The initial weight values are 0.0 for the hidden-output connections, and are decided randomly from -1.0 to 1.0 for the input-hidden connections. As for the feedback connection of the hidden-hidden connections, the weight value is 4.0 for the self-feedback connections, and 0.0 for the others. The reason why the self-feedback connection weight is set to be 4.0 is that the maximum derivative of the output function is 0.25 around input= 0.0, and the error signal goes backward effectively through time without diverging because $0.25 \times 4.0 = 1.0$.

When the mutually-connected NN is utilized for an associative memory, the connection weights are always symmetrical, because the network dynamics always becomes fixed-point convergence when the weights are symmetrical. Hebb learning that is often employed for the learning of associative memory cannot realize asymmetrical connections. However, here, no such constraint is given in advance.

3 Simulation Result

Some simulation results after 1000000 trials of learning are shown in this section. The sequence from the presentation of the arrow pattern to the answer and learning is defined as one trial. If the maximum output supposed to be the answer, a wrong answer appears about once per 10000 trials. Depending on the initial connection weight values in the NN, it sometimes fails to learn.

At the next, the context extraction and associative

memory function is observed. The first presented patterns should be classified into one of the 4 categories, because only the direction of the arrow pattern is necessary to give an attention to the second presented pattern. As mentioned above, totally 234 patterns can be taken as one category. Here, the distance between two patterns in a layer is defined as the sum of the absolute value of the difference of each unit.

Fig. 2 shows the change of the standard deviation of each category σ_i and the distance between the centers of two categories d_{ij}. These values are shown for the input pattern at $t = 0$, and the hidden pattern at $t = 0$ and $t = T - 1$ after being normalized by the standard deviation of all the patterns σ to observe the relative relation. For simplicity, the data from the last 1000 trials were utilized on behalf of observing over all the possible input patterns. The distance between the category i and j, d_{ij} becomes larger at the hidden layer than at the input layer at $t = 0$ for any combinations of the categories. It becomes larger also through time. While, the standard deviation σ_i in one category becomes almost 0.0 at $t = T - 1$ for any categories, and as a result, it is far smaller than the minimum distance between categories $min_{i,j}d_{ij}$. This means that the dynamics of the recurrent network is almost fixed-point convergence and one fixed-point is formed for each category. In the case when the system made a wrong answer, the interval T is 5 or 6. It is supposed that if the remind time is more, the system can generate the correct answer.

Fig. 3 shows the change of the average pattern for each category. It is shown for the input pattern at $t = 0$, and the hidden pattern at $t = 0$ and $t = T - 1$. In the average input pattern, one pixel value at a corner takes 1.0 with a high probability, but since a noise is
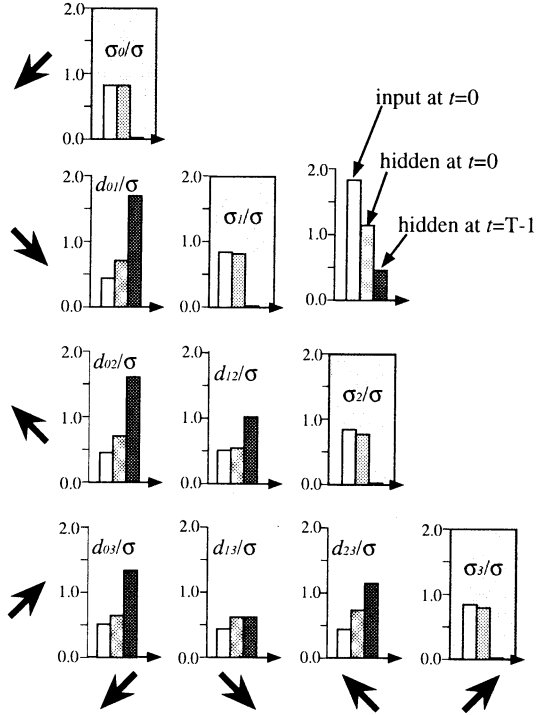
Figure 2: The change of the normalized distances between categories and the standard deviation of one category through time.

hidden state (t=0)   hidden state (t=T-1)
input pattern

Figure 3: The change of the average pattern of input and hidden layers.

Table 1: The size of the basin for each category in the whole input space.

condition \ category	0	1	2	3	others
normal	1075	3853	1160	3912	0
no_shift (category 2,3)	2039	7527	248	186	0
no_shift (all categories)	841	468	(3528)	4670	493
3 categories	2138	2203	5627	-	32
fixed time (T=10)	1286	(5719)	1306	1679	10

added to one pixel, it is not always 1.0. In the average hidden pattern at $t = 0$, no values are so close to 0.5 or -0.5. While at $t = T - 1$, almost all the values are close to 0.5 or -0.5. The dynamics of fixed-point convergence can be observed also from this figure.

Fig. 4 shows two examples of the dynamics. In this example, 15 pixels in total 25 pixels have a different value between the two input patterns even though both were generated from the same original arrow pattern. In the hidden patterns at $t = 0$, the values are different more than 0.5 in 15 units among total 20 units. However, as the lower part of the figure, the both hidden states converge to the same hidden state.

In order to know the size of the basin corresponding to each category, input signals were set randomly and hidden state at $t = 100$ was observed. Table 1 shows the number of hidden states whose distance is less than 1.9 from the average hidden state of one category. 1.9 is the maximum distance from the average hidden state to one hidden state in the same category at $t = T - 1$. It is seen that the number, in other words, the size of the basin varies very much depending on the category. The variety depends on the initial connection weights of the neural network.

When the number of the cropping way into 5 × 5 image is limited to only one in spite of 3 × 3 = 9 for two of the 4 categories, the basins change as the second row of Table 1. It is seen that the size of the

basin became smaller for the category for which the cropping way was limited. In the other two simulations with different initial connection weights, the difference can be observed, but is not so clear. There is one case in which the basin is larger in one category of limited cropping way than in one category of normal way.

When the cropping way was limited to only one for all the 4 categories, the learning was faster and more stable, and 8 basins were formed. The dynamics seems complicated in this case. That may be because the basins do not cover the whole input space. The numbers on the third row of Table 1 are counted when $t = 200$ only for this condition. Some of them do not correspond to any of the 4 categories. The reason why the number for the category 2 is put in parenthesis is that the final convergence point is different from the hidden state at $t = 5..14$.

When the number of the categories reduced to three, the basins changes as the fourth row of Table 1. It is seen that the number of large basins is three, and the other small basin was formed. In the other two simulations with a different random number sequence,
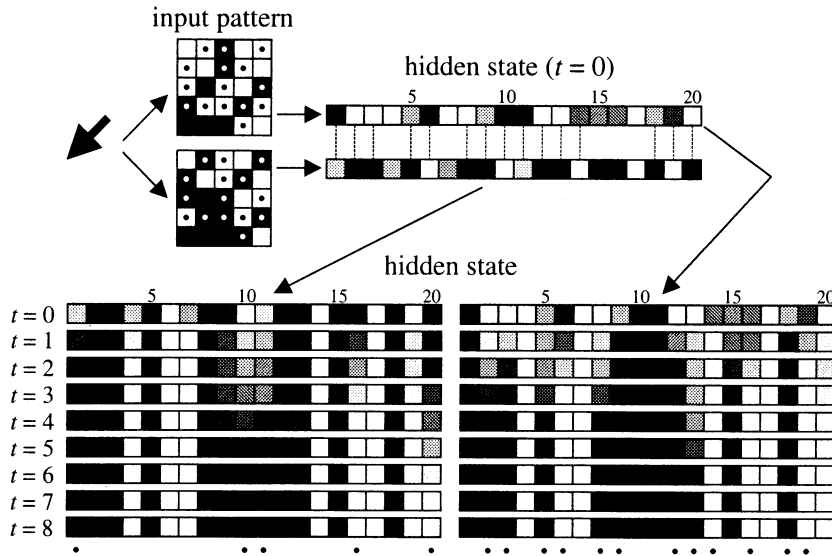
Figure 4: An example of the change of hidden state. The dynamics of fixed-point convergence can be observed.
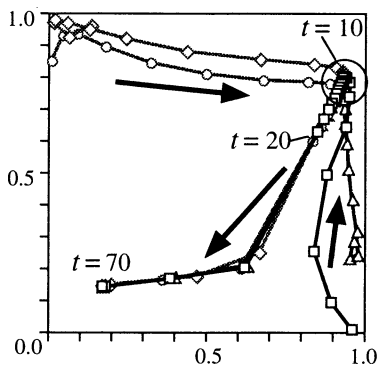
Figure 5: The change of two hidden neuron states when presentation time of the second pattern is fixed at $t = 10$.

the number of the formed basins is just three. It is considered that the the number of the basins becomes equal to the number of categories when the input pattern used in the learning varies in some degree.

When the presenting time of the second pattern is fixed at $t = 10$, the 4 main basins are formed as the fifth row of Table 1, but for the category 1, the hidden state at $t = 10$ is different from the final fixed point. In this case, the values of three hidden neurons changed after $t = 10$. Fig. 5 shows how such neurons change its values. X-axis shows the output of one such neurons, and y-axis the output of another one. Each of four lines in this figure shows the change of hidden neurons output for each of four input patterns in the category 1. It is seen that all the hidden states are the same at $t = 10$, and the speed of hid-

den state change becomes slow. However, after that, the state changes gradually, and finally converges to the real fixed-point. In the simulation with different initial connection weights, four categories can be distinguished with each other as well, but only 3 basins are formed. This means that the hidden state at $t = 10$ for one category changes and finally converges to the fixed-point for another category.

4 Conclusion

The dynamics after the learning of a context-based attention task using Elman-type recurrent network has been observed. When the input patterns cover the input space in some degree, the number of the basins becomes equal to the number of categories required in the task, even though the size of the basin is varied so much. The dynamics was rational and adaptive according to the learning conditions.

Acknowledgement

References

[1] Elman, J. L., "Finding Structure in Time", *Cognitive Science*, **14**, pp. 179–211, 1990.

[2] Zipser, D., "Recurrent Network Model of the Neural Mechanism of Shot-Term Memory" *Neural Computation*, **3**, pp. 179–193, 1991.

[3] Shibata, K., "Formation of Attention and Associative Memory Based on Reinforcement Learning", *Proc. of ICCAS (Int'l Conf. on Control, Automation and Systems*, pp. 9–12, 2001.

Learning method by a statistical approximation for simultaneous recurrent networks

Masao Sakai
Graduate School of Eng.
Tohoku University
05, Aoba, Aramaki, Aoba-ku
Sendai, 980-8579, JAPAN

Noriyasu Homma
College of Medical Sciences
Tohoku University
2-1, Seiryo-machi, Aoba-ku
Sendai, 980-8575, JAPAN

Kenichi Abe
Graduate School of Eng.
Tohoku University
05, Aoba, Aramaki, Aoba-ku
Sendai, 980-8579, JAPAN

(sakai,homma,abe@abe.ecei.tohoku.ac.jp)

Abstract

In this paper, a statistical approximation learning method (SALM) is proposed for a new type of neural networks, simultaneous recurrent networks (SRNs). The SRNs have the capability to approximate non-smooth functions which cannot be approximated by using conventional multi-layer perceptrons (MLPs). However, the most of the learning methods for the SRNs are computationally expensive due to their inherent recursive calculations. To solve this problem, a novel approximation learning method is proposed by using a statistical relation between the time-series of the network outputs and the network configuration parameters. Simulation results show that the proposed method can learn a strongly nonlinear function efficiently.

1 Introduction

In the field of supervised learning, the most popular form of the feedforward neural networks, the multi-layer perceptrons (MLPs) [1], have been proven to approximate smooth functions very well, then many application problems use the MLPs as a model for identifying and controlling nonlinear complex dynamic systems [3]. However there are many practical and difficult applications, e.g., a brain-like intelligent control and planning, where the functions to be approximated are not smooth. In these cases, the MLPs cannot approximate the non-smooth functions accurately[6].

In general, since the recurrent neural networks have some brain-like complex feedback connections, they can provide very complex dynamics. Indeed, in the state space of the dynamic neural system, using the *basin* which separates the initial state vectors from each other according to the corresponding final state vectors, a new type of neural networks, simultaneous recurrent networks (SRNs), has the capability to approximate non-smooth functions[7].

There are several learning methods to train the SRNs. However, the most of these methods including the well-known backpropagation through time (BTT) algorithm are computationally expensive due to calculation of the final state[6]. Also the recursive calculation of the gradients diverges when BTT learns the strongly nonlinear function. To solve these problems, an approximation method, called the *truncation* method (TM), has been proposed[2]. In this method, the recursive terms are simply truncated (assumed to be 0). Therefore, this non-recursive method often leads wrong learning direction.

In this paper, to improve the approximation accuracy without much increase of the computational time, a novel approximation learning method, a statistical approximation learning method (SALM), is proposed. The proposed method can approximate the gradient by using a statistical relation between the networks dynamics and the parameters of the SRNs. Simulation results show that the proposed SALM can efficiently learn a strongly nonlinear function which cannot be learned by the conventional methods. Also, due to the non-recursive formulation of the approximation, it is shown that the SALM can provide a practical computational time compared with that of the conventional BTT.

2 Simultaneous recurrent networks

For a simultaneous recurrent networks (SRNs), the following variables are introduced:

x : The external neural input vector, $x = [x_1 \ x_2 \ \cdots \ x_n]^\top$, n is the number of external inputs.

s_j, z_j : The input and output of the jth neuron in the hidden layer. $j = 1, 2, \cdots, h$, h is the number of neurons in the hidden layer.

y_i : The output of the ith neuron in the output layer. $i = 1, 2, \cdots, m$, m is the number of neurons in the output layer.

$\overline{y}$: The neural output vector, $\overline{y} = [\overline{y}_1 \ \overline{y}_2 \ \cdots \ \overline{y}_m]^\top$,

v_{ij} : The weight connecting z_j to y_i.

$w_{j\nu}$: The weight connecting x_ν to z_j.

$\mu_{j\kappa}$: The feedback weight connecting y_κ to z_j.

θ_i : The bias of the ith neuron in the output layer.

ϕ_j : The bias of the jth neuron in the hidden layer.

Let t be discrete time, $t = 1, 2, \cdots$. The neural input and output at time t in the hidden layer, $s_j(t)$ and $z_j(t)$, $j = 1, 2, \cdots, h$, are defined as

$$s_j(t) = \sum_{\nu=1}^{n} w_{j\nu} x_\nu + \sum_{\kappa=1}^{m} \mu_{j\kappa} y_\kappa(t) + \phi_j, \quad (1)$$

$$z_j(t) = \frac{1}{1 + \exp(-\alpha s_j(t))}, \quad (2)$$

where, α is a gain coefficient of the sigmoid function. The neural outputs at next time $t + 1$ in the output layer, $y_i(t + 1)$, $i = 1, 2, \cdots, m$, are defined as

$$y_i(t + 1) = \sum_{j=1}^{h} v_{ij} z_j(t) + \theta_i. \quad (3)$$

The network outputs of the SRN are given as

$$\overline{y}_i = \lim_{T \to \infty} y_i(T), \quad (4)$$

where, T is the number of iterations. Note that the conventional MLPs are only a special case of the SRNs with a configurations as $T = 1$ and $y_k(0) = 0$.

3　Learning methods for SRN

Supervised learning for the neural networks is the task of learning a nonlinear function in which several sample inputs and outputs of the function are given. Letting $\left\{ \left(x^l, \check{y}^l \middle| l = 1, 2, \cdots, N \right) \right\}$ be the set of desired input-output pairs, the changes of parameters, δp, are given by the learning methods based on a backpropagation algorithm as [4]

$$\delta p = -\eta \frac{\partial e}{\partial p},$$

$$= -\eta \sum_{l=1}^{N} \sum_{i=1}^{m} (\overline{y}_i^l - \check{y}_i^l) \frac{\partial \overline{y}_i^l}{\partial p}, \quad (5)$$

where $e = \sum_{l=1}^{N} \sum_{i=1}^{n} (\overline{y}_i^l - \check{y}_i^l)^2 / 2$ is a square error and η is a positive coefficient. If the gradients $\partial \overline{y}_i^l / \partial p$ are obtained, the change of parameter δp can be calculated. In the following, the calculations of the $\partial \overline{y}_i / \partial p$ by conventional methods, i.e., Backpropagation through time (BTT) and the *truncation* method (TM) are described briefly.

3.1　Backpropagation through time

Backpropagation through time (BTT) is a general method which calculates the gradients exactly. From (3) and (4), the gradients $\partial \overline{y}_i / \partial p$ and $\partial y_i(t + 1) / \partial p$ are given as

$$\frac{\partial \overline{y}_i}{\partial p} = \lim_{T \to \infty} \frac{\partial y_i(T)}{\partial p}, \quad (6)$$

$$\frac{\partial y_i(t + 1)}{\partial p} = \Delta_p(\theta_i) + \Delta_p(v_{ij}) \cdot z_j(t)$$

$$+ \sum_{j=1}^{h} v_{ij} \frac{\partial z_j(t)}{\partial p}, \quad (7)$$

where $\Delta_p(v)$ is defined by

$$\Delta_p(v) = \begin{cases} 1, & (p = v), \\ 0, & (p \neq v). \end{cases} \quad (8)$$

By (1) and (2), $\partial z_j(t) / \partial p$ and $\partial s_j(t) / \partial p$ are calculated as

$$\frac{\partial z_j(t)}{\partial p} = \alpha z_j(t) (1 - z_j(t)) \frac{\partial s_j(t)}{\partial p}, \quad (9)$$

$$\frac{\partial s_j(t)}{\partial p} = \Delta_p(w_{j\nu}) \cdot x_\nu + \Delta_p(\mu_{j\kappa}) \cdot y_\kappa(t)$$

$$+ \sum_{\kappa=1}^{m} \mu_{j\kappa} \frac{\partial y_\kappa(t)}{\partial p} + \Delta_p(\phi_j). \quad (10)$$

Here letting $\partial y_\kappa(0) / \partial p$ be 0, $\partial s_j(0) / \partial p$ are given by (10), then $\partial y_i(1) / \partial p$ are calculated by (7) $\sim$ (9). Finally the gradients $\partial \overline{y}_i / \partial p$ ($\equiv \partial y_i(T) / \partial p$) are obtained by calculating $\partial y_i(t) / \partial p$ recursively.

3.2　Truncation method (TM)

The *truncation* method (TM) is probably the most popular method used to adapt SRNs even though the people who use it mostly just call it ordinary backpropagation. To calculate the gradients $\partial \overline{y}_i / \partial p$, the method uses only one simple pass of backpropagation through the last iteration of the model by truncating $\partial y_i(T - 1) / \partial p = 0$. Therefore the method is the simplest approximation and the least expensive method.

4 Statistical Approximation Learning method (SALM)

4.1 Basic strategy for a statistical approximation

The gradients $\partial \overline{y}_i/\partial p$ calculated by the TM are not accurate. Therefore it often fails to learn the strongly nonlinear models. To improve the accuracy, a statistical approximation learning method (SALM) is proposed. In the proposed method, the truncation, $\partial y_i(T-1)/\partial p = 0$, is redefined using a statistical approximation. That is, the recursive gradients $\partial z_j(t)/\partial p$ in (7) is approximated by using the expectation of z_i, $E[z_i]$, as

$$\frac{\partial z_j(t)}{\partial p} \approx \frac{\partial E[z_j]}{\partial p}. \tag{11}$$

Then, from (7)

$$\frac{\partial y_i(T-1)}{\partial p} \approx \Delta_p(\theta_i) + \Delta_p(v_{ij}) \cdot z_j(T-2)$$

$$+ \sum_{j=1}^{n} v_{ij} \frac{\partial E[z_j]}{\partial p}. \tag{12}$$

To calculate the gradients of the expectation, $\partial E[z_j]/\partial p$, since the SRNs are only a special case of the fully connected recurrent networks, an equivalent notation of the SRNs to the fully connected recurrent networks is developed. In the fully connected recurrent networks, the expectation $E[z_j]$ is represented by a function of a key parameter $\overline{\sigma^2}$, which implies a variance of the inputs s_j in (1)[5]. Using the equivalent notation and the statistical relation between the expectation $E[z_j]$ and the key parameter $\overline{\sigma^2}$, the gradients $\partial E[z_j]/\partial p$ can be approximated by

$$\frac{\partial E[z_i]}{\partial p} \approx \frac{\partial E[z_i]}{\partial \overline{\sigma^2}} \cdot \frac{\partial \overline{\sigma^2}}{\partial p}. \tag{13}$$

Let $\omega_{ji} \equiv \sum_{\kappa=1}^{m} \mu_{j\kappa} v_{\kappa i}$, and $\psi_j \equiv \sum_{\nu=1}^{n} w_{j\nu} x_\nu + \sum_{\kappa=1}^{m} \mu_{j\kappa} \theta_k + \phi_j$, then the SRNs defined by (1) $\sim$ (3) are equivalent to fully connected recurrent networks given as

$$z_j(t+1) = \frac{1}{1 + \exp(-\alpha s_j(t))}, \tag{14}$$

$$s_j(t) = \sum_{i=1}^{h} \omega_{ji} z_i(t) + \psi_j. \tag{15}$$

To apply the statistical relation to the equivalent SRNs, let ω_{ji} be a uniformly distributed random variable. Introducing the following restriction

$$\psi_j = -\frac{1}{2} \sum_{\substack{1 \leq j \leq h \\ 1 \leq i \leq h}} \omega_{ji}^2, \tag{16}$$

the key parameter $\overline{\sigma^2}$ can be defined as[5]

$$\overline{\sigma^2} = \frac{1}{12h} \sum_{\substack{1 \leq j \leq h \\ 1 \leq i \leq h}} \omega_{ji}^2. \tag{17}$$

Here $\partial \overline{\sigma^2}/\partial v_{kl}$ is calculated by the definition of ω_{ji} as

$$\frac{\partial \overline{\sigma^2}}{\partial v_{kl}} = \frac{1}{6h} \sum_{j=1}^{h} \mu_{jk} \left(\sum_{\kappa=1}^{m} \mu_{j\kappa} v_{\kappa l} \right). \tag{18}$$

Also $\partial \overline{\sigma^2}/\partial w_{kl}$ can be calculated similarly. Therefore, if the gradients $\partial E[z_i]/\partial \overline{\sigma^2}$ are calculated, then the target gradients in (13), $\partial E[z_i]/\partial p$ can be calculated.

4.2 Calculation of $\partial E[z_i]/\partial \overline{\sigma^2}$

From (14), the sigmoid function is given as a following power series representation[5].

$$z_i \approx \begin{cases} 1, & (s_i > \epsilon/\alpha), \\ \sum_{\tau=0}^{M_1} F(\tau) \cdot (\alpha s_i)^\tau, & (|s_i| < \epsilon/\alpha), \\ 0, & (s_i < -\epsilon/\alpha), \end{cases} \tag{19}$$

$$F(\tau) \equiv \begin{cases} 1/2, & (\tau = 0), \\ \sum_{k=1}^{\tau} \frac{(-1)^{k+1}}{2k!} \cdot F(\tau - k), & (\tau > 0), \end{cases} \tag{20}$$

where M_1 is a suitable natural number and ϵ is a positive constant. The probability density $g(s_i, 0, \overline{\sigma^2})$ of the expectation of the input s_i can be given as a power series representation[5].

$$g\left(s_i, 0, \overline{\sigma^2}\right) = \frac{1}{\sqrt{2\pi\overline{\sigma^2}}} \exp\left(-\frac{s_i^2}{2\overline{\sigma^2}}\right), \tag{21}$$

$$= \begin{cases} \sum_{\tau=0}^{M_2} R(\tau) \cdot \left(\frac{1}{\overline{\sigma^2}}\right)^{1/2} \cdot \left(\frac{s_i^2}{\overline{\sigma^2}}\right)^\tau, & \left(|s_i| < \beta\sqrt{\overline{\sigma^2}}\right), \\ 0, & (\text{otherwise}), \end{cases}$$

$$R(\tau) \equiv \frac{(-1)^\tau}{\sqrt{2\pi} \cdot 2^\tau \cdot \tau!}, \tag{22}$$

where M_2 is a suitable natural number and β is a positive constant. Supposing $M_1 = 2M_2$ to avoid much complicated representation, $E[z_i]$ can be given

by a power series representation as

$$E[z_i] = \int_{-\infty}^{\infty} z_i(t) \cdot g(s_i, 0, \overline{\sigma^2}) \ ds_i$$

$$\approx \begin{cases} \begin{aligned} & \sum_{\substack{0 \le \tau \le M_2 \\ 0 \le k \le \tau}} Q(\tau, k)\beta^{2\tau+1} \left(\alpha^2 \overline{\sigma^2}\right)^k \\ & + \sum_{\substack{M_2+1 \le \tau \le 2M_2 \\ \tau - M_2 \le k \le M_2}} Q(\tau, k)\beta^{2\tau+1} \left(\alpha^2 \overline{\sigma^2}\right)^k, \\ & \hspace{3cm} \left(\alpha^2 \overline{\sigma^2} < \tfrac{\epsilon^2}{\beta^2}\right), \\[2mm] & \sum_{\substack{0 \le \tau \le M_2 \\ 0 \le k \le \tau}} Q(\tau, k)\epsilon^{2\tau+1} \left(\alpha^2 \overline{\sigma^2}\right)^{(k-\tau)-\frac{1}{2}} \\ & + \sum_{\substack{M_2+1 \le \tau \le 2M_2 \\ \tau - M_2 \le k \le M_2}} Q(\tau, k)\epsilon^{2\tau+1} \left(\alpha^2 \overline{\sigma^2}\right)^{(k-\tau)-\frac{1}{2}} \\ & + \sum_{\tau=0}^{M_2} \tfrac{R(\tau)}{(2\tau+1)} \left\{ \epsilon^{2\tau+1} - \left(\tfrac{\epsilon^2}{\alpha^2 \overline{\sigma^2}}\right)^{\tau+\frac{1}{2}} \right\}, \\ & \hspace{3cm} \left(\alpha^2 \overline{\sigma^2} > \tfrac{\epsilon^2}{\beta^2}\right), \end{aligned} \end{cases} \quad (23)$$

$$Q(\tau, k) \equiv \gamma \frac{2 \cdot F(2k) \cdot R(\tau - k)}{(2\tau + 1)}, \quad (24)$$

where γ is a positive constant. From the above arrangements the $\partial E[z_i]/\partial \overline{\sigma^2}$ can be calculated by (23) and (24) easily. Thus $\partial y_i(T - 1)/\partial p$ in (12) can be calculated by (13) and (18). Finally, using the truncation algorithm and the $\partial y_i(T - 1)/\partial p$ as the initial values, the gradients $\partial \overline{y}_i/\partial p$ can be calculated.

5 Simulation Results

The three neural learning methods, the multi-layer perceptrons (MLPs), the simultaneous recurrent networks (SRNs) trained by backpropagation through time (BTT), and the SRNs trained by the truncation method (TM), have been tested on a design task which requires the networks to learn a following *non-smooth* function (Figure 1(a)).

$$\check{y}_1 = \begin{cases} 1, & (x_1 - 0.5)(x_2 - 0.5) > 0, \\ 0.5, & (x_1 - 0.5)(x_2 - 0.5) = 0, \\ 0, & (x_1 - 0.5)(x_2 - 0.5) < 0. \end{cases} \quad (25)$$

In this task, the networks with $m = 1, h = 30, n = 2, \alpha = 10, T = 10$ were employed, and the constants and the coefficient were decided experimentally: $M_2 = 5$, $\epsilon = 2.4$, $\beta = 2$, $\gamma = 1.2$ and $\eta = 0.0002$. The connecting weights of the networks were initialized randomly, then changed by the learning methods.
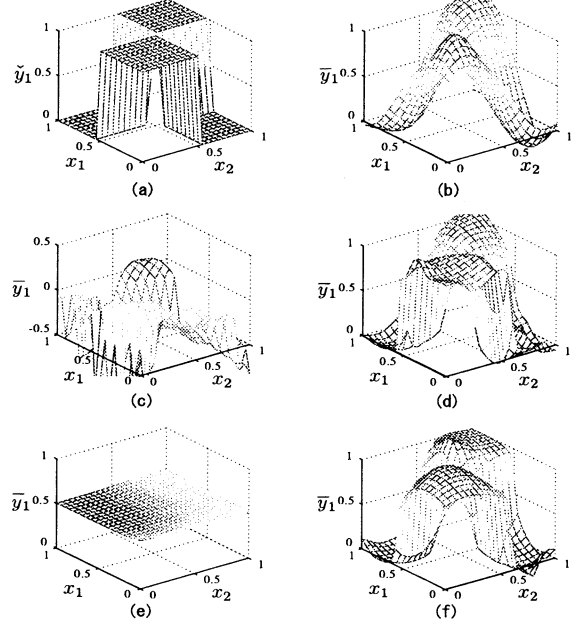
Figure 1: The input-output functions of the target model and neural networks; (a) target model, (b) MLP, (c) SRN with BTT, (d) SRN with TM (a successful case), (e) SRN with TM (a typical failure), and (f) SRN with SALM.
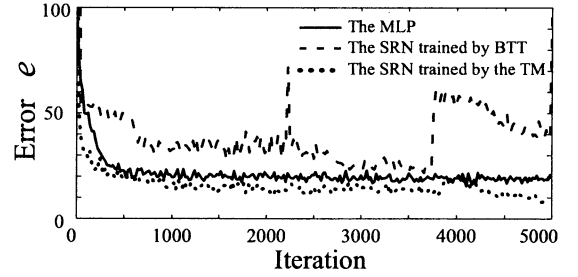
Figure 2: The error e as functions of learning iteration by the three conventional neural methods, the MLP (solid curve), the SRN trained by BTT (dashed curve), and the SRN trained by the TM (dotted curve).

The number of training data was 121 (11 × 11 mesh data in [$x_1 \ x_2$] space). The solid curve in figure 2 shows the simulation result using the MLP, and the input-output function of the MLP (after 5000 learning iterations) is shown in figure 1(b). Note that the MLP cannot learn the target model accurately: the non-smooth parts of the target model are approximated by the smooth functions.

The dashed curve in figure 2 shows the simulation result using the SRN trained by BTT, and the input-output function of the SRN trained by BTT is shown
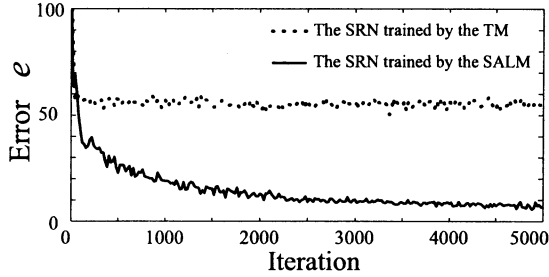
Figure 3: The error e as functions of learning iteration for the SRN. The dotted curve shows the typical failure by the TM, while the solid curve shows the result by the proposed method, the statistical approximation learning method (SALM).

Learning method	CPU-TIME [sec.]
The MLP	0.065
The SRN with BTT	16.72
The SRN with the TM	1.95
The SRN with the SALM	2.58

Table 1: CPU-TIME.

in figure 1(c). Note that the SRN with BTT fails to learn the target model because the recursive calculations of the gradients diverged.

The dotted curve in figure 2 shows the simulation result using the SRN trained by the TM, and the input-output function of the SRN trained by the TM is shown in figure 1(d). Note that the SRN with the TM can learn the target model better than the MLP. However, in the other simulation trials, the networks with different values of the initial states and connecting weights often fail to learn the target model. The typical failure is shown in figure 3 (the dotted curve) and figure 1(e) (the learned input-output function). Note that the SRN trained by the TM converges to the local optimal solution. This is because of the inexact approximation of the TM.

On the other hand, the solid curve in figure 3 shows the simulation result using the SRN trained by the SALM, where the initial network states and parameters of the network are taken to be equal to those of SRN giving the input-output function in figure 1(e) trained by the TM. The trained input-output function is shown in figure 1(f). First, comparing with figure 1(b) $\sim$ 1(f), note that the SRN with the SALM can learn the target more accurately than the three conventional methods. Also, the average error of the SALM over 15 trials is 8.64, less than the average error 11.47 by the TM under the same condition. In brief,

the accuracy of the SALM is better than that of the TM.

Second, Table 1 lists the CPU-TIMEs to calculate the the gradients $\partial \bar{y}_i / \partial p$ in the above simulations. Note that the SALM has computational time almost equivalent to the TM.

6 Conclusions

In this paper, a statistical approximation learning method (SALM) for the simultaneous recurrent networks (SRNs) has been proposed. In a simulation study, the proposed SALM was tested. This simulation results showed that the proposed SALM can efficiently learn the target model consisting of strongly nonlinear and non-smooth functions which cannot be learned by the conventional methods. Also, due to the non-recursive formulation of the approximation, it was shown that the SALM can provide a practical computational time compared with that of the conventional BTT. Further applications using this method are in progress.

References

[1] Barron, A. (1994). Asymptotically optimal functional estimation by minimum complexity criteria. In: *Proc. of 1994 IEEE International Symposium of Information Theory*, IEEE, New York.

[2] Fausett, L. (1994). *Fundamentals of Neural Networks: architectures, algorithms and applications.* Prentice Hall.

[3] Narendra, K.S. and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks* **1**, 4–27.

[4] Rumelhart, D.E., G.E. Hinton and R.J. Williams (1986). Learning representations by backpropagating errors. *Nature* **323**, 533–536.

[5] Sakai, M., N. Honma and K. Abe (2001). Chaos control by stochastic analysis on recurrent neural networks. In: *Proc. of AROB 6th 2001.* Tokyo. **2**, 478–153.

[6] Werbos, P.J (1994). *The Roots of Backpropagation.* John Wiley & Sons. New York.

[7] Werbos, P.J. (1995). Optimization methods for brain-like intelligent control Organization of Behavior. In: *Proc. of the 34th Conference on Decision and Control 1995, IEEE Press.*, 579–584.

A Study of Improvement in Robustness of Sensor Information for Autonomous Robots

Hiromi Takeuchi
Nagoya Inst. of Tech.
Nagoya, Aichi, 466-8555

Koichiro Yamauchi
Hokkaido University
Sapporo, Hokkaido, 060-8628

Naohiro Ishii
Nagoya Inst. of Tech.
Nagoya, Aichi, 466-8555

E-mail: hiromi@egg.ics.nitech.ac.jp

Abstract

In this paper, we propose a robot control system. The proposed system has two features, the sensor information integration and virtual environment model. The sensor integration system is our previous work, we apply it this robot system. Using this system, the robot can remove the noisy sensor input from several sensors information. The virtual environment model is a function that simulate real environment by a robot and is used for a prediction of object which has observed.

In this paper, we show the outline of this robot system and the result of it in computer simulations using sensor information integration only.

Keyword Sensor Integration, Neural Network, Robot Control, Agents

1 Introduction

Recent years, the improvement of robotics makes us feel the robot become more familiar. And, the robot that does the motion similar to human is also developed. However, in order for a robot to play an active part in the medical and welfare spot, the sensor of the robot has to be improved more higher.

Research of these sensors is already done variously. However, our approach is not improvement of sensor, but using intelligent information processing technology. Our purpose, using intelligent information processing technology, we make the recognition of sensor become more higher without expensive sensors.

In this study, we propose a robot system that has two features. First is the integration of sensor information using sensory integrating system (SIS), which was our previous work. To integrate sensor information makes the noise decrease. Second is to construct a virtual environment into the robot. By using this system, the robot predicts an environment in the next

time. Moreover, if the object that has to be observed is out of sensor areas, the virtual environment complements the missed object.

This research aims at building the system which combined these.

2 Sensory Integrating System

The detailed structure of the system[1][2] is illustrated in Fig.1 This system consists of several sets
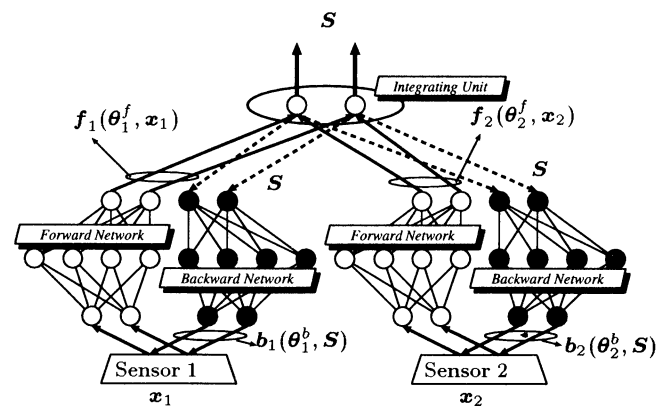
Figure 1: Sensory Integrating System

of multi-layered neural networks and sensors, each of which receives inputs from the corresponding sensor. We assume that all sensors of this system always observe a same object simultaneously. Each neural network consists of two back-propagation networks which we call forward network and backward network. The forward network receives inputs from the corresponding sensor, and predicts the class of the inputs. The outputs of all forward networks are sent to an integrating unit, which integrates all outputs of the forward

networks. The integrating unit basically calculates the averaged outputs of all the forward networks. On the other hand, the backward networks try to reconstruct the sensory input from the output of the integrating unit. This is for calculating the confidence of each sensors. If the reconstructed input is quite different from the current input, the system try to ignore the sensory input, and vice versa. The system optimizes the integrated output and the value of the confidence of each sensor by repeating the forward and backward network calculations.

The integrated output optimized is the final output of the system.

The recognition algorithm was below.

1. Initialize confidence of all sensors c_i to 1.

2. Initialize S as the average of output of forward networks.

3. set X_i to the backward networks output $b_i(S)$.

4. Update c_i

$$c_i := c_i + 2\beta(1 - c_i) - \gamma \|x_i - b_i(S)\|^2 \quad (1)$$

5. Update the vector of X_i as equation below.

$$X_i := X_i + 2\gamma c_i(x_i - X_i) \quad (2)$$

6. Update S as equation below.

$$S := S + 2\alpha \left\{ \frac{1}{m} \sum_i^m f_i[X_i] - S \right\} \quad (3)$$

7. Repeat Step 3 to Step 6 until S converges to a stable point. Therefore, if $\|S^{t-1} - S^t\| < \epsilon$, goto Step 3.

   where ϵ is a small positive constant and $\epsilon \ll 1$, S^{t-1} and S^t denote the outputs before and after the execution of equation (3).

Note that the forward networks predict the solution S from X_i directory so that S converges to an appropriate solution very quickly.

The detail of recognition and learning algorithm of this system is referred by [2].

3 The Robot Control System

3.1 The outline of robot control system

In this section, we explain the robot system which was proposed. Figure.2 shows the outline of this system.
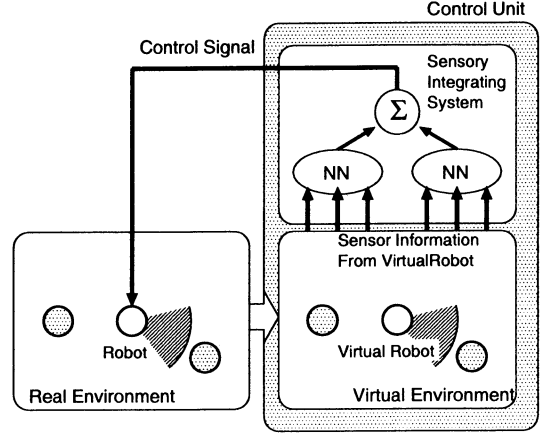
Figure 2: The schematic of this system

This system recognizes environment according to the following procedures.

1. Observe the real world and construct the virtual world.

2. The Control system observe the virtual world and get the sensor information from virtual world.

3. The sensor data which was received from virtual world is integrated by SIS and it outputs a robot control signal.

4. The robot obeys the control signal and goes forward.

3.2 Control Unit

The controller consist of the Sensory Integrating Unit (SIU) and Course Selector (CS), and some elements to make the SIU learn appropriate control signals, a critic, a prediction network and a noise generator. The detail of this controller shows Figure.3. The CS outputs control signal to reach the goal constantly. If the robot use this output, it reach the goal through the best route. The SIU consists of two sets of the forward and the backward neural networks. The two forward networks receive inputs from the visual sensors and the distance sensors, respectively. Both forward networks have a output for the control signal.

The critic evaluates the current action to see if it is good or not with observing the visual sensory inputs. The noise generator adds noise to the 1st element of the output from the SIU for discovering appropriate control signals. The prediction network predicts whether the SIU outputs a good control signal to current sensory inputs or not.
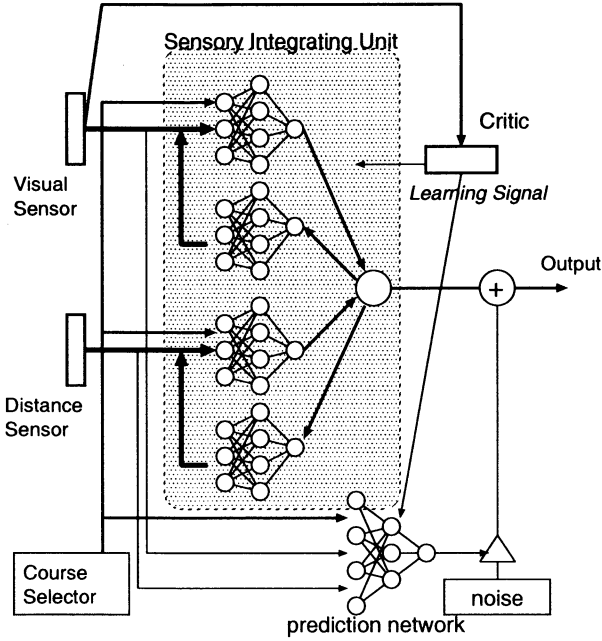
Figure 3: Structure of Control Unit

However, the critic, the noise generator and the prediction network are removed after the learning.

4 Computer Simulation

In this section, we show effects of this proposed system in computer simulation.In this simulation, we use sensor integrating system only. The purpose of this simulation is the robot move forward which avoiding obstacles from start to goal on the field. The simulation which the robot move between start and goal is repeated 2000 times.

4.1 Robot

First, we define the robot which uses by computer simulation.

The detailed structure of robot in this paper is illustrated in Fig4. The robot has two sensers. One of these sensor emulate one-dimensional vision sensor. Another sensor is distance sensor. these sensor which referenced to the paper of real robot[3].

The visual sensor of the robot consist of 12 photo detector. The viewing range of photo detector is 15 degrees. So, the robot can observe in angular measure from -90 to 90 degrees and a total range is 180 degrees. If a target is in the viewing range of the photo detector, it yield 0.9 outputs, otherwise it yield 0.1 output.
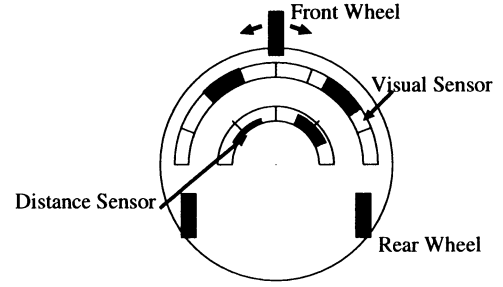
Figure 4: Robot

The distance sensor of the robot consist of 6 elements. The range of the element of distance sensor is 30 degrees. This sensor can get the distance data from targets, however it has direction elements more less than visual sensor has.

This robot always moves forward at a constant speed but with control of the direction of the front wheel. The front wheel is adjusted by a controller which received all sensory inputs.

4.2 Learning and Recognition

During the learning phase, the robot repeats random move fundamentaly. And the robot learn actions which is fit for solving problem from the random actions.

Visual Sensory Input		Output of the Critic
Before the Control	After the Control	
		1.0
		0.0

Figure 5: Examples of Response of Critic

The critic recieves visual sensory inputs. It yields 1 otherwise 0 as shown in Figure 5. The diagram in Figure 5 shows some examples of the visual sensory input and the corresponding output value of the critic. The masked area shows where the photo detector observes the target ahead and the brank area shows where the photo detector observes nothing. The evaluation rules of critic are simple. If there is a obstacle in 90 degrees in front of the robot, it outputs 1 otherwise 0.

The prediction network receives all sensory information. This network controls noise level. For example, if the output value of this network is large, the noise level is reduced. Using this prediction network,

the system learns quickly because it avoids redundant search for appropriate control signals.

A specific condition and processes in this experimentation is as follow.

1. The robot locates a start on the field.

2. Three obstacles locate randam place on the field and one target locates center of the field.

3. The robot searches the location of obstacles using sensory information.

    (a) if it is no existance in sight of the robot, the robot learn a control signal from Course Selector for reach a goal.

    (b) if the obstacles exist in front of the robot, the system starts a process which the robot learns avoiding methods. First, the robot modifies the direction of movement using sensory inputs. After the modify, the robot search the location of obstacles again, and the critic decides whether the robot learn this action or not. When there is no obstacle in front of the robot, The critic starts learning process, and vise versa.

4. During the process 3 was repeated, the experimentation is terminated if the robot reached a goal or contacted obstacles. And if the robot goes out from the field, it is terminated too.

5. Return to process 1 until the error of all neural networks is closed enough.

4.3 Results

To begin with, we show two snapshot of simulations to evaluate the proposed robot system.

Figure 6 shows the snapshot which the robot system after the learning. In this figure, we can see the robot can avoid the obstacles smoothly. Therefore, we denote the availability of this learning method.

Status of the robot	Ratio of Successful
Before learning	0%
After learning	62.4%
Visual Sensor Only	53.5%

Table 1: Ratio of successful to reach a goal, in 2000 trials.

Table 1 shows the ratio of the successful to reach the goal over 2000 trials. We can see from this table that the ratio of After learning is the most large in the
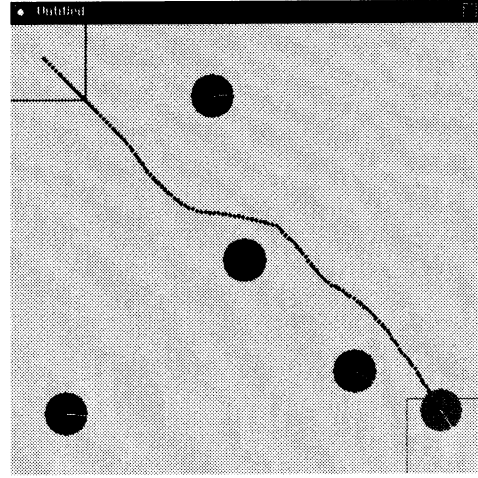
Figure 6: An example of the simulation when the robot was learned enough

table. And, this result denotes the robustness of the system.

5 Conclusion

In this paper, we propose the robot system which includes an avoidance method. Using this system, the robot is able to reach a goal while it avoids obstacles.

In this paper, we cannot show the experiment which combined a virtual environment model and sensor information integration. However, we believe this combine comes into effect for giving a robustness recognition ability to the system.

Using this system, the robot is able to work on complex environment such as life space of humans.

References

[1] Hiromi Takeuchi, Koichiro Yamauchi, and Naohiro Ishii. Self-supervised learning and recognition by integrating information from several sensors. In *IEEE IECON2000*, pages 1195–1200, 2000.

[2] Hiromi Takeuchi, Yoshifumi Terabayashi, Koichiro Yamauchi, and Naohiro Ishii. An avoiding method for autonomous robots based on sensory integrating system using neural networks. In *SNPD'01*, pages 885–891, 2001.

[3] A. Elfes. Using occupancy grids for mobile robot perception and navigati on. *Computer*, 22(6):46–57, 1989.

Superimposing Neural Learning by Dynamic and Spatial Changing Weights

N. Homma[*1,*2]

[*1]College of Medical Sciences
Tohoku University
Sendai, Japan 980-8575

M. M. Gupta[*2]

[*2]Intelligent Systems Research Laboratory
University of Saskatchewan
Saskatoon, SK, Canada S7N 5A9

Abstract

In this paper a new neural network model is presented for incremental learning tasks where networks are required to learn new knowledge without forgetting the old one. An essential core of the proposed network structure is their dynamic and spatial changing connection weights (DSCWs). A learning scheme is developed for the formulation of the dynamic changing weights, while a structural adaptation is formulated by the spatial changing connecting weights. To avoid disturbing the past knowledge by the creation of new connections, a restoration mechanism is introduced by using the DSCWs. Usefulness of the proposed model is demonstrated by using a system identification task.

1 Introduction

Incremental learning schemes add new knowledge to the networks without re-examining the past experiences. To achieve the attractive learning ability, many incremental learning schemes have been proposed [1, 2, 5]. In general, for incremental learning tasks the following three strategies are needed: (i) *Learning strategy*: no changes or only bounded changes for the trained weights; (ii) *Structural adaptation strategy*: creation of new weight connections for storing the new knowledge; and (iii) *Restoration strategy*: restoration of the past knowledge from the disturbances caused by the creation of new connections.

In this paper, to integrate these strategies for the strict incremental learning tasks, a novel neural model with *dynamic and spatial changing weights* (DSCWs) is proposed. This model formulates not only its neural computing (input-output relation) process, but also provides some strength to the above three strategies. Simulation results show that the proposed model is useful for a system identification task.

2 Neurons with Dynamic and Spatial Changing Weights (DSCWs)

We propose a neuron with integrated dynamic and spatial changing weights (DSCWs) as shown in Fig. 1. In this neural model, the following variables are defined:
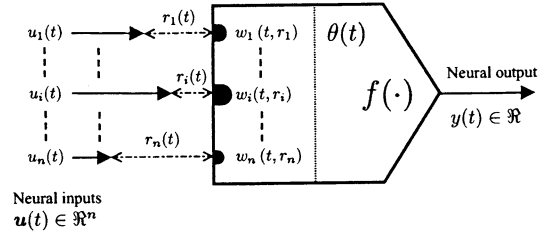
Figure 1: A neuron with integrated *dynamic and spatial changing weights* (DSCWs).

$u(t) = [u_1(t) \cdots u_n(t)]^T$ is the neural input vector, $r(t) = [r_1(t) \cdots r_n(t)]^T$ is the spatial distance vector between the sensory devices or another neuron's axon branches and the corresponding target dendrites (Fig. 1), $w(t,r) = [w_1(t,r_1) \cdots w_n(t,r_n)]^T$ is the weight vector as a function of time t and the spatial distance vector r, $\theta(t)$ is the threshold, and $y(t)$ is the neural output.

The dynamic change in the weight is defined as

$$\frac{\partial w(t,r)}{\partial t} = g_t(t, w, u) \tag{1}$$

where, $g_t = [g_{1t} \cdots g_{nt}]^T$ is a vector function describing *continuous* changes in the weight vector. Also the change in the threshold is defined by a function g_θ

$$\frac{d\theta(t)}{dt} = g_\theta(t, \theta) \tag{2}$$

Note that these rules of dynamic changes are corresponding to the continuous-time *learning* scheme.

As a novel *learning strategy* for the incremental learning tasks, a simple dynamic rule of changing weights is considered here.

$$\frac{\partial w_i(t, r_i)}{\partial t} = -\frac{1}{\tau_i} w_i(t, r_i) + F_i(u_i(t)) \tag{3}$$

$i = 1, 2, \cdots, n$, where τ_i (> 0) is a time constant, and $F_i(u_i(t))$ is an external force.

In this dynamic system, it is commonly well known in control systems theory that the different time constants yield memories with different durations such as *long-term memory* (LTM) and *short-term memory* (STM) in the brains. Note that, the proposed method can provide the

concept of LTM and STM by using different time constants, which cannot be realized by conventional learning schemes including other incremental learning schemes.

On the other hand, since the derivatives $\partial w_i / \partial r_j$ are 0 for $i \neq j$ by the definition, the spatial changes in the weights are defined by the diagonal matrix as:

$$\frac{\partial w(t, r)}{\partial r} = \mathrm{diag}\left(\frac{\partial w_1}{\partial r_1}, \cdots, \frac{\partial w_n}{\partial r_n}\right) \quad (4)$$

To provide a new *structural adaptation* scheme, a dynamic changes in the distances is introduced by a vector function $gr = [g_{r_1} \cdots g_{r_n}]^T$. Thus,

$$\frac{dr(t)}{dt} = gr(r, w, u) \quad (5)$$

Using the dynamic and spatial changing weights, the neural output is defined as

$$y(t) = f(u(t), w(t, r), \theta(t)) \quad (6)$$

where, f is a nonlinear static activation function (Fig. 1).

2.1 Structural Adaptation by Spatial Growing

Here, the growing of an axon branch toward the target dendrites is achieved according to the *spatially diffusing* neural input signals which might be a simple model of the chemical process from axon branches to dendrites or neural soma [3]. However, the diffusing signals without synaptic junctions must decay spatially since synaptic junctions may be a highly efficient transmitter of the chemical matters with a low loss.

Assuming the concept of the spatially diffusing decay signals, the following *equivalent* weight is introduced as the spatial change in (4).

$$\frac{\partial w_i(t, r_i)}{\partial r_i} = -\frac{1}{\mu_i} w_i(t, r_i) \quad (7)$$

where, μ_i (> 0) is a diffusion coefficient and r_i is the corresponding distance (Fig. 1). Then

$$w_i(t, r_i) = w_i(t, 0) e^{-\frac{r_i}{\mu_i}} \quad (8)$$

If a synapse connection exists between an axon branch and the dendrite, then r_i is relatively very small compared with the diffusion coefficient μ_i, and therefore, the signals are transformed to the dendrite potentials without loss.

Note that diffusing signals are different from signals through synaptic junctions, and knowledge or experience information of the neural computing process might not be conveyed by such diffusing signals. Therefore, the diffusing signals $s_i = s(u_i, w_i)$ are used only for the spatial growing scheme in (5), and the growing scheme is not applied for the existing synaptic weights. To realize this distinction, weights w_i are re-defined as

$$w_i(t, r_i) = \begin{cases} w_i(t, 0), & \left(e^{-\frac{r_i}{\mu_i}} > \phi\right) \\ 0, & \text{(otherwise)} \end{cases} \quad (9)$$

where, $\phi(> 0)$ denotes a threshold to decide whether the synaptic junction has been formed or not. By this definition, the knowledge information is conveyed only through synaptic junctions, and thus the dynamic change in (3) is applied for only the existing synaptic weights. Then the dynamic change in (3) is independent of the growing dynamics in (5).

3 Superimposing Learning

A training set of input vectors $\{x_i \in \Re^n, i = 1, 2, \cdots, N\}$ and the corresponding desired signals $\{y_i \in \Re^m, i = 1, 2, \cdots, N\}$ is considered. N is the number of the training vectors, n the number of inputs, and m the number of outputs. Let t_k denote continuous time in the k-th iteration, $kT \leq t_k < (k+1)T$, $k = 0, 1, 2, \cdots$. Here T is a sampling period. During the k-th iteration, the networks are required to learn a given pair of the inputs and the desired signals, $(x(t_k), y(t_k)) = (x_i, y_i)$, $i \in \{1, 2, \cdots, N\}$.

For an input vector $x(t_k) = x_i$ at the k-th iteration, a square error measure is defined as

$$E_i(k) = \lim_{t_k \to (k+1)T} \|\hat{y}(t_k) - y_i\| \quad (10)$$

where, $\hat{y} \in \Re^m$ is the network outputs. Note that the network outputs may be changed by the dynamic weights even if the input vector is unchanged for the iteration.

In this paper, if the input $x(t_k) = x_i$ is learned successfully within an error tolerance γ, i.e, $E_i(k) \leq \gamma$, this knowledge is considered to be stored *stably* as a *long-term memory* (LTM) even after an additional learning of new inputs is carried out.

Three layer feedforward neural networks are considered here, and the number of neurons in the input (1st), the hidden (2nd) and the output (3rd) layer are n, n_H and m, respectively. The outputs of neurons in each layer are defined as

$$y_p^{(1)}(t_k) = x_p(t_k) \quad (11)$$

$$p = 1, 2, \cdots, n \text{ (Input layer)}$$

$$y_h^{(2)}(t_k) = f_h^{(2)}(\sum_p w_{hp}^{(2)} \cdot y_p^{(1)}(t_k), \theta_h^{(2)}(t_k)) \quad (12)$$

$$h = 1, 2, \cdots, n_H \text{ (Hidden layer)}$$

$$y_q^{(3)}(t_k) = f_q^{(3)}(\sum_h w_{qh}^{(3)} \cdot y_h^{(2)}(t_k), \theta_q^{(3)}(t_k)) \quad (13)$$

$$q = 1, 2, \cdots, m \text{ (Output layer)}$$

where $w_{hp}^{(2)} \doteq w_{hp}^{(2)}(t_k, r_{hp}^{(2)})$ and $w_{qh}^{(3)} \doteq w_{qh}^{(3)}(t_k, r_{qh}^{(3)})$. The network output is defined as $\hat{y}(t_k) \doteq y^{(3)}(t_k)$.

In the following, the superscripts denoting the layer number are omitted and the subscripts imply the corresponding layer numbers, respectively. For example, $w_{hp} \doteq w_{hp}^{(2)}$ and $r_{qh} \doteq r_{qh}^{(3)}$, where $r_{qh}^{(3)}$ denotes the distance between the axon branch of the h-th neuron in the hidden

layer and the dendrite of the q-th neuron in the output layer. Also, for simplicity, *representing* subscripts a and b are introduced, where available combinations are only $(a,b) = (p,h)$ and $(a,b) = (h,q)$. For example, w_{ba} denotes w_{hp} or w_{qh}.

Note that, as mentioned in Section 2.1, the dynamic changing rule in (3) is applied for only existing synaptic weights with $r_{ba} = 0$, i.e, $w_{ba}(t,0)$. Let $\Delta w_{ba}(kT)$ denote the changes in weights $w_{ba}(kT,0)$ given by the discrete-time backpropagation algorithm at the k-th iteration, and $w_{ba}(k\infty,0)$ denote the resulting weights at the k-th iteration such that

$$w_{ba}(k\infty,0) \doteq w_{ba}(kT,0) + \Delta w_{ba}(kT) \qquad (14)$$

Using this notation, for $kT \le t_k < (k+1)T$, the changing forces in (3) are defined as

$$F_{ba}(y_a(kT)) \doteq \frac{1}{\tau_{ba}} w_{ba}(k\infty,0)u_0(t_k - kT) \qquad (15)$$

where, u_0 is the unit step function. Substituting this force for (3), we have

$$w_{ba}(t_k,0) = w_{ba}(k\infty,0) - \Delta w_{ba}(kT)e^{-\frac{t_k - kT}{\tau_{ba}}}$$

Here if $\tau_{ba} \to 0$, this dynamic change is obviously equivalent to the discrete-time backpropagation scheme.

The proposed updating scheme of the time constants τ_{ba} is defined as

$$\tau_{ba}((k+1)T) = \tau_{ba}(kT) + \Delta\tau_{ba}(kT) \qquad (16)$$

Here

1. The initial value is very small: $\tau_{ba}(0) \to 0, \tau_{ba}(0) > 0$.

2. The change $\Delta\tau_{ba}$ may satisfy the following.

   (a) No change for the *diffusing* weights with $r_{ba}(kT) \ne 0$: $\Delta\tau_{ba}(kT) = 0$.

   (b) Monotonous change for the *existing* weights with $r_{ba}(kT) = 0$:

   $$\Delta\tau_{ba}(kT) = \begin{cases} 0, & (E_i(k) > \gamma) \\ \infty, & (E_i(k) \le \gamma) \end{cases} \qquad (17)$$

The dynamic changing rule for biases θ_b are the same as that for the weights w_{ba}.

On the other hand, the structural adaptation proposed here is as follows. In this paper, for simplicity, when one of the synaptic connections to a neuron in the hidden layer from the input layer, or to the output layer from a neuron in the hidden layer is formed, the other available synaptic connections to and from the neuron in the hidden layer are also formed. This means that the creation of new synaptic connections is equivalent to the selection of the corresponding *learnable* neurons in the hidden layer.

The following discrete-time structural *growing* adaptation (a monotonous change) scheme is proposed

1. The initial distance is a bounded number B_{ba} but it is large enough compared with the coefficient μ_{ba}: $r_{ba}(0) = B_{ba}$.

2. If a new synapse is needed, for example if no learnable weight (neuron) exists, a weight with the maximum *diffusion* signal is selected to grow

$$r_{ba}((k+1)T) = \begin{cases} 0, & (a = \alpha, b = \beta) \\ r_{ba}(kT), & (\text{otherwise}) \end{cases}$$

$$(18)$$

where α and β are the subscripts of the maximum *diffusing* weights such that $|w_{\beta\alpha} \cdot y_\alpha| = \max_{b,a} |w_{ba} \cdot y_a|$.

3.1 Restoration of the Past Knowledge by Lateral Inhibition

To implement the restoration strategy, a *lateral inhibition* mechanism is introduced in the hidden layer. Assuming that neurons cannot fire when they receive signals less than a negative threshold with a large bounded absolute value, $N_{th} < 0$, the inhibition weights in the hidden layer is initialized by

$$w_{hl}(0,0) = \begin{cases} N_{th}, & (l \ne h) \\ 0, & (l = h) \end{cases} \qquad (19)$$

where, $l = 1,2,\cdots,n_H$, and they are fixed by $\tau_{hl} \to \infty$, i.e, $w_{hl}(t,0) = w_{hl}(0,0)$.

As the structural adaptation selects new *learnable* neurons in the hidden layer, signals from a hidden neuron should be inhibited only if the synaptic weights of the neuron are not storing the *long-term memory* (LTM) and only when the LTM is recalled (the corresponding input is fed). Therefore, when the LTM is stored, all the possible inhibition connections, w_{hl}, from the neurons which have synapses to the neurons which don't have synapses, are formed by changing the distance between these neurons $r_{hl} = 0$.

On the other hand, to decide whether the input is corresponding to the LTM or not, the inner product of the weights and the inputs, $s_l = w_l \cdot x$, is used. The inner product corresponding to the long-term memory, s_l^L, is stored by the same rule as the dynamic changing weights

$$\frac{ds_l^L(t_k)}{dt} = \frac{-s_l^L(t_k) + s_l(kT)u_0(t_k - kT)}{\tau_{s_l}} \qquad (20)$$

where, $l = 1,2,\cdots,n_H$. The updating rule is similar to that of τ_{ba}: $\Delta\tau_{s_l} \to \infty$ if the l-th neuron has synapses when a LTM is formed ($E_i(k) \le \gamma$), and otherwise $\Delta\tau_{s_l} = 0$.

Then, the lateral inhibition signals, s_{hl}, are defined as

$$s_{hl}(t_k) = \begin{cases} N_{th} & (r_{hl} = 0 \text{ and } s_l - s_l^L) \\ 0 & (\text{otherwise}) \end{cases} \qquad (21)$$

and these signals are summed as an external signal to the neuron in the hidden layer.

4 Simulation Study

The proposed *dynamic and spatial changing weight* neural networks (DSCW-NNs), in combination with the tapped delay lines for inputs and outputs, could be applied for system identification (modeling) tasks. The nonlinear plant used for this system identification task [6] is given by

$$y_P(k_P + 1) = f_P(y_P(k_P), y_P(k_P - 1),$$
$$y_P(k_P - 2), u(k_P), u(k_P - 1)),$$

$k_P = 1, 2, \cdots$, where y_P and u are respectively the plant output and input. The function f_P is given by

$$f_P(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}$$

In this task, a training input vector x and the desired output y are $x = [y_P(k_P) \ y_P(k_P - 1) \ y_P(k_P - 2) \ u(k_P) \ u(k_P - 1)]^T \in \Re^5$, and $y = y_P(k_P + 1) \in \Re$, respectively. To make the training data, a random input sequence uniformly distributed amplitude in the range $[-1, 1]$ is fed to the plant [7]. Learning of each set was terminated when $E_i(k) \leq \gamma$.

The DSCW-NN structure used in this simulation study was composed of 5 distributing elements in the input layer, 15 DSCW-neurons in the hidden layer and 1 DSCW-neuron in the output layer. The nonlinear functions of the hidden neurons were the unipolar sigmoidal function $f_h(x) = 1/(1 + \exp(-x))$, but for the single output neuron, the linear function was used, that is $f_q(x) = x$. The input sequence given by $u(k_P) = \sin(2\pi k_P/250)$, for $k_P \leq 500$, and $u(k_P) = 0.8 \sin(2\pi k_P/250) + 0.2 \sin(2\pi k_P/25)$, for $k_P > 500$, was fed to the network with the tapped delay lines. The outputs of the plant and the proposed network, after learning from only 10 data sets, are shown in Fig. 2. The total number of iterations used was 6034 for 10 data sets when $\gamma = 10^{-20}$
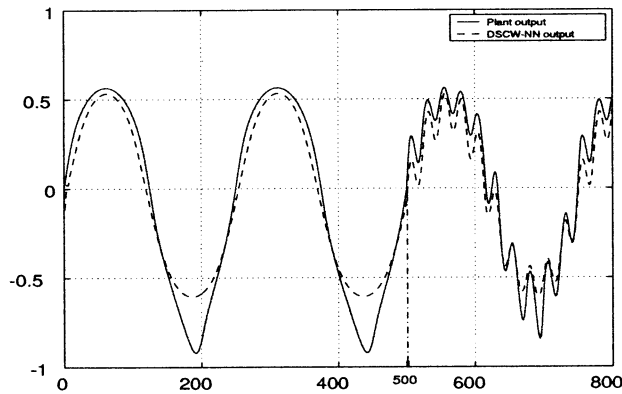
Figure 2: Outputs of the plant (solid line) and of the proposed DSCW-NN (dotted line) after incremental learning of 10 random chosen data sets.

Figure 2 shows approximately the same amount of error between the plant output and the network (model) output as does [7], however, the DSCW-NN required only 10 training data to achieve this degree of accuracy, compared with 1000 training data required for the generalized regression neural network (GRNN) used in [7]. That is, the required number of training data sets was only 1 % to achieve comparable accuracies. Consequently, with the smaller number of training data, for example 10 (Fig. 2), the proposed DSCW-NN is much superior than that of the generalized regression neural network (GRNN) described in [7].

5 Conclusions

In this paper, a new integrated neural model with *dynamic and spatial changing weights* (DSCWs) for incremental learning tasks using a novel learning, structural adaptation and superimposing strategies has been developed. The proposed DSCW-NN never forgets the past knowledge. This is a strict incremental learning that cannot be achieved by the existing incremental learning schemes.

The proposed model is mostly inspired by biological aspects and thus to some extent can be a model of actual brain functions [4]. Possible extensions to the radial basis function (RBF) neurons, dynamic neurons with feedback, and self-learning methods such as Hebbian rule should lead to some interesting investigations in the future work.

References

[1] Fu, L., H.H. Hsu and J.C. Principe (1996). Incremental backpropagation learning networks. *IEEE Trans. Neural Networks* **7**, 757–761.

[2] Fu, L. (1996). Incremental knowledge acquisition in supervised learning networks. *IEEE Trans. Syst., Man, Cybern. A* **26**, 801–809.

[3] Hiramoto, M., Y. Hiromi, E. Giniger and Y. Hotta (2000). The drosophila netrin receptor frazzled guides axons by controlling netrin distribution. *Nature* **406**, 886–888.

[4] Honma, N., T. Kamauchi, K. Abe and H. Takeda (1999). Auto-learning by dynamical recognition networks. In: *Proc. of IEEE SMC*. Vol. III. Tokyo. pp. 211–216.

[5] K.Yamauchi, N. Yamaguchi and N. Ishii (1999). Incremental learning methods with retrieving of interfered patterns. *IEEE Trans. Neural Networks* **10**, 1351–1365.

[6] Narendra, K.S. and K. Parthasarathy (1990). Identification and control of dynamic systems using neural networks. *IEEE Trans. Neural Networks* **1**, 4–27.

[7] Specht, D.F. (1991). A general regression neural network. *IEEE Trans. Neural Networks* **2**, 568–576.

Analysis of Transient Change in Bio Reactive Rate Constants
---Introduction of Method by Zhou and Szabo

Hirohumi, Hirayama and Yoshimitsu Okita*

Department of Public Health Asahikawa Medical College
Higashi 2-1, Midorigaoka , Asahikawa city 078, Japan.
hirayama @asahikawa-med.ac.jp
Shizuoka university*.

Abstract
    We introduce a method developed by Zhou and Szabo (1996) for evaluating the time dependent character of bio chemical reactions. The The probability density P(x,t) for the molecular pair to take the relative configuration x at time t was described by the Green function and reactivity function that depends upon the configuration of the molecular pair. In the present work, we present a computational approximation for the biochemical reaction characterized by the step reactive function. We utilized the numerical inversion technique for the Laplace transformation. The rate constant ks decreased as the time passed and it also decreased as the relative translational diffusion constant decreased. The present method when extended will be avaliable for evaluating the rate constant between bio molecules having an arbitrary configurations.

    key words. Rate constant, Bio chemical reaction, Molecular interaction, Green funciton

1.Introduction.

The time dependent property of biochemical reaction is characterized by rate constant k. Most of theoretical and experimental works utilize the rate constant for the steady state reaction.. In such situation, the concentrations of substrate, enzyme-substrate complex and product are achieving steady values. In actual biological reactions ,however, the concentration of the species is not constant but changes in time and space where the substrate distribute[1]. As the reaction proceeds, the concentration of the substrate decreases. The amount of activated enzyme decreases and the total number of the active site are decreased because the the substrate -occupied sites of enzyme increase. Consequently, the substrate concentration near around the enzyme is decreased while that away from the target enzyme must be changed a little. Hence, rigorously speaking the rate constant is not constant but depends on the time that has passed.

In the present study, we introduce a method developed by Zhou and Szabo in 1996 [2] for simulating the time dependent rate coefficients of diffusion controlled reactions.

2. Mathematical Method.

We utilize the equilibrium distribution approximation technique developed by Zhou and Szabo 1996 [2].
The probability density function for the molecular pair to adopt configuration x at time t is P(x,t). It can be can be expressed by

$$P(x,t) = peq(x) \, w \, (t) \qquad \text{------(1)}$$

where

$$peq(x) = 1/(8 \pi^2) \exp(\, [\, - \beta \, U(q) \,] \, peq(g1,g2) \qquad \text{----(2)}$$

where U is the interaction potential which approaches to zero as the inter molecular distance increases. $q = (\, r, \, \Omega 1, \, \Omega 2)$
r is the displacement vector of the configuration of the molecular pair. $\Omega 1, \Omega 2$ are the orientations of the molecules comprising the molecular pair. g1 and g2 are the gating states of these molecules.

$$\beta = 1/(kB \, T) \qquad \text{------(3)}$$

and

$$peq(g1,g2) \qquad \text{------(4)}$$

is the equilibrium gating distribution. kB is the Boltzmann coefficient, T is the absolute temperature.
w(t) is independent of configuration within the reactive part of the contact surface.

The general relation concerning to the probability density to the non reactive Green function
Go(x,t I,xo, t=o) is

$$P(x,t) = peq(x) - \int dt' *$$
$$* \int Go(x,t \ I \ x',t') K(x') P(x',t') dx'$$
------(5)

K(x) is the reactivity function.

Go(q,t I qo,t=0) is the Green function that satisfies the reflecting boundary condition over an entire contact surface. Operating on the Laplace transformation,

$$P^{\wedge}(x,s) = peq(x) /s -$$
$$\int Go^{\wedge}(x,s \ I \ x') K(x') peq(x') w^{\wedge}(s)dx'$$
------(6)

The radiation boundary condition holds on average over the reactive part of the contact surface,
this gives the identity

$$\int K(x)P(x,t) \ dx = \int K(x) \ peq(x) \ w(t)dx = k(0) \ w(t)$$
------(7)

Substituting the Laplace transform,we have

$$w(s)^{\wedge} = k(0) /s \ [\ k(0) + \int dx \ K(x) \ Go^{\wedge}(x,s \ I \ x') *$$
$$*K(x') \ peq(x') \ dx' \ ]$$
------(8)

The Laplace transform of the rate coefficient is obtained as

$$1/ \ [\ sk^{\wedge}(s) \ ] = 1/k(0) + [\int dx \ K(x) \ Go^{\wedge}(x,s \ I \ x')*$$
$$K(x') \ peq(x') \ dx' \ ] / [\int K(x) \ peq(x) \ dx \ ]^2$$
------(9)

In the present work,we show the simplest example for the step function reactivity. The reactivity function is given by

$$K(r) = 1/r$$

For the simplest case in the absence of gating is given by the Laplace transform of the rate constant is

$$1/[\ s \ k^{\wedge}dc(s) \ ] = (D / s)^{5/2} /[\ 8 \ \pi D \ (R1^3 - R^3 \ )^2] *$$
$$*[\ 1 - s \ R1^2 \ /D + 2 \ (s/D)^{3/2} \ (R1^3 - R^3 \ )/3 -$$
$$(\ 1 -(\ sR^2/*D)^{1/2}/(\ 1 +(\ s \ R^2/D)^{1/2} \ )*$$
$$*(\ 1 + (\ s \ R1^2 \ /D)^{1/2} \ )^2 \ exp(\ -2(s/D)^{1/2} \ \varepsilon \ ) \ ]$$
------(10)

where we set ε /R =1.0. We set following parameters as the standard state,
    D (diffusion constant)=$10^{\wedge}(-6)$ mole/sec,
    t = $10^{\wedge}(-6)$ sec
    R = $10^{\wedge}(-6)$ cm and R1 = 1.5*R cm .

3. Results.

Fig 1 shows the temporal changes in log10[k(t)] as function of changes in the D in log10. The time was set at $10^{\wedge}(-8)$ to $10^{\wedge}(-2)$. As the time in log10(t) increased, the log10(k(t)) decreases rapidly. At an early period, for t = $10^{\wedge}(-8)$, as the diffusion constant decreased from $10^{\wedge}(-4)$

to $10^{\wedge}(-8)$, the rate constant decreased weakly. For the time passed at t = $10^{\wedge}(-2)$,on the contrary, the rate constant increased as the diffusion constant decreased.

Fig 1
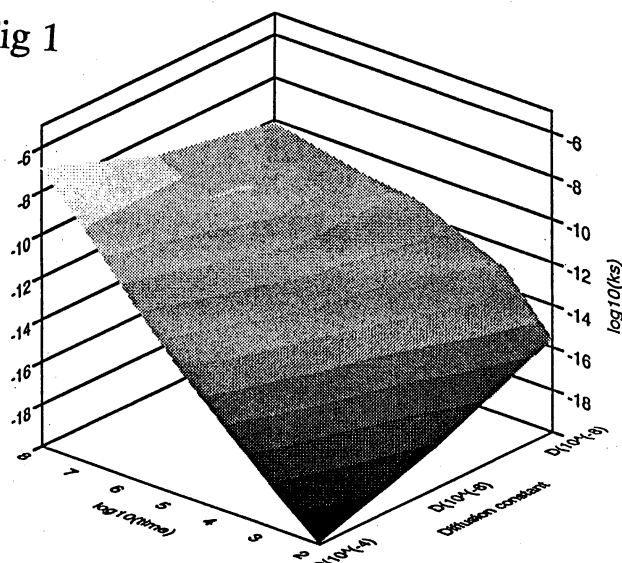
Fig 2
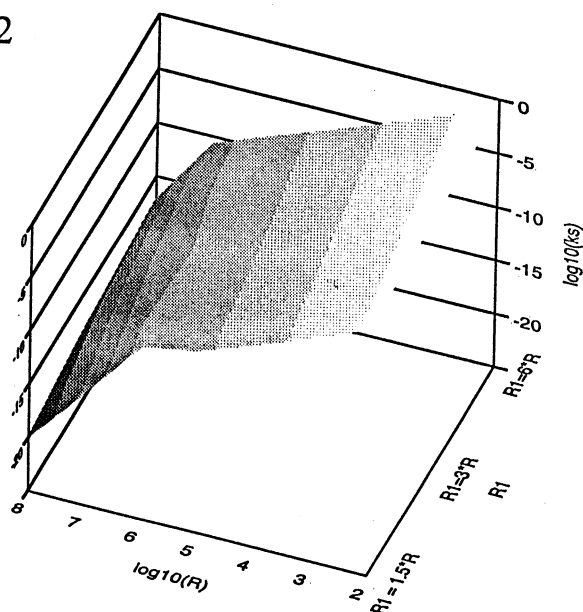
Fig 2 shows dependency of k(t) on the relative positions between R1 and R. The rate constant decreased as R increased.

4. Discussion and Conclusion.

The present work introduced a simple method for approximating the time dependency of rate constant More complex reactivity function other than step function should be employed to describe realistic bio chemical reactions.

5. References.

1. Smoluchowski,M,V.Versuch einer mathematichen theorei der koagulations kinetics kolloider losungen. Z. Phyc. Chem. vol 92. pp129-. 1917.
2.Zhou HX and Szabo A. ; Theory and simulation of the time dependent rate constants of diffusion influences reactions. Biophysical Journal. vol 71. pp 2440. 1996.
3. Stechfest, H. Numerical inversion of Laplace transforms. Comm. ACM. vol 13. pp 47-49. 1970.

Reaction Field Tensor Analysis for a Bio Molecular Particle.

Hirohumi, Hirayama   and Yoshimitsu, Okita*
Department of Public Health Asahikawa medical college.
Tel. 0166-65-2111. ex 2411
    E mail ,   hirayama@asahikawa-med.ac.jp
* Shizuoka University

Abstract

The present technical report inform a mathematical method for computing the electrical field and reaction field tensor induced by a spherical bio molecular particle having finite electrical potential. Molecular dipole moments in a d electric medium correlate over large distances among the bio molecules. On the basis of the electrical continuum theory , the electric Hertz vector was expanded in a polar coordinates. The potential vector was expanded by the series of spherical harmonics and spherical Bessel functions. The coefficients in these functions were determined by the continuity of boundaries among the molecules and the medium. Based on the Maxwell solution, we also have the reaction field potential in a definite form. The present method,when extended will be available for analyzing the electrical constitutional changes in a bio molecular particle.

Electrical potential field. Reaction field tensor. spherical cavity, Hertz vector, spherical harmonics.

1. Introduction.

A complex biological interaction among the bio molecules induces reorganization of the structure of the reacting bio molecules. This may in part originated from the changes in the intra molecular reaction filed. The present paper shows a method introduced by Titulaer (1974).

2. Mathematical method.

Analysis was confined on an arbitrary sample V of a di-electric medium with d- electric constant $\varepsilon\, 2(\omega)$.
The sample is embedded in an infinite di-electric medium with di-electric constant $\varepsilon\, 1(\omega)$.   On this, a spatially homogeneous external field which oscillates with ω acts on a spherical region. (radius a, center at ri)
The target spherical region is entirely within the sample region V. The dependence of the external field Eo(R,t) on the space and time is

Eo(R,t) = Eo(t) h(R;r)

= Eo exp(- i ω t) h(R; r)   --- (2,1)

h(R; r) is the characteristic function of the sphere with radius a   around r.

h(R;r) =1 for I R - r I < a

h(R;r) =0   for I R - r I > a

As a consequence of the imposed field (2-1), another physical region will have a net dipole moment <m(r',t)>Eo which exists in the same region V having radius a and its center at r'.   <m(r',t)>Eo depends

linearly on the components of Eo.

<m(r',t)>Eo = X(r,r' ;ω) · Eo exp(-iω t) (2,3)

In this region, the net dipole moment m(r',t) of the spherical region around r is expressed by

m(r',t) = ∫ dR Σ δ [R - ri(t)] μ i(t) h(R; r') --(2,4)
                v   i

Σ extends over all molecules in the sample.
ri(t) is the vectorial position of i th molecule.
μ i(t) is the dipole moment of i th molecule.

The apparent di pole < m*(r,t)>Eo at r gives rise to an electric field E*(r',t) at r' which is denoted by

E*(r',t) = 1/[ε 2(ω)] D(r,r';ω) . < m*(r,t)>Eo
                                                    --(2-5)

This field evokes a net di pole moment in a small spherical region of radius a about r' which is

< m(r',t)>Eo = {(ε 2(ω)-1)/4 π} 4/3 π a³ E*(r',t)

Combining 2-5 and 2 -6, relation is given

< m(r',t)>Eo ={(ε 2(ω)-1)/3 ε 2(ω)}a³
D(r,r';ω) . < m*(r,t)>Eo   ---- (2-7)

Eliminating of < m*(r,t)>Eo in 2-7 by the relation of

< m*(r,t)>Eo = { 3 ε 2(ω)/ [2 ε 2(ω)+1] } a³ < m(r,t)>

results in the relation of $< \mathbf{m}('\mathbf{r},t)> \mathrm{Eo}$ and Eo

$< \mathbf{m}(\mathbf{r}',t)> \mathrm{Eo} =1/ \varepsilon_2(\omega) [\{(\varepsilon_2(\omega)-1)/3\}a^3]^2$
$\quad\quad \mathbf{D}(\mathbf{r},\mathbf{r}';\omega) . \quad \mathrm{Eo} \quad ----- (2-8)$

The tensor $\mathbf{D}(\mathbf{r},\mathbf{r}';\omega)$ defined by (2-5) depends on the shape of the sample V and the di electric constant of the external region $\varepsilon_2(\omega)$. The tensor D is decompose to

$\mathbf{D}(\mathbf{r},\mathbf{r}';\omega) = \mathbf{D\infty}(\mathbf{r},\mathbf{r}';\omega) + \mathbf{Rv}(\mathbf{r},\mathbf{r}';\omega) --(2-9)$

$\mathbf{D\infty}(\mathbf{r},\mathbf{r}';\omega)$ reflects the direct field
$\mathbf{D\infty}(\mathbf{r},\mathbf{r}';\omega) =- (\partial / \partial \mathbf{r}' \ \partial / \partial \mathbf{r} - k_2^2 \mathbf{I})$
$\quad\quad \exp(i k_2 |\mathbf{r}-\mathbf{r}'|)/|\mathbf{r}-\mathbf{r}'|$
k_2 is the wave number of a wave with frequency ω in the medium with di electric constant $\varepsilon_2(\omega)$ and magnetic constant $\mu_2(\omega)$ as
$k_2 = [\omega/c] [\varepsilon_2(\omega) \mu_2(\omega)]^{1/2}$

**** Expressions for orientational correlations in a spherical sample.

Here, the reaction field tensor $\mathbf{Rv}(\mathbf{r},\mathbf{r}';\omega)$ is determined for the case of a spherical sample V of radius R, with di electric constant $\varepsilon_2(\omega)$ and magnetic permeability μ_2. The sample is embedded in a medium characterized by $\varepsilon_1(\omega)$ and magnetic permeability μ_1.
Starting from the electric Hertz vector associated with a di pole of strength $\mathbf{m}$ at vectorial position $\mathbf{r}$

$\Pi(\mathbf{r}) = 1/ \varepsilon_2(\omega) \exp(i k_2 |\mathbf{r}-\mathbf{r}'|)/|\mathbf{r}-\mathbf{r}'|$
$\quad\quad \mathbf{m} \exp(-i\omega t) \quad ------(3.1)$

From this expression, the electric and magnetic fields are obtained by means of the formulas

$\mathbf{E}(\mathbf{r}') = \nabla \times \nabla \times \Pi(\mathbf{r}')$
$\mathbf{H}(\mathbf{r}') = \varepsilon(\omega)/c \ \nabla \times \partial \Pi(\mathbf{r}')/\partial t \quad ---(3.2)$
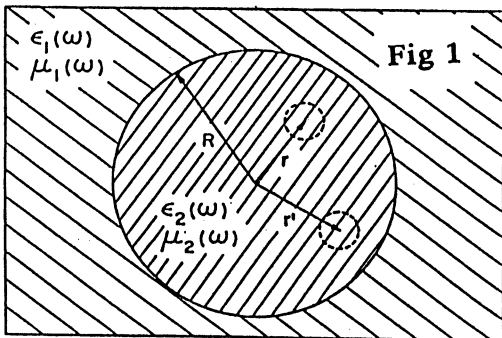
Fig 1

Fig 1. The sample V is taken to be a sphere with radius R. Two spherical regions with radius a around $\mathbf{r}$ and $\mathbf{r}'$ are considered.

For the first step, (3.1) is expanded in terms of a set of vector solutions of Helmholtz's equation. This solution has to be singular at the origin where at the center of the spherical sample under examination. The specific form of those solutions,
$\quad \mathbf{L}^{(3)}_{k,L,m}(\mathbf{r}'), \mathbf{M}^{(3)}_{k,L,m}(\mathbf{r}')$ and
$\quad \mathbf{N}^{(3)}_{k,L,m}(\mathbf{r}')$
are given explicitly in the Appendix. The basic expansion by the spherical harmonics are given by Hansen (1934).

$\exp(i k |\mathbf{r}-\mathbf{r}'|)/|\mathbf{r}-\mathbf{r}'| = \sum \sum 1/[L(L+1)]$
$\quad\quad\quad\quad\quad\quad\quad\quad_{L,m}$
$[\mathbf{L}^{(3)}_{k,L,m}(\mathbf{r}') \mathbf{L}^{(1)*}_{k,L,m}(\mathbf{r})$
$\quad +\mathbf{M}^{(3)}_{k,L,m}(\mathbf{r}') \mathbf{M}^{(1)*}_{k,L,m}(\mathbf{r})$
$\quad\quad +\mathbf{N}^{(3)}_{k,L,m}(\mathbf{r}') \mathbf{N}^{(1)*}_{k,L,m}(\mathbf{r})] --(3-3)$

The electric field of a dipole $\mathbf{m}$ at $\mathbf{r}$ at points $\mathbf{r}'$ with
$|\mathbf{r}'| > |\mathbf{r}|$ can be determined by (3-1) 3-2 and 3-3 conjunction with the following curl vectorial properties.

$\nabla \times \mathbf{L}^{(i)}_{k,L,m}(\mathbf{r}) =0$
$\nabla \times \mathbf{M}^{(i)}_{k,L,m}(\mathbf{r}) = k \ \mathbf{N}^{(i)}_{k,L,m}(\mathbf{r})$
$\nabla \times \mathbf{N}^{(i)}_{k,L,m}(\mathbf{r})= k \ \mathbf{M}^{(i)}_{k,L,m}(\mathbf{r}) ---(3-4)$

Applying these relations result in (3-2)

$\mathbf{E}(\mathbf{r}') = 4 \ \pi i k_2^3 / \varepsilon_2(\omega) \sum \sum 1/[L(L+1)]$
$\quad\quad\quad\quad\quad\quad\quad\quad_{L,m}$
$[\mathbf{M}^{(3)}_{k_2,L,m}(\mathbf{r}') \mathbf{M}^{(1)*}_{k_2,L,m}(\mathbf{r})$
$+\mathbf{N}^{(3)}_{k_2,L,m}(\mathbf{r}') \mathbf{N}^{(1)*}_{k_2,L,m}(\mathbf{r})] . \ \mathbf{m} \exp(-i\omega t)--(3-5)$

The reaction field corresponding to this field is obtained by comparing this expression with the standard Maxwellian solutions. Then,

$\mathbf{E}^{(r)}(\mathbf{r}') = 4 \ \pi i k_2^3 / \varepsilon_2(\omega) \sum \sum 1/[L(L+1)]$
$\quad\quad\quad\quad\quad\quad\quad\quad_{L,m}$
$[R_L^{(m)} (\varepsilon_2, \mu_2 ; \varepsilon_1, \mu_1) \mathbf{M}^{(1)}_{k_2,L,m}(\mathbf{r}') \mathbf{M}^{(1)*}_{k_2,L,m}(\mathbf{r})$
$+R_L^{(e)} (\varepsilon_2, \mu_2 ; \varepsilon_1, \mu_1) \mathbf{N}^{(1)}_{k_2,L,m}(\mathbf{r}') \mathbf{N}^{(1)*}_{k_2,L,m}(\mathbf{r})]$
$\quad . \ \mathbf{m} \exp(-i\omega t) \quad\quad ----(3-6)$

The coefficients of $R_L^{(m)} (\varepsilon_2, \mu_2 ; \varepsilon_1, \mu_1)$ and
$R_L^{(e)} (\varepsilon_2, \mu_2 ; \varepsilon_1, \mu_1)$ are determined by the continuous boundary conditions and are given in the Appendix. The reaction field tensor R for a sphere is given by

$\mathbf{R}^{**}(\mathbf{r}, \mathbf{r}' ; \omega t) = 4 \ \pi i k_2^3 \sum \sum 1/[L(L+1)]$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad_{L,m}$
$[R_L^{(m)} (\varepsilon_2, \mu_2 ; \varepsilon_1, \mu_1) \mathbf{M}^{(1)}_{k_2,L,m}(\mathbf{r}') \mathbf{M}^{(1)*}_{k_2,L,m}(\mathbf{r}')$
$+R_L^{(e)} (\varepsilon_2, \mu_2 ; \varepsilon_1, \mu_1) \mathbf{N}^{(1)}_{k_2,L,m}(\mathbf{r}') \mathbf{N}^{(1)*}_{k_2,L,m}(\mathbf{r}')]$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad -----(3-7)$

For actual computation, we used following physical constants
$\quad$ c (the speed of light) = 2.9979 10^8m/sec
$\quad \mu_0$ (magnetic permeability)= 1.2566 10^{-6} H/m
$\quad \varepsilon$ (di electric constant of water) = 80

Fig 2

Ppyst'*10^15

Fig 3

Pstpy'*10^15

Fig 4

Ppyr'*10^15
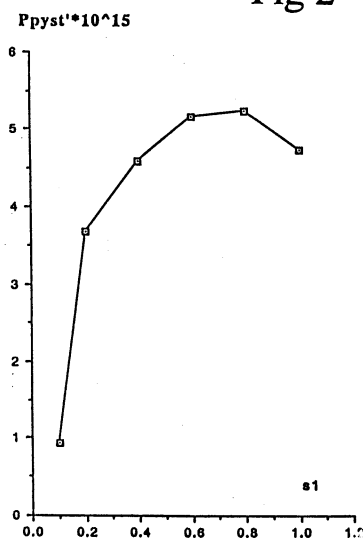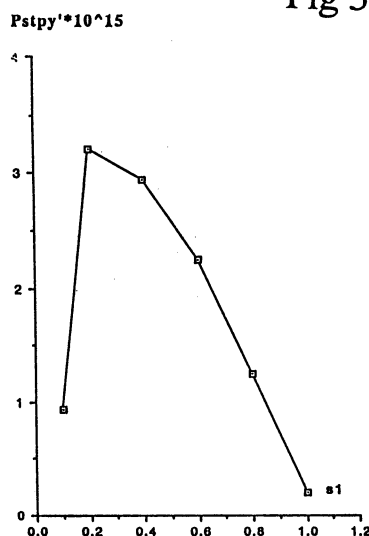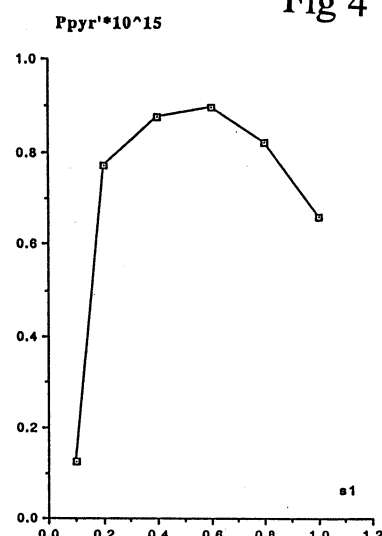
Fig 5

Prr'*10^15
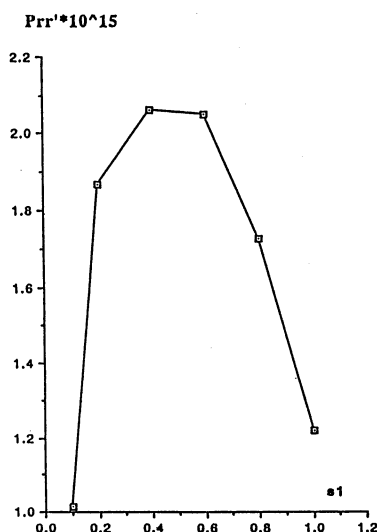
3. Results and Discussion..

Fig 2 to Fig 5 shows $P_{\phi\theta}$, $P_{\theta\phi}$, $P_{\phi r'}$ and Prr' respectively as functions of circumferential angle θ in radius. All the components of the reaction tensor showed definite peak values with different circumferential angles. This tendency indicates that during the molecular structural change, an intra molecular organization will direct to proceed a given definite conformation as the reaction tensor directs. The participation of the magnetic reactive field might be negligible for bio molecular material due to the di electric constant. The present method, when extended will be available for predicting the conformation change in reaction to the bio molecular interaction that produces a reaction field tensor.

4. Conclusion.

Reaction tensor method will be available for predicting the molecular structural change of the bio chemical reaction.

5. Reference

1. 1. Titulaer, U.M. J. Chemical. Physics. vol 60. pp2703. 1974.

----- **Appendix** ----------------------

A complete set of solutions of Helmholtz's equation for a vector field f(r) satisfies

$$\nabla^2 f(r) + k^2 f(r) = 0 \quad --A1$$

A complete set of solutions of this equation is obtained by taking the functions

$$L^{(i)}{}_{k,L,m}(r) = 1/k \, [L(L +1) \,]^{1/2}$$
$$\nabla \, [\, z^{(i)}{}_{L}(kr) \, Y_{L,m}(\theta, \phi) \,] \quad --A2$$
$$M^{(i)}{}_{k,L,m}(r) = \nabla \times [\, r \, z^{(i)}{}_{L}(kr) \, Y_{L,m}(\theta, \phi) \,] --A2$$
$$N^{(i)}{}_{k,L,m}(r) = 1/k \, \nabla \times M^{(i)}{}_{k,L,m}(r) \quad ---A2$$

for all non negative l and all m between - L and + L. The symbol $z^{(1)}{}_{L}(kr)$ denotes the spherical Bessel, Neumann or Hankel functions

$$z^{(1)}{}_{L}(kr) = j_{L}(kr) \quad z^{(2)}{}_{L}(kr) = n_{L}(kr)$$
$$z^{(3)}{}_{L}(kr) = h^{(1)}{}_{L}(kr) \quad z^{(4)}{}_{L}(kr) = h^{[2]}{}_{L}(kr) \quad ---A3$$

$Y_{L,m}(\theta, \phi)$ are the spherical harmonics, normalized to unity on a sphere. The functions M and N are purely solenoidal and are natural basis for the description of electric and magnetic fields. In the spherical coordinates,

$$N_{k2,L,m}(r') = i r \, N_{k2L,m r}(r') + i\theta \, N_{k2L,m\,\theta}(r')$$
$$+ i\phi \, N_{k2L,m\,\phi}(r')$$
$$N_{k2,L,m}(r) = i r \, N_{k2L,m r}(r) + i\theta \, N_{k2L,m\,\theta}(r)$$
$$+ i\phi \, N_{k2L,m\,\phi}(r)$$

$$N_{k2L,m r}(r') = L\,(L+1)/(k2\,r') \, YLm(\theta\,\phi) \, z_{L}(k2r')$$
$$= L\,(L + 1)/(k2\,r') \, \sqrt{(2\,L + 1)/(4\,\pi)}{}^{*}$$

$$* \text{gamma}(L - \text{abs}(m) +1)/\text{gamma}(L + \text{abs}(m) +1)$$
$$P_L^{\text{abs}(m)}(\cos\theta)\,\exp^{(im\phi)}$$
$$j_L(k_2 r')\,(-1)^{\wedge}(m+\text{abs}(m))/2$$
$$= \exp^{(im\phi)}\,N_r(r')$$

$$N_{k_2 L,m\,\theta}(r') = L(L+1)/(k_2 r')\,\partial\, Y_{Lm}(\theta\,\phi)\,\partial\,\theta\,\underline{z_L}(k_2 r')$$
$$= L(L+1)/(k_2 r')\,\sqrt{(2L+1)/(4\pi)}*$$
$$* \text{gamma}(L-\text{abs}(m)+1)/\text{gamma}(L+\text{abs}(m)+1)$$
$$\partial\, P_L^{\text{abs}(m)}(\cos\theta)/\,\partial\,\theta\quad \exp^{(im\phi)}$$
$$1/r'\,\partial\,(r'\,j_L(k_2 r')\,/)\,\partial\,r'\,(-1)^{\wedge}(m+\text{abs}(m))/2$$
$$= \exp^{(im\phi)}N_\theta(r')$$
$$N_{k_2 L,m\,\phi}(r') = L(L+1)/(k_2 r'\sin(\theta))$$
$$\partial\, Y_{Lm}(\theta\,\phi)\,\partial\,\phi\,\underline{z_L}(k_2 r')$$
$$= L(L+1)/(k_2 r'\sin(\theta))\,\sqrt{(2L+1)/(4\pi)}*$$
$$* \text{gamma}(L-\text{abs}(m)+1)/\text{gamma}(L+\text{abs}(m)+1)$$
$$i\,m\,P_L^{\text{abs}(m)}(\cos\theta)\quad \exp^{(im\phi)}$$
$$1/r'\,\partial\,(r'\,j_L(k_2 r')\,/)\,\partial\,r'\,(-1)^{\wedge}(m+\text{abs}(m))/2$$
$$= i\,\exp^{(im\phi)}N_\phi(r')$$

$$M_{k_2,L,m}(r') = i_r\,M_{k_2 L,m r}(r') + i_\theta\,M_{k_2 L,m\,\theta}(r')$$
$$+ i_\phi\,M_{k_2 L,m\,\phi}(r')$$

$$M_{k_2 L,m r}(r') = 0$$

$$M^{(i)}_{kLm}(r) = \nabla \times [\,r\,Z^{(i)}_L(kr)\,Y_{L,m}(\theta,\phi)\,]$$

The present report requires only i =1 and 3
$$M^{(i)}_{kLm}\,;\,r\quad(\text{indicating } r \text{ component}) = 0$$

$$M^{(i)}_{kLm}(r)\,;\,\theta\quad(\text{indicating circumferential component})$$
$$= 1/(\sin\theta\,[\,\partial\,Y_{L,m}(\theta,\phi)/\,\partial\,\phi\,]\,Z^{(i)}_L(kr)$$
$$= 1/(\sin\theta)\,[\,\partial\,/\,\partial\,\phi\,(-1)^{(m+\text{abs}(m))/2}$$
$$\sqrt{(2L+1)\,(L-\text{abs}(m))!\,/(4\pi\,(L+\text{abs}(m))!}$$
$$P_L^{(m)}(\cos\theta)\,e^{im\phi}\,]$$
$$[\,Z^{(i)}_L(kr) = j_L(kr),\,h^{(i)}_L(kr)\,]$$

where $m = L, L-1, L-2, ---0, 1, 2, -L+1, L$

$$= 1/(\sin\theta)\,[\,j_L(kr),\,h^{(i)}_L(kr)\,]$$
$$i\,m\,(-1)^{(m+\text{abs}(m))/2}$$
$$\sqrt{(2L+1)\,(L-\text{abs}(m))!\,/(4\pi\,(L+\text{abs}(m))!}$$
$$P_L^{(m)}(\cos\theta)\,e^{im\phi}$$
$$= i\,e^{im\phi}\,M_\theta(r')$$

$$M^{(i)}_{kLm}(r)\,;\,\phi\quad(\text{indicating polar component})$$
$$= -[\,\partial\,Y_{L,m}(\theta,\phi)/\,\partial\,\theta\,]\,Z^{(i)}_L(kr)$$
$$= -\partial\,/\,\partial\,\theta\,[\,(-1)^{(m+\text{abs}(m))/2}$$
$$\sqrt{(2L+1)\,(L-\text{abs}(m))!\,/(4\pi\,(L+\text{abs}(m))!}$$
$$P_L^{(m)}(\cos\theta)\,e^{im\phi}\,]\,Z^{(i)}_L(kr)$$
$$= -\{\,j_L(kr),\,h^{(i)}_L(kr)\,\}(-1)^{(m+\text{abs}(m))/2}$$
$$\sqrt{(2L+1)\,(L-\text{abs}(m))!\,/(4\pi\,(L+\text{abs}(m))!}$$
$$e^{im\phi}\,\partial\,P_L^{(m)}(\cos\theta)/\,\partial\,\theta$$
$$= e^{im\phi}\,M_\phi(r')$$

$$N^{(i)}_{kLm}(r)\,;\,\theta\quad(\text{indicating circumferential component})$$
$$= 1/(k\,r)\,[\,\partial\,Y_{L,m}(\theta,\phi)/\,\partial\,\theta\,]\underline{Z^{(i)}_L}(kr)$$

In the expression for $N^{(i)}$ we have abbreviated

$$\underline{Z^{(i)}_L}(kr) = 1/r\,d/dr\,[\,r\,Z^{(i)}_L(kr)\,]$$

,for i =1
$$= 1/r\,d/dr\,[\,r\,Z^{(1)}_L(kr)\,]$$
$$= 1/r\,d/dr\,[\,r\,j_L(kr)\,]$$
$$= 1/r\,[\,j_L(kr) + r\,\partial\,/\,\partial\,r\,[\,j_L(kr)\,]\,]$$
where
$$j_L(kr) = \sqrt{\pi/(2\,k\,r)}\,J_{L+1/2}(kr)$$

$$\partial\,/\,\partial\,r\,[\,j_L(kr)\,] = \partial\,j_L(kr)/\,\partial\,z\,\,\partial\,(kr)/\,\partial\,r$$
$$= k\,\partial\,/\,\partial\,z\,[\,\sqrt{\pi/(2\,z)}\,J_{n+1/2}(z)\,]_{z=kr}$$
$$= k\,[\,\sqrt{\pi/2}\,\partial\,/\,\partial\,z\,(z^{-1/2}\,J_{n+1/2}(z)\,)]_{z=kr}$$

$$= k\,\sqrt{\pi/2}\,[\,-1/2\,z^{-3/2}\,J_{n+1/2}(z)$$
$$+ z^{-1/2}\,\partial\,/\,\partial\,z\,J_{n+1/2}(z)\,]_{z=kr}$$

By the Formulae of Bessel functions
$$Z_n'(x) = n\,Z_n/x - Z_{n+1}$$
$$= Z_{n-1} - n\,Z_n/x$$

$$J_{n+1/2}(z)' = (n+1/2)\,J_{n+1/2}(z)/z - Z_{n+3/2}$$
$$= J_{n-1/2} - (n+1/2)\,J_{n+1/2}(z)/z$$

$$h^{(i)}_L(k_2 r) = j_L(k_2 r) + i\,n_L(k_2 r)$$
$$\underline{Z^{(i)}_L}(kr) = 1/r\,d/dr\,[\,r\,Z^{(i)}_L(kr)\,]$$

$$\underline{h^{(1)}_L}(k_2 r) = 1/r\,d/dr\,[\,r\,h^{(1)}_L(k_2 r)\,]$$
$$= 1/r\,j_L(k_2 r) + k_2\,j'_L(x)\,)x=k_2 r$$
$$+ i\,[\,1/r\,n_L(k_2 r) + k_2\,n'_L(x)\,)x=k_2 r$$
Thus,
$$h^{(1)}_L(k_1 r) = j_L(k_1 r) + i\,n_L(k_1 r)$$
$$h^{(1)}_L(k_2 r) = j_L(k_2 r) + i\,n_L(k_2 r)$$
$$\underline{h^{(1)}_L}(k_2 r) = d j_L(k_2 r) + i\,d n_L(k_2 r)$$

where
$$d j_L(k_2 r) = 1/r\,j_L(k_2 r) + k_2\,j'_L(k_2 r)\,(x = k_2 r)$$
$$d n_L(k_2 r) = 1/r\,n_L(k_2 r) + k_2\,n'_L(k_2 r)\,(x = k_2 r)$$
$$j^{(1)}_L(k_2 r) = 1/r\,j_L(k_2 r) + \partial\,j_L(k_2 r)/\,\partial\,r = d j_L(k_2 r)$$

$M^{(i)}$ is purely transversal field while $N^{(i)}$ contains both a longitudinal and a transversal component. With the set (a2) of solutions of Helmholtza equations, we can construct a complete set of Maxwell equations.

The solutions of magnetic type
$$E(r) = A^{(m)}\,M^{(i)}_{kLm}(r)\,\exp(-i\omega t)$$
$$H(r) = A^{(m)}\,(k\,c/(i\,\omega\,\mu))N^{(i)}_{kLm}(r)\,\exp(-i\omega t)$$
The solutions of electric type
$$E(r) = A^{(e)}\,N^{(i)}_{kLm}(r)\,\exp(-i\omega t)$$
$$H(r) = A^{(e)}\,(k\,c/(i\,\omega\,\mu))M^{(i)}_{kLm}(r)\,\exp(-i\omega t)$$
For the calculation of reaction fields inside a sphere, we need solutions of Maxwell's equations for which the electric field has the form

$$E(r) = A^{(m)}\,[\,M^{(3)}_{k_2 Lm}(r) + R_L^{(m)}(\varepsilon_2,\mu_2;\varepsilon_1,\mu_1)$$
$$M^{(1)}_{k_2,L,m}(r)\,]\,\exp(-i\omega t)\quad\text{for } r << R$$
$$E(r) = A^{(m)}\,Q_L^{(m)}(\varepsilon_2,\mu_2;\varepsilon_1,\mu_1)\,[\,M^{(3)}_{k_2 Lm}(r)\,]$$
$$\exp(-i\omega t)\quad\text{for } r >> R$$
or
$$E(r) = A^{(e)}\,[\,N^{(3)}_{k_2 Lm}(r) + R_L^{(e)}(\varepsilon_2,\mu_2;\varepsilon_1,\mu_1)$$
$$N^{(1)}_{k_2,L,m}(r)\,]\,\exp(-i\omega t)\quad\text{for } r << R$$
$$E(r) = A^{(e)}\,Q_L^{(e)}(\varepsilon_2,\mu_2;\varepsilon_1,\mu_1)\,[\,N^{(3)}_{k_2 Lm}(r)\,]$$
$$\exp(-i\omega t)\quad\text{for } r >> R$$

Several Information Space Conditions for the Evolution of Complex Forms of Life

Hideaki Suzuki
Human Information Science Labs.
ATR International
Seika-cho, Soraku-gun, Kyoto 619-0288 Japan

Naoaki Ono
Dept. of Physics, Grad. Sch. of Sciences
Kyoto University
Kyoto 606-8502 Japan

Abstract

In order for an artificial life (alife) system to evolve complex creatures, the information space and the reaction rules of the system have to satisfy several sufficient conditions. From a comparison with the biochemical reaction space in living things, the paper proposes seven conditions or functions for this purpose, and assesses a variety of alife systems in terms of the proposed conditions. It is concluded that having a one-dimensional space with multi-sets of string objects is a promising way to implement an artificial environment.

Keywords: *artificial life, artificial chemistry, information space, reaction rules, molecular representation*

1 Introduction

The evolutionary ability of an artificial life (alife) system is primarily determined by two fundamental factors chosen by the designer: the information space and reaction rules. For example, two-dimensional cellular automata (2D-CA) [1, 2, 3, 4] is made up of a two-dimensional array of automata (information space) and a transition table (reaction rules). The fundamental design of Tierra [5, 6] is given by a one-dimensional memory addressed by a pointer (information space) and elementary machine instructions (reaction rules). The real biochemical reaction space (a solvent and molecules) happens to have appropriate space dynamics and reaction rules, and for this reason, living things have succeeded in evolving a variety of creatures in the history of biological evolution.

Alife is a paradigm in which we design an artificial system just by preparing the sufficient conditions for the evolution of complex forms of life. Specifying the conditions helps us choose an appropriate information space, which has to necessarily satisfy the sufficient conditions.

To approach this goal, the paper presents a candidate set of criteria (conditions) for the information space and discusses their meanings, in comparison with the real biochemical reaction space. Then, in the latter half of the paper, we take several of the artificial information spaces proposed so far (and the space proposed in this paper) and assess them. The advantages or disadvantages of these spaces are discussed in light of the proposed criteria, and we suggest that a one-dimensional lattice with string objects is the most promising way to evolve artificial creatures.

2 Conditions for Information Space

Among various factors in a biochemical liquid space, we specifically focus on two factors: selective pressure enabling cells to evolve larger cells and functions of molecules to maintain and reproduce cells.

First, we consider an important condition for evolution to happen in an alife system. Evolution is a process characterized as the modification of genetic information that determines the functions of a creature. If a system does not allow the existence of entities conveying this information, it cannot evolve any form of life. Therefore, the information space

(1) Should be able to include entities that express genetic information.

Next, we focus on the factor of selective pressure for larger and more complex forms of life. The information spaces of the alife systems proposed so far can be roughly classified as elastic (fluid, like SeMar [7, 8]) or rigid (solid, like Cellular Automata [1, 2, 3, 4]). In an elastic information space, molecules are freely inserted (or deleted), which logically allows an unlimited increase in the creature (cell) size, whereas in a rigid information space, a creature cannot evolve to a larger size without forcibly writing information units in a neighboring creature. This kind of writing, when carried out, necesarily destroys the neighbor (typically, its own daughter). Hence, in order to assure an increase in the cell size while protecting the cell itself, we must

require the following two conditions. The information space

(2) Must create and maintain cell walls to protect cellular structure itself.

(3) Must allow an unlimited increase in (or, have no selective pressure for) the (effective) size of a cell.

One of the most distinctive features of the solvent in a cell is the *transportation* of molecules. Here, we classify such a transportation as active or passive. (The transportation in an artificial system is defined as a change in the order/disposition of information objects (molecules).) According to a recent molecular biology study [9], molecules in a cell are actively transferred by actuators. Recently, a number of different transfer proteins have been identified to move along the rails of microtubules stretched throughout a cell and to help transfer molecules to appropriate places in a cell in a timely manner. A mature living cell cannot maintain nor divide itself without this active transportation of particles. Hence, as conditions for active transportation or equivalent outcomes, we require that the information space and the reaction rules

(4a) Allow the snatching and exploiting (digestion) of particles in a neighboring cell (creature),

(4b) Allow the geographical mingler of offsprings.

The other kind of transportation − passive transportation − is the representation of a chemical diffusion process. The diffusion process is not only useful for in-cell (or between-cell) signal transmission by small molecules (like vitamins or minerals), but is also useful for the exploration of novel combinations of molecules. This exploration is achieved as follows. In a cell, when a metabolic reaction is put into action, a large number of output molecules are produced and scattered in the solvent. Most of these molecules work as ingredients for the next metabolic reaction, but a small portion of diffused molecules escape from this reaction, move around the cell, and participate in another metabolic reaction at another place. This possibility enables a novel biological function to evolve. Hence, as conditions for passive transportation or equivalent outcomes, we require that the information space and the reaction rules

(5a) Allow the possibility of testing novel combinations of particles,

(5b) Allow the transmission of molecules for signals.

3   Assessment

Here we classify alife systems in terms of type of reaction rules (deterministic or probabilistic), type of information object (character or string), and space structure (tank, 1D-rigid, 1D-elastic, 2D, or others). Most of the time, 'deterministic' reaction rules take a 'set' representation of objects, and 'probabilistic' reaction rules take a 'multi-set' representation of objects. In a 'set' representation, one kind of biochemical molecule is represented by a single information object. A 'character' object is defined as an unchangeable entity chosen among prepared elementary units, and a 'string' object is defined as a changeable entity created by a combination of prepared units.

Table 1 shows the advantages or disadvantages of several alife systems. From these tables, we can generally say that 'deterministic' reaction rules are weak in passive transportation, and systems using 'probabilistic' reaction rules operated on 'characters' cannot include genetic information. In addition, a 'tank' space cannot enable active transportation. Among various choices, a one- or two-dimensional grid space with probabilistic reaction rules operated on strings has no serious drawbacks, and provides a promising way to evolve complex forms of creatures.

4   Conclusion

In light of the proposed conditions for the information space of an alife system, various kinds of alife systems were assessed. It was concluded that a one-dimensional grid (rigid) space with a multi-set of string objects or a one-dimensional string (elastic) space with multi-set representation are the most promising ways to evolve artificial creatures.

Acknowledgements

References

[1] Langton, C.G.: Self-reproduction in cellular automata. Physica D **10** (1984) 135–144

[2] Lohn, J.D., Reggia, J.A.: Automatic discovery of self-replicating structures in cellular automata. IEEE Transactions on Evolutionary Computation **1** (1997) 165–178

[3] Sayama, H.: A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. Artificial Life 5 N.4 (2000) 343–365

[4] Sayama, H.: Self-replicating worms that increase structural complexity through gene transmission. In: Bedau, M.A. et al. (eds.): Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life, MIT Press, Cambridge (2000) 21–30

[5] Ray, T.S.: An approach to the synthesis of life. In: Langton, C.G. et al. (eds.): Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Santa Fe Institute Studies in the Sciences of Complexity, Vol. 10), Addison-Wesley (1992) 371–408

[6] Ray, T.S., Hart, J.: Evolution of differentiated multi-threaded digital organisms. In: Adami, C., Belew, R.K., Kitano, H., and Taylor, C.E. (eds.): Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life, MIT Press, Cambridge, MA (1998) 295–304

[7] Suzuki, H.: An Approach to Biological Computation: Unicellular Core-Memory Creatures Evolved Using Genetic Algorithms. Artificial Life 5 N.4 (2000) 367–386

[8] Suzuki, H.: Evolution of Self-reproducing Programs in a Core Propelled by Parallel Protein Execution. Artificial Life 6 N.2 (2000) 103–108

[9] Hirokawa, N.: Kinesin and Dynein Superfamily Proteins and the Mechanism of Organelle Transport. Science 279 (1998) 519-526

[10] Adami, C., Brown, C.T.: Evolutionary learning in the 2D artificial life system "Avida". In: Brooks, R., Maes, P. (eds.): Artificial Life IV: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, MIT Press, Cambridge (1994) 377–381

[11] Adami, C.: Learning and complexity in genetic auto-adaptive systems. Physica D 80 (1995) 154–170

[12] Adami, C.: Introduction to Artificial Life. Springer-Verlag, Santa Clara, CA (1998)

[13] Speroni di Fenizio, P., Dittrich, P., Banzhaf, W.: Spontaneous formation of proto-cells in an universal artificial chemistry on a planar graph. In:

Kelemen, J., Sosik, P. (eds.): Advances in Artificial Life (6th European Conference on Artificial Life Proceedings), Springer-Verlag, Berlin (2001) 206-215

[14] Ono, N., Ikegami, T.: Self-maintenance and self-reproduction in an abstract cell model. J. theol. Biol. 206 (2000) 243-253

[15] Ono, N., Ikegami, T.: Model of Self-Replicating Cell Capable of Self-Maintenance. In: Floreano, D. et al. (eds.): Advances in Artificial Life (5th European Conference on Artificial Life Proceedings), Springer-Verlag, Berlin (1999) 399-406

[16] Ono, N., Ikegami, T.: Artificial Chemistry: Computational Studies on the Emergence of Self-Reproducing Units. In: Kelemen, J., Sosik, P. (eds.): Advances in Artificial Life (6th European Conference on Artificial Life Proceedings), Springer-Verlag, Berlin (2001) 186-195

[17] McMullin, B., Groß, D.: Towards the Implementation of Evolving Autopoietic Artificial Agents. In: ibid. (2001) 440-443

[18] Suzuki, Y., Tanaka, H.: Chemical evolution among artificial proto-cells. In: Bedau, M.A. et al. (eds.): Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life, MIT Press, Cambridge (2000) 54–63

[19] Fontana, W.: Algorithmic chemistry. In: Langton, C.G. et al. (eds.): Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Santa Fe Institute Studies in the Sciences of Complexity, Vol. 10), Addison-Wesley (1992) 159–209

[20] Fontana, W., Buss, L.W.: 'The Arrival of the Fittest': Toward a Theory of Biological Organization. Bull. Math. Biol. 56 (1994) 1-64

[21] Dittrich, P., Banzhaf, W.: Self-evolution in a constructive binary string system. Artificial Life 4 (1998) 203–220

[22] Speroni di Fenizio, P., Banzhaf, W.: Stability of metabolic and balanced organisations. In: Kelemen, J., Sosik, P. (eds.): Advances in Artificial Life (6th European Conference on Artificial Life Proceedings), Springer-Verlag, Berlin (2001) 196-205

Table 1: Assessment of several artificial life systems

Rules	Object	Space	Example	(1)	(2)	(3)	(4a)	(4b)	(5a)	(5b)
Deterministic	Character	1D (rigid)	Tierra [5, 6]	○	○	×	△[1]	○[2]	×	×
		1D (elastic)	SeMar [7, 8]	○	○	○	△[1]	×	×	×
		2D	CA [1, 2, 3, 4]	○	△	△[3]	×	○	×	×
		2.5D	Avida [10, 11, 12]	○	○	○[4]	×	○	×	×
	String	1D		○	○	△	△	△	×	×
		2D		○	○	△	△	○	×	×
		Others	planer gragh [13]	○	○	○	×	○	×	×
Probabilistic	Character	1D (rigid)	Ono & Ikegami [14]	×	○	○	×	×	○	○
		2D	Ono et al. [15, 16, 17]	×	○	○	×	○	○	○
		Others	ARMS (partitioned tank) [18]	×	○	○	△	○	△	△
	String	tank	Fontana et al. [19, 20, 21, 22]	○[5]	×	×	×	×	○	×
		1D (rigid)		○	○	○[6]	△[7]	△	○	○
		1D (elastic)		○	○	○	○[8]	○[8]	△[7]	△[7]
		2D		○	○	△	△[7]	○	○	○

Each column denotes (1) Getnetic information, (2) Cell wall, (3) Free increase in cell size, (4a) Snatching and digestion, (4b) Offsprings mingled, (5a) Passive mixing, and (5b) Signal diffusion. '○' means that the system fully satisfies the required condition, '×' means that the system cannot satisfy the required condition, and '△' means that the system does not fully satisfy the required condition. The judgment was given by an example system if one existed or given by general properties otherwise.

[1] Might be possible but not yet achieved.
[2] By Tierran instruction.
[3] A certain amount of increase in cell size was reported in [4].
[4] A creature program is lined up perpendicularly to the 2D geographical space.
[5] Logically possible but not yet implemented.
[6] Walls cannot move but contents of a cell can increase.
[7] Efficiency is low.
[8] Logically possible by rewriting rules.

Learning Robust Policies for Uncertain and Stochastic Multi-agent Domains

Sachiyo Arai

sachiyo@cs.cmu.edu

The Robotics Institute,

Carnegie Mellon University

5000 Forbes Ave., Pittsburgh, PA 15213 U.S.A.

Kazuteru Miyazaki

teru@niad.ac.jp

Faculty of Assessment and Research for Degrees,

National Institution for Academic Degrees

3-29-1 Otsuka, Bunkyo-ku, Tokyo, 112-0012 JAPAN

Abstract

In this paper, we demonstrate the effectiveness of our method, *First Visit Profit-Sharing* (FVPS) in *NEO (non-combatant evacuation operation)* domain where consists of several competitive agents, which ferry groups of evacuees to one of several shelters. We compare the performance of the FVPS with that of the Sarsa(λ) using a replacing eligibility trace, Q-learning, and other Profit-sharing-based algorithm. We also claim that this approach results in an effective stochastic or deterministic policy which is appropriate for the environment.

1 Introduction

Challenges in Multi-agent Domains: There are three problems which have previously been encountered when reinforcement learning approaches are applied to multiagent domains. The first is due to the "agent's sensory limitation", in which an agent is fooled into perceiving two or more different states as the same state. This is known as *perceptual aliasing* [Chrisman, 1992]. If all these different states require the same action, then perceptual aliasing is desirable, as it results in a generalization of the state space. However, if each state requires a different action, then this can lead to the agent becoming "confused", and hence performing the wrong action.

The second problem is due to the "agents' communicational limitation" under *concurrent learning* [Arai et al., 1997] environment, in which the dynamics of the environment vary unpredictably as each agent modifies its own policies and behaves asynchronously. Thus, midway though the learning process, an agent cannot estimate accurately the model of state transitional probabilities for its environment. These two problems can result in non-Markovian properties within state transitions from the viewpoints of the agents, which have only limited abilities for sensing and communication.

A third problem is known as "curse of dimensionality", and denotes the explosion in required state space to represent policies as a lookup table. The aware of dimensionality is very serious in multi-agent domains because of *existence of multiple entities* whose states and actions should be represented in a state space.

Related Works: FVPS is very similar to the Monte-Carlo (MC) approach[Singh & Sutton, 1996] in that it makes no attempt at satisfying Bellman equations relating the values of successive states and it assumes that there is a goal within a finite number of steps from every initial state. It is different from MC in that the *Weight*[1] of a state (or state-action) pair, which is acquired by our method has no meaning as the estimation of state-action values, whereas MC attempts to estimate the value of the state (or state-action pair in Sarsa(λ)) as an averaged reward using *Widrow-Hoff rule*. FVPS reinforces effective rules to make an agent acquire stochastic policies that cause it to behave very robustly in uncertain and stochastic domain, where complete information will not be available to the agent, and its action will not always effect on it deterministically.

2 Our Approach

Our multi-agent reinforcement learning approach is based on Profit-sharing[Grefenstette, 1988](G-PS), which is a credit assignment procedure in classifier systems. However, G-PS does not referr to any requirement which a credit assignment function should satisfy to acquire a rational policy.

Rationality in the Dynamic Domain: In domains where the three serious problems mentioned in previ-

---

[1] *Weight* is similar to *Value* in the Dynamic Programing approaches, and *Strength* which is used in the classifier systems,

ous section are present, we can hardly expect agents to acquire an optimal policy. Although in general, the acquired policy need not be optimal for multi-agent situations, it is important that this policy makes agents achieve their goals. In this paper, we define a *rational* policy as one that is guaranteed not to become trapped within infinite loops in the state machine. To guarantee convergence to a rational policy in a non-Markovian domain, we introduce the *Rationality Theorem* [Miyazaki et al.,1994], and *First Visit Profit Sharing (FVPS)*[Arai & Sycara,2001] into credit assignment process of Profit-sharing[Grefenstette, 1988]. We call Profit-sharing, which is applied *Rationality Theorem*, R-PS.

First Visit Profit Sharing (FVPS): Though the *Rationality Theorem* can be used to design a credit assignment function that excludes the loops, applying this theorem becomes problematic when the episode is long. In other word, little credit will be assigned to the state-action pairs which are fired in the initial stages (around $t \approx 0$) of the episode when the functions which satisfy this theorem, such as geometrically decreasing function. This may causes that the agent can't help behaving randomly around the initial state. To resolve this problem, we propose the FVPS algorithm (Fig.1) where if the current observation is a state that has been revisited one and the same action is executed, the agent does not stack this state-action pair into the *Episodic Memory*, since the re-executed state-action pair will cause a cyclic loop in the agent's route. FVPS does not require any extra memory other than that used by the current framework of the R-PS. Although FVPS uses an internal one-dimensional episodic memory, this memory is not used to construct the state-transition model but only to discard looping behavior and to acquire an effective stochastic policy to escape perceptual deceptive states. Therefore, FVPS is classified as a model-free approach which does not cost much computation and memory space.

3 The NEO Domain

Non-combatant evacuation operations, or *NEOs*, have been used to test a variety of coordination strategies. Though real-world NEOs have many constraints and resource conflicts, the domain used in this study models multiple transportation vehicles which transfer groups of evacuees to safe shelters. Each transport is operated asynchronously by an autonomous agent, which makes its own decision based on locally available information.

The NEO domain consists of a grid world with multiple transporter agents, each of which carries a group of

```
begin
Initialize W(x,a) arbitrarily and TotalSteps=0.
    Repeat (for each episode n){
        do {
            - get sensory input xt of the environment ;
            - choose at from an available action set At ;
            - take action at ;
            - FirstVisit Routine (episodicMemory[],
              currentStackSize, (xt, at)) ;
            - check reward rt ;
            - TotalSteps ++ ;
            } while (reward ! = 0) ;
        if rt = R(> 0) at time T steps then, T=TotalSteps;
```

$$\forall(x,a) \text{ in the episodic memory}$$

$$W_{n+1}(x,a) \leftarrow W_n(x,a) + R \cdot \beta^T \qquad (1)$$

```
    } until enough number of episodes.
end

FirstVisitRoutine(episodicMemory[], currentStackSize,(xt,at)):
begin
        initialize pointer pt=0,
            do from the first state-action pair of episodicMemory[]{
                search same state-action pair as (xt, at).
                (compare (xt, at) with stacked state-action
pair(x,a)pt of
                episodicMemory[]; pt++;)
                if (xt,at)== episodicMemory[pt=k], break;
            } while (pt == currentStackSize);
            if found the same state-action pair in the episodicMemory
                (pt < currentStackSize) {
                retain whole state-action pairs of episodic memory,
                    do not stack executed state-action pair, (xt, at).
        }
        else {
            stack executed state-action pair, (xt, at) into
episodicMemory[].
            currentStackSize ++;
        }
end
```

Figure 1: FVPS algorithm

evacuees(Fig. 2). The goal of a transporter agent is to ferry its group to one of the shelters as quickly as possible. However, there may be conflicts, as transporters cannot co-exist in the same location at the same time. In addition, the location of the shelters changes over time, the agents have limited visibility, and information which is related to other agents, such as their current positions, goals, and intentions which is based on their current learning situations, is also unknown. An effective approach for agents in such environments is to learn reactive behaviors through trial and error experiences, since it is very difficult to know in advance what effective action should be taken at each possible state of the environment.

Each transporter agent is modeled as a reinforcement learning entity. There is no communication among the agents, and the agents have limited visibility. It should
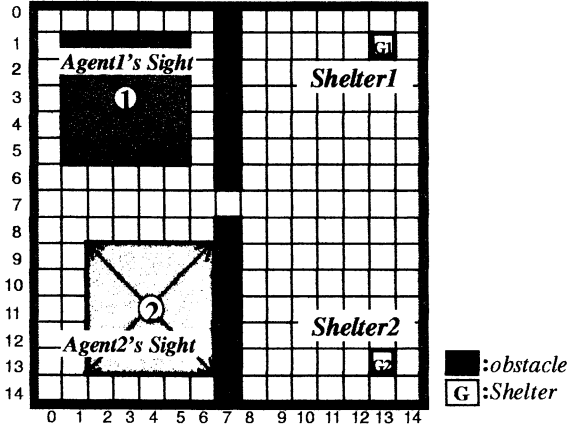
Figure 2: Two Agents move within the grid world.

be noted that there are other agents within the environment that are also learning independently of each other, without sharing sensory inputs or policies. As a result, the other agents appear as additional components within the environment, whose behavior is dynamic and unpredictable. Therefore, the environment is not assumed to be MDP, but partially observable MDP(POMDP).

4 Experiments

4.1 Experimental Design

We constructed a set of experiments in the NEO domain, as shown in Fig. 2, to see the performance of five algorithms, one-step Q-learning[Watkins & Dayan, 1992], Sarsa(0), Sarsa(1)[Sutton, 1998], Profit-sharing applying Rationality Theorem(R-PS) and FVPS. One-step Q-learning and Sarsa(0) are taken as the typical algorithms of DP-based methods, and Profit-sharing and Sarsa(1) are taken as the typical MC-based-methods.

In Fig. 2, two agents started from different locations, and their task was to learn policies for finding each shelter as quickly as possible. There are five actions within the action set, $\mathcal{A} =$ {Stay, Up, Right, Down, Left }. Both agents cannot occupy the same position at the same time, nor may they pass through obstacles. In each episode, the order of their movement is that Agent1 always moves first in each episode. Agents always start in the same location.

The locations of agents' target shelters are varied in every episode within the right half of the grid world. Therefore, there is a possibility that the agents need to go over the conflict to reach each goal. When one agent reaches its target shelter, its episode terminates,

and the agent remains in the shelter until the second agent reaches its goal. In this case the perceptual distance (i.e. the visibility) of each agent is only a 5 × 5 region ; i.e. each agent can see a shelter or the other agent only when they are no more than two moves away.

The learning parameters were selected as follows:

Profit-sharing:
In R-PS, we use $f(R_T^n, t) = R \cdot \beta^{T-t}$, (R=100.0, $\beta = 0.3$, satisfying the Rationality Theorem. In FVPS, the rules which are retained after the *First Visit* routine will be given the value $R \cdot \beta^T = (100.0) \cdot (0.8)^T$. When each agent reaches the goal state (i.e. a shelter), it received a reward of 100.0(= R). Initial weight of each rule is 10.0. Conflict actions were resolved using a weighted roulette selection, $p(a_i|s) = \frac{W(s,a_i)}{\sum_{k \in \mathcal{A}} W(s,a_i)}$ in R-PS, and FVPS.

Q-learning and Sarsa(λ):
We used the same parameters in both Q-learning and Sarsa(λ): *learning rate* $\alpha = 0.05$ and *discounting factor* $= 0.9$, as these were found to be the best parameters in our experiments. When each agent reaches a shelter, it received a reward of 1.0. Initial value of each rule is 0.1. Initial value of each rule is 0.1. The Boltzmann distribution $p(a_i|s) = \frac{e^{Q(s,a_i)/T}}{\sum_{k \in \mathcal{A}} e^{Q(s,a_i)/T}} (T = 0.1)$ is used to select an action.

The evaluation metric is determined by averaging the number of states required per agent to reach each shelter. Experiments consist of 10 trials, each of which consists of 100,000 episodes. The lookup table of each agent is reset for each trial.

4.2 Experimental Result

We use two grid worlds; the 15 × 15 world illustrated in Fig. 2, and a similar but smaller 7 × 7 environment. The average of required steps to reach the shelters of the agents, and the learning curves of the five algorithms are shown in Fig. 3. The x-axis shows the number of episodes and the y-axis shows the average steps to the goal of 10 trials.

4.3 Discussion

Here, we discuss the advantages of our reinforcement approach within the multi-agent domain.

First, for the perceptual aliasing problem, our method acquired effective stochastic polices to escape the perceptual aliasing area. These stochastic policies were built up by reinforcing only state-action pairs which have been used in the successful plans, instead of estimating the values of state (or state-action pairs). Though the Sarsa(1) method has a similar way of reinforcing state-action pairs, Sarsa(1) still trying to estimate the value of
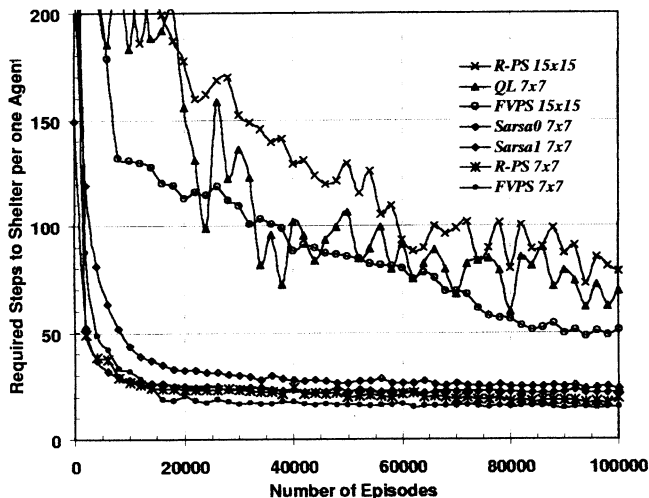
Figure 3: Performance in the Dynamic and Uncertain Domain

the state-action pair using *Widrow-Hoff rule*. This may causes continuous changes of values among the perceptual aliasing states. As a consequence, Sarsa(1) could not show good performance within the perceptual aliasing.

Second, within this simplified Neo domain, the effect of the concurrent learning problem seems very small: i.e,; there is few competitive situation to reach the respective goals of the agents. However, our Profit-sharing performs better than others. Though the Sarsa method has a similar way of reinforcing the state-action pair, Sarsa has no effective way to give a learning bias to the learning agent to reduce its sensitivity to the other agents' behaviors.

Third, for the curse of dimensionality problem, we examine the amount of memory space which was required to acquire the policies of environment. In this case, the whole state space size is around 35,000. After 100,000 episodes, agents have experienced most of the state space. In the case of Profit-sharing learning agent, 99% of rewards, which are given during 100,000 trials, are assigned to reinforce the 50% states of the whole state-spaces. This means that the weight of other 50% states does not need to keep as a lookup table. Therefore, Profit-sharing agents need to keep only half of the whole state space to behave effectively, while Q-learning agents need to keep the whole space to output the effective action. Sarsa(1) is less efficient than Q-learning in practical use, because it needs to keep the eligibility traces. As shown, Q-learning is less appropriate than Profit-sharing in domains which have perceptual aliasing or concurrent learning problems.

5 Conclusion and Future Work

In this paper, we present a variant of the Profit-sharing algorithm and demonstrate its effectiveness within a multi-agent domain characterized by conflicting situation and uncertainty. While Profit-sharing is appropriate for an episodic task where the reward is only given at the end, it is less suited for domains that include intermediate rewards. We plan to combine Profit-sharing with other bottom-up approaches, such as genetic algorithms, and with top-down approaches to develop reinforcement learning system suitable for real world applications.

References

[Arai et al., 1997] Arai,S.,Miyazaki,K.,   and Kobayashi,S. Generating Cooperative Behavior by Multi-Agent Reinforcement Learning. *Proceedings of 6th European Workshop on Learning Robots*, pages 111-120, 1997.

[Arai & Sycara,2001] Arai,S. and Sycara,K. Credit Assignment Method for Learning Effective Stochastic Policies in Uncertain Domains. *Proceedings of Genetic and Evolutionary Computation Conference*, pages 815-822, 2001.

[Chrisman, 1992] Chrisman, L. Reinforcement learning with perceptual aliasing: The Perceptual Distinctions Approach. *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 183-188, 1992.

[Grefenstette, 1988] Grefenstette J. J. Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, *Machine Learning*, Vol.3, pages 225-245, 1988.

[Miyazaki et al.,1994] Miyazaki, K.; Yamamura, M.; and Kobayashi, S. 1994. On the Rationality of Profit Sharing in Reinforcement Learning, In *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing*, pages 285-288.

[Singh & Sutton, 1996] Singh, S.P. and Sutton, R.S. Reinforcement Learning with Replacing Eligibility Traces. *Machine Learning*, Vol.22 : pages 1-37, 1996.

[Sutton, 1998] Sutton, R.S.and Barto, A.G. 1998. Reinforcement Learning, *7.5 Sarsa(λ)* pages 179-181.

[Watkins & Dayan, 1992] Watkins, C. J. H., and Dayan, P. Technical note: Q-learning. *Machine Learning*, Vol.8: pages 55-68, 1992.

Reinforcement Learning in 2-players Games

Kazuteru Miyazaki	Sougo Tsuboi	Shigenobu Kobayashi
National Institution for Academic Degrees	TOSHIBA	Tokyo Institute of Technology
3-29-1 Ootsuka Bunkyo-ku	1 Toshiba Komukai Saiwai	4259 Nagatsuta Midori-ku
Tokyo 112-0012 Japan	Kawasaki 212-8582 Japan	Yokohama 226-8502 Japan
teru@niad.ac.jp	sougo.tsuboi@toshiba.co.jp	kobayasi@dis.titech.ac.jp

Abstract

The purpose of reinforcement learning system is to learn an optimal policy in general. However, in 2-players games such as the othello game, it is important to acquire a penalty avoiding policy. In this paper, we are focused on formation of penalty avoiding policies based on *the Penalty Avoiding Rational Policy Making algorithm* [2]. In applying it to large-scale problems, we are confronted with the curse of dimensionality. To overcome it in 2-players games, we introduce several ideas and heuristics. We show that our learning player can always defeat against the well-known othello game program KITTY.

1 INTRODUCTION

Reinforcement learning (RL) [3] is a kind of machine learning. It aims to adapt an agent to a given environment with a clue to rewards. We can classify RL systems on Markov Decision Processes (MDPs) into two types. One is focusing on identifying state transition probabilities, such as *Q-learning* (QL) [4] and *k-Certainty Exploration Method* [1]. The other is focusing on whether there is the state transition or not. 2-players games discussed in this paper belong to the latter.

Recently, in most RL systems, a positive reward called a *reward* is given to the agent when it has achieved a purpose, and a negative one called a *penalty* is given to it when it has violated a restriction. If we set incorrect reward and penalty values, in the class where it is focused on whether there is the state transition or not, the agent will learn unexpected behavior. Therefore it is important to distinguish a reward from a penalty [2] in the class.

We know the *Penalty Avoiding Rational Policy Making algorithm* (PARP) [2] as a RL system to make a distinction a reward and a penalty in the class. Though it can suppress any penalty as stable as possible and can get a reward constantly, it has to memorize many state-action pairs.

In this paper, we aim to adapt PARP to 2-players games such as the othello game. We introduce several ideas and heuristics to overcome the combinational explosion in large-scale problems. We have confirmed the effectiveness of proposed method in the othello game.

2 THE DOMAIN

Consider an agent in some unknown environment. At each time step, the agent gets information about the environment through its sensors and chooses an action. As a result of some sequence of actions, the agent gets a *reward* or a *penalty* from the environment. We assume that target environments are MDPs. A pair of a *sensory input* (*a state*) and an *action* is called a *rule*. We denote a rule '*if x then a*' as *xa*. where '*x*' is a state and '*a*' is an action. The function that maps states to actions is called a *policy*. We call a policy *rational* if and only if expected reward per an action is larger than zero.

The function that maps a state (or a rule) to a reward (or a penalty) is a *reward function*. We assume that we can know a reward function and a candidate for a descendant state of the state transition. When the agent selects an action a in the state s_t at time t, we can know variation of the state s_{t+1} at time $t + 1$ and its immediate reward or penalty. They are not necessary correct functions. Though it is not confused by incomplete information such that some penalty or state that should be existed on are not given to the agent, it is confused by incredible information such that some penalty or state that should not be existed on are given to the agent. It is natural assumption in 2-players games such as the othello, igo, shougi, backgammon and so on.

We call a sequence of rules used between the previous reward (or penalty) and the current one an
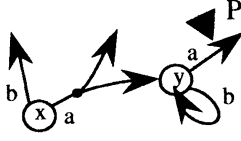
Figure 1: An example of penalty rules (xa, ya) and a penalty state (y)

episode. We call a subsequence of an episode a *detour* when the state of the first firing rule and it of the last firing rule are the same though both rules are different. The rule that does not exist on a detour in some episode is *rational.* Otherwise, a rule is called *irrational.* We call a rule *penalty* if and only if it has a penalty or it can transit to a *penalty state* in which there are penalty or irrational rules. For example, in figure 1 xa and ya are penalty rules, and state y is a penalty state. We call a policy that cannot have any penalty rule *penalty avoiding policy.* For each sensory input, a *deterministic policy* always returns an action. We assume that there is a deterministic rational policy in penalty avoiding policies.

3 PENALTY AVOIDING RATIONAL POLICY MAKING IN 2-PLAYERS GAMES

3.1 The Penalty Avoiding Rational Policy Making algorithm (PARP)

PARP can always learn a deterministic rational policy in the class where there is it in penalty avoiding policies. It uses the *Penalty Rule Judgment algorithm* (PRJ) (Fig. 2) to suppress all penalty rules in the current rule set. After suppressing all penalty rules, it makes a rational policy by the *Rational Policy Improvement algorithm* (RPI) [2]. RPI does not need in 2-players games because suppression of all penalty rules leads to get a reward.

PRJ has to memorize all rules that have been experienced and descendant states that have been transited by their rules to find all penalty rules. In applying PRJ to large-scale problems, we are confronted with the curse of dimensionality. To overcome it in 2-players games, it is important to *save the memory* and *restrict exploration.* They are discussed in section 3.2 and 3.3, respectively.

procedure The Penalty Rule Judgement
begin
    Set a mark on the rule that has been got a penalty directory
    do
        Set a mark on the following state ;
        there is no rational rule or
        there is no rule that can transit to no marked state.
        Set a mark on the following rule ;
        there are marks in the states that can be transited by it.
    while (*there is a new mark on some state*)
end.

Figure 2: The Penalty Rule Judgment algorithm

3.2 How to Save the Memory by Calculation of State Transition

We aim to adapt PRJ to the domain discussed in section 2. Before selecting an action, it tyies to find all penalty rules in the current rule set by calculation of all states that can be transited from the current state. After selecting an action, if the agent gets a penalty, it tries to find a penalty rule again. They are realized by the long-term and the short-term memories.

The long-term memory : An unknown penalty rule is found on the short-term memory, it is memorized on the long-term memory. The long-term memory is holding in learning.

The short-term memory : All states and actions in the current episode are memorized on the short-term memory. After calculating all states and rules that can be transited from the current state, they are memorized on the short-term memory. If there is the states in the long-term memory, PRJ tries to find a penalty rule. The short-term memory is initialized for each episode.

We can save the memory to strage state transitions by these memories.

3.3 How to Restrict Exploration by Knowledge

In applying PRJ to large-scale problems, we need many trials to spread a penalty rule. Especially, it is a serious problem in long episode. To overcome it, we introduce a *semi-penalty* that is defined by each problem to expand the definition of a penalty by knowledge. We call a rule *semi-penalty* if and only if it has a penalty or a semi-penalty, or it can transit to a penalty state or a *semi-penalty state* in which there are semi-penalty,

penalty or irrational rules. After finding penalty rules by PRJ, we use PRJ to find semi-penalty rules. Remark that, if we define incorrect semi-penalty, we need more trial to find penalty rules than PRJ without a semi-penalty since exploration is biased.

Since a semi-penalty does not always cause a penalty, it has a possible that all states are semi-penalty ones even if there is a penalty avoiding rational policy. The problem is avoided by an action selector. Usually, we should select a rational rule that is not a penalty and a semi-penalty rule. If we cannot select any rational rule in semi-penalty states, we should select a rational rule that is not a penalty rule.

4 APPLICATION TO THE OTHELLO GAME

4.1 The Well-known Othello Game Program : KITTY

We implement our method as an othello game player's learning system. We use KITTY[1] by Igor Durdanovic as an opponent player. It is the near-strongest program in open source players. We use *kitty.ios* in KITTY's source code. It has interface of *Internet Othello Server* (IOS) that makes a filed of the othello game. We do not give KITTY learning mechanism. Therefore KITTY's action selection probability is stable. The depth of its search mechanism is 60 that is the maximum value.

4.2 Construction of the RL Player

4.2.1 The Setting of the Game

We describe *our RL player*. It gets the state of the game from IOS. It can calculate variations of actions in the state. It selects an action and returns it to IOS. If it cannot select any action, it returns 'PASS action' to IOS. If it cannot win the game, it gets a penalty from IOS.

We set the size of the short-term memory 1000. It is enough to storage at least one step state transition. It can calculate two or three steps state transitions in first stages and one step it in middle stages. Remark that there is no irrational rule in the othello game.

4.2.2 Knowledge of the Othello Game

It is important to restrict exploration from first to middle stage since there are huge state spaces in middle

[1] http://forum.swarthmore.edu/ jay/learn-game/ systems/kitty.html
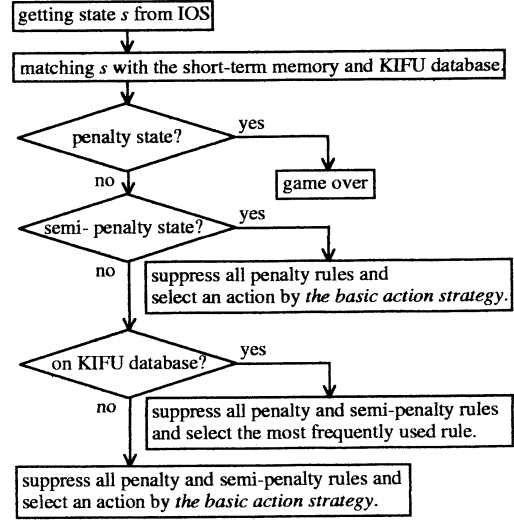
Figure 3: The Action Selector

stages. We use the following two type knowledge to realize it.

i. KIFU database

KIFU database makes to memory steps in previous famous games. We use NEC's KIFU database [2]. It contains about 100,000 games. We can get typical state transitions in first stages from KIFU database. It may contribute to avoid wasteful exploration in first stages.

ii. Evaluation Funcion

We use KITTY's evaluation function that is sent to IOS by KITTY as our RL player's evaluation one. KITTY returns a value from -64.00 to $+64.00$ to IOS as the evaluation value of a state. We define a semi-penalty state as the state whose evaluation value is larger than $+1$.

4.2.3 How to Select an Action

Our RL player selects an action based on figure 3. *The basic strategy* means to select an action whose number of transition states is the least in all actions. It contributes to restrict wastfull exploration.

If the total number of cells is larger than 54, our RL player calculates the end of the game. On the other hand, KITTY calculates it if the number is larger than

[2] ftp://ftp.nj.nec.com/pub/igord/IOS/misc/ database.zip
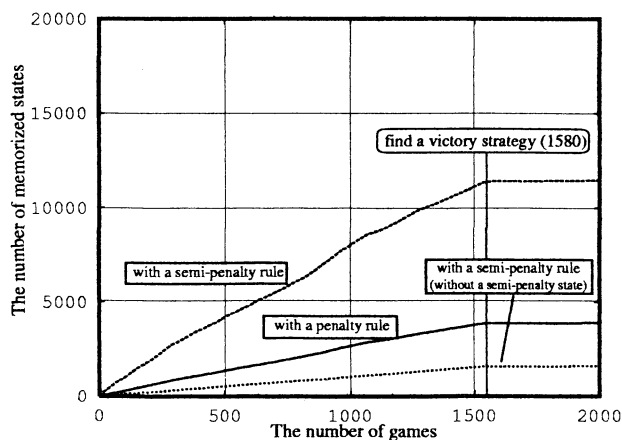
Figure 4: The number of memorized states when we use the semi-penalty discussed in 4.2.2

Figure 5: The number of memorized states when we use inefficient semi-penalty

50 since it can use the min-max exploration with its evaluation function.

4.3 Results and Discussion

Our RL player has learned a deterministic rational policy in 1580 games as the first player (the black player). It means that our method is better than KITTY since winners of KITTY vs. KITTY games are always the second players (the white players).

If we do not use the semi-penalty, the frontier of penalty rules is 34 cells in 2000 games. On the other hand, in the experiment, we can select a semi-penalty rule at 18 cells in 949 games. It means that we can overcome the slow spreads of penalty rules by the semi-penalty.

Figure 4 is the number of memorized states plotted against the number of games when we use the semi-penalty discussed in 4.2.2. It is very compact for the number of non-semi-penalty states with a semi-penalty rule. It means that the semi-penalty is hard to change to non-semi-penalty states.

We have defined another semi-penalty. When the number of action is less than 2 in the next stage, we define the state as a semi-penalty state. Figure 5 is the result of it. It is not stable for the number of non-semi-penalty states with a semi-penalty rule. It means that the semi-penalty is easy to change to non-semi-penalty states. Therefore we should define a semi-penalty that is hard to change to non-semi-penalty states.

5 CONCLUSIONS

In this paper, we adapt the Penalty Avoiding Rational Policy Making algorithm [2] to large scale MDPs such as 2-players games. We have implemented our method as an othello game player's learning system. It can always defeat against the well-known othello game program KITTY.

In the future works, we will compare our method with KITTY with learning mechanism. Furthermore, we will extend our method to Partially Observable Markov Decision Processes and multi-agent systems.

References

[1] Miyazaki, K., Yamamura, M. & Kobayashi, S. k-Certainty Exploration Method : An Action Selector on Reinforcement Learning to Identify the Environment, *Journal of Artificial Intelligence*, Vol.91, pp.155-171, (1997).

[2] Miyazaki, K. & Kobayashi, S. Reinforcement Learning for Penalty Avoiding Policy Making. *2000 IEEE International Conference on Systems, Man and Cybernetics*, pp.206-211, 2000.

[3] Sutton, R. S. & Barto, A. Reinforcement Learning: An Introduction. *A Bradford Book*, The MIT Press, 1998.

[4] Watkins, C. J. H., and Dayan, P.: Technical note: Q-learning, *Machine Learning Vol.8*, pp.55-68, 1992.

Evolutionary Reinforcement Learning for Pursuit Game

Natsumi Okahara   Takeshi Tateyama   Seiichi Kawata   Toshiki Oguchi

Graduate School of Engineering
Tokyo Metropolitan University
1-1 Minami-Ohsawa, Hachioji-shi, Tokyo 192-0397, JAPAN
{okahara, tateyama, kawata, oguchi}@control.prec.metro-u.ac.jp

Abstract

In this paper, we propose an evolutionary reinforcement learning method which combines Q-learning as a learning part and Genetic Algorithm as an evolutionary part, and discuss the result of adapting this method to solve the pursuit game. In our proposed method, Q-function is represented not by a look up table, but by the Q-nets which are neural networks. Each individual learns using Q-learning throughout its life, and apply genetic operator to their chromosomes. Then individuals start to learn again. Experimental results show that, by adding an evolutionary process, individuals can get better adaptational mechanisms than using a non-evolutionary learning method.

Keywords: Evolutionary reinforcement learning, Q-nets, Genetic Algorithm

1. Introduction

When we human or animals adapt to the environments, their adaptational mechanisms have two processes, that is, "Learning" which is a trial and error process occurred at an individual level, and "Evolution" which is occurred at a population level. On the analogy of these mechanisms, some adaptational algorithms for artificial agents which combines learning and evolution have been proposed. For example, Ackley and Littman[1] proposed "Evolutionary Reinforcement Learning (ERL)", in which each individual learns an "action network" which is an action evaluation network using reinforcement learning, and evolves an "evaluation network" which is a state evaluation network using Evolutionary Computation. Unemi and Nagayoshi[2] also proposed the method which obtain a Q-function which is an action evaluation network using neuro Q-learning as well as Genetic Algorithm. On the other hand, Sasaki and Tokoro[3] focused on the mechanisms of genetic inheritance, and evaluated the characteristics of two different types of it, which are so called Darwinian style and Lamarckian style.

In this paper, we present a new evolutionary reinforcement learning method for the pursuit game. The pursuit game is a task which has some prey agents and hunter agents trying to capture the prey agents within some area. The pursuit game is a well-known example for a multi-agent system, and studied by many researchers. Earlier research showed that the pursuit game can be solved by Q-learning or modular Q-learning, but there is no research of the pursuit game applying an evolutionary learning method. So we propose an evolutionary reinforcement learning method which combines Q-learning and Genetic Algorithm, and adapt this method to solve the pursuit game. Experimental results show that our proposed method can solve the pursuit game better than a non-evolutionary learning method. We also show that the hunters operated by our proposed method can capture the prey in more short time by setting a limit to the time to capture the prey, when their life spans are long enough.

2. Learning and Evolutionary algorithm

The learning and evolutionary algorithm proposed here basically combines Q-learning as a learning part and Genetic Algorithm as a evolutionary part.

2.1 Learning algorithm
a) Q-net

Q-learning is a widely used method of

reinforcement learning algorithm, and it generally uses a lookup table for a representation of Q-function. Therefore if the state space is large, learning takes a lot of time. Pursuit game we discuss here has a large state space. In such case, some generalization method can be useful for approximating a Q-function. Q-net proposed by Lin and Mitchell[4] is an neural network representing a Q-function (see Figure 1). The weights of this network are updated by the back propagation algorithm. The update rule is as follows.

$$w \leftarrow w - \eta \frac{\partial E(w)}{\partial w}$$

$$E(w) = r_{t+1} + \gamma \max_b Q(s_{t+1}, b) - Q(s_t, a_t)$$

w : weight, η : learning rate, r : reward,
γ : discount rate,
$Q(s,a)$: Q-value for (state s, action a), t : time
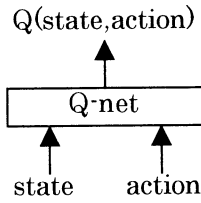
Figure 1: Q-net (monolithic)

Q-net for action1   Q-net for action2
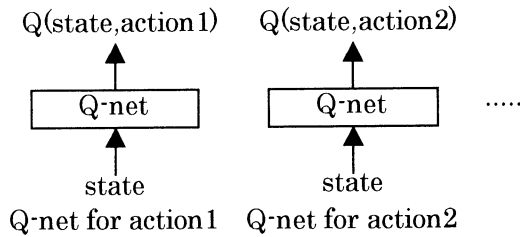
Figure 2: Q-nets (OAON)

b) OAON

Q-net has two kinds of input, which are state and action. Learning by using this network is difficult because it has to learn a highly nonlinear function, that is, given the same state, it has to output very different value depending on the action input. To overcome this difficulty, Lin and Mitchell[4] proposed OAON (One Action One Network) architecture. It uses one network for one action (see Figure 2), then learning process becomes much easier than using a monolithic

Q-net.

In our proposed method, we use Q-nets (OAON) for a representation of Q-function.

2.2 Evolutionary algorithm

In our proposed method, we use Genetic Algorithm(GA) as an evolutionary mechanism. Each individual has Q-nets and its chromosome consists of the weights of these networks. Each chromosome has the fitness and the selection is done depending on this value. The population of individuals evolves through the genetic operation: selection, crossover, and mutation. The selection and crossover here is done by roulette wheel selection and one point crossover respectively, and the mutation is done by adding a random number whose magnitude is within a pre-defined range to some selected genes under the mutation probability.
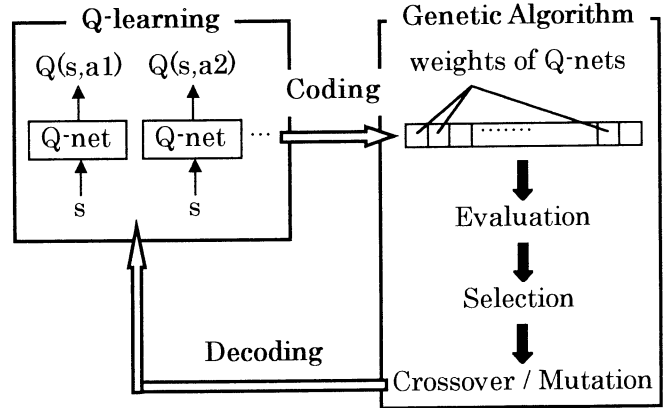
Figure 3: Evolutionary Reinforcement Learning

3. Pursuit Game

A single prey agent and two hunter agents are in a seven by seven grid world (see Figure 4). Two hunters move within this world, trying to capture randomly escaping prey as soon as possible. The settings are detailed as follows.

- The initial assignment of agents' location is determined randomly.
- Each hunter can recognize a relative distance to any other agent within a five by five visual field.
- At each time step, the prey and each hunter choose an action from five actions, that is, to move north, east, south, west or stay still, and act in pre-defined order.

- The prey selects its own action randomly.
- Each hunter has five Q-nets corresponding to five actions, and can learn and evolve the weights of these networks using our proposed method. Each hunter selects its own action by Boltzmann distribution of Q-values, without communicating with the other hunter. Each Q-net consists of three layers, which are input, hidden and output layer.
- More than one agent cannot share the same grid. Therefore the agent that chooses the action to move to the grid already occupied by the other agent cannot move and remains at its current position. When an agent choose the action toward the wall, the agent also cannot move and stay still.
- Hunters capture the prey if all of them occupy the neighbor grid of the prey (see Figure 4 b)). When they capture the prey, each of them receives reward 1 immediately.
- One episode means a series of the pursuit game, which begins with the initial state and ends with the goal state. When an episode ends, that is, the prey is captured, all of the agents are relocated at their new random positions and the next episode starts. The hunter's life ends when it finishes an episode pre-defined times.
- There are two populations corresponding to each hunter. Genetic operator is applied separately to each of these populations.
- Each hunter's fitness is defined as a reciprocal of the average steps to capture the prey on the last five episodes of its life.

● : Hunter   ○ : Prey

a) An initial state   b) A goal state
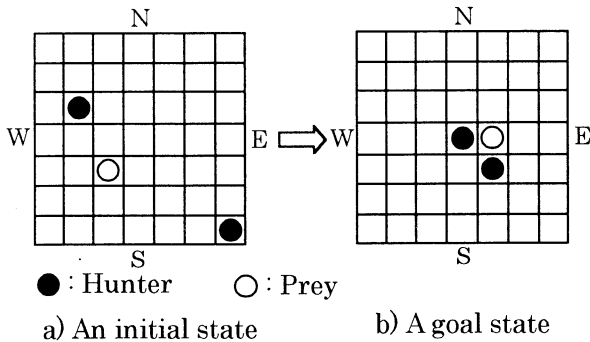
Figure 4: Pursuit Game

Learning and Evolution of the hunter agents is accomplished through the following steps.

1. Initialize the weights of Q-nets for each hunter with random numbers between 0 and 1.
2. Learn the weights of Q-nets of the hunters during their lives using Q-learning.
3. Encode the weights of Q-nets into chromosomes.
4. Calculate each hunter's fitness, and then apply genetic operator to each of two populations of hunters and obtain new chromosomes.
5. Decode new chromosomes into the weights of Q-nets.
6. Back to 2.

4. Experimental Results

4.1 Comparison of a Evolutionary Learning method with a Learning method

We compared the characteristics of two methods to solve the pursuit game, one is our proposed method which combines Q-learning and GA, and the other is Q-learning. Figure 5 shows the experimental results using five episodes as a hunter's life span, whose horizontal axis indicates the number of episodes and vertical axis indicates the number of steps for hunters to capture the prey on each episode. There are two curves in Figure 5, and one curve shows the result of our proposed method, that is a evolutionary learning method, and the other shows the result of Q-learning, that is a non-evolutionary learning method. On these simulations, we use the parameter settings displayed on Table 1. It is shown from Figure 5 that the adaptational speed becomes faster by adding an evolutionary process. It is also shown that the ultimate steps to capture the prey are a little bit less using our proposed method. Generalization by neural networks has the merit of speeding up learning, but also has the demerit to apt to converge to local optima. By adding the GA, global search has been attained.

4.2 Effect of the step limit setting

We set a limit to the number of steps to capture the prey, that is, at any episode, if the hunters cannot capture the prey within some time steps, then the learning is considered to have failed and begin a new episode without giving rewards to the

Table 1: Parameter settings

population size	50	crossover probability	0.9
mutation range	±1.5	mutation probability	0.05
learning rate	0.7	discount rate	0.9
temperature in the Boltzmann distribution			0.3
number of hidden units of a Q-net			15
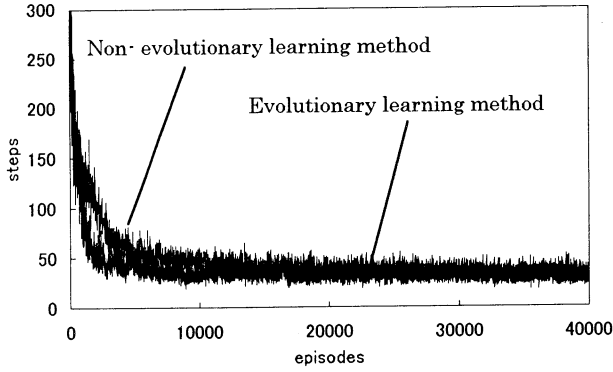
Figure 5: Comparison of a Evolutionary learning method with a Non-evolutionary learning method
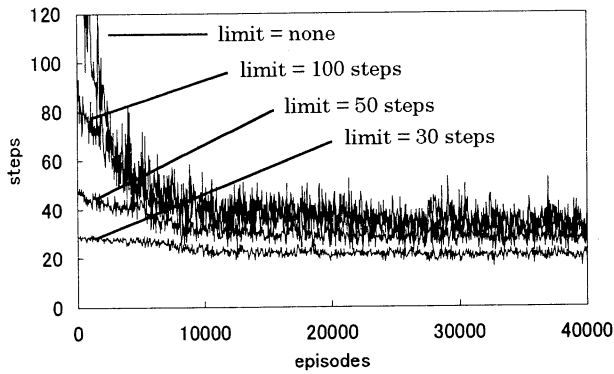
Figure 6: A variety of step limit settings

hunters. To show the effectiveness of this setting, we simulated the pursuit game with various step limit settings. Figure 6 shows the experimental results using fifty episodes as a hunter's life span, whose axes indicates the same as of Figure 5. All four curves in Figure 6 are the results of our proposed method with different step limit settings, that is, none, 100, 50, and 30 steps. It is shown from Figure 6 that learning with a small limit value takes a lot of time, but it can obtain better solution, comparing with a large limit value. So hunters manage to learn the strategy to capture the prey within the limit, even if we set the limit value small to some extent. If the hunter's life span is five episodes, there is not so much effectiveness. That is because the life span is too

short, and hunters hardly capture the prey throughout their lives. Therefore, if the hunter's life span is long enough, setting a proper limit is effective to solve the pursuit game, considering the tradeoff between learning speed and the goodness of the obtained solution.

5. Conclusion

We have proposed an evolutionary reinforcement learning method which combines Q-learning, using Q-nets for a representation of Q-function, and Genetic Algorithm, and discussed the results of using this method to solve the pursuit game. It is shown that, by adding an evolutionary process, the adaptational speed of the hunters becomes faster and they can capture the prey in more short time than using a non-evolutionary learning method. It is shown that, if the hunter's life span is long enough, hunters can learn the strategy to capture the prey in quite a short time, by setting a limit to the time to capture it.

In our pursuit game settings, each hunter has mechanisms both of learning and evolution, but the prey has neither of them and selects its own action just randomly. So we will examine what occurs if the prey, too, can learn and evolve.

References

[1] Ackley D, Littman M (1991), Interactions Between Learning and Evolution. Artificial Life II, SFI Studies in the Science of Complexity, vol.X, Addison-Wesley, pp.487-509

[2] Unemi T, Nagayoshi M (1996), Evolution of Reinforcement Learning Agents — toward a feasible design of evolvable robot team. In Workshop Notes of ICMAS '96 Workshop1: Learning, Interactions and Organizations in Multiagent Environment

[3] Sasaki T, Tokoro M (1997), Adaptation of Evolutionary Agent Group to Dynamical Environment (in Japanese). Computer Software, Vol.14, No.4, pp.33-46

[4] Lin L, Mitchell T (1993), Reinforcement Learning With Hidden States. In Proceedings of the Second International Conference on Simulation of Adaptive Behavior. MIT Press, Cambridge, MA. pp.271-280

A teaching method by using self-organizing map for reinforcement learning

Takeshi Tateyama   Seiichi Kawata   Toshiki Oguchi
Graduate School of Engineering, Tokyo Metropolitan University
1-1 Minami-Ohsawa, Hachioji-shi, Tokyo Japan 192-0397
{tateyama, kawata, oguchi}@control.prec.metro-u.ac.jp

Abstract

In this paper, we proposed a new pre-teaching method for reinforcement learning. The proposed method consists of two parts. In the first part, human teacher selects some trigger point and shows suitable hints for learning. In the second part, self organizing map (SOM) generalizes these hints and produces initial good rule sets for reinforcement learning method. After that, reinforcement learning algorithms learn optimal solution sets. Some simulation results shows the effectiveness of our proposed reinforcement learning algorithm with pre-learning method.

Keywords: reinforcement learning, self-organizing map, teaching, actor-critic method

1 introduction

The main problem of reinforcement learning is that learning converges slowly especially when the search space is large. In this paper, a new teaching method for reinforcement learning using a self-organizing map(SOM)[3] to increase the learning rate is described. Recently, several researchers have presented teaching methods for reinforcement learning. L.Lin's method[2] increases the learning rate by providing the learning agent with "lessons" which are sequences of actions from an initial state to a goal state. However, when the task is large scale and complex, it is hard to generate sequences of actions. J.A.Clouse's method[1] allows a human expert to interact in real-time with a reinforcement learning algorithm. However, a human expert needs to observe the learning agent's behavior at all learning process. And when the agent behaves very quickly, it is difficult for the human expert to advise it in real-time.

In this paper, we propose a new teaching method using SOM to generate initial teaching data from small numbers of teaching data produced by human expert. The outline is as follows. In the first part, a human teacher produces several but not so many teaching data. In the second part, a SOM generates initial teaching data from the above teaching data produced by the human expert. In the third part, we initialize a reinforcement learning agent using the SOM. After that, the learning agent starts reinforcement learning. In this case, the probability of selecting the optimal actions estimated by the SOM becomes high, and the agent can get high rewards. As a result, we can expect to increase the learning rate.

To make sure whether our teaching method increases reinforcement learning's learning rate, we design an experiment on mobile robot simulation. As the result, the learning rate is increased although a teacher showes only a few optimal actions.

2 Actor-critic method

Actor-critic method[4] is one of the reinforcement learning algorithm which is structured by two elements, "actor" and "critic". The actor selects actions, and the critic estimates states. In a reinforcement learning framework, at each time step t, a learning agent observes a state $s_t \in S$ in a environment. Next, the actor of the agent selects an action $a_t \in A$ according to some probability based on the learning method. At next time step $t+1$, the agent transits to a state s_{t+1} and receives a reward r_{t+1} from the environment. After each transition, to evaluate the state s_{t+1}, the critic computes the TD error,

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \qquad (1)$$

where γ is the discount rate($0 \leq \gamma \leq 1$) and $V(s)$ is the value function which estimates how good the state s is. For further details of the value function, the reader should refer to [4]. In case the TD error δ_t is positive, the critic judges that the state s_{t+1} is better than the state s_t and the selected action a_t is a good action. According to this judgement, the tendency to select the action a_t is strengthened. On the other hand, δ_t is negative, the critic judges that the state s_{t+1} is worse than the state s_t and the action a_t is a bad action. In this case, the tendency to select the action a_t is weakened. After updating the actor, the critic updates value function $V(s_t)$ using following update rule,

$$V(s_t) \leftarrow V(s_t) + \alpha_c \delta_t \qquad (2)$$

where α_c is the step-size parameter of critic($0 < \alpha_c \leq 1$). The agent's purpose is to get an optimal policy π by repeating preceding process.

3 A teaching method by using self-organizing map for reinforcement learning

3.1 The purpose of our study

The purpose of this study is designing a new teaching algorithm which satisfies following three requests.

(1)The teaching method should accelerate the learning rate by using a few teaching data generated by teacher against Lin's method[2].

(2)A teacher should not need to observe all learning agent's behavior against Clouse's method[1]. The teaching data is only given to the learning agent before starting reinforcement learning.

(3)The effect of pre-teaching should be reduced in progress of reinforcement learning algorithm to get optimal policy even if the teaching data is wrong.

To meet these requests, we propose a new teaching method using SOM. We will describe the detailed procedure in the next section.

3.2 Detail of the learning algorithm
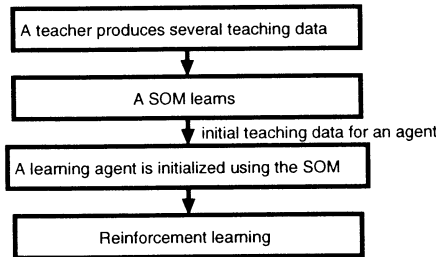
Figure 1: The procedure for our teaching method

The procedure for our teaching method is shown in Figure1. The details will be described as follows.

3.2.1 Making of teaching data

The human teacher produces several but not so many teaching data. The ith($i = 1, 2, ..., N$) teaching data is composed of a pair of two vectors, (s_i, a_i). The teaching data means that an action a_i should be selected at a state s_i.

3.2.2 Learning of SOM
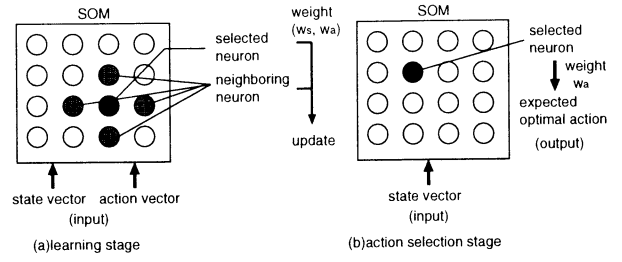
Figure 2: SOM's learning stage(a) and action selection stage(b)

In this study, the SOM is used to generate initial teaching data for the reinforcement learning agent from several teaching data produced by the human expert. The procedure will be described as follows. At learning stage, input vectors to the SOM are teaching data produced by the human expert(see Figure2(a)). The ith input vector is composed of a pair of two vectors, (s_i, a_i). A neuron j of the SOM has a set of weights (w_s^j, w_a^j). To determine "selected neuron" k, we compute the distance d_j between the input (s_i, a_i) and each neuron's weights (w_s^j, w_a^j):

$$d_j = (s_i - w_s^j)^2 + (a_i - w_a^j)^2 \qquad (3)$$

The neuron of which distance is minimum is the selected neuron. The selected neuron's weights and those of its neighboring neuron's weights are updated as follows:

$$w_s \leftarrow w_s + \alpha_S(t)(s_i - w_s) \qquad (4)$$

$$w_a \leftarrow w_a + \alpha_S(t)(a_i - w_a) \qquad (5)$$

where $\alpha_S(t)$ is the step-size parameter of SOM($0 < \alpha_S(t) \leq 1$) that decreases in time. As the result of learning, neurons' weights of the SOM store not only teaching data but also aimilar state-action pairs.

3.2.3 Initializing the learning agent using the SOM

At this stage, we initialize the reinforcement learning agent using the SOM. To be concreate, the agent is initialized to strengthen the tendency to select the optimal action estimated by the SOM. The procedure will be described below. In the first part, we select a state vector $s_n \in S$ randomly. Secondly, the state vector is inputted to the SOM(see Figure2(b)). For the purpose of getting the optimal action estimated by the SOM at the state s_n, we compute the distance d_j' between the input s_n and each neuron's weight w_s^j:

$$d_j' = (s_n - w_s^j)^2 \qquad (6)$$

The optimal action is computed by getting the weight vector w_a of the selected neuron of which distance is minimum. Thirdly, the agent is initialized to strengthen the tendency to select the estimated action before starting reinforcement learning. For example, in case of using Q-learning[4], the estimated state-action pair's Q-value is set to adequate value that is higher than the other action's Q-value.

3.2.4 Reinforcement learning

The learning agent starts reinforcement learning. As a result, the agent can get high rewards from the start of reinforcement learning and we expect to increase the reinforcment learning rate. Moreover, in our method, the effect of teaching is reduced while each Value $V(s)$ of the critic approaches the correct value, and the agent is expected to have optimal policy even if the teaching data is wrong.

4 Experiments and results

4.1 Mobile robot simulation
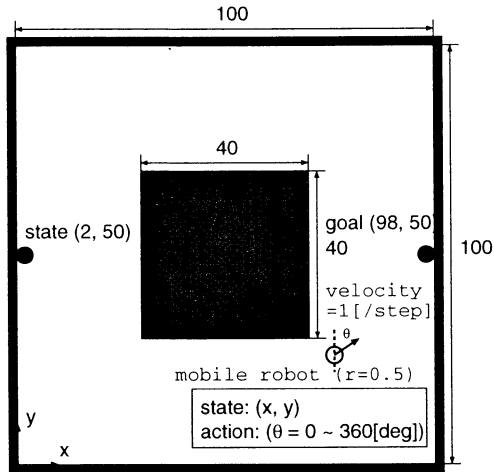
Figure 3: The mobile robot simulation in continuous state and action space

To determine whether our teaching method increases reinforcement learning's learning rate, we designed an experiment on mobile robot simulation. Figre 3 shows the environment of the simulation. The environment consists of a 100×100 room. In the room, a 40×40 square obstacle is placed. The robot is shown as a circle of which radius is 0.5. At each time step t, the robot observes a state (x_t, y_t) in the environment. Next, the robot selects a direction[deg] as an action a_t and moves in the direction at a velocity of 1 per

step. Here, $x_t(0 \le x_t \le 100), y_t(0 \le y_t \le 100)$ and $a_t(0 \le a_t \le 360[deg])$ are continuous. When the robot reach the goal state$(98, 50)$, it receives a reward 1 from the environment. The purpose of the robot is to get the optimal policy to reach the goal state from the start state$(2, 50)$.

4.2 Learning method
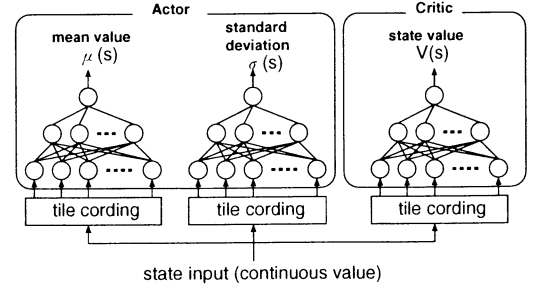
Figure 4: The reinforcement learning system for continuous state and action space

Figure 4 shows the reinforcement learning system for continuous state and action space. The learning algorithm is actor-critic method. At each time step t, the robot observes a state $s_t = (x_t, y_t)$. These continuous values are changed to binary data by tile cording[4]. When the robot tries to select an action, the state (x_t, y_t) as the binary data are put into the actor. The actor consists of two neural networks. The one computes a mean $\sigma(s_t)$ the other computes a standard deviation $\mu(s_t)$. The action value is computed by using the normal distribution. At the next time step $t + 1$, the critic which consists of one neural network computes two state values $V(s_t)$ and $V(s_{t+1})$ to compute TD error. If TD error is positive, neural networks of the actor are updated to strengthen the selected action a_t. Otherwise the actor is not updated.

4.3 Applying our method

Figure 5(a) shows ten teaching data produced by human expert. Each arrow in the figure shows that the robot should move in the direction at the state. Figure 5(b) shows initial teaching data for the reinforcement learning which are generated by the SOM from the above teaching data. In our experiments, the number of neuron of the SOM was $50 \times 50 = 2500$, $\alpha_S(t_0)$ was 0.9 and the SOM learnt 5,000 times.

Next, the agent is initialized to strengthen the tendency to select the action estimated by the SOM
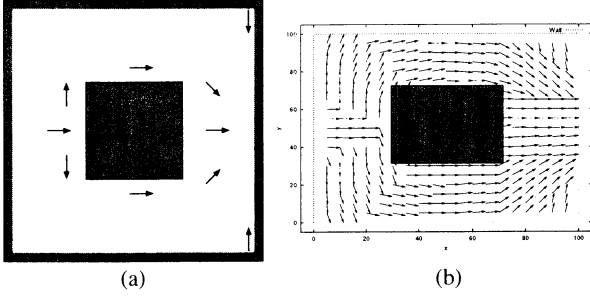
Figure 5: (a)10 teaching data produced by human expert, (b)Initial teaching data for the reinforcement learning which are generated by the SOM

Figure 7: The result of the mobile robot simulation2(X axis: the number of episodes, Y axis: the sum of time steps)

before starting reinforcement learning. In this experiments, the mean value $\sigma(s)$ is changed close to the action value estimated by the SOM using following update rule,

$$\sigma(s) \leftarrow \sigma(s) + \beta(a_{SOM}(s) - \sigma(s)) \qquad (7)$$

where a_{SOM} is the estimated action by the SOM at a state s and $\beta(0 < \beta \leq 1)$ is the step-size parameter.

4.4 Simulation results

In our experiments, the parameter values were $\alpha_m = 0.02, \alpha_s = 0.002, \alpha_c = 0.02, \gamma = 0.999, \beta = 0.01$, where α_m and α_s are the step-size parameter of the mean value's network and the standard deviation's network. The results of the simulation are shown in Figure 6 and 7. In figure 6, the X axis indicates the number of episodes, and the Y axis indicates time steps. The result shows that the agent can get many rewards from the start of the learning. In figure 7, the X axis indicates the number of episodes, and the Y axis indicates the sum of time steps. As is evident from these results, the learning rate of our method is increased although the teacher showed only a few teaching data.

5 Summary

In this paper, we described a new teaching method for reinforcement learning using the SOM to increase the learning rate. And we showed the validity of our method by the mobile robot simulation. The result of the simulation showed that the learning rate of our method was increased although the teacher showed only a few teaching data.

References

[1] Clouse, J.A. and Utgoff, P.E.(1992), A teaching method for reinforcement learning. Proceedings of the Ninth International Conference on machine learning, pp.92-101

[2] Lin, Long-Ji(1991), Programming robots using reinforcement learning and teaching. Proceedings of the Ninth National Conference on Artificial Intelligence, pp.781-786

[3] Kohonen, T.(1996), Self-organizing maps(in Japanese). Springer-Verlag Berlin Tokyo

[4] Sutton, R.S. and Barto, A.(1998), Reinforcement Learning: An Introduction. A Bradford Book, The MIT Press.

Figure 6: The result of the mobile robot simulation1(X axis: the number of episodes, Y axis: time steps)

Changeable graphical images of 3-d systems and means of work with them in machines.

Hanna Astrowskaja, Svetlana Novikava, Kanstantin Miatliuk, Pavel Buka.

The Institute of Mathematics & Cybernetics,
P.O.Box 48, Minsk, 220064 Belarus
+375-17- 277-80-55 E-mail: imc@tut.by

The paper contains cybernetic definition of real changeable systems (made of connected 3-d constructions). Certain constructions (images of machines) can have the mind able to learn the whole system and to define their own actions (behavior) in it. The behavior is realized in agreement with their aims.

Suggested work is originated by theory of Hierarchical Multilevel Systems. This theory [1] allows to connect several levels in system definition: system ω^λ of level λ defined as changeable unit; its construction $\sigma^{\lambda\tau}$ (units $\omega^{\lambda\tau}$ of lower levels λ_τ with their connections $\gamma^{\lambda\tau}$); its activity ρ^λ in the environment σ^λ (other systems of level λ); its coordinating system ω^β of higher level β, and its aim in level β.

The geometrical objects are defined as hierarchical systems. In this way all of them (including 0-d objects – points) have internal constructions, images as units with their actions, environments, aims, and coordinator. Coordinator connects all their levels and has abilities for learning and self-organizing.

Owing to that it is possible carrying out the task of machine activity imaging in 3-d environment by process with slight expenses. Program realization is executed for two types of machines - caterpillars and multi legs insects.

A Review of Identification Methods Applicable
To Artificial Life and Robotics Systems

William R. Wells
University of Nevada, Las Vegas
Las Vegas, NV, USA
Email: wcube@egr.unlv.edu

Key words: systems identification, H_∞ identification, worst-case identification

Abstract

A review of new methods of systems identification (deterministic) which are relevant to the identification of systems concerning artificial life and robotics is given. The motivation for these methods follows the need for such methods in the design of robust non-linear adaptive controllers. The two methods which are highly relevant to artificial life and robotics are the worst-case methods (H_∞ and l_1 norms) and nonlinear black-box methods (regression methods).

Introduction

The techniques developed in the 60's, 70's and 80's in the area of systems identification and parameter estimation are the standard for present applications and theoretical developments. Many of these developments extend and reformulate such classical methods as least mean square parameter estimates, maximum likelihood estimates, deterministic state observers, discrete Kalman filters and extended Kalman filters. Variations of these new methods have been found in many recent applications concerning the design of controllers for dynamical systems. One area in particular is that of the system design of adaptive control of model reference adaptive control (MRAC) systems. In this application a controller is parameterized by a number of adjustable parameters which are updated by on-line system identification models. Examples and tutorials on the application of systems identification to problems in non-linear control design can be found in a textbook published in the last decade [1].

Probably the most important "new" method of systems identification is that of "worst-case estimation" including H_∞ and l_1 filtering and estimation. Research in the areas involving state and parameter estimation and its usefulness to the design of robust controllers is reported in [2].

Worst – case control relevant identification

These system identification methods are developed to be compatible with the design of modern robust controllers. They are deterministic and their use is to estimate model uncertainty based on identification in H_∞ and l_1. With no assumptions on the plant other than stability these methods attempt to produce an approximate model of the system with guaranteed error bounds expressed in terms of maximum noise level. A brief overview of the method is given below for H_∞ estimation in the frequency domain.

Given an unknown stable transfer function $G(s)$ and n corrupted frequency response measurements, $a_{n,k} = G(i\omega_{n,k}) + \eta_{n,k}$ for $k = 1, 2, ...n$, construct a model G_n for G using an algorithm with the property that $\lim_{\substack{\varepsilon \to 0 \\ n \to 0}} \sup \|G_n - G\|_\infty \to 0$ for all stable G.

The noise level, ε, is defined as $\varepsilon_n = \max\left\{ \left| \eta_{n,k} \right| : k = 1, 2, ...n \right\}$ and the H_∞ norm $\|G\|_\infty$

is defined as $\|G\|_\infty = \sup_\omega |G(i\omega)|$. For details of this method and examples see [3].

Nonlinear black-box modeling identification

This methods is used to describe most any nonlinear system dynamics when little physical insight is available. The method utilizes nonlinear regression methods and can include structures based on neural networks, radial basis networks, wavelet networks as well as models based on fuzzy sets and fuzzy rules. A brief overview of the method is given below.

Given observed inputs $u(t)$ and outputs $y(t)$ from a dynamical system, find a relationship between past observations $\left[u^{t-1}, y^{t-1}\right]$ and future outputs $y(t)$ expressed as

$y(t) = g(u^{t-1}, y^{t-1}) + v(t)$ where $v(t)$ represents the error between the next output and past data. Next, it is necessary to find a best value of g within a family of functions through the

parameterization of g with a parameter vector Θ expressed as $g(u^{t-1}, y^{t-1}, \Theta)$. The choice of Θ is determined by the fit between the model and the data according to the quadratic norm

$\sum_{t=1}^{t=N} \|y(t) - g(u^{t-1}, y^{t-1}, \Theta)\|^2$. A tutorial on this method can be found in [4].

Conclusions

Based upon a review of the properties of some of the recent nonlinear deterministic identification techniques it appears that two in particular are quite relevant to application to artificial life and robotics due to the difficulty in mathematically modeling these systems. The two systems discussed are the worst-case methods utilizing H_∞ and l_1 norms and the nonlinear black-box method (regression methods). The properties that make them so relevant to these systems are those of robustness in the presence of model uncertainty and system nonlinearity and the avoidance of noise detail modeling as a consequence of a deterministic formulation.

References

[1] Slotine JE, Li W (1991), Applied nonlinear control. PrenticeHall, New Jersey
[2] Kwawernaak H (1995), Automatica, Vol. 31, No. 12, pp1691-1901
[3] Makila PM, Partington JR, Gustafsson TK (1995),Worst-case control-relevant identification. Automatica, Vol. 31,No. 12,pp1799-1819
[4] Sjoberg J, Qinghua Z, Ljung L, etal (1995), Automatica, Vol. 31, No. 12, pp1691-1724

Life-Based Paradigm in System Design

Sung-Bum Ko and Gi-Young Lim
Chonan National Technical College, 275 Budae-dong, Cheonan, Korea, sbko@dragon.cntc.ac.kr
Hanbat National Univ., 16-1 Deokmyeong-dong, Yuseong-gu, Daejeon, Korea, limgy@hanbat.ac.kr

Abstract--It is reasonably anticipated that many of the system would be run in more dynamic and complex environment in the days to come. In that situation, such intelligent abilities are necessarily required as learning, evolution, security, reasoning, emotion, planning, self-organization or treating the fuzzy concept. In this paper, we suggested a life-based paradigm for general system design, called SAL (System As a Life). We showed that most of the above intelligent models can be deduced from SAL by adding some constraints.

Index Terms—life paradigm, system as a life

I. INTRODUCTION

RECENTLY, whole scale entering into the information age, the environment around us is rapidly having dynamics and complexity in it. Following the trends, the demands of the system with more powerful abilities on intelligence has been increased day by day. These abilities surely contains learning, evolution, security, reasoning, emotion, planning, self-organization or treating the fuzzy concept. The common old approach still used for this is to design local solutions independently and bring all together, which is so called reduction paradigm[2]. But, in this rigid paradigm, it is structurally inevitable to lose much of information during the split and merge process, which surely be an obstacle to design really complex system. In this paper we suggested a new paradigm SAL(System As a Life) for general system design. SAL is a theoretical system formulated by the essential principles of the lives which therefore is called 'life paradigm'. In such a life paradigm as SAL, unlike reduction paradigm, the information processing is performed in holistic-based pattern, which means no loss of information. Thus, especially in case of the complex system design, life paradigm can reflect the domain better than the reduction paradigm. But the holism in terms of information processing, due to its emergent characteristic, is not easy to prove analytically. In this paper instead, we showed that most of the intelligent models can be deduced from SAL by adding some constraints.

II. SYSTEM AS A LIFE

SAL has a hierarchical structure. The top level is called SA (Super Agent) which consists of multiple agents. And the agent is a system which consists of multiple cells. And the cell is a system which consists of two sub systems; PM and TM. Here, the cell is built on the so called components in the Internet environment. Component is a general task processor based on the object model. That is, SAL can be defined as an expanded structure with three higher levels over the object.
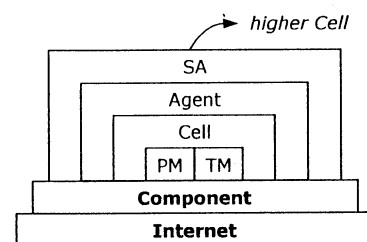
Fig. 1. Structure of SAL

A. Cell

It is the cell that corresponds to the real cell in the living thing of the physical world. That is, SAL-based system is a kind of virtual life composed of virtual cells.

A.1 Structure of Cell [2]

It is the cell that corresponds to the real cell in the living thing of the physical world. That is, SAL-based system is a kind of virtual life composed of virtual cells.

Cell consists of PM and TM which take charge of the interchanging potential among cells and processing the given task respectively. In terms of structure, TM corresponds to the cytology and PM corresponds to the cytoplasmic membrane of the cell. The basic paradigm of the information processing of PM and TM is clearly different. PM basically uses analog oriented type while TM uses digital oriented type. Potential is an extremely abstract energy concept which can be expressed as an analog quantity in mathematics. PM has four input ports such as PF, LF, NF, AF and two out ports such as CF, OF in all six ports. Each port has its own filter by which analog-based control can be available over the input or output. And each filter has one or more gates which can also restrict the condition of input or output.

Cell implements abstract correlations through dynamic potential exchange. There are two kinds of potential such as dynamic potential and static potential. And the potential also can be split into inner potential and outer potential. The inner potential belongs to static potential and the outer potential belongs to dynamic one. While in the DNA of the PM layer, the transcendental information is stored.
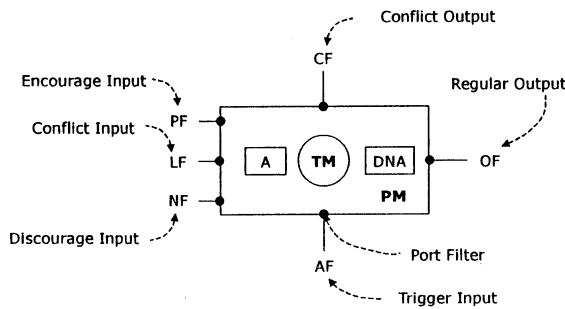
Fig. 2.   Structure of CELL

A.2 Function of Port

The input of PF port, called PIP (Positive Input Potential), is a potential that has positive nature against the inner potential. And the input of NF port, called NIP (Negative Input Potential), is a potential that has negative nature against the inner potential. PIP added to NIP makes IP (Input Potential). LF means conflicting potential imported from outside which is, therefore, called 'outer conflicting potential'. In fact, conflicting potential is normally generated from the collision between two input potentials of PF and NF. The conflicting potential thus generated is called 'inner conflicting potential'. The total conflicting potential is a function of these two confliction potentials; inner one and outer one. the output potential determined by input potential and inner potential outputs at OF port. Formally all the potentials are supposed to have regular value ranging from 0.0 to 1.0. We may define many types of relation model in terms of potential which define the system's features depending on the applications. One of the simplest one can be expressed as

OF = Inner_P + Outer_P
Inner_P = A
Outer_P = MAX((PIP - NIP), 0)
CF = LF + MIN(PIP, NIP)

A.3 Trigger of PM

PM is usually triggered by AF port. There are two types of triggering methods such as one-step trigger and two-step trigger. In case of one-step trigger mode, both PM-layer and TM-layer activate at the same time. In case of two-step trigger mode, the PM-layer activates first and the TM-layer activates next. That is, there are two kinds of threshold values in the two-step trigger mode. The activation of the PM-layer means that the potential exchange begins while the activation of the TM-layer means that the digital information processing begins. It is also possible to impose an additional condition by which the activation of the PM-layer is restricted, which is called 'conditional one-step trigger'. This special mode is especially useful in case of the hybrid reasoning. We use another classification terms of 'A-type trigger' and 'B-type trigger'. The former does not contain any port but AF while the latter contains PF and NF besides AF.

B.   TM (Task Module)

TM takes charge of the task processing directly and shapes itself the core part of the cell. TM vertically interacts with PM through the special interface.

B.1 Structure of TM

TM consists of control-layer and domain-layer, Control-layer performs control function related with potential exchange and runs a special knowledge system for this purpose. Here, the correlation among PMs set by potential exchange becomes a very important factor in deciding the system state. And domain-layer with its own knowledge and information system deals with its own task practically. The control-layer can recognize which gate the potential passed through, which allows us to impose a special meaning to each gate of the port. The result of the task processing is evaluated periodically and rewarded properly based on the evaluation. The unit of the reward is potential which can be stored as the form of inner potential in the cell. The transcendental information called 'Hard-DNA' which is emotional type is corrected by the practical experience. This information, called Soft-DNA, is stored in the KB of TM.

B.2 Characteristic Function

Cell model is defined as a kind of expanded model of the typical object model. In this expanded model, TM is imposed to have specially designed two functions called performance function and interest function.

[1] Interest Function

A TM is supposed to have certain amount of interest in a given task. The degree of the interest is defined as the interest function of the TM to the given task. The value of the interest function varies between 0.0 and 1.0. Here, the 1.0 means that the object has the perfect interest in the given task while the 0.0 means opposite.

[2] Performance Function

The performance function is similar to the interest

function except that the performance function tells about performance instead of interest against the given task.

[3] Normal Distribution Curve

The characteristic function has to follow the Normal Distribution Curve (NDC). NDC has enough diversity in terms of standard deviation, which is necessary to get the diversity of agent's ability pattern. That is, the learning mechanism, peculiar to agent, is based on this NDC property. This special constraint promotes understanding among PMs and allow them to have more flexible communication in certain limited range.

C. Agent

We call a system 'CBA (Cell-Based Agent)' that is composed by Cell network. This term shares main characteristics with commonly used word 'agent' from the point of function. That is, CBA is a kind of agent specially implemented by using the cell concept. Only for convenient sake, we use the term 'agent' instead of 'CBA' in this paper.

C.1 Structure of Agent

Agent mainly consists of control-layer and domain-layer. The control-layer takes charge of controlling or coordinating the whole system of the agent. And the domain-layer is the global network itself of multiple cells. The task domain of an agent can be computed as the vector sum of the TM's characteristic functions.

C. 2 Potential Injection

We can inject certain amount of potential into certain port of all PMs or subset of PMs), which we call "Potential injection". There are three types of potential injection; P-type, N-type and L-type depending on which port is used for injection.
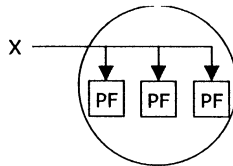
Fig. 3   P-type potential injection

C.3 Entropy

We introduce a new term 'entropy' in this paper to express the agent state effectively. If we perform potential injection with injection value x, the output value of the system $P_\omega(t)$ varies and converges into $P_s(x,t)$. Then we are able to find out the best one among $P_s(t)$ called $P_s^*(t)$. Now, with these assumptions, let's think about a situation that satisfies

following conditions:

[1] $P_s^*(t) > P_\omega(t)$

[2] $P_s^*(t) = P_s(x,t)$

Using these conditions, we can define entropy of a system. First, in case of positive x the system belongs to 'excess state of entropy' and in case of negative x the system belongs to 'lack of entropy'.

III. DEDUCTION OF MODELS

Here, we will show that some of the intelligent models such as emotion and reasoning. Learning, self-organization and fuzzy concept can be designed (or deduced) naturally from SAL by adding only some constraints. In order to deal with the intelligent models it is very helpful to define a SAL-based Information System (SIS) first. SIS can be simply implemented by assigning the facts or rules to the TM. in our model.

A. Emotion

In our model, the emotion can be classified into two types; inner type emotion and outer type emotion. The value of emotion expresses the degree of unhappiness. In case of inner type emotion, the value of it expresses the degree of 'do our best'. And in case of outer type emotion the value of it expresses the degree of 'the result of the task' in terms of the expectation by planner. Unlike outer type emotion the inner type emotion is very difficult to define. But in our model, the inner type emotion can be defined easily by using entropy concept. Fig. 4 shows all the possible emotion types in our emotion model.
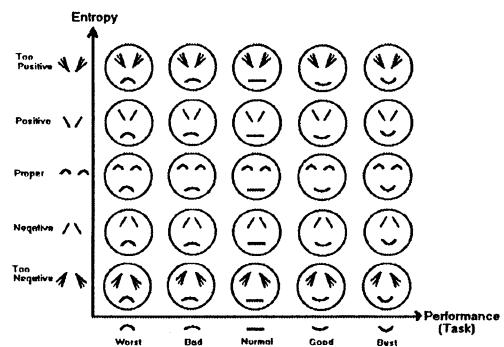
Fig. 4   Emotion Space

B. Reasoning

The reasoning mechanism of cell is based on the very

simple principle. That is, cell basically depends on emotion-based operation and realize reasoning mechanism through investing his received potential to a cell whom he emotionally like. By this means, the typical logic rule 'modus ponens' can be executed in sequence based on emotion. Here, we will explain the process of forward reasoning by using figure3.2 First. two fact-cells [A] and [B] are activated independently. Then, rule-cell [A∧B→X] choose the smaller fact-cell between them and output it through the OF port. Then the output potential triggers two rule-cells and a fact-cell; [X → Y], [X → Z] and [X]. In this case, the output potential is not always restricted by the input potential and not always divided equally among n related cells. In principle, The amount of the potential and the distribution method are wholly delegated to each cell's self-imposed control. Both emotion and reason functionate together in determining how to distribute the potential. For example, rationally, the cell [A∧B→X] should trigger [X→Y], [X→Z] and [X]. But frequently, under the influence of emotion, some of the above rational actions can be forced to be coordinated.
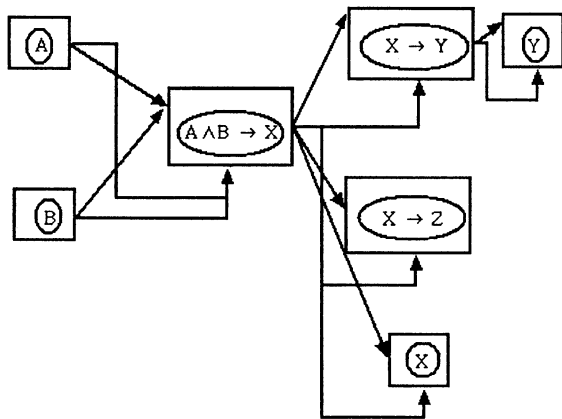
Fig. 5   Example of reasoning

C. Learning and Evolution

The knowledge base of TM is defined as a characteristic function over the component-sequence. In normal condition, the characteristic function forms regular curve. In fact, the ability of task processing depends on the pattern's quality of the characteristic function. In the cell model, the TM's breeding process occurs periodically through six steps, which includes learning process automatically. Fig. 6 shows these six breeding steps in order.

[Step 1]Birth
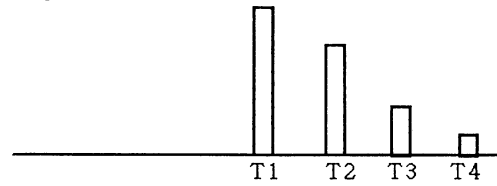
[Step 2]Copy
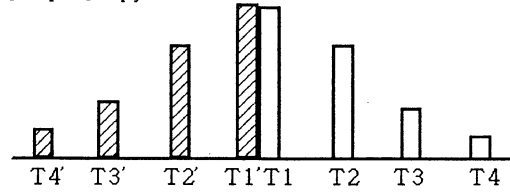
[Step 3]Learn
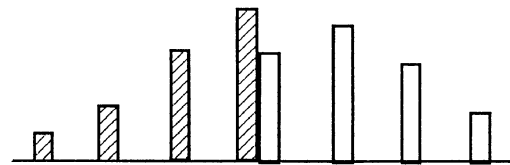
[Step 4]Sort & Modify
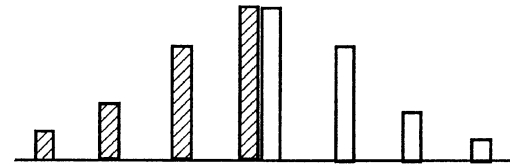
[Step 5]Update OLD-DNA
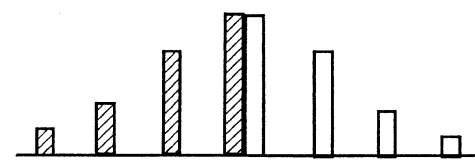
[Step 6]Reproduction
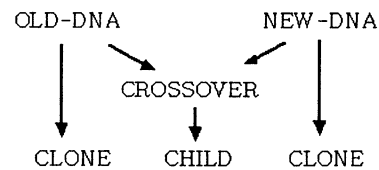
Fig. 6   Breeding cycle of TM

D. Self Organization

Self organization is defined as the process through which the system(agent in this paper) dynamically increase(or decrease) the complexity of the structure. We will explain it by using the case seen in figure 3.8. First the control-cell [IE1] triggers the [Process1] and

the triggered [Process1] triggers [Process2] continuously. Then the [Process2] posts a solution on the BB space. Till this time, the initiative of the system control is surely taken by the sub-system [System 1]. While the conflicting potential which is caused by the collision between NF and PF of [Process1] outputs through CF port. As the CF port has a link with AF port of [IE2], IE2 may be triggered with fully increased input value. And the triggered [IE2], by the similar way, generates Solution2. Then the generated [Solution2] inputs certain amount of potential(outer conflicting potential) in order to strengthen the total conflicting potential while suppressing the [Solution1]. That is, IE2 takes advantage of the positive feedback principle to obtain its safety. Thus the concept of confliction is very important in our model especially from the point of self organization. In theory, as this process can continue infinitely it is possible for the system to control its complexity level properly.
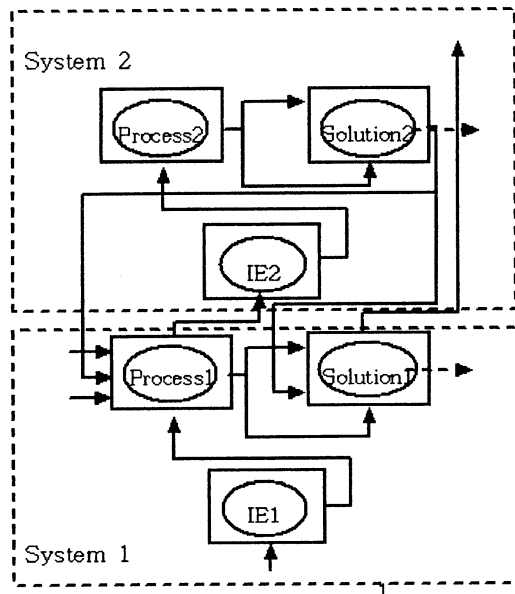
Fig. 7   Example of self organization

E. Fuzzy Concept

A cell should hang out his own title for cooperation or performing joint-project with others. The title of a cell mainly represent his main task, in other word 'major'. The abstract level of the title is decided by himself. This means that the interpretation of the same title may be different each other depending on his own DNA and experience. Originally, cell is defined as a kind of autonomous living thing who is responsible of his own decision or behaviors. Through competition among cells for a time there come into various different interpretations with different strength level, which in many way corresponds to the typical Fuzzy model in AI.
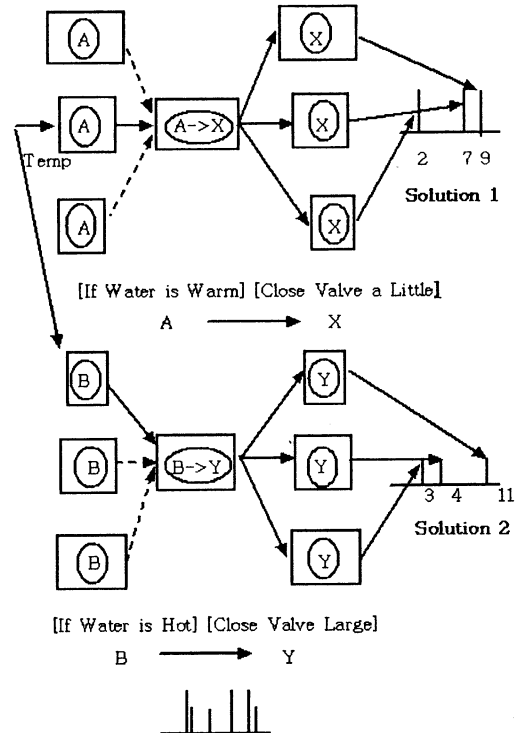
Fig. 8   Example of fuzzy concept

IV. CONCLUSION

In this paper, we suggested a life paradigm SAL. We showed that most of the AI models can be logically deduced from SAL paradigm by adding proper constraints. This means that a system based on SAL naturally have intelligent abilities it really want to have. That is, SAL-based system implicitly contains all the intelligent abilities in itself, which is not at all available in case of reduction paradigm. We assert that the current object-oriented workbench should be evolved into life-oriented one by using SAL concept, For example, the LOL (Life-Oriented Language) using the SAL paradigm, if developed, would be useful enough for the really complex system design, which will be our next study.

REFERENCES

[1]   Sung-Bum Ko, "Internet As a Ecology", *The Korean Society for Biotechnology and Bioengineering*, pp. 306, 2000.
[2]   Sung-Bum Ko, Gi-Young Lim "Life-Based System Design", *2000 International Workshop on Advanced Intelligent Systems*, p. 9, 2000.

[3] R.A. Brooks, "A Robust Layered Control System For A Mobile Robot", *IEEE Journal of Robotics And Automation*, Vol. RA-2, No.1, 1986.

[4] Pattie Maes, "Modeling Adaptive Autonomous Agents", *Artificial Life*, pp. 135-162, 1994.

[5] Sung-Bum, Ko, Gi-Young Lim "The development of EBO System as a Life", *Proceedings of KFIS Fall Conference*, pp. 295-306, 1999.

A Stochastic Model on Lattice Spaces and Antigenic Diversity Threshold

Hiroshi UEDA and Yoshiteru ISHIDA
Department of Knowledge–based Information Engineering
Toyohashi University of Technology
Hibarigaoka 1–1, Tempaku, Toyohashi, Aichi 441–8580

Abstract

A stochastic model based on Nowak–May's model for interaction between HIV (Human Immunodeficiency Virus) and the immune system is presented. By simulations on the stochastic model expressed on lattice spaces, an *antigenic diversity threshold* of HIV is shown. Mutation of HIV causes *antigenic diversity threshold* to decline. However, most of the behavior of this stochastic model strongly depends on initial conditions.

Keywords HIV, Stochastic model, Antigenic diversity threshold, Lattice model, Monte Carlo Simulation

1 Introduction

Many models on the interaction between HIV and the immune system have been proposed[1]. The HIV infection process is characterized by a long and variable asymptomatic incubation period. Another feature of HIV is its enormous genetic diversity by mutation. Mutation of HIV leads to variance of number of HIV strains, hence high diversity.

Nowak and May proposed a dynamical model to account for their *antigenic diversity threshold* theory [2]. According to the theory, there is an *antigenic diversity threshold* of HIV strains over which HIV infection leads to AIDS (Acquired Immunodeficiency Syndrome)[3]. This threshold can be considered a stability condition of HIV population.

Progression to disease is essentially stochastic process, however Nowak–May's model focuses on the deterministic model. The motivation of this study is to build a model that can deal with mutation of HIV thoroughly.

In this paper, we propose a stochastic model based on the Nowak-May's model[2], to examine existence of the threshold in a situation with higher dimension allowing mutation (hence higher diversity) during the interaction.

For this purpose, we devise a stochastic model on two *lattice spaces*: one for characterizing HIV and another for helper T lymphocytes.

In section 2, outline of Nowak–May's model and their *antigenic diversity threshold* theory is given. Section 3 describes basic idea underlying our stochastic model. Section 4 concentrates on a simulation results. Section 5 makes a summary of this study.

2 Nowak–May's model

Nowak and May proposed the model similar to the following[2].

$$\dot{v}_i = v_i(b_i - p_i x_i), \qquad (1)$$

$$\dot{x}_i = k_i v_i - u_i(\sum_{i=1}^{N})x_i \quad (i = 1, \cdots, N). \qquad (2)$$

where v_i and x_i are time dependent variables indicating the population size of HIV type i strain and that of its corresponding clones of T helper lymphocytes stimulated by the HIV type i, respectively. N is number of HIV strains. z is population size of non-specific T helper lymphocytes. The following parameters are all positive;

- r : the net replicating rate of HIV,

- s : the rate of elimination of HIV by non-specific immune response,

- p : the rate of elimination of HIV by immune response specific to type i,

- k : the rate of stimulation of i-specific T-cell,

- k' : the rate of stimulation of non-specific T-cell,

- u : the rate of depletion of T-cell by HIV.

Table 1: Transition types, rates and events. Parameters of in the transition rate correspond to parameters of differential equations (1), (2). μ denotes a mutation rate.

k	Transition	Rate	Event
1	$v_i \rightarrow v_i + 1$	$(1 - \mu)b_i v_i$	HIV Replication
2	$v_i \rightarrow v_i - 1$	$p_i v_i x_i$	HIV Elimination
3	$x_i \rightarrow x_i + 1$	$k_i v_i$	T-cell Stimulation
4	$x_i \rightarrow x_i - 1$	$u_i(\sum v_i)x_i$	T-cell Depletion
5	$v_i \rightarrow v_j + 1$	$\mu b_i v_i$	HIV Mutation

The stability condition for HIV population size can be formulated as follows:

$$\sum_{i=1}^{N} \frac{b_i u_i}{p_i k_i} < 1. \tag{3}$$

Thus, stability can be characterized by the *antigenic diversity threshold* (left–hard-side of (3))[2].

3 A Stochastic Model

Our stochastic model uses *Poisson Process* in which the event (transition) occurs according to the rules shown in Table 1 [4]. Variables; x_i, v_i and parameters b_i, p_i, k_i, u_i are the same as those of the model (1), (2). Mutation is the event that does not appear is the model (1), (2). As in Table 1 ($k = 5$), the mutation occurs with the mutation rate μ. Let the rate of the event k shown in Table 1 be r_k, then the probability p_k of the event k can be given by:

$$p_k = r_k/r \tag{4}$$

where $r = \sum r_k$.

Our model is a fully stochastic model in the sense that not only mutation but other events are probabilistic. Another feature of this model is using two lattice spaces: one for HIV strains and another for T-cell clones. As shown in Figure 1, both HIV lattice and T-cell lattice are arranged based on their characteristic parameters.

For HIV lattice, the population size of v_{kl} characterized by the parameter p_k and b_l is placed at the site (k, l). T-cell lattice is also arranged similarly.

In our stochastic model, the mutation of HIV means increasing of population at other site. In other words, HIV can proliferate to a neighboring site (e.g. 8-neighboring sites) on lattice space.
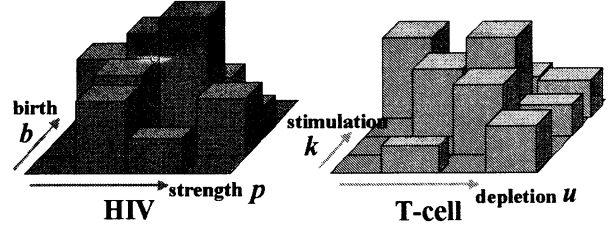
Figure 1: HIV Lattice Space (Left) and T-cell Lattice Space (Right) are sorted by parameters. Each lattice has population size. For example, in HIV lattice, HIV placed on the upper right site has a high birth rate and a strong immune response.

4 Simulations Results

In this section, we confirm that our stochastic model also has an *antigenic diversity threshold* similar to Nowak–May's model. We also investigate an effect of the change of the mutation rate and the mutation range.

4.1 Explosion of HIV population size

In the series of simulations, we use the following rule for identifying *relative explosion of HIV population size*.

We identify *relative explosion of HIV population size* with the fact that total population size of HIV exceeds 1,000 times as many as the total T-cell population.

4.2 Comparison with Nowak–May model

In order to pursue a parallelism between the stochastic model and Nowak–May's model, we performed a Monte Carlo simulation under the following conditions:

- No mutation ($\mu = 0$),

- initial population size of HIV strain type kl is 10 ($v_{kl} = 10$),

- initial number of HIV strains (diversity) N.

Figure 2 shows a simulation result. The fraction of relative explosion of HIV population remains constant until HIV diversity becomes 20, however, it increases suddenly at the value. As shown in Figure 3, *antigenic diversity threshold* is observed as the value $\sum \frac{b_i u_i}{p_i u_i} = 1$, which is consistent with *antigenic diversity threshold* condition (3).
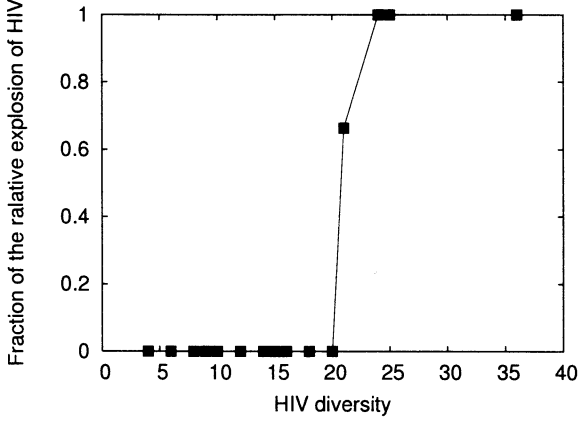
Figure 2: Fraction of the relative explosion of HIV population size versus initial number of HIV strain (diversity). Each plot represents the average of 1,000 runs. *Antigenic diversity threshold* is at the point about 20.

4.3 Effect of increasing mutation rate

We performed a computer simulation with the initial condition similar to that in section 4.2 but varying the mutation rate. Figure 4 shows the effect of the change of the mutation rate. Four cases: $\sum \frac{b_i u_i}{p_i u_i} = 0.85, 1.0, 1.08$, and 1.33 are considered. In the case close to *antigenic diversity threshold*, increase of the fraction of relative explosion of HIV population size is observed.

4.4 Sensitivity to the initial condition of HIV

Since our model is a stochastic implementation of Nowak–May's model, its behavior is excepted to be sensitive to the initial distribution of HIV lattice. Sensitivity to the initial condition is investigated by performing a computer simulation under the following condition:

- the mutation rate $\mu = 0.5$,
- the initial population size of HIV $v_0 = 10$,
- mutation allowed to 8-neighboring sites,
- range of HIV lattice space: 20×20.

Figure 5 shows that HIV on high birth rate and strength on lattice space is predominant to the immune system. In Nowak–May's model, the initial distribution of HIV means not only initial conditions but
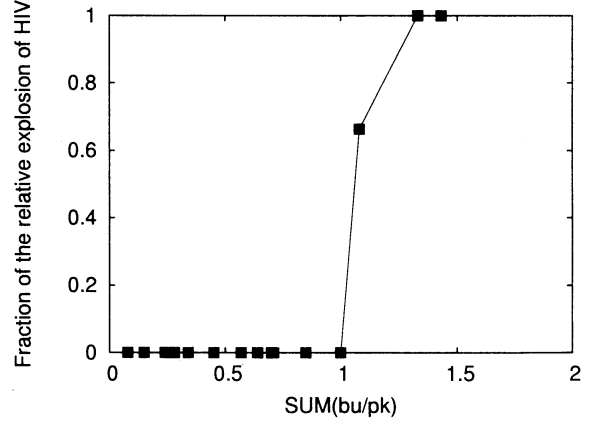
Figure 3: Fraction of the relative explosion of HIV population size versus $\sum \frac{b_i u_i}{p_i k_i}$. The horizontal axis is Nowak–May's diversity measure (3). The *antigenic diversity threshold* is at the point around 1 by the measure.

the difference in the equation itself, this is a natural result.

4.5 Effect of the change of mutation range

Figure 6 shows the effect of the mutation range. A computer simulation is performed under the condition similar to these in section 4.4. Slight fluctuations of fraction of the relative explosion of HIV population size are observed. Change of the mutation range did not affect the behavior of the stochastic model.

5 Summary

A stochastic model expressed on lattice spaces was proposed. Our model corresponds to Nowak–May's model from the viewpoint of the *antigenic diversity threshold*. By performing computer simulations, we investigate the effects on the stochastic model by the following factors: (i)mutation rate, (ii)initial HIV, and (iii)mutation range. The following observations are obtained:

- The behavior is sensitive to the initial distribution on HIV lattice,
- In the case close to the *antigenic diversity threshold*, sudden increase of fraction of the relative explosion of HIV population size is observed.
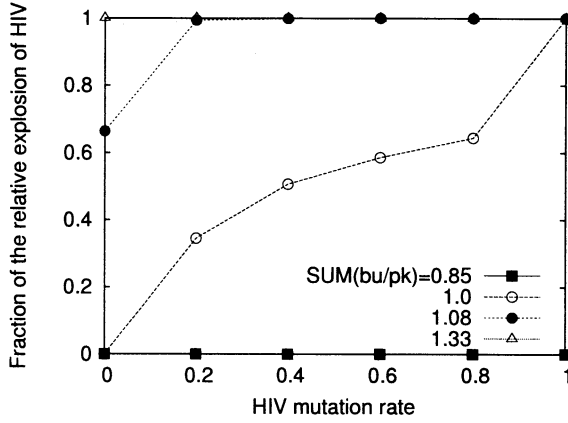
Figure 4: Fraction of the relative explosion of HIV population size versus μ. In the cases $\sum \frac{b_i u_i}{p_i k_i} = 1.0, 1.08$, Increasing of the fraction is observed.

Acknowledgement

References

[1] Angela R. Mclean. The balence of power between HIV and the immune system. *Trends in Microbiorogy*, Vol. 1, pp. 9–13, 1993.

[2] Martin A. Nowak and Robert M. May. Mathematical biology of hiv infections: Antigenic variation and diversity thresholds. *Mathematical Biosciences*, Vol. 106, pp. 1–21, 1991.

[3] Martin A. Nowak and Andrew J. McMichael. How hiv defeats the immune system. *Scientific American*, pp. 58–65, August 1995.

[4] Roy M. Anderson and Robert M. May. The invasion, persistence and spread of infectious diseases within animal and plant communities. *Phil. Trans. R. Soc. Lond. B*, Vol. 314, pp. 533–570, 1986.

Fraction of the relative explosion of HIV
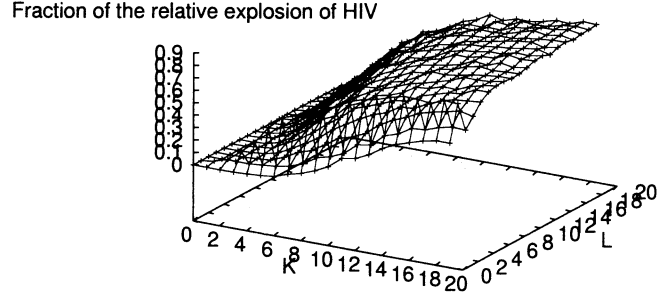
Figure 5: Plots of the initial HIV parameters (p_k, b_l) and the fraction of the relative explosion of HIV population size. HIV parameters are: $p_k = k + 10$ and $b_l = l + 1$ $(k, l = 0, 1, 2, \cdots, 19)$.
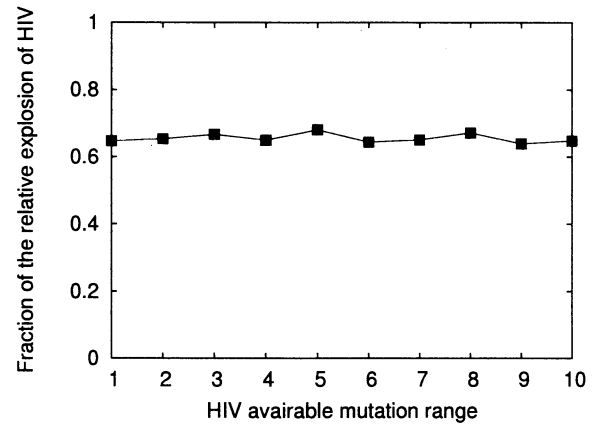
Figure 6: Fraction of the relative explosion of HIV population size versus mutation range. Initial HIV population is at the center of lattice and the size: $v_0 = 10$.

Development of the Basic Structure for an Exoskeleton Cyborg System

T.Onishi, T.Arai, K.Inoue, Y.Mae

Tsuyama National College of Technology, Osaka University, Osaka University, Osaka University

The Robotics Society of Japan, The Robotics Society of Japan, The Robotics Society of The Japan, Robotics Society of Japan

624-1,Numa,Tsuyama-City,708-8509   1-3,Machikaneyama,Toyonaka,560-0043
1-3,Machikaneyama,Toyonaka,560-0043   1-3,Machikaneyama,Toyonaka,560-0043

Abstract

The human internal skeleton is referred to as an endoskeleton, while oyster shells and the cuticles of lobsters and crabs are referred to as exoskeletons. A great deal of research has been conducted in an attempt to combine these endoskeletons and exoskeletons. However, there are hardly any instances of research being undertaken to create a cyborg by combining endoskeletons and exoskeletons. The human endoskeleton consists of three kinematic pairs that are joined together with a maximum of three degrees of kinematic freedom. These are:

1. A turning pair on a lateral-bending axis
2. A turning pair on a longitudinal-bending axis
3. A turning pair on the central axis of a skeleton

If the strength of the muscles on a human skeletal decline, or if component parts of a human skeleton are lost, then a module-type exoskeleton designed with a maximum of three degrees of kinematic freedom would be able to contain an entire body such that it could assist the function of the arms, hands, legs and other body parts to complement the declined or lost physical functions (kinematic or biological functions). Based on this concept, it is possible to develop an exoskeleton cyborg system that behaves in much the same way that a human does in daily life. The functions that this system should be capable of performing include allowing the user to stand up from a bed, to move around the floor to help the elderly or disabled excrete or take a bath, to manoeuvre around stairs and other obstructions, to get in and out of cars, to move to a workplace at a remote location and so on.

1. Introduction

Although freedom of movement forms the basis of human physical functions, it is often restrained by juvenility, illnesses, accidents, wars, ageing, etc. On the other hand, there are cases where a level of power or performance beyond normal human capability is required. The exoskeleton cyborg system described in this paper proposes a technology to complement degenerated or lost physical functions or to augment existing physical functions. It consists of an exoskeleton unit, an exoskeleton cyborg unit, and an exoskeleton cyborg system, which together can perform given functions to complement or augment the existing kinematic and biological power or performance of the human skeleton (including the trunk, arms, hands, legs, head, etc.) and its associated joints. In this system, all the component parts of the endoskeletons (trunk, arms, hands, legs, head, etc.), which are designed to simulate parts of a human body, are contained in a series of exoskeletons that correspond to the cuticles of a lobster or a crab, and they operate in co-ordination with each other. A set of these exoskeletons contains a living organism (human body), and therefore it can be described as an exoskeleton cyborg system.

In this study, a prototype of an exoskeleton cyborg system was made and the overall structure and configuration were examined. As shown in Figure 1, human physical functions change with age. The elderly and disabled are taken care of by parents, families, public nursery staff, nurses, care workers, etc. The work of taking care of them is often relatively hard labour.
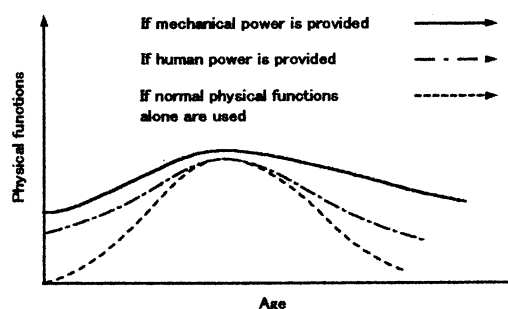
Fig.1. Change in human physical functions

There are some types of work that require a level of power beyond an individual's physical

capability, and the use of additional mechanical power is sometimes indispensable in these cases. However, there are only a limited number of implements that can provide the necessary supplementary mechanical power under these circumstances. Examples of this would include elderly-care beds, various types of wheelchair, toilet support implements, bathing support implements, stair lifts, man-lifts for cars, manually-operated agricultural and construction machinery, etc.

Research and development has been conducted to in an effort to lessen the hard labour involved in the work of helping the elderly or disabled stand up from a bed, move around the floor, excrete, take a bath, avoid stairs, get in and out of cars, move to a workplace, etc. [1], [2]. However, no implements that could be universally applied or universal designs have yet been developed. Although a study of exoskeletons has been initiated at the Department of Defence of the US [3], it is still in the planning stage.

In this paper, we would like to propose an exoskeleton cyborg system. This exoskeleton cyborg system is designed to complement or augment human kinematic and biological functions, and it consists of a body-supporting unit [4] designed to be worn on the trunk and other body parts such as the arms, hands, legs, head, etc. to be attached to the body-supporting unit. In this system, human behaviour is taken into account as media and behavioural data is smoothly and efficiently transmitted between units that are similar in shape in order to augment the power generated by each body part and to enable each part to move speedily and accurately.

2. Configuration

The exoskeleton cyborg system is configured using either all or part of its units or components as required, including a body-supporting unit, a module-type exoskeleton unit, a flexible-wire drive unit with a driver, four pairs of omni-directional independent-suspension rotary wheels (each pair having two wheels, one set opposite to the other), a three-point suspension coalescing unit, and a communicating control unit.
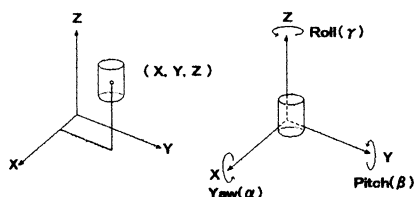
Fig.2 Pose relative to the human skeleton

To determine the configuration of the exoskeleton cyborg system, one pose should be examined relative to the human skeleton. As shown in Figure 2, a pose can be expressed using six generalised co-ordinates, x, y, z, alpha, beta, and γ.

As shown in Figure 3, the human skeleton was approximated using a total of 77 skeletal parts. If these skeletal parts can be held in place using appropriate exoskeletons, it is possible to build a skeleton capable of retaining the whole of a human body in position.
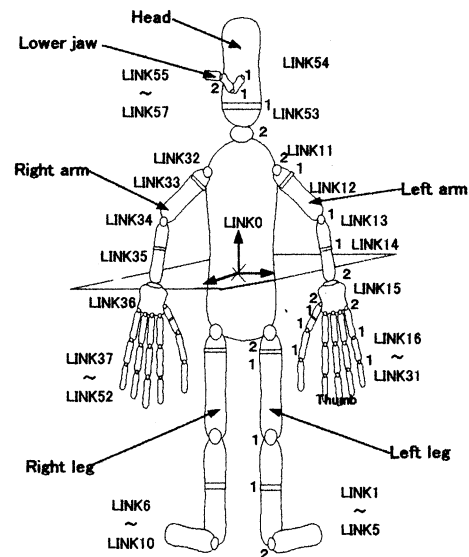
Fig.3 Human skeletal parts approximated using links

The module-type exoskeleton unit shown in Figure 4 is used to contain endoskeletons.
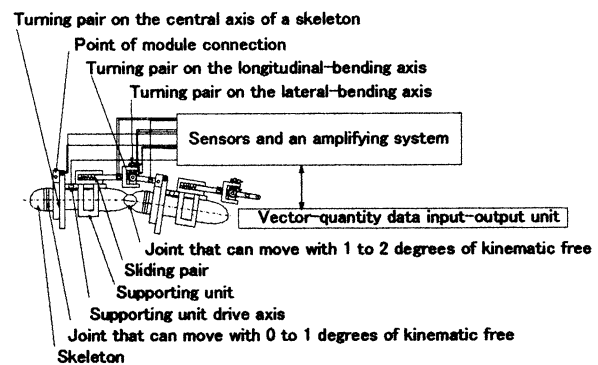
Fig.4 Module-type exoskeleton

This exoskeleton unit consists of a first turning pair for lateral bending at the joint with the exoskeleton, a second turning pair for longitudinal bending at a point perpendicular to the exoskeleton, and a third turning pair for trochoidal movement on the central axis of a skeleton. If necessary, a sliding pair is mounted on the frame to support the motion generated by

the first or second turning pair. Force and displacement sensors are also mounted on a drive unit to achieve a specified purpose.

2.1 Arm- and hand-supporting units

Multiple exoskeleton units are fitted to the trunk, arms, hands, legs and head to build an exoskeleton cyborg.

Exoskeletons with 7 degrees of freedom ⟶

Endoskeruto with 7 degrees of freedomn ⟶
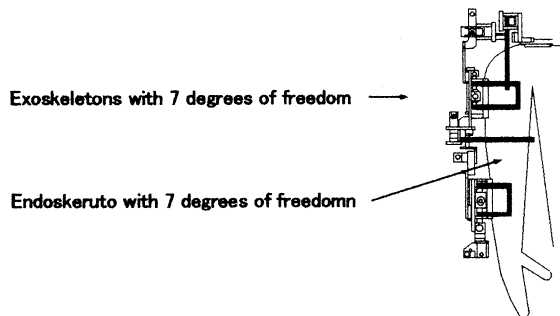
Fig.5 Arm-supporting unit

As shown in Figure 5, exoskeleton units are mounted on each of the shoulder, elbow and wrist so that they can function as a single arm-supporting unit. The first exoskeleton unit mounted on the shoulder is actuated by three turning pairs and two sliding pairs. The second one mounted on the elbow is actuated by one turning pair and one sliding pair. The third one mounted on the wrist is actuated by one turning pair on one axis, two turning pairs at right angles to one another along the skeleton, and two sliding pairs. The arm-supporting units on both the right- and left-hand sides can move with seven degrees of kinematic freedom, and they are the exoskeleton cyborg units capable of status detection and drive control.
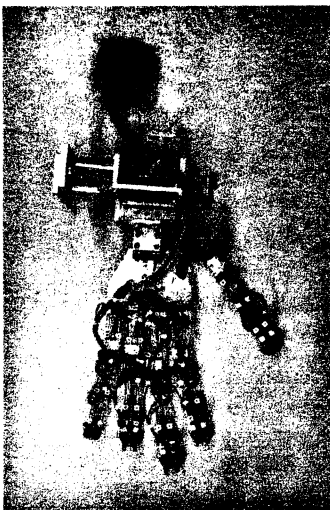
Fig.6 Hand-supporting unit

The hand-supporting unit shown in Figure 6 is set on the joints of the thumb and other fingers and is then mounted on the hand endoskeletons. The exoskeleton at the thumb is actuated by five turning pairs and three sliding pairs. Each exoskeleton unit at each finger is actuated by four turning pairs and three sliding pairs. The hand-supporting units on both the right and left sides can move with 21 degrees of kinematic freedom, and they are the exoskeleton cyborg units capable of status detection and drive control.

2.2 Leg-supporting unit

As shown in Figure 7, exoskeletons each with seven degrees of kinematic freedom are installed and endoskeletons are mounted on them to build the leg-supporting unit.

The first exoskeleton unit at the hip joint is actuated by three turning pairs. The second one at the knee is actuated by two turning pairs and one sliding pair. The third one at the ankle is actuated by two turning pairs and one sliding pair. The leg-supporting units on both right and left sides can move with seven degrees of kinematic freedom, and they are the exoskeleton cyborg units capable of status detection and drive control.

14 degrees of freedom
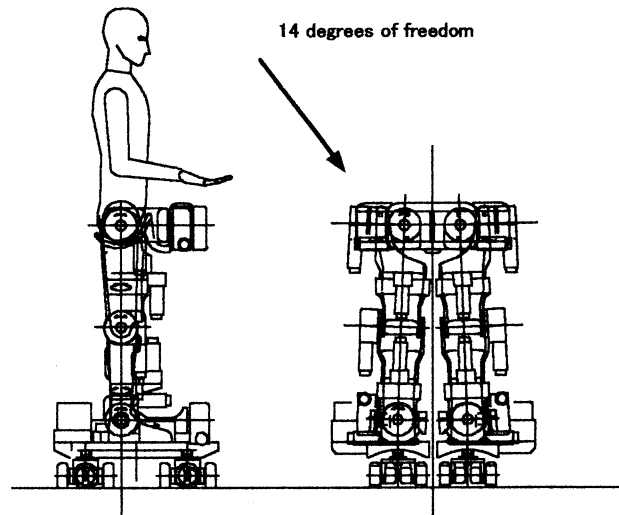
Fig.7 Leg-supporting unit

1.3 Head-supporting unit

The head-supporting unit consists of an exoskeleton unit with three degrees of kinematic freedom at the neck, one with four degrees of kinematic freedom at the lower jaw, and endoskeletons each at the head and the neck. This exoskeleton cyborg unit can move with seven degrees of kinematic freedom.

2.4 Body-supporting unit

The body-supporting unit is made-up as shown in Figure 8(a),(b). The trunk is contained inside the body-supporting unit shown in Figure 8 (c). The body-supporting unit consists of a leg-opening frame, an arm-opening frame and a shoulder-opening frame. With the lever set in the closed position, the body-supporting unit can be mounted on the trunk and locked, as shown in Figure 8 (d).

(a)   (b)

(c)

(d)
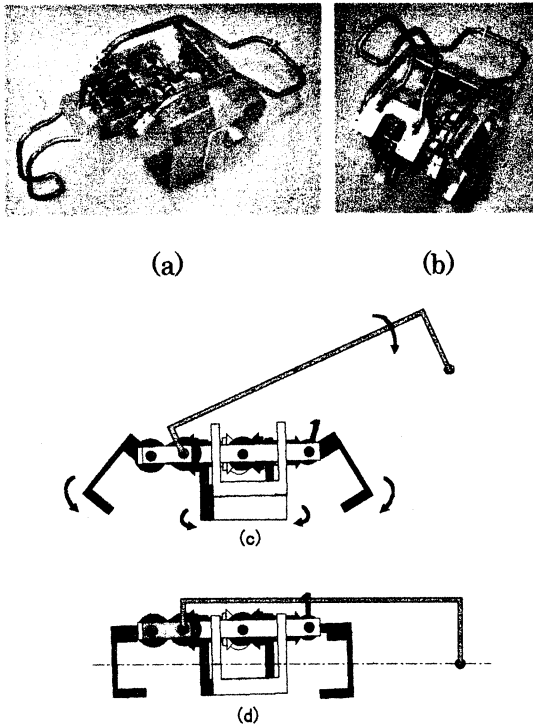
Fig. 8 Body-supporting unit

The previously mentioned arm, hand, leg and head-supporting units are coupled to the body-supporting unit such that they together contain and hold the trunk, arms, hands, legs, and head.
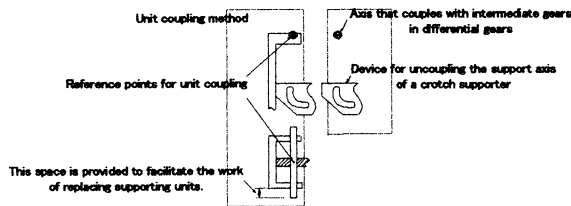
Fig.9 Three-point suspension coalescing unit

Each unit is coupled using the three-point suspension coalescing unit shown in Figure 9. By coupling all of these units, the exoskeleton unit shown in Figure 10 can be built.

2.5 Hoist unit

A host unit for hoisting the body-supporting

unit is shown in Figure 11(a),(b). The body, arm, hand, leg and head-supporting units must be placed in the hoist unit before they are raised. The body-supporting unit can move both upward and downward and rotate. The hoist unit is equipped with four pairs of wheels (each pair with two wheels, one set opposite to the other) and can move in all directions. After all of the units are coupled to perform the full range of functions, the data transmission function is activated to achieve co-ordinated movement.
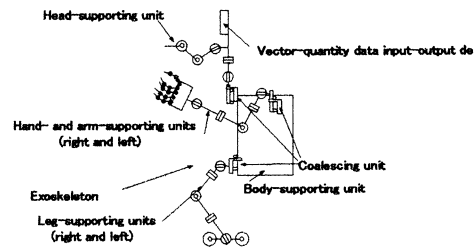
Fig.10 Exoskeleton unit
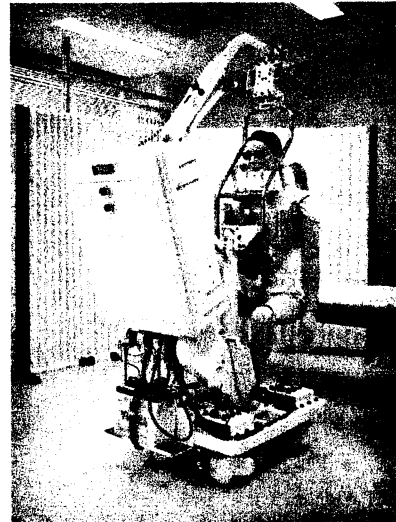
(a)
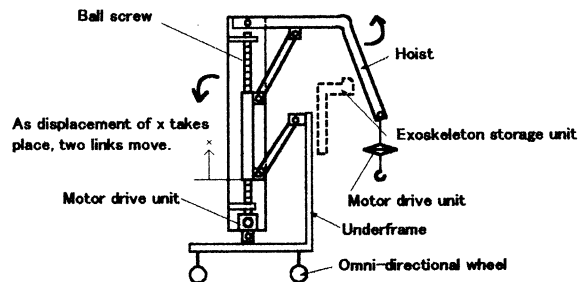
(b)

Fig.11 Hoist unit

3. Control Method

3.1 Wheel control

The principles of the omni-directional movement of the four pairs of wheels (each pair with two wheels, one set opposite to the other) are shown in Figure 12. The movement to a target point D (x, y,??) is controlled at the centre of rotation C (xc, yc). A hoist unit that has four pairs of wheels (each pair with two wheels, one set opposite to the other) is used, and therefore the centre of rotation of each pair of wheels must be the same so that all wheels can turn smoothly without friction,

Fig.12 Principles of omni-directional movement

Steering and travelling are controlled by the rotation of two wheels, one set opposite to the other, and therefore the right- and left-hand wheels must be controlled independently. This independent wheel control is shown in Figure 13.

Fig.13 Control of opposing sets of two wheels
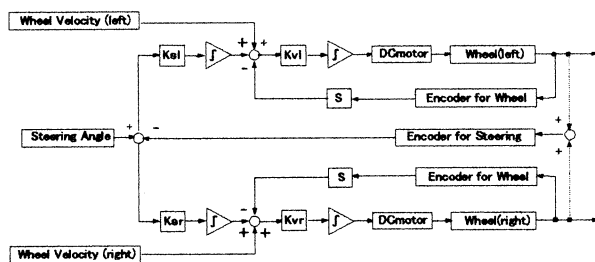
The leg-supporting unit uses the same four pairs of wheels. It can travel at different rates, from a normal walking pace to a fast walking speed.
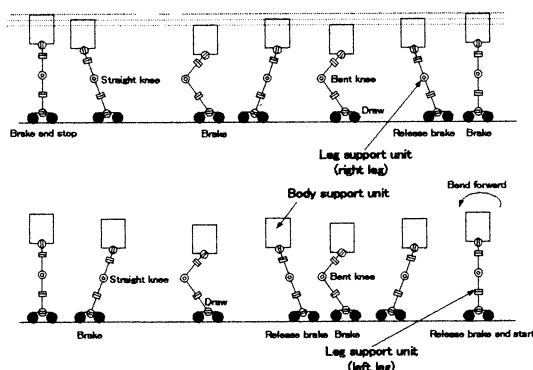
Fig.14 Fast walk

3.2 Joint control

The data is subjected to co-ordinate conversion using the DH method. The values obtained from this co-ordinate conversion are fine tuned, as shown in Figure 15, and they are used to actuate the joints.

Fig.15 Data tuning

The data is tuned so that the joints can be actuated with the most appropriate data values for each specified condition. That is, an amplifying circuit is used to convert the minimum and maximum signals of the input data to the minimum and maximum output data, and to supply the most appropriate levels of drive signals to each joint drive unit.

3.3 Raising up

Raising up is accomplished using the hoist unit. To raise a person from a supine position up into arms, the exoskeleton units are consecutively linked and coalesced. For this linking and coalescing to take place properly, the co-ordinates of all relevant data must be converted using the solution of the DH method shown in Figure 16.

$$Ai = \begin{bmatrix} \cos\theta i & -\cos\alpha i \sin\theta i & \sin\alpha i \sin\theta i & a_i \cos\theta i \\ \sin\theta i & \cos\alpha i \cos\theta i & -\sin\alpha i \cos\theta i & a_i \sin\theta i \\ 0 & \sin\alpha i & \cos\alpha i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r(0) = A_1 A_2 \cdots A_i \ r(i)$$

Fig.16 Solution of the DH method

The action of raising a person up can be analysed, as shown in Figure 17.

i	θi	di	ai	αi
1	$\theta 1$ (0)	0	a1 (x)	$-\pi/2$
2	$\theta 2$(0)	d2 (y)	0	$\pi/2$
3	$\theta 3$	d3	0	$-\pi/2$
4	$\theta 4$	0	a4	0
5	$\theta 5$	0	a5	0
6	$\theta 6$	d6	0	$-\pi/2$
7	$\theta 7$	d7	0	$\pi/2$
8	$\theta 8+\pi/2$	0	a8	$-\pi/2$
9	$\theta 9$	0	a9	0
10	$\theta 10 +\pi/2$	d10	0	$\pi/2$
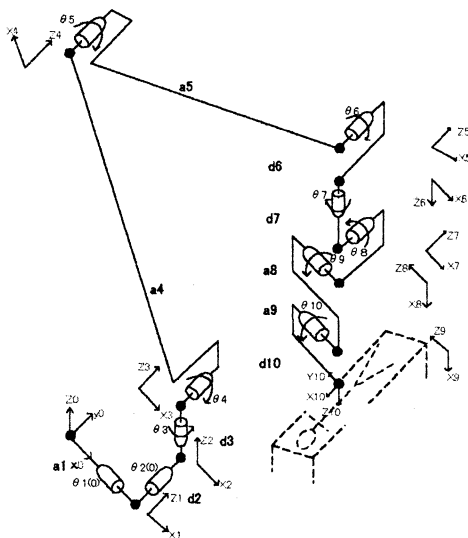
Fig.17 Action of raising a person up and results of analysis

3.4 Signal transmission

The vector-quantity data input-output unit shown in Figure 18 is used to co-ordinate the movements of the trunk and the arm, hand, leg and head support units.
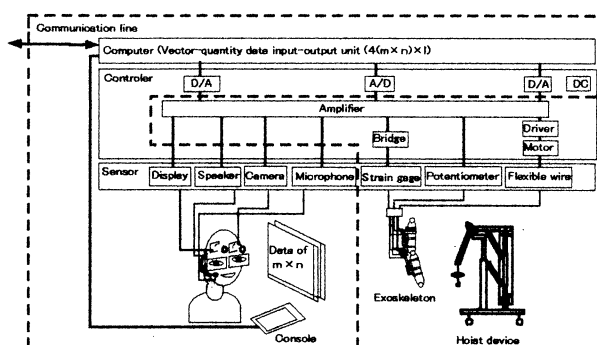
Fig.18 Vector-quantity data input-output unit

The vector-quantity data input-output unit converts the angle, force, image and audio signals to vector-quantity data and transmits them.

4. Conclusion

Appropriate exoskeleton units are installed to form arm- and hand-supporting units, leg-supporting units, head-supporting unit, etc. They are coupled with the body-supporting unit to build an exoskeleton cyborg system.

The exoskeleton cyborg system is controlled using sensors, actuators and a vector-quantity data input-output device so that it can support the elderly or disabled in performing physical functions or kinematic and biological functions.

If it is used together with the hoist unit, it can provide the elderly or disabled with additional power so that they can stand up from a bed, move on the floor, excrete, take a bath, avoid stairs, get into and out of cars, move to a workplace, etc.

Furthermore, data on the motion of each joint in the exoskeleton cyborg system can be smoothly and efficiently transmitted to another exoskeleton cyborg system similar in shape at a remote location via a communication line. Using this feature of the exoskeleton cyborg system, it is possible to demonstrate how the exoskeleton cyborg system can assist the elderly or disabled in complementing or augmenting human power and performing physical functions with a satisfactory level of speed and accuracy.

Acknowledgements

References

[1]H.Kazerooni,J.Guo,"HumanExtenders",Journl of Dynamic Systems,Measurement,and, Control,Vol.115,pp.281-290,1993.

[2]A.Nakai,Y.Kunii,H.Hashimoto, "Development of DOF Exoskelton Type Haptic Interface",JRSJ Vol.17,pp.74-81.

[3]DARPA,"Exoskeleton For Human Performance Augmentation",2000.

[4]T.Onishi,"Device For Holding Human Body", United States Patent, PatentNumber ,6105184, 2000.

Physically grounding the lexical semantics of words in a robot visual perception

Nicolas Bredèche
LIMSI-CNRS, Univ. Paris 11
Orsay, France
LIP6-CNRS, Univ. Paris 6
Paris, France

Yann Chevaleyre
LIP6-CNRS, Univ. Paris 6
Paris, France

Abstract

In this paper, the problem of anchoring objects' names from a human lexicon to a robot perceptions is studied. An open problem is to define a good perceptual system supporting efficient anchoring of these symbols. To study this problem, we consider a detection task used and we study and show results from an approach based on an iterative reformulation of raw percepts into a structural description of perceptions. The goal of this perceptual learning is to provide the robot with the ability to achieve tasks in the environment needing object detection such as object tracking or simple interaction with humans about objects in a real-world environment.

1 Introduction

In a real world environment, an autonomous mobile robot can be confronted to countless situations where various objects or other agents (human or robot) are likely to stimulate its percepts. Because of the richness of the environment, programming may require robots to build their own representation from their perceptions and not from a priori knowledge. Moreover, the robot must take into account noise in order to build a perceptual system.

In this paper, we study how a robot builds a perceptual system for a specific task : anchoring a human lexicon for robot-human communication. From the robot's standpoint, this task is about learning to bind symbols to data from a video camera. This link is not easy to build because the video camera provides images as raw perceptions that are difficult to handle.

In order to ground this lexicon, we consider building a perceptual system with the ability to verify object's occurrences in the environment. Detection provides an efficient framework to achieve this verification task and can be made possible thanks to an appropriate reformulation of raw data from the percepts. Raw data are thus reformulated several times to build a structured description of the environment. This reformulation mechanism is meant to build the most fitted perceptual system for the given detection task.

From the robot standpoint, The ability to verify an object occurrence in the environment through detection can be useful to achieve object tracking, object identification or simple human-robot grounded communication tasks.

2 Grounding a lexicon in images

To share a lexicon of objects' names with a human, the robot must make sure that it is capable of grounding these symbols in its perceptual system. Moreover, there is no way to identify explicitly which percepts are relevant to detect a specific object. Anchoring in this setting may be seen as a supervised learning task with a set of images labelled positive or negative according to the presence of a specific object in a small part of the given image (i.e. the target concept).

Several research domains are concerned with the problem of identifying objects in images. On the first hand, the vision and pattern recognition community have investigated the problem of object detection hidden in images using model based approaches. While works in this field lead to efficient algorithms for complex object recognition, these methods are not well suited to autonomous robotics. On the second hand, many researches in the field of image retrieval and classification are interested in classifying labelled images. In this case, algorithms rely on image descriptions that are both simple and efficient. As a matter of fact, a global colour histogram on the whole image captures enough relevant information and is enough to yield good results [8].

In situated perception, the robot builds its own representation from its perception of the world and has to deal with raw and noisy data from the percepts. The

Image from the camera | Grid-based description | Two examples of multiple instance description
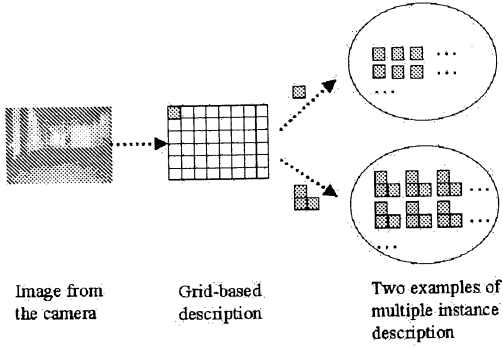
Figure 1: A simple grid-based description of the image and two possible corresponding multiple instances descriptions.

mechanism that consists in a long-term building of a link between symbols and raw data from the robot's perception is referred to as anchoring [3] and is different from the object recognition and the image indexing tasks. Anchoring does not consist in explicitly identifying an object, but is closer to a detection task, regardless of the orientation, size or position of the object. In fact, the object to be detected is often hidden in the images. In this paper, we study how a robot can build a perceptual system to extract structured data out of the raw data form the percepts to perform an efficient anchoring.

This work is done in the context of the MI-CROBES collective robotics project[7]. The robot we have used is a PIONEER2DX mobile robot with an LCD camera (the images we use are limited to 160*120 in resolution with a 24 bits colour information).

3 Detecting concept occurrences in perceptions

3.1 A definition of detection

As described in the previous section, the robot has to detect objects hidden in a small part of images. The robot first gets the image from the video camera, and divides it according to a *grid* with fixed dimension in order to get a set of *elements*. For example, figure 1 shows a $160 * 120$ image divided into 48 elements thanks to a $8 * 6$ grid with every elements covering $20 * 20$ pixels. In this case, *elements* of the grid corresponds to *parts* of the image.

We define the detection task as follow : in order to detect an *object* inside an image, it is enough to detect a relevant part of the object in one part of the image.

Here, the term *object* means both an inanimate object (door, window, ashtray...), another robot or a human. Thus, the problem is to find a relevant part of the image with the right data among the set of parts to conclude that an object is present.

3.2 A multiple-instance representation to describe images

In order to describe an image as a set of parts, the robot first extract elements from the image and then create a *multiple instance* description that will be used for a learning task. The multiple instance (MI) learning problem was first introduced by Dietterich et al. [4].

Within the multiple instance framework, objects are represented as *bags of vectors* of variable size. Vectors are also called *instances*. The size of a bag b is noted $\sigma(b)$. Its instances are noted $b_1 \ldots b_{\sigma(b)}$. Let $\mathcal{X}$ be a feature vector space, and $\mathcal{Y}$ the finite set of labels or classes. The multiple instance induction task consists of finding a classifier $H : 2^{\mathcal{X}} \to \mathcal{Y}$, which accurately predicts the label $F(b)$.

The multiple instance learning problem has been associated to a bias introduced by [4], which will here be refered to as the *single-tuple bias*. It can be formally defined as follows [1] :

Definition 1 *The single-tuple bias is a restriction on the set of functions* $H : 2^{\mathcal{X}} \to \{\oplus, \ominus\}$ *to those for which there exists a function* $h : \mathcal{X} \to \{\oplus, \ominus\}$ *such that* $H(b) \equiv \exists i, h(b_i)$.

The underlying idea is that for certain learning tasks, if a bag is labeled positively, then at least one of its instances must be responsible for this.

Another way to explain the *single-tuple bias* within the multiple instance framework is to learn with ambiguity. In this paper, we consider a single part of the image to be enough to detect an object while all the other parts are irrelevant. Thus, the problem is to find the relevant instance in the multiple instance description of the image. In figure 1, the image is described by a set of elements and two possible multiple instance description are shown. From an instance structure, the description generated from the image is a set of all the possible way to apply this instance. The image can be described by an instance embedding 1 to n elements since embedding more than one element into each instance enables to explicit relations between elements in the image and is a good compromise between complexity and richness since only selected relations are taken into account during learning. In fact, the goal

is to find a good instance's structure to describe the image for this detection task.

In order to evaluate our work, we have used RIP-PERMI [1], a multiple instance extension of the fast efficient rule learner RIPPER[2]. RIPPERMI returns a set of rules that cover the positive examples.

4 Perceptual learning : an iterative reformulation-based approach

4.1 A structural reformulation

For living things, perceptual learning [6] corresponds to mechanisms by which perceptual capabilities are adapted in the short and long term to the environment and to the symbol system. In the same way, the robot tries to build a perceptual system which gives a reliable description out of raw perceptions in order to perform a given learning task.

In machine learning, term-abstraction [5] refers to the creation of new kinds of instances to enhance the accuracy of the learning task. This abstraction technique can find its application in detection as it is defined in this paper, since the robot has to find the appropriate instance's structure through iterative reformulation of the data. The robot reformulates the data by using neighbourhood relations among elements in the images and tries to find a good trade off between the number of neighboured elements embedded in the instance and efficiency. This is known as Propositionalization[9], as opposed to a relational representation of the data where complexity due to exploring all known relations is difficult to handle. In figure 1, we can see two possible descriptions each using a specific instance's structure. The first instance covers one element, while the second instance covers three neighboured elements. Term-abstraction in this case should help the instance's structure to converge toward goal-adapted structure (e.g. 2 vertically connected elements for face detection (face and hair)).

4.2 The PLIC implementation

In order to experiment this approach, we defined and implemented the PLIC system (Perceptual Learning by Iterative Construction), which is both a structural reformulation tool as well as a wrapper that explores the possible structures for defining the instance according to given search hypotheses. PLIC can be used with or without the wrapper tool.

PLIC generates a multiple instance description out of a proprietary Datalog description of an example.
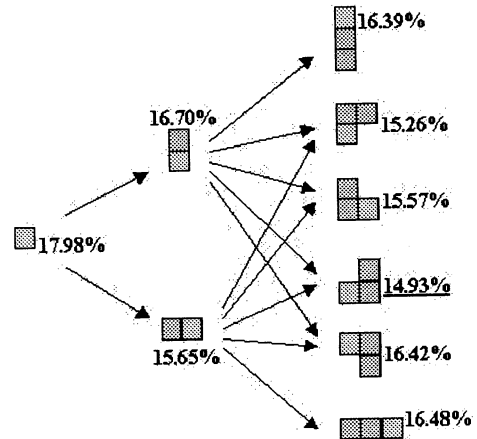
Figure 2: Experiments done with the wrapper on 3 levels of structure syntax (with error in %)

This Datalog description is divided into 3 data description sets: the first concerns global data for the example, the second concerns data related to a part of the example (e.g. here, the elements of an image), and the last is about relations between parts as described in the previous section. An instance is then defined by a set of construction rules which is applied starting from every elements of the image. Filter rules can also be added in order to select instances while building the description.

The wrapper tool in PLIC starts with a basic definition of the instance, and then build the new structurally enhanced instances thanks to the given construction rules. The descriptions generated by all of the different instances are then evaluated and the process is repeated until a predefined termination state is reached.

5 Experiments on structured data

The wrapper tool of PLIC generated descriptions from 3 levels of structured instances with the obligation of extending an instance only from the last added element. These descriptions are based upon a 8 * 6 grid division of a 160 * 120 image (i.e. granularity of 48 elements). We use RIPPERMI with a 10 * 10 fold cross validation to evaluate these descriptions.

Figure 2 shows the results produced by the wrapper. Starting from the simplest element (which leads to a non structural description), the wrapper extends the structure of the original instance into 2 structurally distinct instances. Then, six other structurally new instances are extended from the 2-elements structured

instances. These instances are then applied from every single elements of the images, always leading to a 48 instances description for each image. To summarize, there are three different structure definitions which correspond to nine different structures.

Results from the experiments show that the lowest error rate is obtained with one of the most complex instance structure definition explored (14.93%), which clearly outperforms the results from the non structural description (17.98%) given the standard deviation is approx. 0.5% for all results. However, other results with similar structure syntax (i.e. with every instances embedding three elements) do not always give better results than those obtained at the previous level of structure syntax complexity (i.e. with every instances embedding two elements). This is due to the fact that adding a new element to an instance's structure only improves efficiency if relevant information concerning a relation between all the elements of the instance is found. On the contrary, if adding a new element does not lead to producing rules that takes into account relations among all the elements of the instance, the added attributes of the new elements will make it more difficult to find relevant rules by expanding the search space within every instances because of overfitting. Thus, the best structure obtained represents a trade off between its complexity and the accuracy it leads to.

6 Conclusion

In this paper, we address the problem of a robot anchoring a human lexicon to its sensory data. To achieve this task, the robot has to build a perceptual system that extracts relevant perceptual information out of raw data from the percepts.

We have proposed a perceptual learning approach to build this high level perceptions which goal is to create a structured description out of what the robot sees. We defined and implemented the PLIC system that iteratively reformulates raw data into structured data according to given rules. The experiments done with data generated by this tool yielded good results compared to a non-structural approach.

This work finds its application in a real-world robotic environment and is embedded in a Pioneer 2DX mobile robot that is able to build a high level perceptions adapted to its environment and to the concepts to learn (i.e. names of objects). Applications for this work can be acquiring and tracking targets, looking for objects, describing the environment with a human lexicon...

Future works include using the robot's motor capabilities in order to perform active learning (looking for examples, confirming intuitions about detecting an object by getting closer, etc.). This would extend our work towards a better use of the action-perception loop to anchor a lexicon.

Acknowledgements

References

[1] Chevaleyre Y. and J.D. Zucker. "A Framework for Learning Rules from Multiple Instance Data", *ECML*, 2001.

[2] Cohen, W. W.. "Fast effective rule induction". *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995.

[3] Coradeschi S., A. Saffiotti. "Anchoring symbols to sensor data: preliminary report". *Proceedings of AAAI-2000*, Austin, Texas, July 2000.

[4] Dietterich, T. G., R. H. Lathrop and T. Lozano-Pérez. "Solving the Multiple-Instance Problem with Axis-Parallel Rectangles", *Artificial Intelligence*, volume 89, no 1-2, 1997.

[5] Giordana A., L. Saitta. "Abstraction: a general framework for learning". *In Working Notes of the AGAA-90 Workshop*, Boston, MA, 245-256.

[6] Goldstone, R. L. "Perceptual Learning". *Annual Reviews of Psychology*, 49:585-612, 1998.

[7] Picault S., A. Drogoul. "The MICRobES Project, an Experimental Approach towards Open Collective Robotics". *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems*, Springer-Verlag Tokyo Inc., 2000.

[8] Stricker M. and M. Swain. "The Capacity and the Sensitivity of Color Histogram Indexing", *Technical Report 94-05*, Communications Technology Lab, ETH-Zentrum, 1994.

[9] Zucker, J.-D. and J.-G. Ganascia. "Changes of Representation for Efficient Learning in Structural Domains", *International Conference in Machine Learning*, Bari, Italy: Morgan Kaufmann, 1996.

Effectiveness of Simplified Geometric Models in Planning Using Manipulation Skills

°**Akira NAKAMURA**   **Takashi SUEHIRO**

National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan
E-mail: a-nakamura@aist.go.jp, t.suehiro@aist.go.jp

Abstract

Dexterous manipulation plays an important role in working robots. Manipulator tasks such as assembly can generally be divided into several motion primitives. We call these "skills" and explain how most manipulator tasks can be composed of skill sequences. Skills are also used to compensate for errors both in the geometric model and in manipulator motions. There are dispensable data in the shapes, positions and orientations of objects when achieving skill motions. Therefore, we can simplify geometric models by considering the dispensable data in a skill motion. This paper describes simplified geometric models in planning of skill-based manipulation and its effectiveness.

Key words: manipulation skill, geometric model, planning

1. Introduction

For manipulating robots to be useful in several fields, it is necessary to realize various tasks by using special techniques. By analyzing human motions during tasks such as assembly and disassembly, movements were found to consist of several significant motion primitives. We called these "skills" and showed that most of the tasks of the manipulator can be composed of sequences of skills. That is, we demonstrated that robots can perform various human tasks by using the concept of skill.

Skills in which the contact states vary during assembly and disassembly tasks are particularly significant. We considered three skills of "move-to-touch," "rotate-to-level" and "rotate-to-insert," which play an important part in such tasks. Most assembly tasks can be composed of these three skills.

Skills are also used to compensate for errors both in the geometric model and in manipulation behavior. There are dispensable data in the shapes, positions and orientations of objects when achieving skill motions in a task. Therefore, it is possible to change simply the shapes of objects when planning manipulation by considering the dispensable and indispensable data in

a skill motion.

In this paper, we show the relation between manipulation skills and simplified geometric models. Then, the relation is explained clearly using many examples of tools and parts. We also show the influence on simplified geometric models of different skill sequences in the same task.

2. Manipulation Skills

This section explains our concept of skills. See References [1]-[4] for more details.

2.1. Skill primitives

In assembly and disassembly tasks, skills in which the contact states vary are particularly significant. In References [5]-[8], we considered three skills, "move-to-touch," "rotate-to-level" and "rotate-to-insert," which play an important part in such tasks. For simplicity, we assume that a grasped object, another object and the hole are rectangular parallelepipeds.

(1) Move-to-touch Skill

The move-to-touch skill is defined as the transition of a grasped object P in a constant direction that continues until a contact with another object Q occurs (Fig. 1(a)). This skill is performed in velocity control mode. Not only the transition in free space (Fig. 1(a)), but also the sliding while keeping contact in a different direction of motion (Fig. 1(b)) is included in this skill. These two transitions are represented respectively by the move-to-touch$_f$ skill and the move-to-touch$_c$ skill. The achievement of this skill can be detected by instantaneously increased resistance in the direction of the transition.

(2) Rotate-to-level Skill

The rotate-to-level skill is defined as rotation motion around either a contact point or a contact edge to match the face of the grasped object P with the face of another object Q (Fig. 2). This skill is performed with a pushing force. The achievement of this skill can be detected by the change of the instantaneous center position.

(a) Move-to-touch$_f$ skill (b) Move-to-touch$_c$ skill
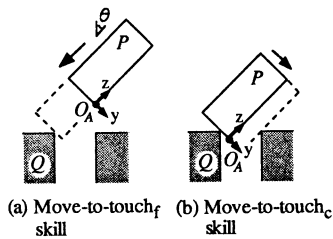
Fig. 1 Move-to-touch skills
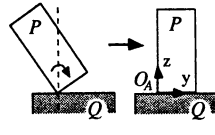
Fig. 2 Rotate-to-level skill

Fig. 3 Rotate-to-insert skill

Fig. 4 Rotate-to-bite skill

Fig. 5 Rotate-to-loosen skill

Fig. 6 Move-to-touch skill

(3) Rotate-to-insert Skill

In an insertion task it is generally difficult to achieve the state in Fig. 3(b) directly. The state in Fig. 3(a) is achieved first by using the skill sequence of Fig. 1(a) and (b). The state in Fig. 3(b) is then accomplished by gradually raising the object while maintaining contact as in Fig. 3(a). The rotate-to-insert skill is this motion of rotating the object P obliquely into the hole in another object Q to insert it accurately. In our study, we assume that the rotate-to-insert skill also includes the pressing motion required to achieve the goal of the insertion task (Fig. 3(c)). The achievement of each rotate-to-insert skill is detected by the variation of the instantaneous center position.

Furthermore, we will consider two skills related to the tasks in the examples mentioned later. In these two skills, we use a cross-head screwdriver and screw according to tasks as examples.

(4) Rotate-to-bite Skill

This skill is a rotation motion around the axis of the screwdriver to fit the tip of the screwdriver into the flutes of the screw head (Fig. 4). This skill is performed with pushing force.

(5) Rotate-to-loosen Skill

This skill is defined as a rotation motion to loosen the screw (Fig. 5). This is performed by matching the axes of rotation of a part and a tool. If these axes do not correspond, the tool is moved to the position before execution of rotation. In this paper, we assume that this skill also includes the transition to remove an error before rotating.

2.2. Composition of Skill Sequence

A specific task is composed of sequences of skill primitives such as move-to-touch, rotate-to-level and rotate-to-insert skills. The skill sequences can be decided by several methods. We showed a method using variations of the number of contact points in skill primitives [5].

3. Dispensable and Indispensable Data for Skills

In this section, we will consider the environment data necessary to achieve manipulation skills [7].

(1-1) Move-to-touch$_f$ Skill

We assume that the Y, Z-axes of coordinates are positioned such that the direction of motion of an object P corresponds to $-Z$ (Fig. 6). This skill is executed after the following condition of objects P and Q is checked.

(a) P is just over the goal of Q with respect to the Y-direction.

The depth of the Z-direction is unnecessary.

(1-2) Move-to-touch$_c$ Skill

This skill is executed after the following condition is checked (Fig. 1(b)).

(b) P has a contact point on Q.
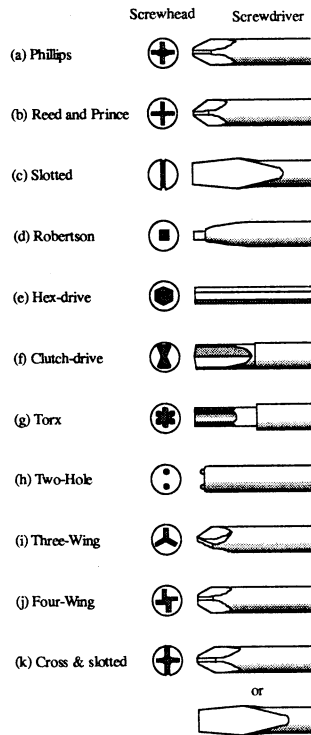
Fig. 7 Screwdrivers
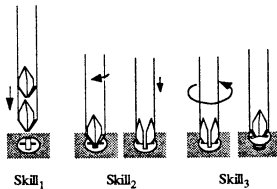
Fig. 8 Skill sequence of loosening with
a cross-headed screwdriver

(2) Rotate-to-level Skill

This skill can be executed if an object P has the following relationship to an object Q (Fig. 2).

(c) P has a contact point on the upper side of Q

It is not necessary to measure accurately in the direction of rotation.

(3) Rotate-to-insert Skill

This skill can be executed if an object P has the following relationship to an object Q (Fig. 3).

(d) P has two contact points at the edges of the holes of Q as shown in Fig. 3(a).

Accurate measurement in the direction of rotation is not needed.

(4) Rotate-to-bite Skill

This skill can be executed if the screwdriver P has the following relationship to the screw Q (Fig. 4).

(e) Either the tip of P is fitted into the flute of Q, or the tip of P gets into a groove where the

fitting condition is induced by rotation.

(5) Rotate-to-loosen Skill

This skill can be executed if the screwdriver P has the following relationship to the screw Q (Fig. 5).

(f) The tip of P is fitted into the flute of Q.

4. Simplified Geometric Models in Skill-Based Planning

4.1. Simplified Geometric Models

In planning for manipulation, it is not necessary to use geometric models which express completely real objects. By constructing geometric models using only the data necessary for planning, it is possible to facilitate these processes. We call such robust models "simplified geometric models," which are defined as follows.

Simplified geometric models are geometric models composed of the necessary and minimum data for shape, position and orientation to perform skill-based planning.

4.2. Examples

We will consider the task of loosening a screw using a screwdriver [8]. Although there are many kinds of screws and screwdrivers as shown in Fig. 7, we will consider this task by unifying skill sequences in the following examples.

We assume that the task of loosening a screw using a screwdriver is composed of the following skills.

Skill1 : Move-to-touch skill
Skill2 : Rotate-to-bite skill
Skill3 : Rotate-to-loosen skill

Next, we will consider simplified geometric models of screws used in modeling in each skill.

4.2.1 Case of a cross-head screw

The task for a cross-head screw (Fig. 7 (a), (b)) is carried out as shown in Fig. 8. *Skill1* is achieved if the tip of the screwdriver gets into a groove in the neighborhood of the center of the screw-head. The simplified geometric model in *Skill1* is described by the shape of the groove (Table 1). The shape becomes a circle in order not to depend on the relative orientation between the screw and the screwdriver. The radius in a rare Reed and Prince type screw having little clearance (Fig. 7(b)) is smaller than that in a general Phillips type screw (Fig. 7(a)). The region of the initial state of the move-to-touch skill of *Skill1* is obtained as a set of trajectories projected inversely from this circle. Simplified geometric models in *Skill2*, *Skill3* are not needed since these can be executed continuously (Table 1).

- 221 -

Table 1 Simplified geometric models in the task of loosening a screw using a screwdriver

		Real Objects	Simplified Geometric Models		
			$Skill_1$	$Skill_2$	$Skill_3$
I	(a) Phillips	⊕	○	Not needed	Not needed
	(b) Reed and Prince	⊕	○	Not needed	Not needed
II	(c) Slotted	⑪	⑆	Not needed	·
III	(d) Robertson	□	□	Not needed	Not needed
	(e) Hex-drive	⬡	⬡	Not needed	Not needed
	(f) Clutch-drive	⊗	⊗	Not needed	Not needed
	(g) Torx	✲	✲	Not needed	Not needed
	(h) Two-Hole	⦂	⦂	Not needed	Not needed
	(i) Three-Wing	⨁	⨁	Not needed	Not needed
	(j) Four-Wing	⊕	⊹	Not needed	Not needed
IV	(k) Cross & slotted	⊕	(a) ○	Not needed	Not needed
			(c) ⑆	Not needed	·
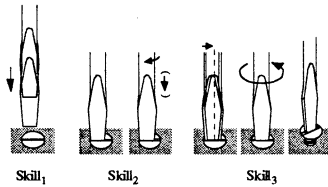
Fig. 9 Skill sequence of loosening with a slotted screwdriver

4.2.2 Case of a slotted screw

The task for a slotted screw (Fig. 7(c)) is carried out as shown in Fig. 9. The simplified geometric model in $Skill_1$ is described by a flute (Table 1), since it is necessary to check the relative position and orientation between the screw and the screwdriver. There is a little room in the direction of the flute. The simplified geometric model in $Skill_2$ is not needed (Table 1), since $Skill_2$ is meaningless. The axes of the rotations between the screw and the screwdriver have to correspond for the achievement of $Skill_3$. Therefore, the simplified geometric model in $Skill_3$ can be described by the center of the circle (Table 1).

4.2.3 Cases of Robertson, hex-drive, clutch-drive, torx, two-hole, three-wing and four-wing screws

The procedure of the task for these screws (Fig. 7 (d)-(j)) is the same as for a cross-head screw (Fig. 8). Simplified geometric models in $Skill_1$ take the shape
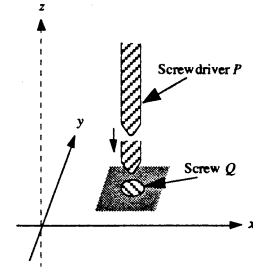
Fig. 10 Dispensable data in $Skill_1$
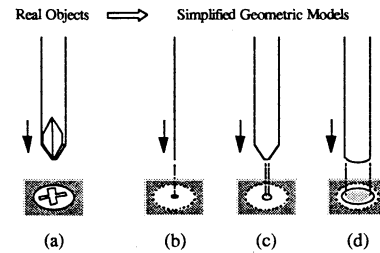
Fig. 11 Simplified Geometric Models of Parts and Tools

of grooves, since it is necessary to equalize the position and orientation between the screw and the screwdriver. Simplified geometric models in $Skill_2$, $Skill_3$ are not needed (Table 1).

4.2.4 Case of a cross and slotted screw

This screw has two kinds of simplified geometric models. Simplified geometric models correspond to the case of a cross-head screw when using a cross-head screwdriver, while simplified geometric models correspond to the case of a slotted screw when using a slotted screwdriver (Table 1).

Simplified geometric models of several kinds of screws have been derived as shown in Table 1. Next, we consider dispensable and indispensable data in the working environment of manipulator. Since $Skill_1$ is move-to-touch skill, the working space in $Skill_1$ can be simplified. According to (1-1) in section 3, it is not necessary to take into account the Z-axis of the coordinates as shown in Fig. 10. Simplification of the geometric models involves transforming the object models and varying the dimensions of space.

4.3. Simplification of tools

The simplification of parts has been taken into account in section 4.2. If the shapes of parts are changed in the step of deriving simplified geometric models, the geometric models of tools may have to be changed also. This means mostly simplifying the geometric models of tools. For example, geometric models of cross-head screws can be simplified in $Skill_1$, and then simplified geometric models of screwdrivers can be derived as shown in Fig. 11, too.

Table 2 Simplified geometric models of cross-head screws without Skill₂

		Real Objects	Simplified Geometric Models	
			Skill₁	Skill₃
I	(a) Phillips			Not needed
	(b) Reed and Prince			Not needed

5. Effectiveness of Skill on Models

5.1. Effectiveness in Shapes of Simplified Geometric Models by Using Skills

In this section, we consider the influence of changing the skill sequence on the simplification of geometric models. Therefore, we will review the change of shape of simplified geometric models by reduction of skill primitives in a task.

Let us consider the task of loosening a cross-head screw. In section 4.2.1, the task is carried out by a skill sequence composed of *Skill₁*: move-to-touch skill, *Skill₂*: rotate-to-bite skill and *Skill₃*: rotate-to-loosen skill. The data in the direction of transition is dispensable in *Skill₁*, and models of screws can be changed to simplified geometric models shown in Type I ((a) and (b)) of Table 1. Next, we consider the case of not using *Skill₂*, that is, when the task is composed of *Skill₁*: move-to-touch skill and *Skill₃*: rotate-to-loosen skill. The rotate-to-bite skill means fitting to grooves by (4) in section 2 and (4) in section 3. Therefore, the screw and the screwdriver must fit at the sub-goal of *Skill₁* in order to fit at the initial state of *Skill₃*. That is, the orientation of the screwdriver must correspond to that of the screw in *Skill₁*. The simplified geometric models in *Skill₁* are shown in Table 2. This means that simplification of models in the task without *Skill₂* almost correspond to the case of Type III ((d)-(j)) in Table 1. In addition, when not using *Skill₁*, that is, the manipulator is controlled by point-to-point, data in the direction of transition in *Skill₁* cannot be reduced.

Changing the skill sequence greatly influences the simplification of geometric models.

5.2. Correlation between Skill Sequence and Simplified Geometric Models

The simplified geometric models are classified into the following categories.

(1) The geometric model in *Skill_i* is simplified by *Skill_j* ($j=i$).

(2) The geometric model in *Skill_i* is simplified by *Skill_j* ($j>i$).

(3) (1) and (2) occur simultaneously.

For example, in the case of a cross-head screw in section 4.2.1, the geometric model in *Skill₁* is simplified by *Skill₂* as shown in Table 1, and data in the direction of transition in *Skill₁* is dispensable by *Skill₁* as shown in Fig. 10. Therefore, the case is classified into category (3).

6. Conclusion

We have proposed simplified geometric models for planning skill-based manipulation. It is possible to carry out planning efficiently, since the simplified geometric models are composed of the only indispensable data. Furthermore, we showed the effectiveness of the simplified geometric models by manipulation skill, using specific examples of tasks both with skill and without skill.

In the future, we will further study simplified geometric models for various skill-based tasks, and attempt to apply our method to real systems.

References

[1] Hasegawa T, Suehiro T, Ogasawara T, Matsui T, Kitagaki K, Takase K (1990), An integrated tele-robotics system with a geometric environment model and manipulation skills. In: Gotoh T, Kato I, Paul RP (eds) Proceedings of the IEEE International Workshop on Intelligent Robots and Systems (IROS '90), Tsuchiura, Japan, July 3-6, 1990, pp 335-341

[2] Suehiro T, Takase K (1990), Skill based manipulation system (in Japanese). J Robotics Soc Jpn 8:551-562

[3] Hasegawa T, Suehiro T, Takase K (1991), A model-based manipulation system with skill-based execution in unstructured environment. In: Dario P (ed) Proceedings of the 5th International Conference on Advanced Robotics, Pisa, Italy pp 970-975

[4] Hasegawa T, Suehiro T, Takase K (1992), A model-based manipulation system with skill-based execution. IEEE Trans Robotics Autom 8(5):535-544

[5] Nakamura A, Ogasawara T, Suehiro T, Tsukune H (1996), Skill-based backprojection for fine motion planning. In: Asada M, Arai T, Kak A, Sandini G (eds) Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96), Osaka, Japan, Nov. 4-8, 1996, pp 526-533

[6] Nakamura A, Ogasawara T, Suehiro T, Tsukune H (1998), Fine motion strategy in three-dimensional space using skill-based backprojection. Artificial Life and Robotics 2(3):134-137

[7] Nakamura A, Ogasawara T, Suehiro T, Tsukune H (2000), Simplification of visual sensing in skill-based manipulation. In: Sugisaka M, Tanaka H (eds) Proceedings of the International Symposium on Artificial Life and Robotics (AROB 5), Oita, Japan, Jan. 26-28, 2000, vol 2, pp 466-469

[8] Nakamura A, Ogasawara T, Kitagaki K, Suehiro T (2001), Simplified geometric models in planning of skill-based manipulation. In: Sugisaka M, Tanaka H (eds) Proceedings of the International Symposium on Artificial Life and Robotics (AROB 6), Tokyo, Japan, Jan. 15-17, 2001, vol 2, pp 466-469

The Influence of Counterbalances on the Dissipated Energy of a Vertically Articulated Robot Manipulator

Jun AMANO
Dep. of Mechanical Engineering
Matsue National College of Technology
14-4 Nishi-ikuma, Matsue, Shimane 690-8518, Japan
E-mail: amano@matsue-ct.jp

Teruyuki IZUMI
Dep. of Electronic and Control Systems Engineering
Shimane University
1060 Nishi-Kawatsu, Matsue, Shimane 690-8504, Japan
E-mail: izumi@riko.shimane-u.ac.jp

Abstract

This paper discusses the influence of counterbalances in order to save the dissipated energy of a vertically articulated robot manipulator. It is seen from the analysis that the dynamical equation can be linearized by adjusting the center of mass of each link. As the result, the optimal path and the minimum energy are expressed by a function of position of the counterbalances, and the configuration of the counterbalances can be designed so that the dissipated energy is minimized. The minimum dissipated energy is compared with a formerly proposed optimal path of the vertically articulated manipulator without the counterbalances. The comparison by simulations shows that the counterbalances makes the dissipated energy increase due to the increment of the inertia in the links.

Keywords: Dissipated Energy, Manipulator, Counterbalance, Inertia, Gravity

1 Introduction

There are many papers on a robotic manipulator. The major of the papers is concerned with upgrading the function and workability of the mechanical system[1]. However the manipulator must also be taken a side view of energy conversion. It is important to save the dissipated energy even of a manipulator for improving the environment of the earth that is warmed up by CO_2 gas emitted from thermal power plants.

The authors have proposed an optimal path and operating time which minimizes the dissipated energy in PTP motions of a vertically articulated manipulators [2]. A globally optimal path is roughly estimated so that the heavier link is accelerated toward gravitational direction and decelerated toward anti-gravitation.

This paper discusses the influence of counterbalances on the dissipated energy of a vertically articulated manipulator. Lagrangian is estimated as a function of position of counterbalances which are mounted on each link. It is seen from the induction of Lagrange's equation that the dynamical equation can be linearized by adjusting the center of mass of each link. Namely, not only gravity force but also Coriolis's and centrifugal forces can be vanished by taking the counterbalances into the consideration. And the inertia

term is turned to a constant matrix. As the result, the optimal path, which minimizes the dissipated energy, is obtained analytically by solving a linear two-point boundary-value problem. As the dissipated energy can be expressed by a function of positions of the counterbalances, the configuration of the counterbalances is designed so that the dissipated energy can be minimized. The minimum dissipated energy of the manipulator with the most suitable counterbalances is compared with the dissipated energy of a formerly proposed optimal path of the vertically articulated manipulator without the counterbalances. The comparison by simulations shows that the counterbalances increase the dissipated energy due to the increment of the inertia in the manipulator.

2 Equation of Motion of the Manipulator

Let the angular vector of an n-links manipulator be denoted by $\theta(t) = [\theta_1(t), \theta_2(t), \cdots, \theta_n(t)]^T$, and its angular velocity vector by $\dot{\theta}(t) = [\dot{\theta}_1(t), \dot{\theta}_2(t), \cdots, \dot{\theta}_n(t)]^T$. If the frictional forces are neglected, the equation of motion of the manipulator is given by

$$H(\theta(t))\ddot{\theta}(t) + C(\theta(t), \dot{\theta}(t)) + G(\theta(t)) = Ki(t) \quad (1)$$

where $H(\theta(t))$ is the inertia matrix, $C(\theta(t), \dot{\theta}(t))$ is the centrifugal and Coriolis's force vector, $G(\theta(t))$ is the gravitational force vector, $K = diag(K_1, K_2, \cdots, K_n)$ is the torque constant matrix and $i(t) = [i_1(t), i_2(t), \cdots, i_n(t)]^T$ is the armature current vector of the DC motors driving the links. Figure 1 shows a vertically articulated manipulator moving in a plane. It is
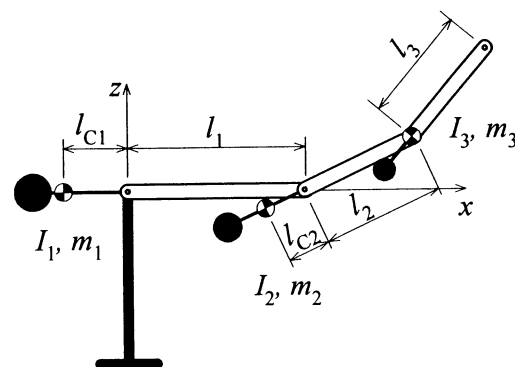
Fig. 1 Configuration of three-link manipulator with counterbalances.

interesting the counterbalances of each link to be adjusted so that the center of the mass of each link included all upper links is equal to the center of the rotation. In this configuration, the kinetic energy is represented by the function of only $\theta(t)$, the potential energy is also vanished. As seen from Lagrange's Equation, three terms in the left hand side of Eq. (1) can be written as

$$G(\theta(t)) = 0 \tag{2}$$

$$C(\theta(t), \dot{\theta}(t)) = 0 \tag{3}$$

$$H(\theta(t)) = H \tag{4}$$

Therefore, the Equation of motion of the manipulator is reduced to

$$H\ddot{\theta}(t) = Ki(t) \tag{5}$$

3 Dissipated Energy

Since the friction is neglected, the dissipated energy of the manipulator becomes only a loss by the Joule heat as follow:

$$J = \int_0^{t_f} i(t)^T Ri(t)dt \tag{6}$$

where $R = diag(R_1, R_2, \cdots R_n)$ is an armature resistance matrix of DC motors. The optimal current, which makes J of Eq. (6) minimum under the condition of Eq. (5), can be analyzed easily due to the linearity of Eq. (5) and be expressed as

$$i(t) = \frac{6(t_f - 2t)}{t_f^3} K^{-1} H(\theta_f - \theta_0) \tag{7}$$

where θ_0 and θ_f are angle vectors at the initial time t = 0 and the final time $t = t_f$, respectively. By substituting Eq. (7) into Eq. (6), the minimum value of the dissipated energy can be expressed by

$$J = \frac{12}{t_f^3}(\theta_f - \theta_0)^T H^T K^{-1} R K^{-1} H(\theta_f - \theta_0) \tag{8}$$

4 Optimal Design of Counterbalances

It is seen from Eq. (8) that the dissipated energy J of a manipulator is influenced remarkably by the inertia matrix H. It is, therefore, important to design the counterbalances so that the dissipated energy can be made minimum by changing the balancer position.

In a vertically articulated manipulator as shown in Fig. 2, the counterbalances are installed in the opposite side to the fulcrum. The length of i-th link is denoted by l_i ($i = 1, \cdots, n$). The moment of inertia about the center of gravity, mass and the distance to the center of gravity in the link side and each balance side are denoted by I_i, I_{ci}, m_i, m_{ci}, l_{gi}, l_{ci} ($i = 1, \cdots, n$).
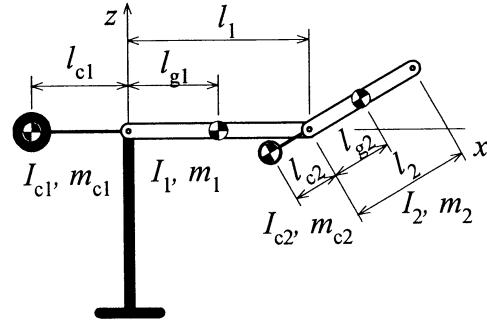
Fig. 2   Configuration of two-link manipulator with counterbalances.

Furthermore, the new parameter about the position of the counterbalance

$$C_{ki} = l_{ci} / l_{gi}, \quad (i = 1, \cdots, n) \tag{9}$$

are introduced. Then the inertia matrix H is given as a function of the parameter C_{ki}. Therefore optimal parameter C_{ki} that make J minimum must satisfy the following simultaneous algebraic equation system.

$$\frac{\partial J}{\partial C_{ki}} = 0, \quad (i = 1, \ldots, n) \tag{10}$$

4.1 One-Link Manipulator

In the case of one-link manipulator, Eq. (8) is simplified to

$$J = \frac{12}{t_f^3} \frac{RH^2}{K^2}(\theta_f - \theta_0)^2 \tag{11}$$

where

$$H = I_1 + I_{c1} + m_1 l_{g1}^2 + m_{c1} l_{c1}^2$$

The counterbalance is assumed to be made of a steel ball which is mounted on the beam without mass, and to have the following relation.

$$m_{c1} l_{c1} = m_1 l_{g1} \tag{12}$$

This means that the balance of links is kept strictly.

When the density of the steel ball is ρ, the moment of inertia of the steel ball is related by its mass as

$$I_{c1} = \frac{2}{5}\left(\frac{3}{4\pi\rho}\right)^{\frac{2}{3}} m_{c1}^{\frac{5}{3}} \tag{13}$$

Substituting these equation's into Eq. (10), the optimal parameter C_{k1} of the counterbalance is obtained as

$$C_{k1} = \left(\frac{m_1}{\sqrt{6}\,\pi\,\rho\,l_{g1}^3}\right)^{\frac{1}{4}} \tag{14}$$

If the parameters of the link are specified as Table 1 and the density of steel is $\rho = 7.85$, the optimal

parameter is $C_{k1} = 0.59$. This means that the optimal position of the counterbalance is about half of the link length.

Figure 3 shows the relation between the dissipated energy J and the position parameter C_{k1}, when $t_f = 0.5$ [sec], $\theta_{10} = -\pi/2$ and $\theta_{1f} = -3\pi/2$. It is seen from this figure that the minimum energy is taken at the position parameter $C_{k1} = 0.59$, which is the same as the result of Eq. (14).

Table 1   Parameters of one-link manipulator.

Moment of inertia	I_1, [kgm^2]	0.43
Mass	m_1, [kg]	14.25
Link length	l_1, [m]	0.25
Center of gravity	l_{g1}, [m]	$l_1/2$
Resistance	R [Ω]	1
Torque constant	K [Nm/A]	1

4. 2   Two-Link Manipulator

As same as the one-link manipulator, steel balls are mounted on each link and have the effect of balance strictly by relations

$$m_{c1}l_{c1} = m_1 l_{g1} + (m_2 + m_{c2})l_1$$

$$m_{c2}l_{c2} = m_2 l_{g2} \qquad (15)$$

Each element of the inertia matrix H in a two-link manipulator with counterbalances is

$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \qquad (16)$$

where

$$h_{11} = I_1 + I_{c1} + m_1 l_{g1}^2 + m_{c1}l_{c1}^2 + (m_2 + m_{c2})l_1^2$$
$$+ I_2 + I_{c2} + m_2 l_{g2}^2 + m_{c2}l_{c2}^2$$

$$h_{12} = h_{21} = h_{22} = I_2 + I_{c2} + m_2 l_{g2}^2 + m_{c2}l_{c2}^2$$

In this case, even if the specifications and the performance parameters of the manipulator are designated as Table 2 and $\rho = 7.85$, the optimal positions of the counterbalances are not so decided uniquely as a one-link manipulator. It also needs the rotation angle of each link, because the dissipated energy of each link affects mutually. Therefore the initial angles and the final angles are put to be $\theta_{10} = -\pi/2$, $\theta_{1f} = -3\pi/2$, $\theta_{20} = 0$, $\theta_{2f} = 2\pi$, respectively. In this case, the optimal parameters of the counterbalances is obtained from Eq. (10) to be $C_{k1} = 0.74$ and $C_{k2} = 11.7$.

The two-link manipulator with an above mentioned parameters is simulated at the operation time $t_f = 0.5$ [sec]. Figure 4 shows the result about the relation between C_{k2} and the dissipated energy in the condition at $C_{k1} = 0.74$. In this figure, J_1, J_2 and J are energy dissipated at the first joint, the second joint and these

sum, respectively. It is seen from this figure that the total energy is minimized at $C_{k2} = 11.7$. Therefore the optimal parameter obtained by Eq. (10) theoretically is valid. However, the value $C_{k2} = 11.7$ is too large for a practical use.

Table 2   Parameters of two-link manipulator.

Moment of inertia	I_1, I_2 [kgm^2]	0.43, 0.244
Mass	m_1, m_2 [kg]	14.25, 10
Link length	l_1, l_2 [m]	0.25, 0.16
Center of gravity	l_{g1}, l_{g2} [m]	$l_1/2$, $l_2/2$
Resistance	R [Ω]	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Torque constant	K [Nm/A]	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
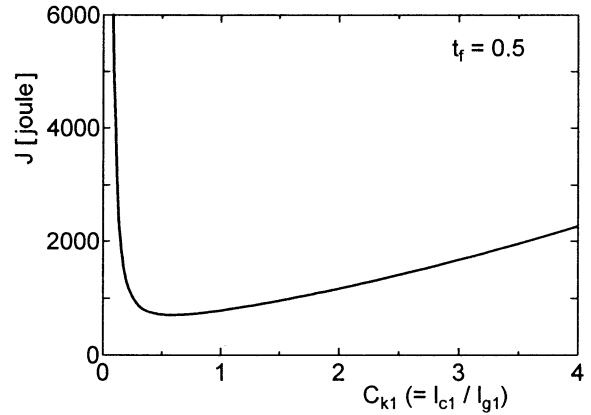
Fig. 3   Energy $J(C_{k1})$ of one-link manipulator.
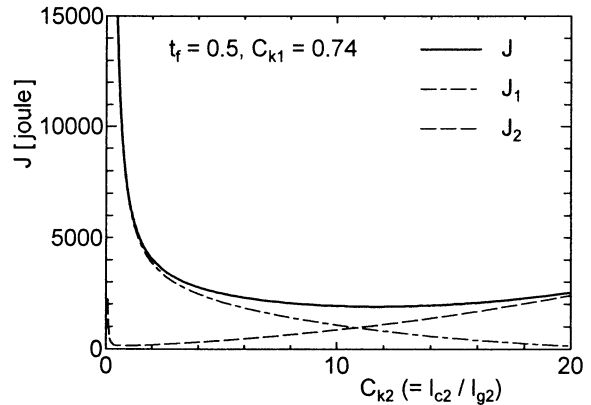
Fig. 4   Energy $J(C_{k2})$ of two-link manipulator.

5   Influence of Counterbalances for Dissipated Energy

In order to make the influence of the optimal counterbalances clear, the dissipated energy are compared on the following three conditions: a manipulator with optimal counterbalances, without counterbalances and without both counterbalances and gravity.

5. 1   One-Link Manipulator

For the vertically articulated one-link manipulator without the counterbalance, each term of Eq. (1) is given by

$$H = I_1 + m_1 l_{g1}{}^2, \quad C = 0, \quad G = -m_1 g l_{g1} \sin\theta_1 \qquad (17)$$

Equation (1) becomes nonlinear due to gravity force. Therefore the optimal current corresponding to Eq. (7) must be solved a nonlinear two-point boundary-value problem numerically [2]. The simulation was performed by using the same parameter in section 4.1. The trajectory of the link in the simulation are shown in Fig. 5. A dot-dash line in Fig. 6 illustrates the relation between the operation time t_f and the dissipated energy J. This line shows that there exists an optimal operating time which minimizes the dissipated energy. A solid and dotted lines are the simulation results in the same manipulator with counterbalance and without both counterbalance and gravity. It is seen from the comparison among three curves that the dissipated energy can be decreased extremely by using a manipulator without counterbalance in the condition of the optimal operating time.

5.2 Two-Link Manipulator

For the vertically articulated two-link manipulator without the counterbalances, each term of Eq. (1) is

$$H = \begin{bmatrix} I_1 + I_2 + m_1 l_{g1}{}^2 + m_2(l_1{}^2 & I_2 + m_2(l_{g2}{}^2 + l_1 l_{g2}\cos\theta_2) \\ + l_{g2}{}^2 + 2l_1 l_{g2}\cos\theta_2) & \\ I_2 + m_2(l_{g2}{}^2 + l_1 l_{g2}\cos\theta_2) & I_2 + m_2 l_{g2} 2 \end{bmatrix}$$

$$C = m_2 l_1 l_{g2}\sin\theta_2 \begin{bmatrix} -(2\dot\theta_1\dot\theta_2 + \dot\theta_2{}^2) \\ \dot\theta_1{}^2 \end{bmatrix}$$

$$G = -g \begin{bmatrix} m_1 l_{g1}\sin\theta_1 + m_2\{l_1\sin\theta_1 + l_{g2}\sin(\theta_1 + \theta_2)\} \\ m_2 l_{g2}\sin(\theta_1 + \theta_2) \end{bmatrix} \qquad (18)$$

The simulation was performed by using the same parameter in section 4.2. The optimal trajectory in the simulation are shown in Fig. 7. A dot-dash line in Fig. 8 illustrates the relation between the operation time t_f and the dissipated energy J. It is also seen from Fig. 8 that the curves of the dissipated energy are similar to the case of the one-link manipulator. In a long path, the counterbalances increase the dissipated energy even if the counterbalances are designed optimally.

6 Conclusions

The relation of counterbalances and dissipated energy was considered in order to design a manipulator saving energy. The optimal configuration of the counterbalances which makes the dissipated energy minimum can be discovered analytically. However, considering the motion of a manipulator, the optimal configuration cannot always be practical. In addition, a comparison by simulations shows that the counterbalances increase the dissipated energy due to the increment of the inertia in the links in a long path.
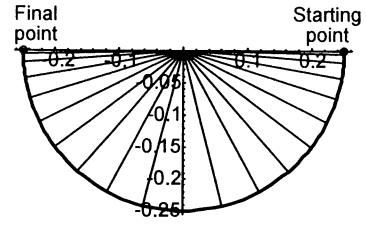
Fig. 5 Trajectory of one-link manipulator.
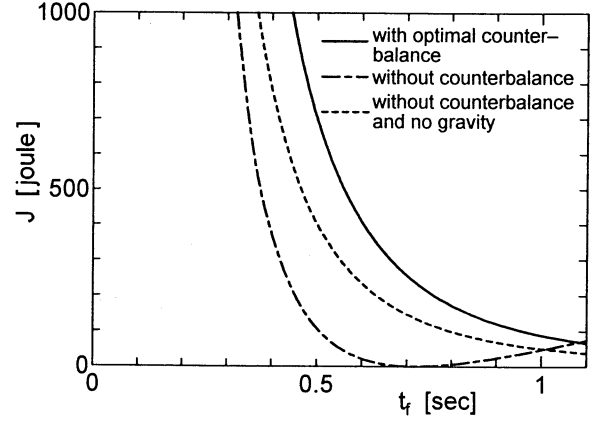
Fig. 6 Energy $J(t_f)$ of one-link manipulator.
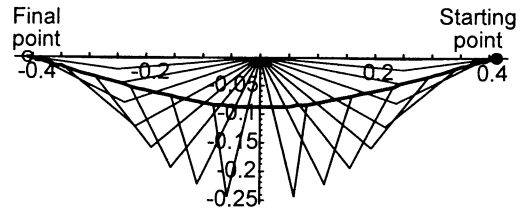
Fig. 7 Trajectory of two-link manipulator.
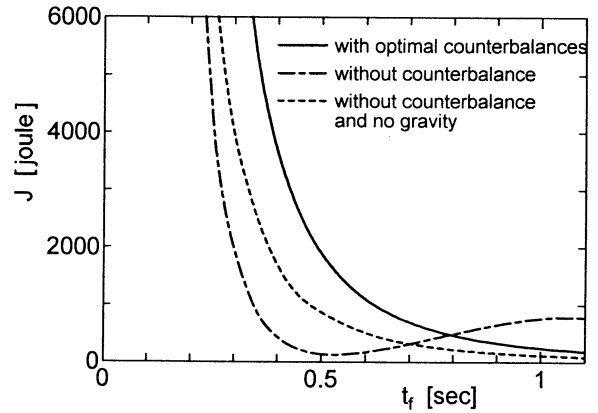
Fig. 8 Energy $J(t_f)$ of two-link manipulator.

References

[1] Paul, R. P., "Robot Manipulator: Mathematics, Programming, and Control", MIT Press (1981).
[2] Izumi, T., Boyagoga, P., Satoh, T., et al., "Optimal Time for Minimizing the Dissipated Energy of a Manipulator in PTP Motion under Gravity", Journal of Robotics Society of Japan, Vol. 15, No. 7, 1082-1088 (1997).

Robotic Cane for Blind Travelers Based on Interactive Technology Paradigm

Mr.   Shim   Inbo
Pusan National University   Korea
nainbow@hotmail.com

Prof.   Yoon   Joongsun
Pusan National University   Korea
jsyoon@pnu.edu

A human-friendly interactive system, based on the harmonious symbiotic coexistence of humans and robots, is explored. Based on this interactive human-robot teaming paradigm, a robotic cane is designed to help blind or visually impaired travelers to navigate safely and quickly among obstacles and other hazards faced by blind pedestrians.

Robotic cane based on mobile robot platforms are introduced as a guide for the blind by utilizing mobile robot obstacle avoidance technologies. The proposed robotic cane consists of a long handle and a sensor head unit that is attached at the distal end of the handle. Photo sensors mounted on the sensor head detect obstacles and steer the device around it. The user feels the steering command as a very noticeable.

The proposed robotic cane makes independent decisions concerning the path it takes. However, the user and the cane may wish to go in different directions. The normal operating mode of the constructed robotic cane must be overridden to allow the blind person to be in control when the need arises.

Unique features of our robotic cane include the active sensor unit, the antennas for contact sensing, and the powered wheels. This active sensing unit, which scans the area ahead by a servo motor, can detect obstacles efficiently. It can reduce the missing areas caused by the narrow coverage due to fixed sensor arrangement. Additional antennas for contact sensing could effectively compensate mid-range and/or long-distance sensing. Various schemes to use antennas for contact sensing are being investigated. Antennas are especially suitable for blind user's superb tactile information processing capabilities. Our robotic cane is driven and steered by two powered motors. Thus, it can guide the user autonomously with sufficient powers.

Also, we apply newly developing intelligent control schemes for autonomous mobile robots to cope with the dynamic situations. To properly respond under dynamically changing environments, reactive planning systems implementing new principles, called behavior-based robotics or emergent computation, have been proposed and confirmed their usefulness. As another alternative, biological information processing systems may provide many feasible ideas to these problems. Immune system, among these systems, plays important roles to maintain its own system against dynamically changing environments.

We propose a new decentralized behavior arbitration mechanism for the robotic cane based on the immune network architecture. There are some similarities between an autonomous mobile robot with a behavior selection mechanism and immune network architecture. That is, current situations detected by installed sensor work as multiple antigens, and a prepared competence module is regarded as an antibodies. Artificial immune system equipped with the autonomous mobile robot selects a competence module(antibody) suitable for the detected current situation(antigens) in a bottom-up manner.

In addition, we shall discuss the application method by interactive technology to construct an appropriate artificial immune network for complex system.

Resolved Acceleration Control of an Underwater Robot with Vertical Planar 2-Link Manipulator

Shingo Yamada   Shinichi Sagara

Department of Control Engineering

Kyushu Institute of Technology

Tobata, Kitakyushu 804-8550, Japan

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 (RAC) with the manipulator's end-tip position and velocity feedback and a Resolved Motion Rate Control (RMRC) with the end-tip position feedback have good control performances in spite of using an inaccurate hydrodynamic model. In this paper, for a floating underwater robot with 2 dimensional and vertical planar 2 link manipulator we construct the robot base and end-tip control system using the RAC and verify the effectiveness of the control system by experiments.

1 Introduction

This paper is concerned with a Resolved Acceleration Control (RAC) of an underwater robot with vertical planar 2-link manipulator. Since underwater robots are necessary for ocean exploration, many studies have been done about Underwater Robotic Vehicle (URV) including Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) [1]. Furthermore, recent years some studies have been done about dynamics and control of Underwater Vehicle-Manipulator System (UVMS) [2, 3, 4, 5, 6, 7]. However, there are few experimental results using robotic arm.

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 and a Resolved Motion Rate Control (RMRC) with the end-tip position feedback have good control performances in spite of using an inaccurate hydrodynamic model [8, 9].

In this paper, for a floating underwater robot with 2 dimensional and vertical planar 2 link manipulator we construct the robot base and end-tip control system using the RAC and validate the effectiveness of the
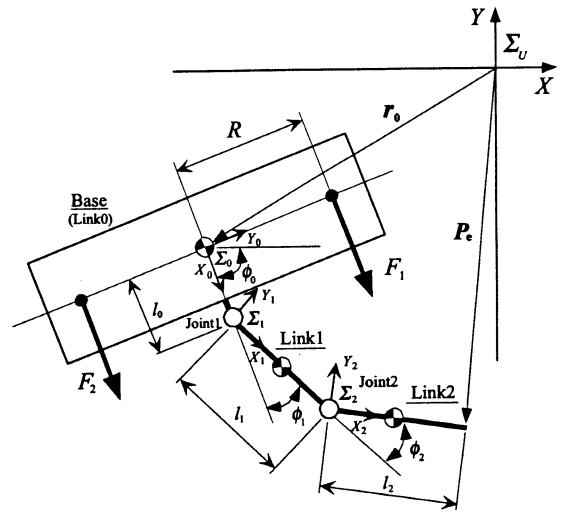
Fig. 1: 2-link underwater robot manipulator model

control system by experiments. Experimental results demonstrated the effectiveness of the control method.

2 Kinematics and Dynamics

2.1 Modeling

The underwater robot model used in this paper is shown in Fig. 1. It has a robot base and 2-DOF manipulator which can move in a vertical plane. Two thrusters are mounted on the base to provide propulsion for position and attitude control of the base.

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)

$^U R_i$: coordinate transformation matrix from Σ_i to Σ_U

r_0 : position vector of origin of Σ_0 with respect to Σ_U

l_i : length of link i

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

P_e : position vector of end-tip with respect to Σ_U

P_0 : position vector of center of gravity of base with respect to Σ_U

X_e : position and attitude vector of end-tip with respect to Σ_U ($= [P_e^T, \ \theta_2]^T$)

X_0 : position and attitude vector of center of gravity of base with respect to Σ_U ($= [P_0^T, \ \theta_0]^T$)

ϕ : joint angle vector ($= [\phi_1, \ \phi_2]^T$)

m_i : mass of link i (link 0 means the robot base)

M_{a_i} : added mass tensor of link i

I_i : inertia tensor of link i

I_{a_i} : added inertia tensor of link i

$\hat{x}_i$: position vector from joint i to joint ($i+1$) with respect to Σ_i

$\hat{a}_i$: position vector from origin of Σ_i to center of gravity of link i with respect to Σ_i

$\hat{b}_i$: position vector from origin of Σ_i to center of buoyancy of link i with respect to Σ_i

E : unit matrix

g : grabitational acceleration vector

2.2 Kinematics

First, a time derivative of the end-tip position vector P_e is

$$\dot{P}_e = A\dot{X}_0 + B\dot{\phi} \qquad (1)$$

where $A \in R^{2\times3}$ and $B \in R^{2\times2}$ are matrices including attitude angle of base and joint angles.

Next, let Q and L be a linear and an angular momentum of the robot including hydrodynamic added mass M_{a_i} and added inertia I_{a_i}, then

$$Q = [Q_x, \ Q_y, \ 0]^T = \sum_{i=0}^{2} {}^UR_i(m_iE + M_{a_i})\hat{\alpha}_i, \qquad (2)$$

$$L = [0, \ 0, \ L]^T$$
$$= \sum_{i=0}^{2} [(I_i + I_{a_i})\omega_i + \hat{x}_i \times \{{}^UR_i(m_iE + M_{a_i})\hat{\alpha}_i\}] \quad (3)$$

where

$$\hat{\alpha}_i = v_i + \omega_i \times \hat{a}_i, \quad \omega_i = [0, \ 0, \ \dot{\phi}_i]^T \ (i = 0, 1, 2).$$

From Eqs. (1), (2) and (3) the following equation can be obtained:

$$F = [Q_x, \ Q_y, \ L]^T = C\dot{X}_0 + D\dot{\phi} \qquad (4)$$

where $C \in R^{3\times3}$ and $D \in R^{3\times2}$ are matrices including the added mass M_{a_i} and the added inertia I_{a_i}. Here, we assume that the added mass and added inertia are constant.

2.3 Hydrodynamic forces and moments

Generally, the drag force and moment of the joint i can be represented as follows [2, 3]:

$$f_{d_i} = \frac{\rho}{2}C_{D_i}D_i \int_0^{l_i} \|w_i\|w_i d\hat{x}_i, \qquad (5)$$

$$t_{d_i} = \frac{\rho}{2}C_{D_i}D_i \int_0^{l_i} \hat{x}_i \|w_i\|w_i d\hat{x}_i \qquad (6)$$

where $w_i = v_i + \omega_i \times \hat{x}_i$, and ρ is the fluid density, C_{D_i} is the drag coefficient, D_i is the width of link i.

The gravitational and buoyant forces acting link i are discribed as follows:

$$f_{g_i} = ({}^UR_i)^T(\rho V_i - m_i)g, \qquad (7)$$

$$t_{g_i} = ({}^UR_i)^T(\hat{b}_i \times \rho V_i g - \hat{a}_i \times m_i g). \qquad (8)$$

2.4 Equation of motion

Considering the hydrodynamic forces and moments described above and using Newton-Euler formulation, the following equation of motion can be obtained:

$$(M + M_a)\ddot{q} + b(q, \dot{q}) + F_{DG} = [f_x, \ f_y, \ \tau_0, \ \tau_1, \ \tau_2]^T \qquad (9)$$

where $q = [X_0^T, \ \phi^T]^T$, and M is the inertia matrix, $b(q, \dot{q})$ is the vector of Coliolis and centrifugal forces, M_a is the matrix consisting of the added mass and inertia, F_{DG} is the external force vector consisting of the drag, grabitational and buoyant forces and moments. f_x, f_y and τ_0 are forces and torque provided by two thrusters and τ_i is the joint torque.

3 Resolved Acceleration Control

Differentiating Eqs. (1) and (4) with respect to time, the following equation can be obtained:

$$H\ddot{X}_0 + D\ddot{\phi} = \ddot{X}_0 + \dot{F} - (\dot{C}\dot{X}_0 + \dot{D}\dot{\phi}), \qquad (10)$$

$$A\ddot{X}_0 + B\ddot{\phi} = \ddot{P}_e - (\dot{A}\dot{X}_0 + \dot{B}\dot{\phi}) \qquad (11)$$

where $H = C + E$, $\dot{F}$ is the external force including hydrodynamic force and thrust of the thruster which act on the robot.

From Eqs. (10) and (11), the following equation can be obtained.

$$\begin{bmatrix} \ddot{X}_0 \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} H & D \\ A & B \end{bmatrix}^{-1} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \qquad (12)$$

where

$$z_1 = \ddot{X}_0 + \dot{F} - \left(\dot{C}\dot{X}_0 + \dot{D}\dot{\phi}\right),$$

$$z_2 = \ddot{P}_e - \left(\dot{A}\dot{X}_0 + \dot{B}\dot{\phi}\right).$$

If the values of the external forces are exact, using Eq. (12) and the desired values of the base position and attitude $\boldsymbol{X}_d$ and the end-tip position $\boldsymbol{P}_d$, the desired acceleration values of the base $\ddot{\boldsymbol{X}}_d$ and joint $\ddot{\phi}_d$ can be obtained. However, the complete hydrodynamic model cannot be obtained. Furthermore, we assume that the added mass and inertia are constant. Therefore, instead of $\ddot{\boldsymbol{X}}_d$ and $\ddot{\boldsymbol{P}}_d$ the following modified desired values of the base $\ddot{\boldsymbol{X}}_d^*$ and the end-tip $\ddot{\boldsymbol{P}}_d^*$ are utilized:

$$\ddot{\boldsymbol{X}}_d^* = \ddot{\boldsymbol{X}}_d + \boldsymbol{K}_{v0}(\dot{\boldsymbol{X}}_d - \dot{\boldsymbol{X}}_0) + \boldsymbol{K}_{p0}(\boldsymbol{X}_d - \boldsymbol{X}_0), (13)$$

$$\ddot{\boldsymbol{P}}_d^* = \ddot{\boldsymbol{P}}_d + \boldsymbol{K}_{ve}(\dot{\boldsymbol{P}}_d - \dot{\boldsymbol{P}}_e) + \boldsymbol{K}_{pe}(\boldsymbol{P}_d - \boldsymbol{P}_e) \quad (14)$$

where $\boldsymbol{X}_d$ and $\boldsymbol{P}_d$ are the desired values of the base and end-tip, and $\boldsymbol{K}_{v*}$ and $\boldsymbol{K}_{p*}$ ($* = 0$, e) are velocity and position feedback gain matrices, respectively.

Substituting $\ddot{\boldsymbol{X}}_d^*$, $\ddot{\boldsymbol{P}}_d^*$ for $\ddot{\boldsymbol{X}}_0, \ddot{\boldsymbol{P}}_e$ on the right side of Eq. (12), the acceleration $[\ddot{\boldsymbol{X}}_0 \ \ddot{\phi}]^T$ that should be provided with the robot are calcurated. The robot base is actively moved by two thrusters. Therefore, using Eq. (9) and the following relation, the thruster forces F_1 and F_2 are obtained.

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} \sin\phi_0 & \sin\phi_0 \\ -R & R \end{bmatrix}^{-1} \begin{bmatrix} f_y \\ \tau_0 \end{bmatrix}. \quad (15)$$

Here, f_x is ignored because of arrangement of two thrusters. On the other hand, since the manipulator is drived by velocity control type actuators, integrating joint angular acceleration the joint angular velocities are obtained.

4 Experiment

4.1 Experimental system

Fig. 2 shows a configuration of experimental system. A robot has a 2DOF manipulator that joints are actively rotated by a velocity control type servo actuator consisting of DC servo motor, harmonic drive gear and incremental type encoder. Two thrusters are attached to vertical direction on the robot base. Physical parameters of the underwater floating robot is shown in Table 1 and the distance from center of gravity of the base to the thruster is $R = 0.3$m.

Measurement and control system are consist of CCD camera, Video Tracker and 32bit personal computer. Two light emitting diodes (LED) is attached to the base, and its motion is monitored by a CCD camera. 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
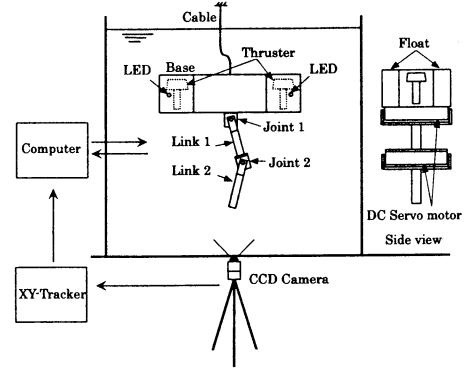
Fig. 2: Configuration of underwater robot system

Table 1: Physical parameters of the underwater robot

	Base	Link 1	Link 2
Mass [kg]	26.04	4.25	1.23
Moment of inertia [kg m^2]	1.33	0.19	0.012
Link length (x axis) [m]	0.2	0.25	0.25
Link length (y axis) [m]	0.81	0.04	0.04
Link width [m]	0.42	0.12	0.12
Added mass(x) [kg]	72.7	1.31	0.1
Added mass(y) [kg]	6.28	3.57	2.83
Added moment of inertia [kg m^2]	1.05	0.11	0.06

positions and attitude angles of the robot base and manipulator are computed in the personal computer that is also used in controller.

4.2 Thruster

Two 40W thrusters which provide propulsion for controlling the position and attitude angle of the base are utilized. Static characteristics of the thruster is shown in Fig. 3. From this figure, the following relation is obtained:

$$F_{CW} = 1.341v^2 - 1.363v - 0.026, \quad (16)$$
$$F_{CCW} = 0.763v^2 - 8.345v - 0.019. \quad (17)$$

where v is the input voltage to the power amplifier of the thruster.
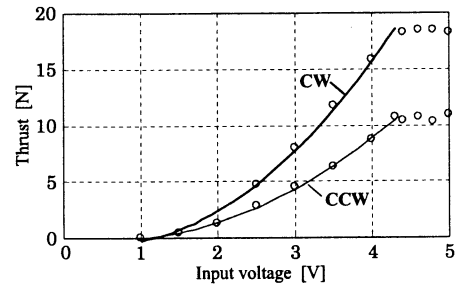
Fig. 3: Thruster characteristics
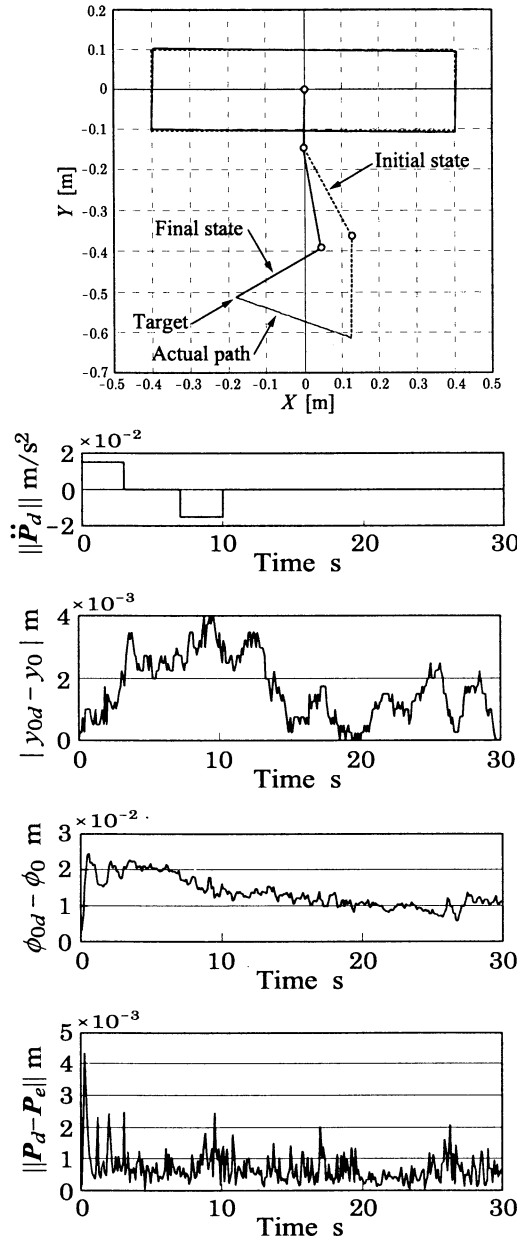
Fig. 4: Experimental result

4.3 Experimental result

The initial relative angles of the robot are $\phi_0 = -\pi/2$rad, $\phi_1 = \pi/6$rad and $\phi_2 = -\pi/6$rad. The desired end-tip position P_d was set up along a straight path from the initial end-tip position $P(0)$ to the target calculated from trapezoidal velocity patern. At the same time, the desired base position and attitude are set up to keep initial state.

The typical experimental result is shown in Fig. 4. From the figure it can be seen that the end-tip and

base follow the reference trajectories in spite of the influence of the hydrodynamic forces and the tracking errors are very small. Experimental result shows that the control performance can be improved by using the resolved acceleration control method.

5 Conclusion

In this paper, for the underwater robot with vertical planar 2-link manipulator, resolved acceleration control method was constructed. The experimental result demonstrated the effectiveness of the control method.

References

[1] J. Yuh ed., *Underwater Robotic Vehicles:Design and Control*, TSI Press, 1995.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] N. Sarkar and T. K. Podder, "Coordinated Motion Planning and Control of Autonomous Underwater Vehicle-Manipulator Systems Subject to Drag Optimization", *IEEE Trans. on Oceanic Engineering*, Vol. 26, No. 2, pp. 228-239, 2001.

[8] S. Sagara, T. Tanikawa, M. Tamura and R. Katoh, "Experiments of a Floating Underwater Robot with 2 Link Manipulator", *Proc. of AROB 5th*, pp. 367-370, 2000.

[9] S. Sagara, T. Danjoh, M. Tamura and R. Katoh, "Resolved Motion Rate Control of a Free-Floating Underwater Robot with Horizontal Planar 2-Link Manipulator", *Proc. of AROB 6th*, pp. 113-116, 2001.

Improvement of the precision of the gradient method and object tracking using optical flow

Masanori SUGISAKA and Shinobu SATO

Department of Electric and Electronic Engineering, Faculty of Engineering, Oita University

700 Dannoharu Oita 870-1192, JAPAN

E-mail: msugi@cc.oita-u.ac.jp ssato@cc.oita-u.jp

Abstract

In this research, we improve the spatial local optimization method to obtain high precision of optical flow in the part where the object movement changes largely and consider the method to trace loci of moving objects. The spatial local optimization method is a kind of local optimization method. In the spatial local optimization method, the precision of the optical flow in the part where the object movement changes largely becomes a problem. Therefore, to make the object movement small relatively, we obtain flow vectors from the image sequence, which drops the resolution of the original input image sequence on the first half resolution, and then we get flow vectors, which is smaller than a threshold value, from the original input image sequence. We show that the precision of the optical flow in the part where the object movement changes largely is improved by this method. And then, we show the method to trace loci of moving objects. We search clusters from histograms of flow vectors and pursue each cluster. We show that it is possible to trace moving objects by this method.

Keywords: optical flow, local optimization method, object tracking, clustering

1. Introduction

To obtain optical flow is a big theme in the field of image measurement and its application is expected in various field such as robot vision.

Typical methods to obtain optical flow are the matching method and the gradient method. We choose the spatial local optimization method, which is a kind of local optimization method. We consider improving the precision of this method in the part where the object movement changes largely.

Next, we consider tracing moving objects using optical flow. By using optical flow to trace moving objects, we consider that it is possible to trace many objects independently. Therefore, to trace moving objects, we use a 2-D histogram of flow vectors, and extract regions of objects, and trace loci of moving objects.

2. Optical flow detecting methods

Detecting optical flow using the gradient method is a way of finding out a constraint equation based on the supposition that the distribution of brightness is kept constant between frames when moving objects in an image sequence are rigid

bodies. Here, a general constraint equation is shown in (1).

$$\frac{\partial \rho}{\partial x}u + \frac{\partial \rho}{\partial y}v + \frac{\partial \rho}{\partial t} = 0 \quad \left(u = \frac{\partial x}{\partial t}, v = \frac{\partial y}{\partial t}\right) \quad (1)$$

Because a flow vector has two dimensions, only in the (1) formula, it cannot fix the flow vector uniquely. Therefore, another constraint equation is used together usually. Generally, the global optimization method and the local optimization method are used as this equation.

Next, we compare the computational cost of each technique. Both the matching method and global optimization method have much cost. Therefore, we choose the local optimization method to obtain optical flow.

The local optimization method gives a constraint condition that flow vectors v are constant in some local area S. Here, a general constraint equation, which is used in the local optimization method, is shown in (2).

$$\frac{\partial \mathbf{v}}{\partial x} = \frac{\partial \mathbf{v}}{\partial y} = 0 \quad (x, y \in S) \quad (2)$$

In this research, we obtain flow vectors by formulas (1), (2) above.

Next, when object movement is big, it becomes a problem when using the local optimization method. In this experiment, we confirmed a tendency that the disorder of flow vectors becomes big at movement above 4-5 pixels per frame. Because, in such case, we consider that it doesn't meet the constraint condition to get optical flow from a local area. Therefore, first, we considered finding big flow vectors above 4 pixels from the image sequence, which dropped the resolution of the original input

image sequence on the half resolution. Next, we thought of getting other small flow vectors using the original image sequence. Because, when dropping the resolution of the original input image sequence on the half resolution, we consider that the look object movement becomes 1/2, too. Incidentally, we did the smooth filtering of 9×9 as a pre-process. We consider that this is effective to remove noises.

3. Object tracking using optical flow

Next, we try to trace loci of moving objects. At first, we obtain flow vectors using the method described section 2, and then, we find out clusters from an image sequence using histograms of flow vectors. Because flow vectors that are in the region of a moving object have nearly value each other, we consider that it is possible to extract regions of moving objects by clustering.

(a)   (b)
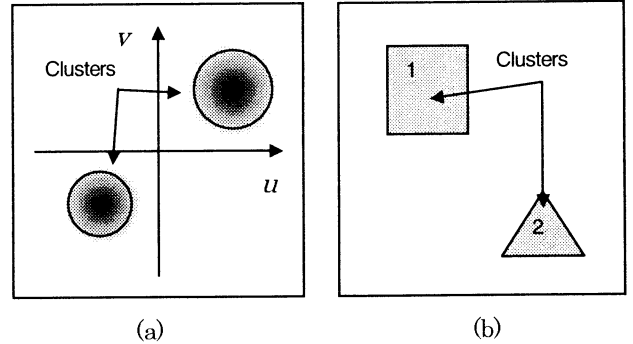
Fig.1   Detecting clusters of moving objects

At first, we remove components of flow vectors less than 1 to reduce noises as a pre-process. Next, as shown in Fig1 (a), we get a 2-D histogram of optical flow. We cluster each region that has flow vectors less than ± 2 in the difference of size and

less than ±45 in the difference of degree with each mode vector of histogram. As shown in Fig.1 (b), we rank clusters in the big order of each cluster so that we can trace moving objects. Therefore, we can trace loci of objects by getting the correspondence of each cluster between frames.

4. Experiments

4.1 Improvement of the precision of the local optimization method

In this experiment, we used the image sequence shown in Fig.2. Flow vectors when making the sphere move to right in 5 pixels in this image sequence are shown in Fig.3 (a)(b).

Here, (a) shows the flow vectors obtained by a general local optimization method, and (b) shows the flow vectors obtained by the method using two kinds of image sequences described in section 2. As the result, (b) shows that the disorder of flow vectors in the part in which object movement changes largely is improved compared with (a). Mean absolute errors between experimental data and ideal data are shown in Table 1. Here, u is the error of x components of flow vectors and v is the error of y components of flow vectors. We found that (b) was better than (a).

Fig.2   Image sequence

(a)   (b)

Fig.3   Optical flow

4.2 Object tracking

Next, we try to trace loci of moving objects on the image sequence shown in Fig.4. Here, the object 1 moves up 3 pixels and moves left 3 pixels per frame. The object 2 moves up 3 pixels and moves right 3 pixels per frame. The object 3 moves down 3 pixels per frame.

Table 1   Mean absolute errors

	u [pixels]	v [pixels]
(a)	1.9	0.5
(b)	1.4	0.3
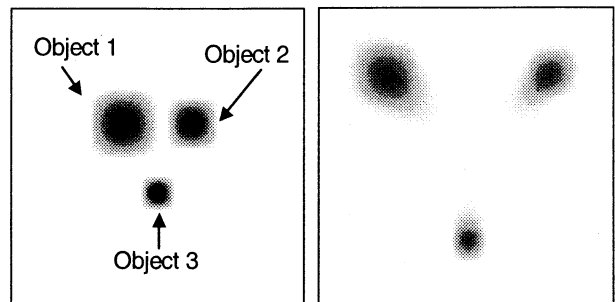
Fig.4   Image sequence

The result of clustering is shown in Fig.5. It shows that each region of three moving objects is clustered well.
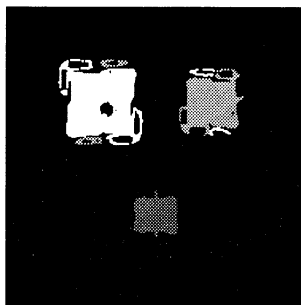
Fig.5   Result of clustering

Fig.6 shows that loci of three moving objects can be pursued well. In this method, we found that the precision of this object tracking method depends on the precision of the clustering.
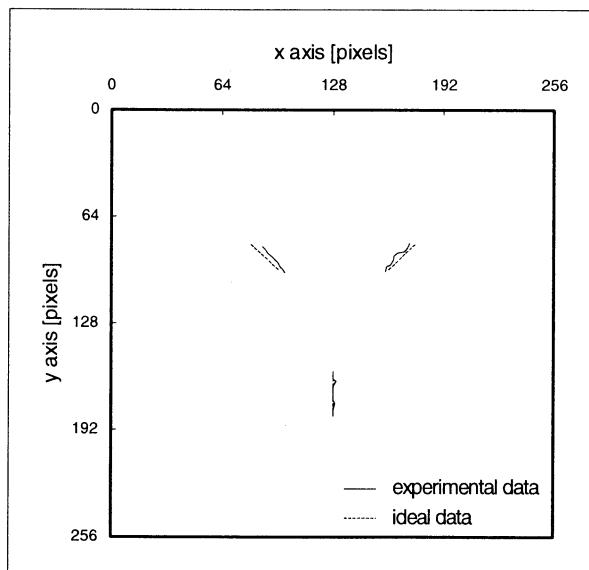
Fig.6   Loci of moving objects

In this experiment, we sometimes couldn't pursue correctly since the ranking of clusters interchanged. It is considered to be cause that we don't take the correlation between frames into consideration because we obtain clusters using independent histograms for every frame. As an improvement method, it is possible to include the temporal and spatial correlation between each frame in the constraint condition of numbering.

5. Conclusion

By this experiments, we found that precision of optical flow when object movement changed largely was improved when using the technique to have been described in section 2. In the future, we consider that it is necessary to verify about whether or not this technique is appropriate about different image sequences.

Next, we consider improving the precision of object tracking. We have to improve the precision of clustering. In this method, when objects translate, it is possible to trace loci of their objects. However, we haven't taken account of movements such as turn or approach yet. Therefore, we have to be possible to trace such movements.

References

[1] H.Miike, K.Koga, et al., "Video image processing using a personal computer", pp.133-143, Morikita Publishing, 1993

[2] G.Xu and S.Tsuji, "3-D vision", pp.111-118, Kyoritsu Publishing, 1998

[3] T.Agui and T.Nagao, "Image processing and image recognition", pp.92-98, Shokodo, 1992

Obstacle Avoidance in the Mobile Robot Using the CCD Camera

Masanori SUGISAKA and Toshio MAEYAMA

Department of Electric and Electronics Engineering, Faculty of Engineering, Oita University

700 Dannoharu, Oita 870-1192, JAPAN

Tel : +81-975-54-7831   Fax : +81-975-54-7818

E-mail : msugi@cc.oita-u.ac.jp   tmaeyama@cc.oita-u.ac.jp

Abstract

The research of robot vision which is named from the meaning to give the robot the feature of the sight which the human being has is studied in the various theories and methods at present, and there are image processing, image recognition, image understanding and so on. The purpose of this research is the autonomous avoidance running of our mobile robot in the area where there is an obstacle by using image processing of binarization and edge detection. We find the area where the robot can run safely in the area where there is an obstacle by using these methods and we have done an autonomous avoidance running experiment of our mobile robot. Therefore, in this paper, we describe the way of the autonomous avoidance running of the mobile robot in the area where there is an obstacle.

Keyword : obstacle avoidance, binarization, edge detection, mobile robot, robot vision

1. Introduction

The problem of the aging society is one of the problems that Japan has at present. The support of the elderly people by the robot can be considered as one of the ways of solving this problem. Therefore, the mobile robot which is in our laboratory was made as nursing robot for the elderly people support. In the present research, our mobile robot can track the line which was put on the floor and can also avoid an obstacle using the ultrasonic sensor. However, in case of tracking the line, it is impossible for our mobile robot to run when there is an obstacle on the line. Also, when using the ultrasonic sensor, our robot cannot sense the sensor which is reflected from the obstacle if the sensor doesn't hit to the obstacle from the front. These problems are reflected in the way described above.

Therefore, in this research, we show that the autonomous avoidance running of our mobile robot is possible in the area where there is an obstacle by using the image processing of binarization and edge detection from the image that is obtained from CCD camera which is in the part of the eyes. If the obstacle avoidance running in a narrow place is possible, the place where our mobile robot is used for spreads and the utilization in a hospital and so on can be considered.

2. Experimental equipment

The robot is composed of two CCD camera located in its head, it has six ultra sonic sensors (two facing forward and rear legs and the rest in the robot body). The robot has four wheels. Two wheels in right and left are the driving wheels, and two wheels in front and rear are auxiliary wheels. The encoder which is used to count the number of pulses is attached to the driving wheels which are rotated by DC motor. We controlled the speed using the number of pulse obtained from this encoder.

3. Image processing method

3.1 Binarization

Binarization makes the pixel "0"(black) if each pixel of an image has the gradation value smaller than a threshold value, and the pixel "1"(white) if each pixel of an image has the gradation value bigger than a threshold value. Because binarization expresses each pixel at two values ("0" or "1"), the after processing becomes easier. There are some ways of deciding threshold such as P-tile method, mode method, distinction analyzing method and so on. In this research, we used the distinction analyzing method.

3.2 Distinction analyzing method

When all the pixels of an image are divided into two classes with a threshold value t (greater than t or less than t), this method is the way of deciding the threshold value t as the separation between two classes becomes the maximum.

When the maximum of bright level is L and all the numbers of the pixels is N and the numbers of pixel of bright level i is n_i, the frequency probability p_i of the bright level is $p_i = \dfrac{n_i}{N}$ ($\sum_{i=1}^{L} p_i = 1$) and the average bright level m of whole image is $m = \sum_{i=1}^{L} ip_i$.

It suppose that the bright level t is the threshold value and it classified all pixels into two classes of $C_1 = \{1 \sim t\}$ and $C_2 = \{t+1 \sim L\}$. Here, we calculate the following two amounts of total bright level t beforehand.

$$\omega_t = \sum_{i=1}^{t} p_i \quad \cdots (1) \qquad s_t = \sum_{i=1}^{t} ip_i \quad \cdots (2)$$

Then, the occurrence probabilities of the pixel on each class are $\omega_1 = \sum_{i=1}^{t} p_i = \omega_t$,

$\omega_2 = \sum_{i=t+1}^{L} p_i = 1 - \omega_t$. Also, the average bright levels of each class are obtained as follow

$$m_1 = \sum_{i=1}^{t} ip / \omega_1 = s_t / \omega_1 \qquad m_2 = \sum_{i=t+1}^{L} ip_i = \frac{m - s_t}{1 - \omega_t}$$

Because $\omega_1 m_1 + \omega_2 m_2 = m$ which always forms, the dispersion σ^2 between two classes is

$$\sigma^2 = \omega_1 (m_1 - m)^2 + \omega_2 (m_2 - m)^2 = \frac{(m\omega_t - s_t)^2}{\omega_t (1 - \omega_t)} .$$

It decides the threshold value t that maximizes σ^2 by using the simple total amount which was obtained by these formulas (1), (2) while changing the value of t . Since it decides the threshold value of t of which the separation between two classes become the best. Therefore, this method brings a good result.

3.3 Labeling processing

If there are some areas in one image, the same label is attached to the point belonging to the same area, and other labels are attached to the point belonging to different areas. We should consider only the point which has the same label when analyzing various nature of each area. This is called the labeling processing.

It carries out the raster scanning of the *run* of value "1" calculated by binarization from the upper left of the image to the lower right. It gives a label if "1" appears. It gives the same label as far as "1" is repeated. It puts another label to the point on the *run* if "1" breaks off and "1" appears again on the same line. If this operation is repeated to the end of the first line, a label which is different to the first line will be attached. In the following, it searches for "1" and it sets a label in the following way. ① A new label is attached if not all *run* of the previous line adjoins. ② If only one *run* of the previous line is adjoined, the label of the *run* is attached. ③ If more than two *runs* of the previous line are adjoined, the label of the smallest *run* will be attached and this label will

edge detection. We ignore the edge which is contained inside the image which is obtained by the binarization processing and express the part of the edge except it black part. Therefore, the place where is expressed in the white part is considered as the safe running area (d).

Here, we consider the running condition of robot in the area where there is an obstacle. In the area which is the most nearest from the robot, we consider both the area of 65 pixels which is equivalent to 45 cm which is the width of the robot and two areas of 22 pixels which are equivalent to 15 cm from this both edges. We make our robot move if there is no obstacle in the area of 65 pixels which is the most nearest from the robot, and we stop our robot if there is an obstacle in this area. We make our robot turn to the left if there is no obstacle in these 65 pixels areas which are the most nearest from the robot and if there is an obstacle in the area of 22 pixels on the right, and we make our robot turn to the right if there is no obstacle in these 65 pixels areas which are the most nearest from the robot and if there is an obstacle in the area of 22 pixels on the left. We experiment the autonomous avoidance running of our mobile robot in the area where there is an obstacle using this condition.

4. Experimental result

We controlled the speed of our mobile robot by P control using the number of pulse obtained from the encoders that are attached to the right and left driving wheels which are rotated by DC motor. The desired value is set to eight pulses (8pulses/0.1s). The result is shown in the Fig.2. We experimented the autonomous avoidance running of our mobile robot in the area where there is an obstacle by using these parameters. We have considered that the obstacle avoidance running of our mobile robot is possible by using this method.
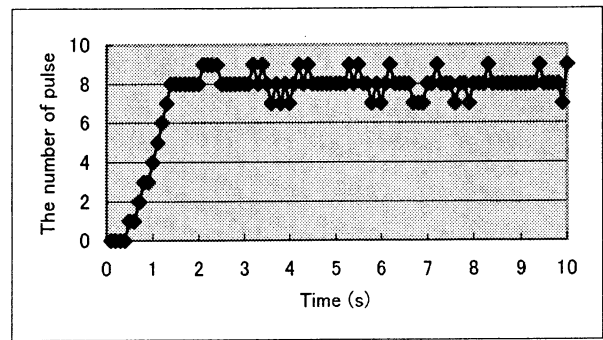
(a) Right wheel parameter (Kp = 0.143)
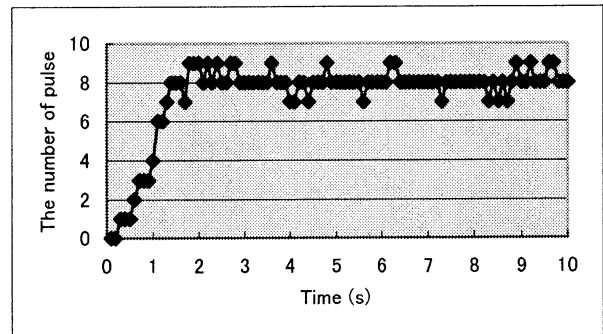
(b) Left wheel parameter (Kp = 0.154)

Fig.2 Proportional parameter of right and left wheel

5. Conclusion

In this paper, we have described the method of the autonomous avoidance running of our mobile robot in the area where there is an obstacle. The obstacle avoidance running of our mobile robot is possible by using this method. However, our robot can't run yet if a black tape is taped on the floor and also if the floor is black. It is necessary to consider the method of solving these problems in the future.

Reference

[1] M.YACHIDA, "Robot vision", P1-2, 40-45, Shoukoudou (1990)

[2] T.AGUI and T.NAGAO, "Processing and recognition of the image", P40-41, Shoukoudou (1992)

[3] H.OZAKI, K.TANIGUCHI, "Image processing –From the foundation to the application-", P212-214, Kyoritsu suppan Incorporated company. (1983)

be memorized as the same label. After processing to the lower right in this way, it scans the image again and rewrites the label which is an identical label.

3.4 Edge detection

What is often attached to our eyes when we look at an object is the part of the edge which is a boundary. The brightness is often changing rapidly in the place of the edge whereas the point on the same surface has resembling brightness. Therefore, it should find the place where the light is changing to detect an edge. In this experiment, we use range filter for the edge detection. The range filter is the filter which takes the difference between the maximum and the minimum value of gradation value in the neighborhood of the remarkable pixel. Since the difference of the maximum and minimum value is taken, the range filter has the nature that the value becomes big in the part which the gradation value is changing. The formula which calculates the new gradation value $g(i, j)$ from the present gradation value $f(i, j)$ of the pixel near a pixel (i, j) is shown as the follow.

$$g(i, j) = \max\{ f(i+m, j+n) \mid -1 \leq m, n \leq 1\}$$
$$- \min\{ f(i+m, j+n) \mid -1 \leq m, n \leq 1\}$$

4. Experimental method

First, we supposed that the area where is the nearest from our robot can be safely run and it takes in an image with the CCD camera which is in the part of the eyes (a). Since the obtained image is a color image, we change into the monochrome gradation image and do the image processing of the binarization (b). If there is an obstacle, a shadow appears and the part of this shadow is expressed in black. Therefore, since the place where there is not an obstacle is expressed in white, we consider that the white area is the safe running area and we make our robot run in

this area. However, when an obstacle is white, the robot recognizes by mistake that the place where there is an obstacle is a safe running area in spite of being an obstacle. Therefore, it is impossible for our robot to do autonomous avoidance running in the area where there is an obstacle by using only the image processing of the binarization.

Next, we considered doing the image processing of the edge detection using the range filter. If there is an obstacle, the part of edge appears and it is expressed in white (c). Therefore, since we consider that the part which is expressed in black when doing the image processing of the edge detection is a safe running area, we make our robot run in this area. However, if a tape is put on the floor, the edge appears in the safe place that is not an obstacle. Therefore, it is impossible for our robot to do autonomous avoidance running in the area where there is an obstacle by using only the image processing of the edge detection. In other words, it is impossible for our robot to run safely by using only the binarization and the edge detection independently.

(a) Original image   (b) Binarization image

(c) Edge image   (d) Integration image
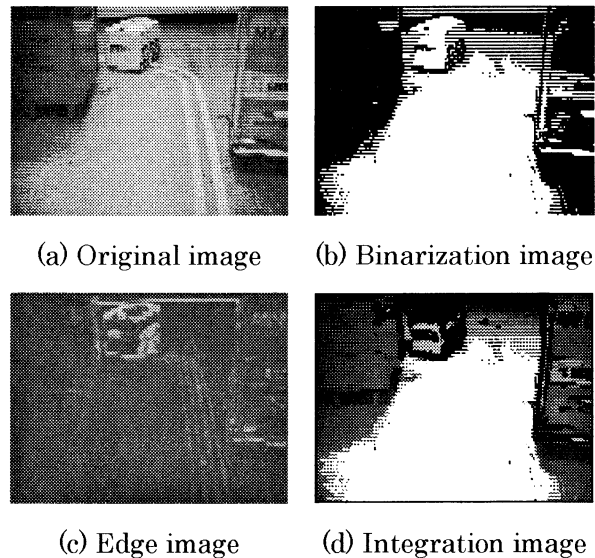
Fig. 1 Processing image

Then, we consider the area that is obtained by integrating the image which is labeling to the image of the binarization with the image of the

Safe and Efficient Navigation for an Intelligent Wheelchair Robot with Human-robot Interaction

Inhyuk Moon   Sanghyun Joung

Department of Medical Engineering, Yonsei University

234 Maeji, Wonju, Kangwon 220-071, KOREA

Abstract

This paper proposes a safe and efficient navigation method for the intelligent wheelchair robot with the Human-robot Interaction (HRI). Since the purpose of the wheelchair robot is mobility aids for the handicapped, the wheelchair robot must guarantee the safety while considering the user's intention. A color C-CD camera is used for input user's intention directed by face directional gestures, and an array of ultra sonic sensors is used for sensing external environment configuration. By combining the sensory information with the user's intention, the planner of the wheelchair robot selects an optimal motion. From experimental results using a wheelchair robot, *MR. HURI*, we show that the proposed navigation method is useful for the MR. HURI.

1   Introduction

Recently, as the silver generation has been exponentially increasing, the social demands about the quality of life also have been increasing proportionally to them. A wheelchair robot for mobility aids for the handicapped was introduced[1]. Since the structure of the wheelchair robot is similar to the mobile robot, many successful results were introduced[2]. However er such researches have been concentrated on sensing and path planning because localization and obstacle avoidance are important problems in the the mobile robotics. Since most user of the wheelchair robot is the handicapped, the wheelchair robot must guarantee the safety and reliability while considering the user's intention. Accordingly, recent studies adopt a human-robot interface in order to include a human into the robot feedback control loop for the safety. We can see that a biosignal such as the electromyogram(EMG)[3], and human motion such as gaze direction[5] and facial direction[4] were utilized as the interface. But the handicapped such as the disabled and the elderly is abnormal. Therefore, the user's command may be incorrect. And the sensory information obtained by external sensors such as vision, sonars, and laser range finder also includes uncertainty. In these reasons, the safety is the most considerable problem in the welfare system.
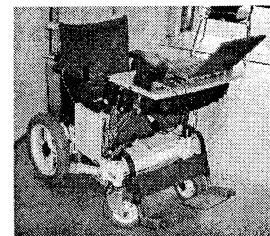
Figure 1: MR. HURI.

In this paper, we propose a safe and efficient navigation method for the intelligent wheelchair robot with the Human-robot Interaction (HRI). The wheelchair robot, MR. HURI (Mobile Robot with HUman Robot Interaction) can avoid obstacles and select motion direction autonomously (see Fig. 1). In addition, it can consider user's intention for the motion direction. We use 13 ultra sonic sensors for detecting obstacles, and the wheelchair robot can avoid obstacles using the sensory information. When the robot motion is incorrect, the user can correct the motion direction by the gesture command. From user's command and the sensory external configuration information by sonars, the robot planner performs an optimal motion planning. As a result, the wheelchair robot can perform a safe and navigation motion while considering the user's intention.

In section 2, we propose a human-computer interface scheme for the HRI. In section 3, we show the motion planner of the MR.HURI. Lastly, we conduct experiments using the MR. HURI, and then conclude this study.

2   HRI by face directional gesture

2.1   Command for HRI

We first define user's commands based on the face directional gesture for the HRI. Nakanishi et al.[4] used only the face direction as user's command, but the robot could not discriminate whether the given command is intentional or unintentional behavior. In this paper, all commands except for "stop" are completed

when the nodding motion is given in order to decide upon the user's intention. The commands for the HRI are listed in Table 1. O means a valid gesture is given, but X is not. For instance, when the user's head turns to left and gives the nodding motion, "left turn" command is completed.

Table 1: HRI commands using face directional gestures.

command	face direction	shaking	nodding
go	forward	X	O
left turn	left	X	O
right turn	right	X	O
stop	X	O	X

2.2 Recognition of face direction

Fig. 2 shows the processing flow for recognizing face directional gesture. A facial region is first extract from input image using the skin color information. Input image is transformed to hue(H), saturation(S), and intensity(I) components.

At the initial state of processing, we first determine the threshold values of the skin color. We manually select a special face area, and calculate the mean and variance of the color components of H and S using initially taken 10 frame images. Assuming that the skin color follows Gaussian distribution[6], we set 3σ around the mean value of H and S to the valid skin color threshold values.

Labeling, shrinking and dilation processing are then applied to the extracted skin color image. From the labeling result, we select large clusters as the face candidates. Facial features such as eyes, eyebrows, mouth, and nose are existed more than five features in the face, and such facial features have relatively low intensity than facial region. Therefore, we can select a facial region from the face candidates by counting such features numbers in each face candidate.

Fig. 3(a) and (b) show the initial state for calculating the threshold value of the skin color and an extracted result by applying the obtained threshold value. Fig. 3(c) and (d) show the selected large clusters and the obtained result by comparing the number of facial features, respectively.
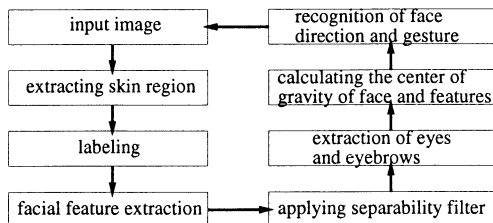
Figure 2: Processing flow for recognizing face directional gesture.

Figure 3: Processing results for detecting face region.

Face direction is recognized by comparing positions of the facial region and the facial features such as eyes, eyebrows, nose, and mouth. However we only use eyes and eyebrows features for the recognition of face direction because nose and mouth features are easily affected by the illumination condition. Eyes and eyebrows features are extracted by applying a separability filter based on a difference of intensity variance between two regions. Fig. 4 shows a template of the separability filter. The template size can be changed by the facial region size. Let $\sigma_T{}^2$ and $\sigma_b{}^2$ be the intensity variance of the whole region of the template and the variance between two regions, respectively, which are expressed by the following equations.

$$\sigma_b{}^2 = n_1(\bar{P}_1 - \bar{P}_m)^2 + n_2(\bar{P}_2 - \bar{P}_m)^2 \quad (1)$$

$$\sigma_T{}^2 = \sum_{i=1}^{N}(P_i - \bar{P}_m)^2 \quad (2)$$

where N denotes the number of all pixels in the whole template region, and n_1, n_2, and P_i denote the number of pixels in each region $1, 2$ and a pixel intensity, respectively. And $\bar{P}_1, \bar{P}_2, \bar{P}_m$ are the mean of intensity in the region $1, 2$ and the whole region. The separability n is expressed by the following equation.

$$n = \frac{\sigma_b{}^2}{\sigma_T{}^2} \quad (3)$$

From Eq. (3) the separability n is a value within $0 < n < 1$. If two regions are completely separated, n is 1. When $\sigma_T = 0$, n is 0.
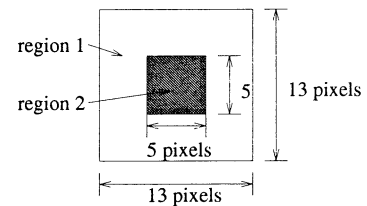
Figure 4: Template of separability filter.
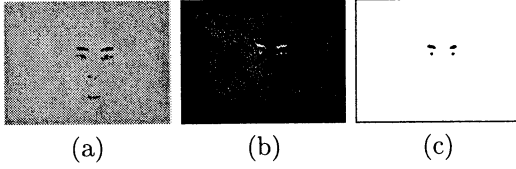
Figure 5: Extracted results by the separability filter.

Fig. 5(a) shows extracted facial features in the facial region, and Fig. 5(b) is a result obtained by the separability filter. By applying the dilation and shrinking processing to the Fig. 5(b), we can extract eyes and eyebrows as shown Fig. 5(c).

In this study, we set the center of gravity of the features to the feature position. Therefore, we first obtain histograms of the extracted facial features projected on the horizontal and vertical axis, respectively. And then, we obtain the mean position by applying the 1D moment.

Face direction is determined by comparing the positions between the facial region and features. Fig. 6 shows recognized results of the facial direction. The black rectangular area is the extracted facial region, and the white lines are positions of the features on the vertical and horizontal axis. When the face direction is left as shown in Fig. 6(a), the difference between positions of the face region and the facial feature is minus, but plus when the face direction is right as shown in Fig. 6(c). By tracking the face direction continuously, the shaking gesture is recognized when the face direction is changed from left to right, alternately.

As the same manner, the nodding gesture is recognized if the position is changed up and down with relatively large differential value. Fig. 7 shows the tracking result of the face directional gesture. RS and LS denote left and right direction selection, and ESC denotes the shaking gesture, respectively.
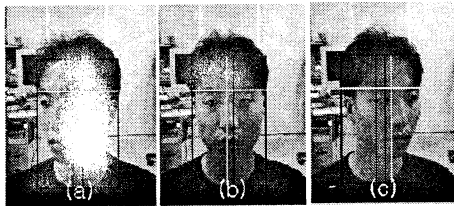
Figure 6: Recognized results of the face direction.

3 MR. HURI and motion planning

3.1 Motion planning

The MR. HURI plans the motion direction based on the user's command and the detected obstacle configuration. If no obstacles are detected within 200[cm]
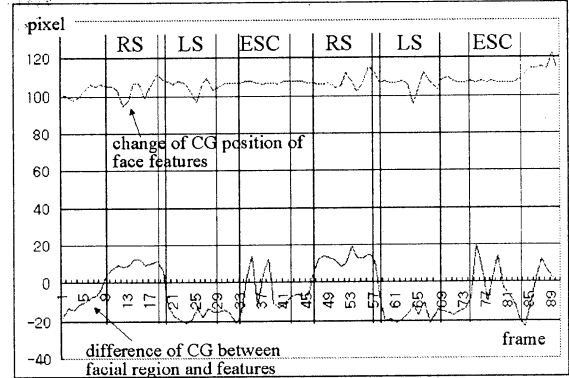
Figure 7: Tracking result of the face directional gesture.

distance, MR. HURI regards such space as the *free space*. In the free space, the motion direction is the same as the straight forward direction as shown in Fig. 8(a). If any obstacle is detected within 120[cm] distance in ±30[deg] from his current forward direction, MR. HURI regards such space as the *narrow space*. In the narrow space, he searches for the largest free space, and then selects the center of the free space as the next motion direction, autonomously. At this time, the turning direction is no matter whether left or right. However, if the user's command is given, the turning direction depends on the directed orientation.

Given user's command in the free space, MR. HURI turns 20[deg] for each user's command to the directed orientation, but in the narrow space the turning angle is depends on the center of the free space in the directed orientation.

In the *middle space* as shown in Fig. 8(c), the planning is different from above cases. MR. HURI moves to the straight forward direction until he meets the narrow space. If the user's command is given in the middle space, MR. HURI selects a motion direction as the same method like the narrow space. If MR. HURI cannot determine the motion direction autonomously, he stops and waits a new user's command. After turning by user's command, MR. HURI keeps the motion direction to the straight forward, but he tries to return

(a) free space   (b) narrow space   (c) middle space
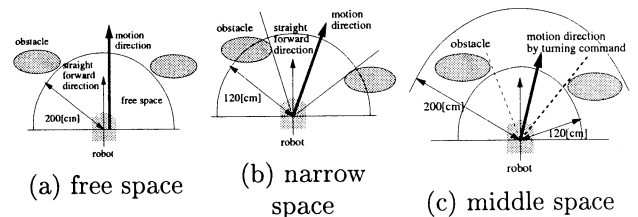
Figure 8: Planning of motion direction.

the original motion direction before turning if turning motion is occurred autonomously without any user's command.

4 Experiments

A user sits on the MR. HURI, and he is separated about 70[cm] distance from the CCD camera. Fig. 9 shows sensing and motion results by the ultra sonic and the robot planner. In Fig. 9(a), MR. HURI starts to the forward direction by user's start command. At (1) position, he meets the narrow space, and an obstacle is detected in ±30[deg] by sonar sensors as shown in Fig. 9(c)(1). Note that the black point in Fig. 9(c) denotes MR. HURI position. Therefore, MR. HURI searches for the largest free space, and then selects the center of the free space as new motion direction. After turning, MR. HURI moves to the straight forward direction. Because the turning motion is autonomously performed, MR. HURI returns to the previous motion direction. At (2) position, MR. HURI can turn the original direction as shown in Fig. 9(c)(2). Lastly he turns and moves to the straight forward direction.

At (3) position in Fig. 9(b), MR. HURI needs to select a motion direction by the user's command because the movable free space is almost same. The user gives the turning right command by gesture. Then, MR. HURI searches for a movable free space in the directed space. At (4) position, MR. HURI finds out a free space, and turns to the center of the free space as shown in 9(c). At (5) position in Fig. 9(b), MR. HURI turns to left direction because obstacle is detected at the forward direction.

In this experiments, the image size for the user's command input is 320x240, and the input image is processed by using the host's CPU performance without any special hardware. The processing speed is 6.57 [frame/sec], and the success rate for recognizing the face directional gesture is 93%.

5 Conclusion

In this paper, we proposed a prototype of the wheelchair robot with human-robot interaction function. The implemented wheelchair robot, MR. HURI, has functions of the mobile robot and HRI, so we can see that such wheelchair robot is more safe and reliable for the robot motion. Since the safety and reliability is the most important problem in the welfare robotics, the proposed wheelchair robot is feasible for the handicapped.

In this study, we just used the hand gesture for calling and directing the wheelchair robot. If the two gestures, face and hands, are combined for the HRI, the human-robot interaction will be achieved more various types.
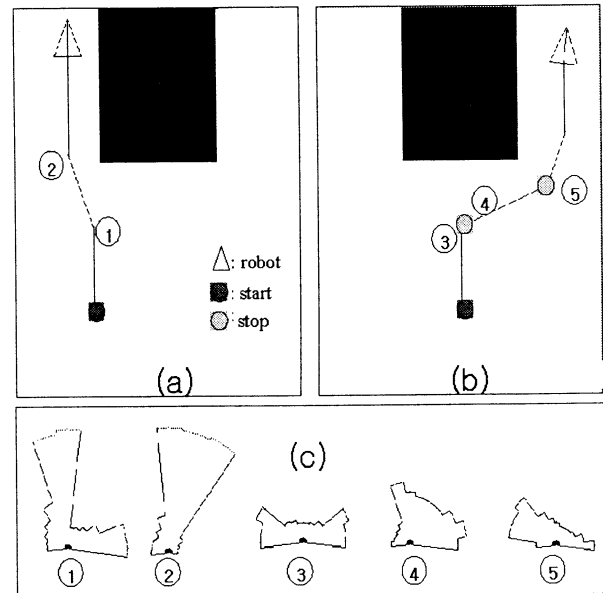
Figure 9: Motion results of MR. HURI.

Acknowledgement

References

[1] S.P. Levine, D.A. Bell, L.A. Jaros, R. C. Simpson, and Y. Koren, "The NavChair Assistive Wheelchair Navigation System," *IEEE Trans. on Rehabilitation Eng.*, Vol. 7, No. 4, 1999.

[2] V. Mittal, et al. eds., *Assistive Technology and Artificial Intelligence* , Springer, 1999.

[3] O. Fukuda, et al., "EMG-based Human-Robot Interface for Rehabilitation Aid," in *Proc. of I-CRA'98*, pp. 3492-3497, 1998.

[4] S. Nakanishi, Y. Kuno, N. Shimada, and Y. Shirai, "Robotic Wheelchair Based on Observations of Both User and Environment," in *Proc. of IROS'99*, pp. 912-917, 1999.

[5] Y. Matsumoto and A. Zelinsky, "An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement," in *Proc. IEEE Int. Conf. on Face and Gesture Recognition*, pp. 499-505, 2000.

[6] J.C. Terrillon and S. Akamatsu, "Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images," in *Proc. Conf. on Vison Interface*, pp. 180-187, 1999.

Obstacle Avoidance Algorithm for Visual Navigation Using
Ultrasonic Sensors and a CCD Camera

Su Hyun Choi   Il Myung Kim   Tae Seok Jin   Jang Myung Lee

Department of Electronics Engineering Pusan National University, Pusan, 609 -735, Korea
{suhyunch, ilmyoung, jints, jmlee}@hyowon.pusan.ac.kr

Abstract

In this paper, an obstacle avoidance algorithm is presented for autonomous mobile robots equipped with a CCD camera and ultrasonic sensors. The approach uses segmentation techniques to segregate floor from other fixtures and measurement techniques to measure distance between mobile robot and obstacles. It uses a simple computation for the selection of a threshold value. And the approach uses a cost function, which is combined with image informatio n, distance information and a weight factor to find an obstacle free path.

This algorithm, which uses a CCD camera and ultrasonic sensors, can be used for the case of including shadow region which is conceived obstacle in visual navigation and in various lighting conditions.

Keywords : Mobile robot, CCD camera, Ultrasonic sensor, Shadow region, Visual navigation

1. Introduction

Robots are already in used in the industry to replace human beings for repeated and dangerous workload. Robot's usefulness are v aried, we can used from them to transport objects. In transporting an object, there may exist some obstacles scattered on its way to its destination. For the robot to arrive at its destination, it must be capable in avoiding the obstacles. These obstacles that are normally found in the real world can either be stationary or moving. Obstacle avoidance algorithm can be divided into model-based system and sensor based system.

The advantage of having the model -based system is its capability to generate an optimal obstacle free path. Its advantage is that it cannot be used in an environment where an unknown objects or moving obstacle present.

In a sensor based system, the sensor current output is either recorded in a defined map and process it to avoid collision, or it could let the robot react to whatever is the current data without talking into account the previous one.

Since the problem of obstacle avoidance can also be conceived as finding the obstacle free area in the environment, then, locating the free spac e and redirecting the robot into that space will obviously avoid the obstacles. Though, the path may not be an optimal one.

In visual navigation using CCD camera, the robot's direction is determined by only input image. But, if the image includes shadow re gion, the robot conceives shadow region obstacle. Though, the shadow region may not be an obstacle. This is a difficult problem in visual navigation.

Ultrasonic sensor have been used widely in robotics for obstacle avoidance mainly due to their simplicity and relatively low cost. These types of external sensors are very good in distance measurement of the obstacle, but they also suffer from specular reflection and insufficient directional resolution due to their wide beam opening angle.

In this paper, a simple algorithm that segregates floors from objects regardless of existing shadow region using ultrasonic sensors and CCD camera to avoid collision is used. The segregation of floors from objects is processed as followings. First, it obtains input image and transforms binary image using threshold value that is determined by arithmetic mean, standard deviation, constant scalar value. Next, it obtains distance information around robot using ultrasonic sensors. Finally, it combines image information and distance information which were formulated by cost function and segregates floors from objects and determine the obstacle free path.

2. Obstacle Avoidance Algorithm

Since the obstacle free path will be generated in terms of sensed data, no information of the ge ometry or absolute positions of the obstacles is taken. Suppose in Figure 1 an obstacle is present in front of the robot. The robot should only find the free space and redirect itself into obstacle. This means that if the robot is in the middle of the obstacles, it only needs to find the flank of the obstacles. The redirection path is represented by the dotted lines. And if the robot is in an environment where a multiple obstacle is present, it can still apply the redirection path by finding the flank or the free space.
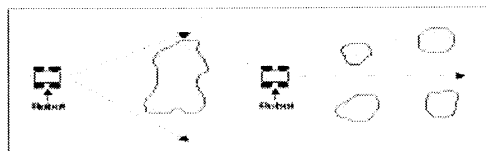
Figure 1. Sample of redirection via flanking

But, in the case of including shadow region by external illumination, robot conceive shadow region obstacle. In figure 2, obstacle free path is $\vec{r}_2$ but robot conceive obstacle free path $\vec{r}_1$ or $\vec{r}_3$.
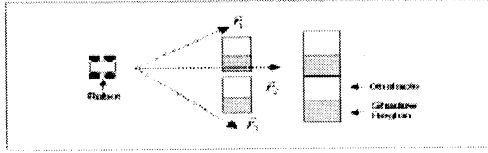
Figure 2. Sample of redirection in the case of including shadow region

This is important problem in visual navigation. In this paper, it uses ultrasonic sensor in order to solve this problem and discussed on Section 5(Computation of obstacle free path).

3. Segmentation

Gray level segmentation or thresholding is a conversion between a gray level image and a bilevel image.

Most images are represented by function of two spatial variables, where $I(x, y)$ is the brightness of the gray level at a spatial coordinate (x, y).

The most common way to convert between gray level to bilevel images is to select a threshold value T. All of the gray level below this threshold value will be classified as black(0) and all of the gray level above will be white(1). This means that the new data will be classified into two groups.
That is,

$$I(x, y) = \begin{cases} 0 & if \ \tilde{I}(x, y) < T \\ 1 & if \ \tilde{I}(x, y) \geq T \end{cases} \qquad (1)$$

where $\tilde{I}(x, y)$ is the original intensity value, x and y are the location of the pixels in the image.

The main problem of segmentation is on how to compute the threshold value which will be used in segmenting the image.

In this paper, threshold value is computed as follow, where M is the number of pixel, X is the intensity value in the image, μ is the arithmetic mean, σ is the standard deviation. T is threshold value and k is a constant scalar.

$$\mu = \frac{\sum X}{M} \qquad (2)$$

$$\sigma = \sqrt{\frac{(\sum X - \mu)^2}{M}} \qquad (3)$$

$$T = \mu - k\sigma \qquad (4)$$

Figure 3, 4 shows the original image and its bilevel image by setting constant scalar $k = 1$.
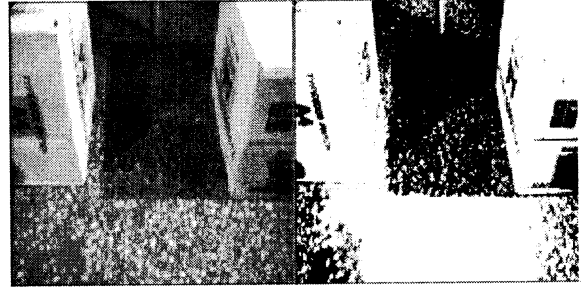
Figure 3. Original image and its bilevel image
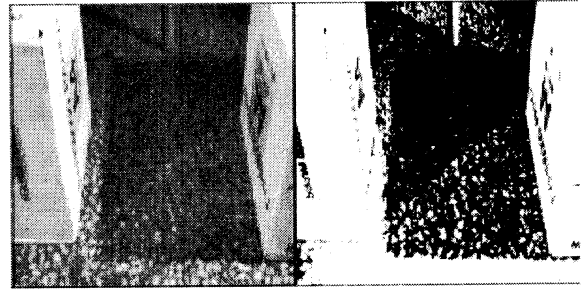
Figure 4. Original image and its bilevel image

4. Ultrasonic Sensors

The basic principle for measuring distance using ultrasonic sensor is to record the time difference between the emitting of transmission wave and the receiving of echo. The measurement system for distance used here has three small wide angle ultrasonic detectors and see figure 5.
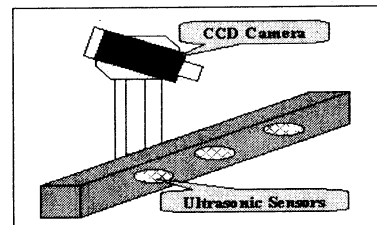
Figure 5. Layout of distance measurement system

The distance between obstacle and robot is equal to

$$D(k) = \sqrt{\frac{c^2}{4} t_d(k)^2 - d^2}$$ (5)

where,
- $D(k)$ is the distance between obstacle and robot at time k.
- c is the speed of sound in air (≈ 343 m/s at $20\,^{\circ}C$).
- $t_d(k)$ is the measured time delay between emission and reception of signal at time k.
- d is the half distance between the transducers.

5. Computation of Obstacle Free Path

Figure 6, 7 shows an area which is defined as an obstacle free area and in this figure, y axis is the height of the image while the x axis refers to the width of the image.
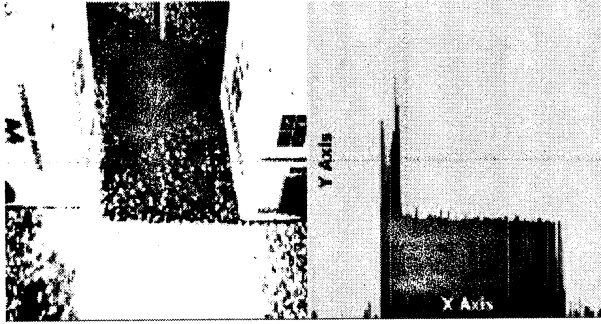
Figure 6. The plot of the obstacle free area

Figure 7. The plot of the obstacle free area

This figure is taken by calculating the distance from the bottom in the bilevel image until it encounters a black pixel(0) which represents obstacle on the bilevel image,

then it records the pixel count and start to count again on its next adjacent pixel. The image used here is 512×512 pixel image, the calculation stops after it reaches the 512th vertical line.

The distance of obstacles from the bottom of the bilevel image is represented by H,

$$H(x) = \min(\,y),\quad for\ I(x,y) = 0$$ (6)

and $0 \le x \le 511$, $0 \le y \le 511$.

With this calculation, we calculate again the obstacle free path. This is done by locating the biggest area on the obstacle free area with respect to the robot's width. The computed areas $A_1, A_2, ..., A_k$ from k to $k + RobotWidth$ is

$$A_k = \sum_{x=k}^{k+RobotWidth} H(x)$$ (7)

where k is the number of computed areas. The obstacle free path is the biggest computed area,

$$\max[\,A_k\,].$$ (8)

Figure 8, 9 represents a calculation of the biggest area and obstacle free path in the bilevel image.
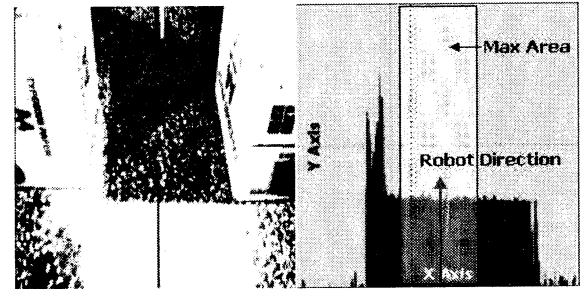
Figure 8. Calculation of the biggest area and obstacle free path in the bilevel imgae
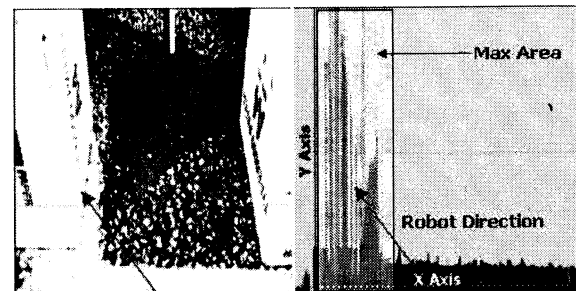
Figure 9. Calculation of the biggest area and obstacle free path in the bilevel image

But, obstacle free path from this figure is not proper because of shadow region. Robot conceives shadow region obstacle and robot converts its direction though the shadow region may not be an obstacle.

Thus, in this paper we propose following cost function which consists of A_k, $D(k)$ and weight factors.

The obstacle free path is determined in the direction of minimizing cost function. The cost function is

$$J = \alpha\ A_k + \beta\ D(k) \qquad (9)$$

Robot's Direction = $\min (J)$

where A_k is obstacle free area, $D(k)$ is the distance between obstacle and robot at time k and α, β are weight factors.

Figure 10, 11 shows the obstacle free path in the case o f including shadow region which is conceived obstacle in visual navigation where $\alpha = 0.5$, $\beta = -1$, $D = 3m$.
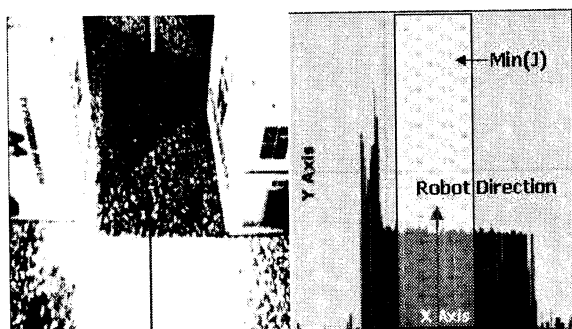
Figure 10. The obstacle free path
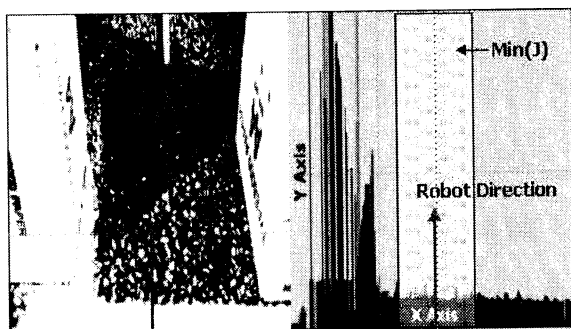
Figure 11. The obstacle free path

In these figures, robot moves in shadow region because of using cost function and this path is the obstacle free path which is determined by minimizing proposed cost function.

6. Conclusion

An obstacle avoidance algorithm was presented which is based on segmentation of input image, distance measurement using ultrasonic sensors, cost function to determine obstacle free path and this algorithm is based on sensory feedback without taking into account the previous images. But this algorithm may not be find the optimal obstacle avoidance path.

The major point of this algorithm is its simplicity for implementation and this algorithm also provides the basic technique in visual navigation in the case of including shadow region and a possible combination of other visual cues in the future will produce a more robust algorithm.

Study of the number of optimal ultrasonic sensors, method of deciding optimal weight factors to seek the optimal obstacle avoidance path will be continued.

7. References

[1] Rafael C. Gonzalez, Richard E. Woods, "Digital image processing," Addison Wesley, 1993.

[2] Kai-Tai Song, Wen-Hui Tang, "Environment perception for a mobile robot using double ultrasonic sensors and a CCD camera," IEEE Transactions on Industrial Electronics, Vol. 43, No. 3, pp. 372-379, 1996.

[3] Tsung Nan Chou, Catherine Wykes, "An integrated vision/ultrasonic sensor for 3D target recognition and measurement," IPA 97, pp. 189-193, 1997.

[4] JM. Maja, T. Takahashi, Z. D. Wang, E. Nakano, "Real time obstacle avoidance algorithm for visual navigation," Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 925-930, 2000.

[5] Anders Nilsson, Per Holmberg, "Combining a stable 2D vision camera and an ultrasonic range detector for 3D position estimation," IEEE Transactions on Instrumentation and Measurement, Vol. 43, No. 2, pp. 272-276, 1994.

[6] Konstantin Kondak, Sven Horstmann, et al., "Computation of Optimal Collisionfree Movements for Mobile Robots," Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1906-1911, 2000.

[7] Siripun Thongchai, Kazuhiko Kawamura, "Application of Fuzzy Control to a Sonar-Based Obstacle Avoidance Mobile Robot, " Proceedings of the 2000 IEEE International Conference on Contro l Applications, pp. 425-430, 2000.

[8] Wang, ZhiDong et al., "A Multiple Robot System for Cooperative Object Transportation with various requirements on Task performing, " ICRA 99, Vol. 2, pp. 1226-1233, 1999.

Causality-Driven Motion Intelligence of Free-wheeled Mobile Manipulator to Ride over Waved Road

Atsushi TAMAMURA

Graduate School of Engineering

Fukui University

Bunkyo3-9-1, Fukui, 910-8507

Mamoru MINAMI

Faculty of Engineering

Fukui University

Bunkyo3-9-1, Fukui, 910-8507

Toshiyuki ASAKURA

Faculty of Engineering

Fukui University

Bunkyo3-9-1, Fukui, 910-8507

Abstract

In this paper, we propose a method to acquire a kind of machine intelligence to utilize dynamically interfered motion of a free-wheeled 1-link mobile manipulator, which we call as motion intelligence. The motion intelligence is defined here as an ability that the machine can find by itself a way to use dynamical interferences and nonlinear friction exerting between robot's tires and road. The desired motion of the 1-link mobile manipulator which does not possess driving motors nor brakes is to travel to one direction and to ride over waved road by only using dynamical interfering motion of the mounted link, that is swinging motion. Simulations show that proposed GA-based causality-driven system could find an effective way to make oscillate the mobile manipulator and finally to ride over the waved road.

1 Introduction

A mobile manipulator consists generally of a vehicle to move and a mounted manipulator to work, those are correspond to leg and arm in case of a human. The dynamical interferences of the mobile manipulator have been thought that they should be erased to improve control accuracy of moving and working operation. However, it is a fact that a human is able to run faster by using arms' swinging motion. A human is using the dynamical interferences of arms' swinging motions to run faster. The improvement of arms' swinging motions for the purpose of fast running is able to be thought as a motion intelligence, which is defined here as an ability to utilize effectively nonlinear dynamics such as the dynamical coupling, coulomb friction and so on. We think that the mobile manipulator possessed a kind of motion intelligence if it can realize a function by itself to achieve a given objective in dynamical behaviour.

In this paper, we aimed to give a 1-link mobile manipulator the motion intelligence. The desired motion of the 1-link mobile manipulator which does not have driving motors nor brakes is to travel to one direction and to ride over waved road by only using dynamical interfering motion of the mounted link, that is swinging motion. The proposed method is composed of a function generator to give the mounted link sample motions using Fourier Series and evaluation system by Genetic Algorithm (GA) to improve the Fourier coefficients based on a fitness function, that is here evaluation of increasing rate of amplitude of oscillation. Simulation experiments show that proposed system could find an effective swinging way to make oscillate the mobile manipulator and finally to ride over the waved road.

2 Mobile Manipulator Model

Figure 1 shows the actual model of the 1-link mobile manipulator, which is used for the simulation experiments in this paper. The machine does not have driving motor to travel nor break to stop. The traveling motion, which is basically just oscillation, is induced by the swinging motion of the arm. For modeling, we
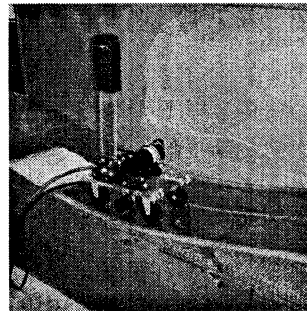
Fig. 1: MobileManipulator
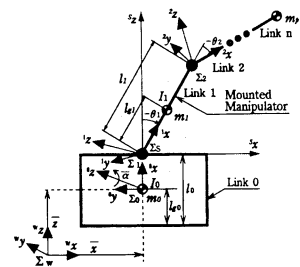
Fig. 2: Model

assumed that the wheel doesn't slip. To discuss the motion intelligence while avoiding the non-holonomic character of mobile robot, it is also assumed that the mobile manipulator does not steer, that means it moves in vertical plane. The vehicle's posture is represented by x, z, α and mounted link's angle by θ, then p is defined as $p = [x, z, \alpha, \theta]^T$. The α means tangential direction of the road. The the equation of motion is given by Eq.(1)[1].

$$
\begin{aligned}
&M\ddot{p} + h + g + D \\
&= \tau_M + U_t f_t + U_n R^{-1}\Big[-\frac{\partial}{\partial p}\big(\frac{\partial C}{\partial p}\dot{p}\big)\dot{p} \\
&\quad + \frac{\partial C}{\partial p}M^{-1}\{h + g + D - \tau_M - U_t f_t\}\Big]
\end{aligned} \tag{1}
$$

, where the waved road is represented by $z = f(x)$ and, $\tau_M = [0, 0, 0, \tau_1]^T$, $M(p)$: the inertia matrix, $h(p, \dot{p})$: the vector of coriolis force and centrifugal force, $g(p)$: the gravity. Besides, we express the frictional force $D(\dot{p}) = [d_x, d_z, 0, d_1]^T$ approximately as,

$$
d_v = \mu_v v + \mu_{sc} N \, sign(v) \tag{2}
$$

$$
N = mg\cos(\alpha) \tag{3}
$$

$$\mu_{sc} = \begin{cases} \mu_c + \mu_s exp(-a \mid v - b \mid) & \mid v \mid \geq b \\ \mu_c + \mu_s & \mid v \mid < b \end{cases} \quad (4)$$

$$v = \sqrt{\dot{x}^2 + \dot{z}^2}, d_x = d_v \cos(\alpha), d_z = d_v \sin(\alpha), d_1 = \mu_1 \dot{\theta}$$

where μ_v and μ_1 are viscous friction coefficients for the mobile robot and the arm, m means the weight of the mobile manipulator.

3 Motion Intelligence

3.1 Search method by GA

Periodical desired angle function $\theta_{1d}(t)$ of mounted arm is represented by using Fourier Series as shown in next equation.

$$\theta_{1d}(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \{a_n \cos(\frac{2n\pi}{T}t) + b_n \sin(\frac{2n\pi}{T}t)\} \quad (5)$$

Then the $\theta_{1d}(t)$ means a cycle function with a period T, and $a_0/2$ is the average value of the function, a_n and $b_n (n = 1 \sim \infty)$ express Fourier coefficients. Since it is impossible that a computer calculates infinitive sammation, we limited it by a number m. The coefficients are $T, a_0, a_1, b_1, \cdots, a_m, b_m$, which are treated as components of genes in GA process. Then the j-th indivisual in the i-th generation, which is represented by g_j^i, is given by Eq.(7).

$$g_j^i = [T_j^i, a_{0j}^i, a_{1j}^i, b_{1j}^i, \cdots, a_{mj}^i, b_{mj}^i]. \quad (6)$$

It is necessary for GA a standard for selection to evolve the population. The fitness function used as the standard in GA, which evaluates how the selected motion corresponding to the indivisual fits to the enviroment, that is here, the waved road and wheels wituout motor. The motions purpose is to ride over the waves, and to travel as fast as possible. Each indivisual in the population, g_j^i, determines the desired motion of the arm by Eq.(6), then in succesion, input torque to the arm, τ, is calculated by PD controller as follows,

$$\tau_1 = d_p(\theta_{1d} - \theta_d) + d_d(\dot{\theta}_{1d} - \dot{\theta}_d) \quad (7)$$

Next, the indivisual g_j^i have to be evaluated by the following fitness function,

$$f_j^i = \int_0^H x_j^i(t)dt \quad (8)$$

, where the H means trial time to evaluate the motion and, the $x_j^i(t)$ represents the solution of Eq.(1) when the input torque τ_1, generated by g_j^i, is exerted. The initial condition is shown by Fig.1, that is the position of mobile manipulator is the bottom of the wave and the arm is upright. The function of Eq.(9) takes bigger value when the mobile manipulator rides over waved road and travels to plus direction.

3.2 Search Simulation

We prepared wave road $z = -0.1 \cos(4x)$, and the GA searched best swinging function of arm for riding over the waved road as fast as possible, from the initial position $[x, y] = [0.0, -0.1]$. The parameters of GA are as follows;

population number $p=10$, 2-points crossover rate $c(\%)=50$, selecting rate $s(\%)=60$ and mutation rate $z(\%)=30$. In order that the GA can search the optimum solution, the parameters mentioned above should be set appropriately. The above parameter values obtained through preliminary experiments preserve the fastest convergence speed in GA process to solution. In the following simulations, the length of the gene is 6 bits and m=1, that is, T, a_0, a_1, b_1, therefore, the binary length of chromosome is 24 bits.

The motion results caused by the swinging motion of the mounted link, which is got by GA searching, are shown in Fig.3 - 6, which are the results when the best indivisual, which is the result found after 95 times generations, is used to determine the arm's swinging motion.

Figure 3 presents time responce of position x, and Fig.4 and 5 do also time responce of position z and arm's angle θ_1. Figure 6 expresses the trajectory of the mobile robot by dots, which means the position at descrete time. Used Fourier coefficients are $T = 1.75$, $a_0 = -0.875$, $a_1 = -0.25$, $b_1 = -2.25$. From Fig.3 and 4, it can be seen that the mobile manipulator rode over three hills at time $12[s]$.
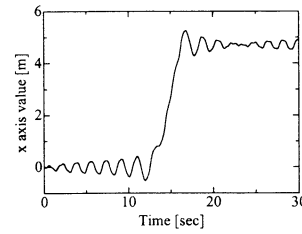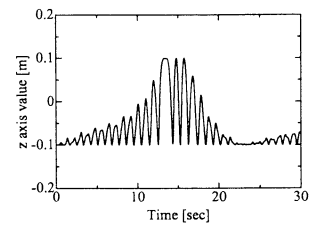
Fig. 3: result x   Fig. 4: result z
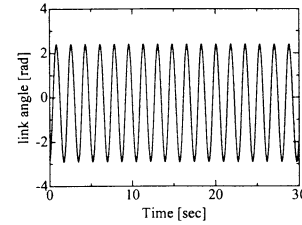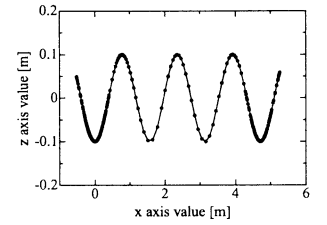
Fig. 5: result link   Fig. 6: result trajectory

4 Conclution

It has been checked by simulation that a mobile manipulator found how to ride over waved road by giving sample motions to the robot in simulations, which seems that the mobile manipulator might possess some kind of motion intelligence. Next, we will try to check it using real experimental machine.

references

[1] M.Minami, T.Asakura, T.Hatano: A study of Mobile ManipulatorTraveling On Irregular Terrain; J. of JSME, 615-63 (C), pp.235-275, 1997.

[2] M.Minami, T.Asakura, M.Asakura, A Tamamura: A Machine Learning Method Mobile Robot with Mounted Link to Utilize Dynamical Interfered Motion and Nonliner Friction; J. of RSJ, 19-7 , pp.97-104, 2001.

Concurrent Map Building and Localization Using the Quantification in the Mobile Robot Navigation

Kyoung-Sik Yun
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea

Jin-Woo Park
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea

Jang-Myung Lee
Dept. of Electronics Eng.
Pusan National University
Pusan, 609-735, Korea

{shadow, jinupark, jmlee}@pnu.edu

Abstract

In this paper, we suggest a method for the performance improvement in the calculation of the position of a mobile robot, which is crucially needed for the concurrent map building and localization. The mobile robot is designed for operating in a well-structured environment that can be represented by planes, edges, corners and cylinders in the view of structural features. In the case of ultrasonic sensors, these features have the range information in the form of the arc of a circle that is generally named as RCD (Region of Constant Depth). Using the exclusive characteristics of the ultrasonic sensors, the simulated RCD from the modeled sensor data generation is matched with the real RCD from the measured sensor data. In this step, we define QOMB(Quantity Of Map Building) in order to reduce the calculation time by notifying the flag of the localization procedure. In addition to that, the localization has inevitable uncertainties in the features and in the robot position estimation. So, we define a physically based sonar model and employ an extended Kalman filter to estimate position of the robot. By redefining the uncertainty ellipsoid in a new point of view, QOL (Quantity Of Localization) is introduced in this paper. Finally, QOMB and QOL show the plausibility in the concurrent map building and localization through experiments with the above mentioned mobile robot system.

Keywords : Map Building, Localization, Mobile Robot, Autonomy, Sonar sensor, Matching, Extended Kalman Filter

1. Introduction

In Map Building, ways of representing the map are divided into 2 methods [1]. One is grid representation method and another is feature representation method. Since the former uses the movable space between the robot and walls and obstacles, it is desirable for obstacle avoidance or autonomous local navigation. But, it is difficult to include the accurate information that is used for robot localization process. The latter is more preferable to the former in the global path planning or localization, however it requires the real-time computation load for the movable space. In general, with the help of rapid improvement of micro processor, the latter is more preferable.

As well as Map Building, the localization is fundamentally required for the mobile robot navigation, but it does not yet have the creditable solution [2]. The localization is largely classified into 2 categories: the relative localization and the absolute localization. The relative localization called dead reckoning method uses the distance, velocity, angular velocity during the movement based on the known position. It is subject to the sensor data, so it requires the absolute localization process periodically. Different of the relative localization, the absolute localization uses the geometric information of the distance and angle from the features of environment with the known map that if well structured, consist of the modeled components of planes, corners, edges and cylinders.

In this paper, we focused our endeavor on the concurrent map building and localiza tion by adopting the feature representation method and absolute localization method for the robot autonomy. And in the midst of concurrent processing, through the quantification for each process, we showed the performance improvement and systematic relatio nship between the map building and localization.

In section 2 we present the method of environment acknowledge using RCD of the ultrasonic sensors. Section 3 contains the localization from the map built during the environment acknowledge step. In section 4 we describe the defined quantities, QOMB and QOL. Section 5 includes impact of the defined quantities through the experiment and finally section 6 includes the conclusions from this research and our future work.

2. Environment Acknowledge using RCD

It is possible to represent the well-structured indoor environment with the modeled components like planes, edges, corners and cylinders. Because of the wide angle opening characteristics of the ultrasonic sensors,

environment elements can be characterized as the circular arc, RCD (Region of Constant Depth)[3] as shown in Fig. 1. Plane elements are represented by $P_L = (p_R, p_\theta, p_V)$, p_R is the shortest distance from the line segment in the plane to the origin in the global coordinates. p_θ is the angle between the line segment and the global x coordinate, p_V represents the valid region of the line segment of plane by the value among 1 and -1. With the similar way, cylinder is represented by $P_{CYL} = (p_x, p_y, p_R)$ with the center point p_x, p_y, the radius p_R and corner and edge are represented by $P_C = (p_x, p_y)$, $P_E = (p_x, p_y)$ respectively.
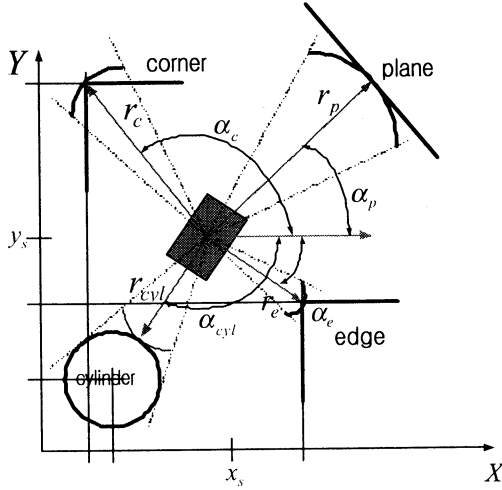
Fig. 1. RCD types of the environment elements

For the comparison between the real sonar data and simulated sonar data of the known environment element, we generated the simulated sonar data using the simple model of the ultrasonic sensor characteristics [3]. And for acquiring the real sonar data, we designed the rotating structure with one ultrasonic sensor as shown in Fig. 2.
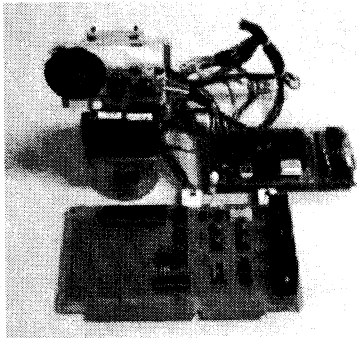
Fig. 2. Rotating structure with ultrasonic sensor

3. Localization from the sensor data and map

When the mobile robot starts from the known initial point, in the view of dead reckoning, it collects the data from the motor encoder and angle potentiometer. And then the extracted position information is used to generate virtual map with the simulated RCDs. These simulated RCDs are used together with real RCD data for map matching process.

In other words, when the real sonar data is acquired at time step k, the position is estimated for the most similar pattern of virtual RCD and based on that position, the virtual RCD pattern is predicted at time step k+1 and then the real RCD pattern at time step k+1 is compared with the virtual RCD pattern. This routine is continuously processed in chain type. Fig. 3 shows this process.
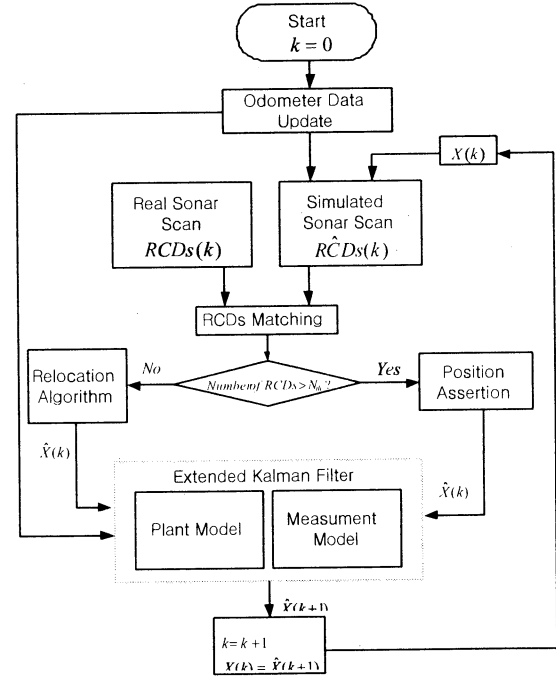
Fig. 3. Localization process chained with map building

In the localization process, we adapted the Extended Kalman Filter. The position and angle of the robot can be described $x(k) = [x(k), y(k), \theta(k)]^T$ as shown in Fig. 4. Initially the robot starts from the known position with the prior modeled information about the geometric environment features $\{p_t \mid 1 \le t \le n_T\}$ and acquires the measured data $z_j(k+1)$ every sampling time. The purpose of the EKF for the localization is to compute the updated estimate $\hat{x}(k+1 \mid k+1)$ from the information of environment features p_t and measured sensor data $z_j(k+1)$
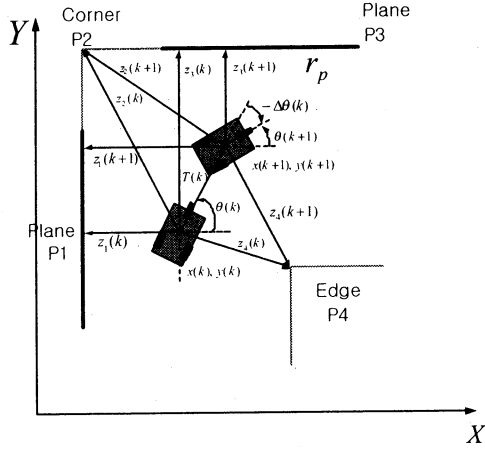
Fig. 4. Localization using measured and predicted data

Generally Kalman Filter is based on 2 models, plant model and measurement model.

Plant model

Control input $\mathbf{u}(k)$ and disturbance $\mathbf{v}(k)$ determines the robot configuration as shown in (1).

$$\mathbf{x}(k+1) = f(\mathbf{x}(k),\mathbf{u}(k)) + \mathbf{v}(k) , \quad \mathbf{v}(k) \sim N(0,\mathbf{Q}(k)) \qquad (1)$$

where $\mathbf{u}(k) = [T(k), \Delta\theta(k)]^T$ and $T(k)$ is the moved distance and $\Delta\theta(k)$ is angular change. $\mathbf{v}(k) \sim N(0,\mathbf{Q}(k))$ is the gaussian noise with zero mean covariance $\mathbf{Q}(k)$. Finally, $f(\mathbf{x}(k),\mathbf{u}(k))$ is state transition function as shown in (2).

$$f(\mathbf{x}(k),\mathbf{u}(k)) = \begin{bmatrix} x(k) + T(k)\cos\theta(k) \\ y(k) + T(k)\sin\theta(k) \\ \theta(k) + \Delta\theta(k) \end{bmatrix} \qquad (2)$$

Measurement model

Measurement model relate the measured sensor data robot position with geometry of environment features. This is explained by (3)-(6).

$$\mathbf{z}_j(k) = \mathbf{h}_{st}(\mathbf{x}(k),\mathbf{p}_i) + \mathbf{w}_j(k), \quad \mathbf{w}_j(k) \sim N(0,\mathbf{R}_j(k)) \qquad (3)$$

where $\mathbf{h}_{st}(\mathbf{x}(k),\mathbf{p}_i)$ is measurement function and $\mathbf{z}(k)$ is measurement data, $\mathbf{x}(k)$ robot position and feature vector $\mathbf{p}_i(k)$.

plane:

$$\mathbf{h}_P(\mathbf{x}(k),\mathbf{p}_i) =$$
$$p_v[p_R - (x(k) + x_s' \cos(\theta(k)) - y_s' \sin(\theta(k)))\cos(p_\theta)$$
$$- (y(k) + x_s' \sin(\theta(k)) + y_s' \cos(\theta(k)))\sin(p_\theta)] \qquad (4)$$

corner and edge:

$$\mathbf{h}_{C,E}(\mathbf{x}(k),\mathbf{p}_i) =$$
$$[(p_x - x(k) - x_s' \cos\theta(k)) + y_s' \sin(\theta(k)))^2$$
$$+ (p_y - y(k) - x_s' \sin(\theta(k)) - y_s' \cos\theta(k)))^2]^{1/2} \qquad (5)$$

cylinder:

$$\mathbf{h}_{CYL}(\mathbf{x}(k),\mathbf{p}_i) =$$
$$-p_R + [(p_x - x(k) - x_s' \cos\theta(k)) + y_s' \sin(\theta(k)))^2$$
$$+ (P_y - y(k) - x_s' \sin(\theta(k)) - y_s' \cos(\theta(k)))^2]^{1/2} \qquad (6)$$

4. QOMB and QOL

Most of localization problems have been computed in off-line using the environment features and data from the robot. This shows on-line computation complexity and load is larger in real experiments because of iterative method from nonlinearities in equations. So, some criterion value is required during the computation process when on-line experiments are executed in real time. Therefore, specifically we propose the method of quantifications that make the process to get out of iterations.

QOMB (Quantity Of Map Building):

$$N = \frac{N_{matched}(\mathbf{RCD}_{observed}, \mathbf{RCD}_{simulated})}{N_{th}} \qquad (7)$$

where $\mathbf{RCD}_{observed}$ is measured RCD and $\mathbf{RCD}_{simulated}$ is simulated RCD based on predicted position and N_{th} is the threshold number of specific local environment features in the step of map building.

In the case of $N \geq 1$, this has much effect on reducing the corresponding covariance in the step of localization and $N < 1$ represents the low reliability of map building so densely sampling process is executed. Engaged with this quantification, in the step of localization, current estimate $\hat{\mathbf{x}}(k+1|k+1)$ based on sensor data in time step k+1 is compared with the former estimate $\hat{\mathbf{x}}(k+1|k)$ in time step k.

$$L = \begin{bmatrix} \hat{x}(k+1|k+1) - \hat{x}(k+1|k) + C_x(k) \\ \hat{y}(k+1|k+1) - \hat{y}(k+1|k) + C_y(k) \end{bmatrix} \qquad (8)$$

From (8), we can get the error ellipsoid of the position of the robot [4]. Here, we define the quantification of localization by computing the multiplication (S_{ellip}) between the largest axis and the shortest axis. This is again compared with the 2-dimensal size of the robot (S_{robot}). The case of $S_{ellip} > S_{robot}$ notifies the conversion of relocation process, $S_{ellip} < S_{robot}$ shows the plausibility to proceed to the next map building process.

5. Experiment

The following experiments show the difference between the existing method and proposed method. Fig. 6 is monitoring part in the view of observer and Fig. 7 is the monitoring part in the view of the robot itself under experiment with the mobile robot, RRC-AGV shown in Fig. 5.
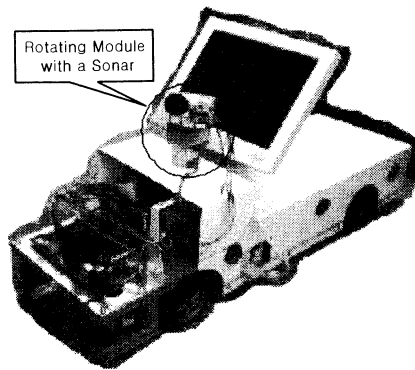
Fig. 5. Mobile robot, RRC-AGV
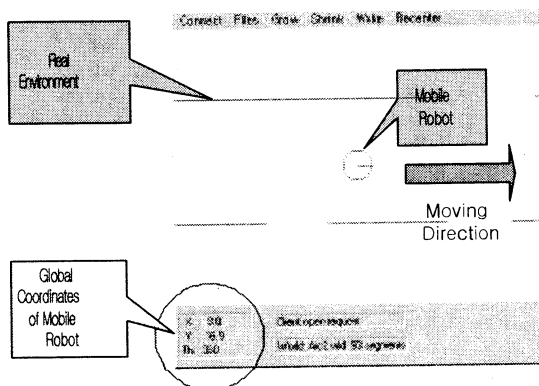
Fig. 6. Global monitoring of the robot for observer
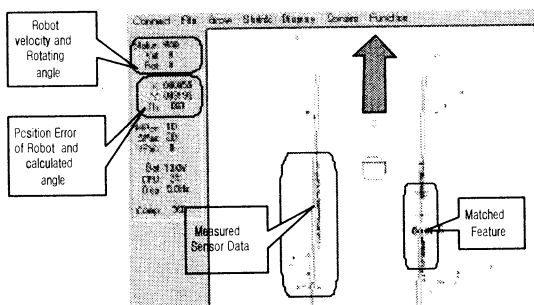
Fig. 7. Local monitoring of the robot itself

In the midst of moving from start point to target point under the same condition, compared to the case without considering the proposed quantification, the mobile robot moves smoothly to the target point without getting lost.
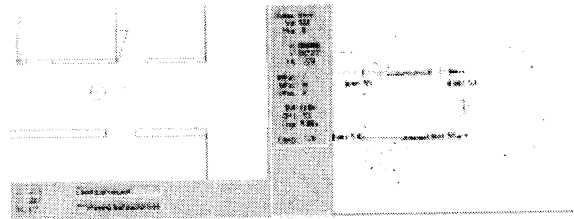
Fig. 8. When applying the QOMB and QOL
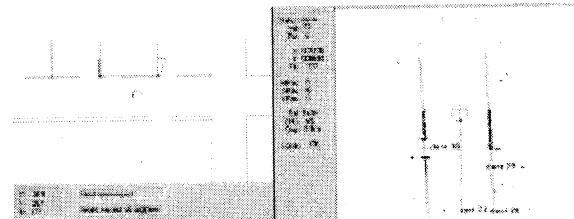
Fig. 9. When not applying the QOMB and QOL

As shown in Fig 9, it happens that the robot gets lost or build map incorrectly without considering QOMB and QOL.

6. Conclusion

To reduce the computation complexity in the steps of localization and map building, some criterion method is required then we proposed 2 quantification criteria that make each process to get out of the respective iteration procedure. QOMB and QOL show the plausibility in the concurrent map building and localization for the above mentioned mobile robot system. Furthermore, by applying this concept to the robot systems with various sensors, we will study the generality of this concept.

References

[1] T.Tsubouchi, "Nowadays Trends in Map Generation for Mobile Robots," in Proc. of IEEE/RSJ int. Conf. on Intelligent Robots and Systems, pp. 848-833, 1996.

[2] Borestein, J. and Feng, L. 1996c, "Where am I ? Sensors and Method for Autonomous Mobile Robot Positioning - 1995 Edition".

[3] John J. Leonard and Hugh F. Durrant-Whyte. "Direct Sonar Sensing for Mobile Robot Navigation," Kluwer Academic Publisher

[4] Y. Nakamura, Advanced Robotics : Redundancy and Optimization, Addison-Wesley, 1991.

[5] Jean-Paul Laumond, Robot Motion Planning and Control, Springer, 1998

[6] W. D. Renken, "Concurrent Localization and Map Building for Mobile Robots Using Ultrasonic Sensors", Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and System, Yokoham, Japan, July 26-30, pp. 2192-2196, 1993.

Map Building for Mobile Robots with Ultrasonic Sensors Using Neural Networks

Chang-Hyun Kim and Ju-Jang Lee

Department of Electrical Engineering and Computer Science

Korea Advanced Institute of Science and Technology

373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Korea

TEL:+82-42-869-3432,5432, FAX : +82-42-869-3410

E-mail : sunnine@odyssey.kaist.ac.kr, jjlee@ee.kaist.ac.kr

Abstract

In mobile robot application, many map building algorithms using ultrasonic sensors were introduced. In this paper, a new map building method using neural networks is proposed. In indoor environments, the ultrasonic sensor system has some uncertainties in its data. To overcome this situation, a neural network approach is used. Basically we use an algorithm based on grid-based map building method. The update grid size is changeable according to distance data of sensors. Also target differentiation method is implemented using fewer ultrasonic sensors. The reflection wave data patterns are learned using neural networks. So the targets are classified to plane, corner and edge roughly that frequently occur in indoor environment. We construct our own robot system for the experiments and some experiments are carried out to show its performance.

1 Introduction

In order to navigate autonomously and perform useful task, a mobile robot needs to know the environmental map. Doing this job, the simplest case is that the map is constructed by human. But in most cases, this method is not available. So the mobile robot must construct the map itself using its own sensors.

Many Sensors, such as vision, infrared, laser sensor and so on, are used to recognize the environment. Among these sensors, ultrasonic sensors are most popular because it has advantages of low cost, low power, small size, and simple hardware implementation.

Basically to present the map, grid-based mapping using *occupancy grid* is introduced by Elfes [1]. Ultrasonic sensor data is used to update the probability of cell based on Bayesian rule in this method. To reduce computational burden, *certainty grid* method that store a single number in a cell present the cell

is occupied is proposed [2]. Global map can be obtained integrating local maps successively. Therefore, the construction of local map around the robot is the first job.

Because of the uncertainty in ultrasonic sensor data, to obtain the exact local map is quite difficult. Many researches about this topic were carried out. Santos divided the area around the robot into some quite big blocks and train the patterns by neural networks [3]. Toledo uses the grid-based method and neural networks [4]. The output of the network is the occupancy probability of the point. Also many feature differentiation method were proposed. They try to differentiate the circumstances into planes, corner and edges. The method using more than one sensors are introduced [5] [6]. Peremans utilize three ultrasonic sensors, one transmitting and three receiving. To increase the differentiation perfomance, the amplitude of the reflection wave is very useful. But this method is very sensitive to environment conditions and seldom used.

In this paper, the method build the local map using neural networks is proposed. We develop the robot system for experiment. It has several ultrasonic sensors. We train a neural network to detect some features seen frequently in indoor environment from data of ultrasonic TOF(time of flight) information.

2 Robot Model

Robian, the robot constructed in our laboratory, is shown in Fig. 1. It has 30 cm diameter circular base and two wheels enable it to move forward or backward, and rotate. *Robian*'s wheels are equipped with shaft encoders that measure the distance moved and the rotation angle of a turn. It is also equipped with a ring of 12 ultrasonic sensors and gyroscope for experiment. We use Polaroid ultrasonic sensors (600 and 6500 se-

ries ranging unit) at center frequency $f_0 = 50$ kHz. Ultrasonic sensors is located at circular base keeping separation of 15 degree to each other symmetrically. Gyroscope can be used to reduce accumulative odometry error. A notebook PC was loaded as a processing unit that analyzes the acquired sensory information and controls the robot. All the controllers communicate the main processing unit through the universal serial bus(USB).
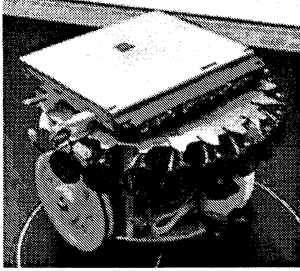
Figure 1: The Robot System (*Robian*)

3 Sensor Readings and Neural Networks

The ultrasonic sensors are used widely due to their simplicity, robustness and low cost. In the general application using ultrasonic sensor, an echo is produced when the transmitted ultrasonic wave encounters an object. Basically the ranging unit measure the elapsed time from transmission until the reflection wave comes back to the sensor and calculate the distance using the speed of sound in air (at room temperature, 343 m/s).

But there is an uncertainty in this system. The transmitted beam diverges with some beam angle. So we do not know the exact position of the obstacle. And even if specular reflection or multiple reflection occurs, the reflection wave can not come back or flies longer distance. Besides, diffuse reflections and fluctuations in the propagation medium cause the error in distance measure.

To reduce these uncertainty, neural networks are adopted. Neural network is powerful tool in classification and function approximation where the input is high dimensional and noisy. Also it has an advantage in computation time comparing with geometrical method. The neural network associated the sensor data with the appropriate feature in the environment.

The commonly occurring features in an indoor environment will be chosen for recognition. The neural networks are used to recognize 3 features, that is,

plane, corner and edge. But corner and edge are limited to the one has right angle. These features are shown in Fig. 2 The training patterns are collected where these obstacle is located with rotation and distance offset.
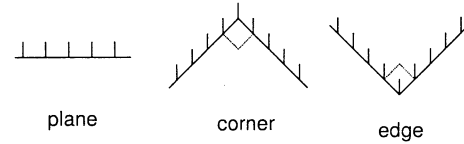
Figure 2: The Features Occurring in an Indoor Environment

The neural network employed is feed-forward multi-layer perceptron one and has one input, 2 hidden and 1 output layer. The input are five neighboring 5 sensor's TOF data at the same time and these data are normalized and re-scaled with the reading of the middle sensor. The neural network output represent the feature. Due to the beam angle and the arrangement of ultrasonic sensors, the area that 5 sensors can cover is proportional to the distance. So the basic feature map is builded using the normalized and re-scaled input. After that, the basic feature map is enlarged or reduced. So the outputs are the center of feature, the scale factor and angle of two branches of wall. The Fig. 3 will help understanding. Neural network is trained using error back-propagation algorithm. The neural network training is supervised.
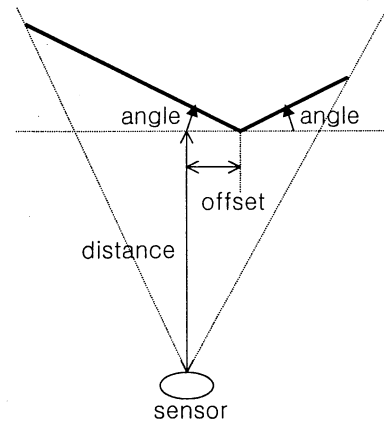
Figure 3: The Output of Neural Network

The neural networks can be used repeatedly due to the symmetry of arrangement. Once the neural network is trained, the network can be applied to another sensor array.

4 Experiment Result

The experiment was carried out in quite large room. The robot was placed appropriately to collect the training data. The distance between obstacle and robot is varying from 50 cm to 1.5m and the angle and position of robot has 5 steps respectively.

At first, the one sensor array is tested. Due to the generalization property of neural network, it shows quite good recognition ability. It also make a view of features that was not trained (does not have right angle) approximately like in Fig. 4.
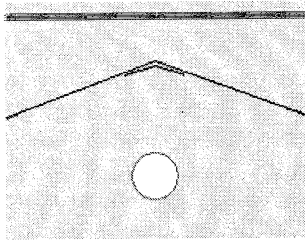
Figure 4: Generalized Property of Neural Network

Next, all sensors data are used. The neural network is used repeatedly and the features built is mainly shown in front of the middle one of input sensors. This is also shown in Fig. 5. But this map is more accurate if the obstacles are quite big. If obstacle is too small to reflect ultrasonic wave and received from sensors, the system can not recognize that. In experiments, filtering of sensor data and some heuristics about exclusion of specular reflection help building more accurate map.
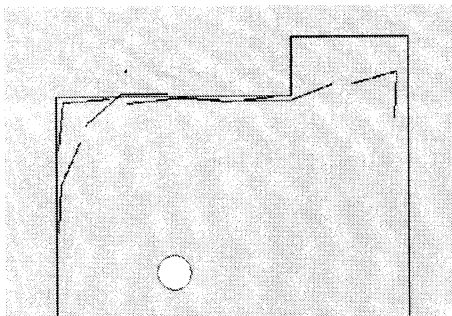
Figure 5: Local Map using All Sensor Data

5 Conclusion

In this paper, we presented a neural network based technique that enable the robot differentiate the features. With the only TOF information of ultrasonic sensors and neural network, we have reduced the uncertainty in data and computation time. The key point is that the size of update map region is proportional to the distance with defining the input of neural network properly and due to the symmetry the neural network can be re-used. With smaller efforts, the robot can differentiate the features into plane, corner and edge roughly.

However, the global map building has not been take into account in this paper. Future works will be the researches about navigation with low odmetry error and build a good global map using constructed system.

References

[1] A. Elfes, "Sonar-Based Real-World Mapping and Navigation," *IEEE J. Robotics and Automation*, Vol. RA-3, No.3, pp. 249-256, 1987.

[2] J. Borenstein and Y. Koren, "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE Trans. on Robotics and Automation*, Vol. 7, No. 4, pp. 535-539, 1991.

[3] V. Santos, J. Goncalves and F. Vaz, "Perception Maps for the Local Navigation of a Mobile Robot: a Neural Network Approach," *IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 2193-2198, 1994.

[4] F. J. Toledo, J. D. Luis and L. M. Tomas, "Map Building with Ultrasonic Sensors of Indoor Environments Using Neural Networks," *IEEE Int. Conf. on Systems, Man, and Cybernatics*, Vol. 2, pp. 920-925, 2000.

[5] B. Barshan and R. Kuc, "Differentiating Sonar Reflections from Corners and Planes by Employing an Intelligent Sensor," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6, pp. 560-569, 1990.

[6] H. Peremans and J. Van Campenhout, "Tri-aural perception on a mobile robot," *IEEE Int. Conf. on Robotics and Automation*, Vol. 1, pp. 265-270, 1993.

[7] B. Barshan, B. Ayrulu and S. W. Utete, "Neural Network-Based Target Differentiation Using Sonar for Robotics Applications," *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 4, pp. 435-442, 2000.

Study on calibration of stereo cameras for a mobile robot prototype

Jiwu Wang   Masanori Sugisaka

Department of Electrical and Electronics Engineering,
Oita University, 700 Dannoharu, Oita 870-1192, Japan
Email: wangjiwu_jf@sina.com.cn,   msugi@cc.oita-u.ac.jp

Abstract: Stereovision is an effective technique to determine the 3D position of objects from two or more simultaneously views of the scene. Camera calibration is a central issue for positioning objects in a stereovision system. After calibration, recovering 3D structure from 2D images becomes simpler. It is usually carried out by calibrating each camera independently, and then applied geometric transformation of the external parameters to find out the geometry of the stereo setting. Based on the ideal pinhole model of a camera, formulas to calculate intrinsic parameters that specify the camera proper characteristics, and extrinsic parameters that describe the spatial relationship between the camera and the world coordinate system were given. Finally, the calibration results of our CCD video cameras were given.

Key Words: Camera calibration, Intrinsic parameters, Extrinsic parameters, Pinhole model

1.Introduction

For mobile robots, stereovision, as a passive sensor, is a reliable and effective method to get range information from environments, and it can be easily integrated with other sensors. More and more robots are designed and implemented with a stereovision system. Camera calibration is a process of relating the ideal model of a camera to the actual physical device and of determining the position and orientation of the camera with respect to a world reference system. Calibration is important for accuracy in 3D reconstruction. In particular it is a critical task for the stereovision analysis.

In this paper, based on the ideal pinhole model of a camera, the principle to calculate intrinsic and extrinsic parameters of stereo cameras was given and a corresponding experiment was built and carried out. Although in practical situations, real cameras deviate the pinhole model, such as lens distortions, imaged rays that are not necessary intersected at a point, and changes of focus, etc. they are only the factors that influence the precision of camera calibration and will be considered in the following further research. At last, the experiment results of camera calibration were given.

2. The principle to calculate intrinsic and extrinsic parameters of stereo cameras

Camera parameters are divided into intrinsic parameters and extrinsic parameters. Intrinsic parameters are used to specify the camera characteristics. These parameters are: 1) focal length, the distance between the camera lens and the image plane; 2) the location of the image center in pixel coordinates; 3) the radial distortion coefficients of the lens. Here, the axis of camera is thought as being perpendicular, so the angle between the optical axes is not considered. Extrinsic parameters are used to describe the relative spatial position and orientation between the camera and the world. They are the rotation matrix and translation vector determining the transformation between the camera and world reference frames.

Firstly, we assume that the position and orientation of the camera is canonical. The intrinsic transformation is determined by the camera design and construction, including image sensor resolution

and optical components. As shown in Fig.1, the coordinate system (C, X, Y, Z) is called the standard coordinate system of a camera, in which Z axis is along the optical axis and origin C is at the center of projection. Point M represents any point of the target object in the standard coordinate system. Point m indicates the corresponding position of point M in the image plane. The coordinate system (c, x, y) is used to identify any points in the image plane, whose origin is at the intersection of the optical axis and x and y axes are parallel to X and Y axes respectively.
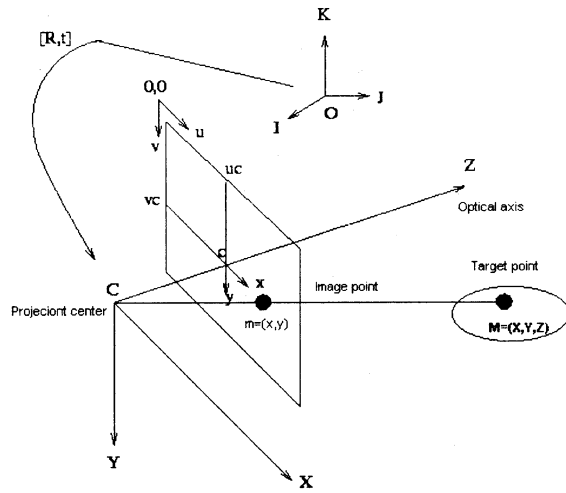
Figure 1 The illustration of the pinhole principle for camera calibration

Based on the pinhole model of a camera, the relationship between point M and m can be expressed as follows in homogeneous coordinates:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dfrac{f}{Z*(pixel\_width)} & 0 & u_c \\ 0 & \dfrac{f}{Z*(pixel\_height)} & v_v \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

where, (u, v) are the coordinates of point m in the homogeneous coordinate system; (u_c, v_c) are the coordinates of the origin c; f is the focal length of the camera, the distance between the camera lens and the image plane. Then intrinsic parameters f, u_c, v_c can be calculated.

Till now, we have not considered lens distortion of a camera. Due to the lenses are not flat surfaces,

the world points are not projected on a plane, but in a surface which can be considered to be spherical. Therefore, lens distortion of a camera is another factor that has to be considered in the positioning from 2D images to 3D space.

Here, radial distortion is assumed to be satisfied. Given (u, v) are the ideal image coordinates of a point, (u', v') are the distorted image coordinates, the relationship between (u, v) and (u', v') is

$$u' = u + k(u - u_c)(x^2 + y^2)$$
$$v' = v + k(v - v_c)(x^2 + y^2)$$

where k is the distortion coefficient.

In general, the three dimensional coordinates of a point will not be the same as the frame whose origin is the center of the projection and whose Z axis lies along the optical axis. Therefore, we should build the relationship between any specified world coordinate system and the standard coordinate system. In other words, it is used to describe the orientation and translation change with respect to the specified world coordinate system. These parameters are called extrinsic parameters. They can be calculated as the same as the transformation of different coordinate systems.

Let the symbol T_O^C denote the homogenous transformation that transfers the coordinates of a point p^O defined with respect to frame O to a coordinate representation p^C with respect to frame C (the standard coordinate system of a camera), the relationship between two coordinate system can be expressed as:

$$p^C = T_O^C p^O$$

In fact, you can build more coordinate system to conveniently express the locations of target objects. However, the transformation between any two coordinate systems is the same as the above formula.

3. Experiments

Before the calibration experiments, you must make pattern first. In our experiments, pattern is a part of a flat checkerboard with sizes of 8×6 black and white squares on the white background, as shown in figure 2.
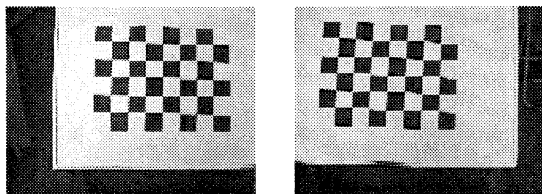
Figure 2 Two pieces of image of the same pattern taken at the same time by left and right cameras respectively.

The first part of the calibration process is to carry out the distortion correction. This is because straight lines in the world coordinate system are not straight lines in the image after perspective mapping. It will bring errors when we locate a point from 2D image to 3D space.

In this undistortion mapping process, we transform the distorted image taken directly with a CCD video camera into an undistorted image based on the radial distortion coefficient of the lens. In order to illustrate conveniently, we mark the frame of the original image with white straight lines and draw the white straight central line at the same time, as shown in Fig.3 a). After undistortion mapping, we found the straight lines at the edges of the image have been curved, and the distance of each central line became shorten, as shown in Fig. 3 b). This phenomenon indirectly verified the undistortion transformation.

In order to further verified the difference between the original image and the undistorted image, we firstly subtract the original image from the undistorted image, as shown in Fig. 3 c), then we subtract the undistorted image from the original image, as shown in Fig. 3 d). From results of the comparison, we find the difference in the central part is very small, nearly to zero. The difference increased when the pixels are far away from the center. Some pixels were missed because the

subtraction was performed in two binary images. This is reasonable for distortion correction. However, from fig.3 b), we found the symmetry for the left part and right part is not good. One of the reasons is that we changed the original image sizes when we drew the mark lines. Another reason is that we assumed that the lens satisfied radial distortion. In fact, it is impossible.
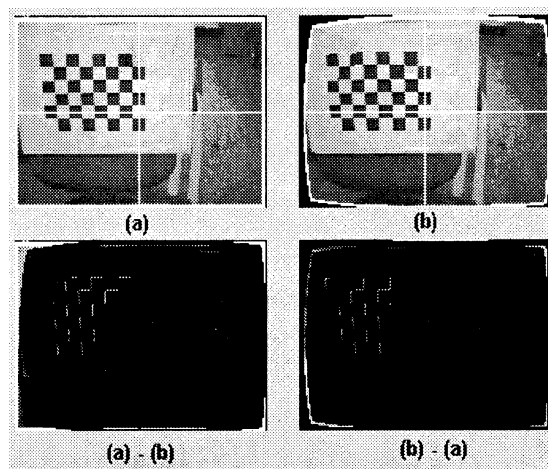
Figure 3 The illustration of image distortion problem in the camera calibration

Once the distortion pattern of a lens is known, we can use any image taken with that lens, undistorted it to make a new image in which straight lines in the world coordinate system will project to straight lines in the image. At this moment, the projection is a true perspective projection, and it is possible to characterize the intrinsic parameters of a lens.

When we calibrate the camera, the image processing has to be done first to find the inner corners of the checkerboard. In order to make the solutions converge and all corners of the checkerboard to be found, ten pieces of image of the pattern at different positions were taken for each camera respectively. The results for the intrinsic matrix for two cameras are:

Left Camera intrinsic matrix:

$$\begin{bmatrix} 223.30 & 0.00 & 90.25 \\ 0.00 & 222.45 & 63.81 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

Right Camera intrinsic matrix:

$$\begin{bmatrix} 223.41 & 0.00 & 86.89 \\ 0.00 & 204.05 & 62.84 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

In above matrices, the unit of parameter f, u_c and v_c is in pixels.

4. Conclusions

In this paper, the calibration of stereo cameras was studied, especially for the calculation of its intrinsic and extrinsic parameters. At the same time, the influence of lens distortion was explored too. There are many factors that influence the precision of the calibration, such as the change of focus, the angle between optical axis and lens's physical characters etc. If for general applications, maybe they do not need high precision calibration. Our hypothesis for camera calibration in this paper can satisfy their requirements. However, it will be of benefit to high precision calibration if the relationships between the precision of camera model and its influence factors can be built. The further tests will be carried on in the following research of positioning from 2D image to 3D space.

Reference

[1] Faugeras, O.D., Toscani, G. (1987), Camera calibration for 3D computer vision. Proc. International workshop on industrial applications of machine vision and machine intelligent, Silken, Japan, pp. 240-247

[2] Tsai, R.Y. (1987), A versatile camera calibration technique for high-accuracy 3D machine metrology using off-the –shelf TV cameras and lenses, IEEE journal of robotics and automation RA-3 (4): pp323-344

[3] Heikkila, J., Silken, O. (1996), Calibration procedure for short focal length off-the-shelf CCD cameras, Proc. 13th international conference on pattern recognition, Vienna, Austria: pp.166-170

A selective vision and landmark based approach to improve the efficiency of probabilistic localization

○ Andrey A. Loukianov   Masanori Sugisaka

Department of Electrical and Electronic Engineering,
Faculty of Engineering, Oita University
Dannoharu 700, Oita 870-1192, Japan.

Abstract

This paper presents a vision and landmark based approach to improve the efficiency of probability grid Markov localization. The proposed approach uses visually detected landmarks to define selected regions in the state space where the location probability density function is updated. Probabilistic landmark-based localization method and details of the map and robot perception are discussed. A technique to compute the update regions for selective computation and their parameters is introduced. Simulation results are presented to show the effectiveness of the approach.

Keywords: navigation, probabilistic localization, visual landmarks

1 Introduction and discussion

Robust and efficient navigation is the key issue in control and operation of autonomous mobile robots [2]. Localization is one of the principal problems in robust navigation, which task is to estimate robot location using sensor data. Reliable and effective localization method has to deal successfully with several issues. The localization approach must be able (1) to stably track current robot position, (2) to globally locate the robot without prior information on its whereabouts and (3) to autonomously recover from the errors in position estimation or incorrect previous information.

Probabilistic Markov localization approaches offer a good framework to deal with all described issues allowing to handle any degree of uncertainty in robot location. The robot's belief about its current posture is represented by a probability density function (PDF) which is spanned over all open environment space and updated by iterative Bayesian *predict-and-match* technique. In the environment the location PDF is usually approximated by a 3D probability grid [4] which can be very large in size for large environments and fine approximation resolution. Therefore, probability grids require large amounts of mem-

ory and significant computing power to represent and process them. It makes them inefficient and limits their use in real-time applications. This problem was addressed by several researchers who suggested to use selective computations [4] and dynamic environment representations [3] for general open-space surroundings or effective models for specific environments [6].

This paper presents a selective approach that improves the efficiency of grid-based probabilistic Markov localization algorithm without degrading its ability to successfully deal with all the issues of the localization problem. The approach makes use of visually detectable landmarks of several classes with known positions in the environment. The information about detected landmarks – their relative arrangement and combination of their types – is used to selectively update the posterior location PDF in an efficient way. This information allows to define the regions in the map for selective computations. Only in these regions the conditional perceptual model is computed and the posterior location PDF is updated, thus significantly reducing computational workload. When more than one landmarks are detected our approach becomes similar to triangulation localization methods. However it does not require the robot to detect three or more landmarks at a time to localize itself - the probabilistic framework allows the method to make use of any available landmark sightings to refine its belief about current position.

Generally, the vision system can detect landmarks more reliably in contrast to the ultrasonic or range sensors which are easily confused in dynamic environments. If we assume that a landmark can be reliably extracted from the camera image and classified, then we can avoid the violation of the Markov assumption about conditional independence of sensor readings.

The results of simulations are presented to show the effectiveness and robustness of the method. In particular, the solutions of the global localization and the so-called "kidnapped robot" problems are presented.

2 Landmark-based localization

A mobile robot's state can be directly defined by its pose vector $\boldsymbol{\xi} = [x_p \ y_p \ \vartheta]^T$. The environment can be represented then as a state space Ξ which spans over all possible robot postures in the given map. The map also contains a set of landmarks $\mathfrak{L} = \{l_1, \ldots, l_{N_L}\}$ whose positions in the map are known with some accuracy. In general, on t-th step of operation, the robot changes its state by executing a sequence of movements, denoted as a, during which the odometry is used to estimate a relative change in position. The robot also utilizes a video camera to detect a set of landmarks $\hat{\mathfrak{L}}_t(\boldsymbol{\xi}) = \{\hat{l}_1, \ldots, \hat{l}_{N_t}\} \subset L$. Each detected landmark is described by some parameters that allow to classify it and to estimate its location relative to the robot. If we assume that the motions and landmark detections are performed in sequence, then the localization problem can be stated in the following general terms:

$$\text{Estimate} \quad \boldsymbol{\xi}_t,$$
$$\text{given} \quad \hat{\mathfrak{L}}_t, \hat{\mathfrak{L}}_{t-1}, \ldots, \hat{\mathfrak{L}}_0, \text{ and } a_t, a_{t-1}, \ldots, a_0.$$

3 Markov localization method

According to Markov localization approach the robot's confidence for being at a definite location is expressed by a probability distribution over all state space $P(\boldsymbol{\xi}), \boldsymbol{\xi} \in \Xi$. To obtain a 'true' posture $\boldsymbol{\xi}^*$ one may use an appropriate statistical estimator, such as maximum likelihood estimator

$$\boldsymbol{\xi}^* = \text{argmax}_{\boldsymbol{\xi}} P(\boldsymbol{\xi}). \tag{1}$$

The distribution of probability density in $P(\boldsymbol{\xi})$ is conditioned on the history of movements and sensor readings based on the *Markov assumption* which, in general, does not hold for dynamic environments [4]. In other words, $P(\boldsymbol{\xi})$ is estimated recursively by applying Bayes formula with conditional probability distributions calculated for the current motion or visual data. For each motion $P(\boldsymbol{\xi})$ is updated as

$$P(\boldsymbol{\xi}_t) = \int_{\Xi} P_a(\boldsymbol{\xi}_t | \boldsymbol{\xi}_{t-1}) \, P(\boldsymbol{\xi}_{t-1}) \, d\boldsymbol{\xi}_{t-1}, \tag{2}$$

where $P_a(\boldsymbol{\xi}_t | \boldsymbol{\xi}_{t-1})$ is the conditional transition probability for a move a that changed the state by $\Delta\boldsymbol{\xi}$ from previous state $\boldsymbol{\xi}_{t-1}$ to $\boldsymbol{\xi}_t$. Equation (2) is integrated over all previous possible locations $\boldsymbol{\xi}_{t-1}$. When a set of visual landmarks $\hat{\mathfrak{L}}_t$ is observed the update equation becomes:

$$P(\boldsymbol{\xi}_t) = \eta \, P(\hat{\mathfrak{L}}_t | \boldsymbol{\xi}) \, P(\boldsymbol{\xi}_{t-1}), \tag{3}$$

where $P(\hat{\mathfrak{L}}_t | \boldsymbol{\xi})$ is the likelihood of observing $\hat{\mathfrak{L}}_t$ conditioned at $\boldsymbol{\xi}$, called sometimes *perceptual model*; and η is the normalization factor.

4 The map and robot perception

In Markov localization the state space (map) is usually represented by a 3D data grid which piecewise approximates $P(\boldsymbol{\xi})$, though more complex and dynamic representations also can be adopted [3]. Here we use the probability grid representation superposed by a 2D logical occupancy grid of the same resolution.

The proposed method makes use of visually detectable landmarks. Any environment feature (natural or artificial) can represent a landmark if it has known map coordinates $\boldsymbol{p}_l = [x_l \ y_l]^T$, can be captured by video camera, reliably recognized and classified. Robust and reliable recognition of the landmark in a video image helps to avoid the violation of the Markov assumption for dynamic environments.

For the better performance of landmark-based localization method the environment should contain many landmarks. Obviously, when the number of landmarks is large it is not always possible to make all of them unique. Here we assume that landmarks can be classified into several unique classes but within each class the landmarks are indistinguishable. Landmark classes correspond to various types of landmarks: markers of certain shape or color, stationary objects (door frames, corners, power sockets), etc.

A conventional single video camera with pan-tilt capabilities mounted on a rotating setup is used to detect landmarks. Each detected landmark is classified and its position relative to the robot is measured. A variety of image processing methods exists to accomplish recognition and classification tasks. The class of the landmark is used to assist in estimating relative position since the single camera cannot give necessary depth information. For that purpose we assume that all landmarks of the same type are located at the same height above floor level. This assumption gives the necessary depth constraint. So every observed landmark is described by the following parameters: its type and its relative position in polar coordinates $\boldsymbol{b} = [r_i \ \tau_i]^T$. For more accurate measurement the optical and kinematic parameters of the rotating camera have to be determined through a calibration procedure [5].

As in real world, we consider the robot perception and environment map to have some errors and inaccuracies. Here we assume that these errors may be represented by zero-mean normal distributions. The position $\boldsymbol{p}_l$ of any landmark l is given with some ac-

curacy expressed by variance σ_l^2 or covariance matrix $C_l = \sigma_l^2 \, \boldsymbol{I}^{2\times2}$. The relative position $\hat{\boldsymbol{b}}$ of observed landmark $\hat{l}$ is also inaccurately estimated. It is expressed by covariance matrix $\hat{C}_b$.

5   Selective computation approach

To improve the computational efficiency of Markov localization one may use selective computation techniques. One, fairly straightforward, way is to set a minimal probability threshold for the grid cell to be updated in Eq. (2) and (3). In addition to this we suggest to employ the information about types of observed landmarks and their local arrangement to update location PDF in (2) and (3) even more selectively - only in certain zones on the map, which we call the *update regions*. The examples of such update regions are shown in Fig. 1.
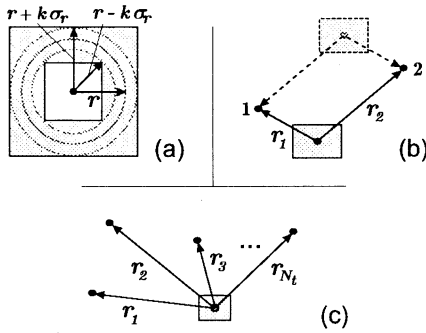
Figure 1: Examples of update regions calculated for (a) one, (b) two and (c) multiple observed landmarks.

Each update region is attached to one or more appropriate landmarks in the map. To select these appropriate landmarks the class of observed landmarks and the distance between their pairs are used. For example, when a single landmark is detected the update regions, similar to Fig. (1a), are attached to all landmarks of the same type as the detected one. For two or more landmarks the combination of their types and the distance similarity criteria $S < S_{max}$ condition are considered. The similarity criteria S is calculated as

$$S = |d - \hat{d}| / \sqrt{\sigma_d^2 + \hat{\sigma}_d^2},$$

where $\hat{d}$ and d are observed and map distances between the pair of landmarks, $\hat{\sigma}_d$ and σ_d are those distances uncertainties, calculated from C_l and $\hat{C}_b$.

The centers of update regions, $\boldsymbol{p}_r$, except for a simple case of a single landmark, are computed using triangulation techniques [1]. The size of update region is defined by the uncertainty given by covariance

matrix C_r. This matrix is computed differently in each case. For single detected landmark $C_r \in R^{2\times2}$ is

$$C_r = \hat{C}_b + \boldsymbol{J}\, C_l\, \boldsymbol{J}^T, \qquad (4)$$

where $\boldsymbol{J} = \partial\boldsymbol{p}_l / \partial\boldsymbol{b}$. For two or more detected landmarks $C_r \in R^{3\times3}$ can be obtained as

$$C_r = \sum_{i=1}^{N_t} \left(\boldsymbol{J}_{bi}\, \hat{C}_{bi}\, \boldsymbol{J}_{bi}^T + \boldsymbol{J}_{pi}\, C_{\sigma i}\, \boldsymbol{J}_{pi}^T \right), \qquad (5)$$

where $\boldsymbol{J}_{bi} = \partial\boldsymbol{p}_r / \partial\boldsymbol{b}_i^T$ and $\boldsymbol{J}_{pi} = \partial\boldsymbol{p}_r / \partial\boldsymbol{p}_{li}^T$.

Before updating the location PDF in Eq. (3) the conditional probability distribution $P(\hat{\mathfrak{L}}|\boldsymbol{\xi})$ has to be defined. This probability distribution is calculated for each update region using the following formulas

single landmark $\quad p(\hat{\mathfrak{L}}|\boldsymbol{\xi}) = \phi_2\left(\Delta\boldsymbol{b}, C_r\right),$ (6)

otherwise $\quad p(\hat{\mathfrak{L}}|\boldsymbol{\xi}) = \phi_3\left(\Delta\boldsymbol{p}_r, C_r\right),$ (7)

where ϕ_2 and ϕ_3 are the 2D and 3D normal Gaussian distributions defined in polar and cartesian reference systems, respectively. The similarity factor S, discussed above, may be used to weight the likelihoods in Eq. (7). The overall conditional perceptual model for any given location $\boldsymbol{\xi}$ is defined by the expression

$$P(\hat{\mathfrak{L}}|\boldsymbol{\xi}) = 1 - \prod_{N_u} \left(1 - p(\hat{\mathfrak{L}}|\boldsymbol{\xi})\right), \qquad (8)$$

where N_u is the number of update regions overlapping the location $\boldsymbol{\xi}$ in 3D space.

Since $P(\hat{\mathfrak{L}}|\boldsymbol{\xi})$ is zero outside any update region the result of Eq. (3) is also zero in these locations. Therefore the location PDF needs to be updated in the match step (3) only in locations spanned by at least one update region in the map. Other locations in the probability grid are assumed to have zero probability and not accessed by the algorithm. On the next $(t+1)$ iteration the same update regions can be used to selectively designate possible locations $\boldsymbol{\xi}_{t-1}$ for the prediction step update equation (2). Because of this the overall computational workload is significantly reduced.

6   Simulation results

To test the robustness and efficiency of the proposed approach two simulations of different localization problems were performed. The global and so-called 'kidnapped robot' localization problems were considered. The robot's motions are modelled to be inexact, so the uncertainty related to odometry predicted location is growing along the course of motion. This uncertainty defines conditional transition
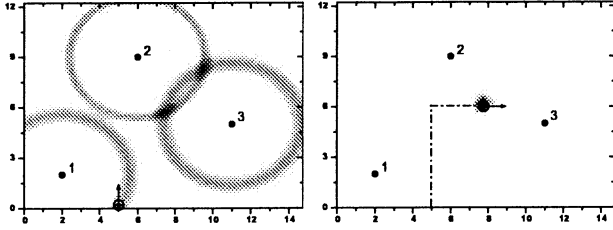
Figure 2: Global localization problem simulation: initial (after first landmark detection) and final robot (after 4 iterations) location beliefs.

Figure 3: 'Kidnapped robot' localization simulation: initial erroneous and final correct (after 5 iterations) robot's beliefs.

probability $P_a(\xi_t|\xi_{t-1})$ in Eq. (2) and its calculation is not considered here to conserve the paper space. Simulation map 15×12.5 m contained three indistinguishable landmarks that can be visually detected by the robot. The landmarks positions are known with limited accuracy: $\sigma_{l1}^2 = 0.08$ m^2, $\sigma_{l2}^2 = 0.03$ m^2, $\sigma_{l2}^2 = 0.1$ m^2. The map was represented by a probability grid with linear resolution of 0.25 m and angular resolution 15° that contained totally 72000 possible states. In both simulations the robot was able to observe only one landmark at a time, so the area of update regions was the largest possible.

In the first simulation the proposed method was used to solve the global localization problem which is to estimate robot location in the map without knowing initial position. The results of simulation for this problem are shown in Fig. 2. Simulation results for the so-called 'kidnapped robot' localization problem are shown in Fig. 3. The approach took four and five iterations in the first and latter problem, respectively, to converge the estimated posture to the true one. Only average of 5.3 percent of all states in the first problem and 5.6 percent of all states in the second problem were accessed and updated in (2) and (3).

7 Conclusion and remarks

In this paper we presented a selective vision and landmark based approach for effective Markov localization with probability grid approximation. The approach uses visually detectable landmarks of several unique classes which are undistinguishable within each class. Effectiveness is achieved by updating the probability grid only in a number of given update regions in the map. The update regions are defined using the information about types and relative arrangement of landmarks detected by the robot. The inaccuracies in robot perception and map information (landmark coordinates) are considered. The results of simulations show that the approach is effective and is able to solve all range of issues in the localization problem.
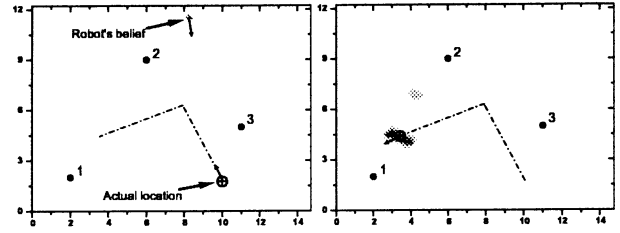
References

[1] M. Betke and L. Gurvits (1994), "Mobile robot localization using landmarks," In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, May, pp 135–142.

[2] J. Borenstein, B. Everett, and L. Feng (1996), *Navigating Mobile Robots: Systems and Techniques*, A. K. Peters, Ltd, Wellesley, MA.

[3] W. Burgard, A. Derr, D. Fox, and A. Cremers (1998), "Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach," In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[4] W. Burgard, D. Fox, D. Hennig, and T. Schmidt (1996), "Position tracking with position probability grids," In *Proc. of 1st Workshop on Advanced Mobile Robots*.

[5] J. Heikkilä and O. Silvén (1996), "Calibration procedure for short focal length off-the-shelf CCD cameras," In *Proc. of 13th Int. Conf. on Pattern Recognition*, Vienna, Austria, pp. 166–170.

[6] S. Thiebaux and P. Lamb (2000), "Combining kalman filtering and markov localization in network-like environments," In *Pacific Rim Int. Conf. on Artif. Intelligence*, pp. 756–766.

[7] S. Thrun, D. Fox, W. Burgard, and F. Dellaert (2001), "Robust monte carlo localization for mobile robots," In *Artif. Intelligence* 128:1-2, pp. 99–141.

Colour-based landmark tracking for use in a mobile robot indoor navigation

Masanori SUGISAKA and Renping CHEN

Department of Electrical & Electronic Engineering, Oita University

700 Dannohara, Oita, 870-1192, Japan

Abstract: As the area of mobile robot indoor navigation is smaller, we can make an accurate map of this area, and can set up landmarks with specific colour in this area. So, we can use the landmark track for a mobile robot indoor navigation. In our case, we use the hue's stability of the same colour for tracking landmark. The azimuth angle of the mode (peak) of landmark colour probability distributions within a video scene as robot's move direction, in another word; robot is always moving toward the place that corresponding to the mode within a video scene.

Keyword: landmark tracking, HSV color system, and probability distribution

1. Introduction

This paper is part of a program that map-based indoor navigation of a mobile robot. The feature of indoor navigation is that the region of the robot moving is finite, so we can make an accurate topological map of this region, and can set up landmarks with specific colour in this region. In order to track the landmarks in real time, at the same time, other tasks are able to run, we therefore chose to use a nonparametric technique for climbing density gradients to find the mode (peak) of probability distributions called the mean shift algorithm. The mean shift algorithm is based on a nonparametric technique for climbing density gradients.

The mean shift algorithm operates on probability distributions. To track colored objects in video frame sequences, the color image data has to be represented as a probability distribution; we use color histograms to accomplish this. Color distributions derived from video image sequences change over time, so the mean shift algorithm has to be modified to adapt dynamically to the probability distribution it is tracking. The new algorithm that meets all these requirements is called the Continuously Adaptive Mean Shift (CAMSHIFT) algorithm.

Figure 1 summarizes the algorithm described below. For each video frame, the raw image is converted to a color probability distribution image via a color histogram model of the color being tracked (landmark's color). The center of the color object (landmarks) is found via

the CAMSHIFT algorithm operating on the color probability image. The current location of the mode (peak) of the landmarks are reported and the azimuth angle of the mode be used to set the robot's move direction, in another word; robot is moving toward the place that corresponding to the mode within a video scene. The process is then repeated for continuous tracking.
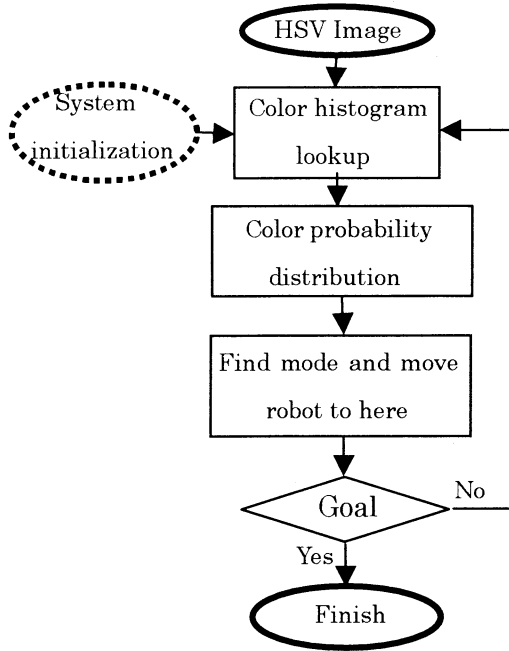
Figure 1: Block diagram of indoor navigation

2. Color Probability Distributions

2.1 Landmark Color Histogram

In order to use CAMSHIFT to track colored objects in a video scene, a probability distribution image of the desired color (landmarks color in our case) in the video scene must be created. In order to do this, we first create a model of the desired hue using a color histogram. We use the Hue Saturation Value (HSV)

color system to do it. Figure 2 shows the HSV color system.
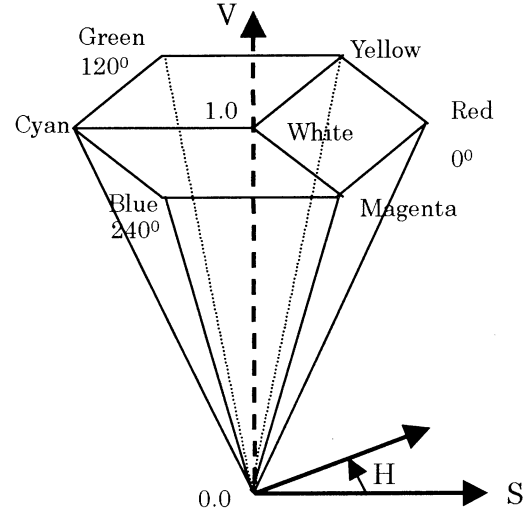
Figure 2: HSV color system

HSV space separates out hue (color) from saturation (how concentrated the color is) and from brightness. We create our color models by taking 1D histograms from the H (hue) channel in HSV space. We first choose a small landmark region in video frame as a sample, and separate out Hue of this region each pixel from Saturation and Value with HSV color system. For 8-bit hues, its range be created in discrete steps from zero to the maximum 255, then according to the scale between the sum of each hues value and the sum of pixel in this sample region, we create the landmark hue using a 1D histogram. We think that the hue of the biggest scale furthest denote the landmark's color. The biggest scale is converted to 1.0; other scales are converted to corresponding value with the same scale. Using this method, we get

the landmark color histogram. Figure 3 shows landmark color histogram.
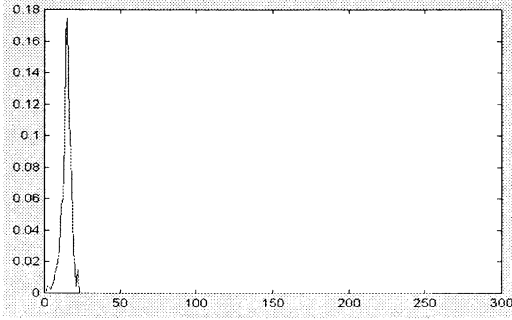
Figure 3a: hue value scale histogram
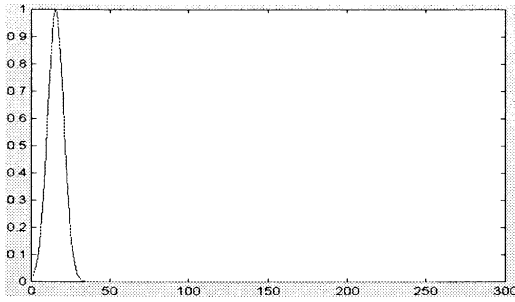
Figure 3b: landmark color histogram

2.2 Color Probability Distributions

The stored landmark color histogram is used as a model, or lookup table, to convert incoming video pixels to a corresponding probability of landmark image, so that we get the probability distribution image. Using this method, probabilities range in discrete steps from zero (probability 0.0) to the maximum probability pixel value (probability 1.0). For 8-bit hues, this range is between 0 and 255. In order to increase the accuracy of the landmark tracking, we convert the probability of landmark image to binary image. In this process, we ignore those pixels that are isolated and very small piece region. And then we track

landmark using CAMSHIFT on this binary image. Figure 4 shows a video image and its binary image of landmark probability image.
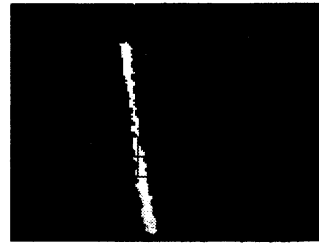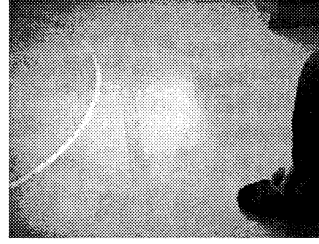
Figure 4: A video image and its binary image of landmark probability image

3. Landmark Tracking and indoor navigation

In order to track landmark, we first must compute the mode (peak) of the binary image (or probability of landmark image). The mode (peak) of the binary image is found as follows:

Find the zeroth moment

$$M_{00} = \sum_x \sum_y I(x, y)$$

Find the first moment fir x and y

$$M_{10} = \sum_x \sum_y x I(x, y)$$

$$M_{01} = \sum_x \sum_y y I(x, y)$$

Then the mode (peak) of the binary image is:

$$x_m = \frac{M_{10}}{M_{00}} \quad ; \quad y_m = \frac{M_{01}}{M_{00}}$$

The process of landmark tracking is continuously moving the mode (peak) of the binary image to the geometric center of the video frame. But then, for the indoor navigation of the mobile robot, we use that the robot moving toward the mode (peak) of the binary image instead of the moving of the mode (peak). Figure 5 shows the principle of the indoor navigation. l denotes the middle line of the video frame. θ denotes the azimuth angle of the mode. The robot is always moving toward the place that corresponding to the mode of the landmark color probability distributions in each video scene, thus accomplish the mobile robot automatic navigation's purpose.
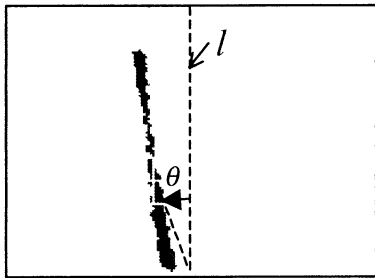
Figure 5: The principle of the indoor navigation

4. Conclusion

The method that color-based landmark tracking for use in a mobile robot indoor navigation is simple and efficient, and it yet works well in noisy environments. In the future, the mobile robot will move based the landmark tracking, and automatically arrived the place that the people designate.

5. References

[1] Gary R.Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," Intel Technology Journal Q2'98.

[2] Y. Cheng, "Mean shift, mode seeking, and clustering," IEEE Trans. Pattern Anal. Machine Intell. , 17:790-799, 1995.

[3] M. Isard, A. Blake, "Contour tracking by stochastic propagation of conditional density," Proc. 4th European Conf. On Computer Vision, Cambridge, UK, April 1996.

[4] K. Sobottka and I. Pitas, "Segmentation and tracking of faces in color images," Proc. Of the Second Intl. Conf. On Auto. Face and Gesture Recognition, pp. 236-241, 1996.

[5] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," In Proc. Of IEEE CVPR, pp. 21-27, 1997.

[6] W.T. Freeman, K. Tanaka, J.Ohta, and K. Kyuma, "Computer Vision for Computer Games," Int. Conf. On Automatic Face and Gesture Recognition, pp. 100-105, 1996.

Solving problems of noise and uneven illumination for line navigating visual mobile robot duality in multi-edge detection and texture analysis

Xin Wang*   Masanori Sugisaka**

*Department of Electronic Engineering, Oita Institute of Technology,
27-407 Daijihigasihama Nakatsu, 871-0006, Japan
(Tel: +81-979-235500, Fax: +81-979-23-7001, E-mail: wang@oita-it.ac.jp; wangxin_inf@yahoo.com)
**Department of Electrical and Electronic Engineering, Oita University
700 Dannoharu, Oita 870-1192 Japan
(Tel: +81-975-547831, Fax: +81-547841, E-mail: msugi@cc.oita-u.ac.jp)

Abstract: *Noise, uncertain and inhomogeneous environment have to be considered very carefully when a visual robot is put into line navigation in a real world instead of ideal experimental conditions. These environmental factors in a image sometimes are hardly to deal with. Our goal in this paper is to detect line landmarks in above environments correctly while a robot moves from one room to other rooms for the purpose of application. In such a dynamic process, the image is the only information and plays a very important role to all of the control algorithms. Aiming at solving the problems resulting from noise and uneven illumination, we used multi-edge detection and texture analysis, and the results are satisfied. This approach is hoped to give a practical conclusion to all this kind of problems.*

Keywords: Line Guidance, Environment, Noise, Edge, Structure Tensor

1. Introduction

Every real-world application has to contend with uneven illumination of the observed scene. Even if we spend a lot of effort optimizing the lighting system, it still very hard to obtain a perfectly even illumination. Such kinds of problems will cause that a visual robot could not works well. Our problems are how to detect these landmarks [1] in noise and uneven illuminant conditions correctly when a robot moves from one place to another place, for example, from a room to other rooms for the purpose of application. The difficulties are, first, when a robot navigate in the real world, that is in a dynamic environmental condition, how to deal with over-light, reflection of light, contrary light to the camera lens and so on; the second is how to handle the noises, such as statistical noise or other cross-line guidance on the road, or type-like marks. We also have to decrease the computing time as short as possible because of the constraints of sampling time of whole the system. Those problems are equivalent to design good image analysis approaches for a visual robot to some extent.

There are some ways, for example, contrast enhancement, contrast stretching or range compression, to make a image look good. But these ways only improve the appearance of the image but not the image quality itself. [2] Moreover, it is possible to correct the effect of inhomogeneous illumination by either taking a reference image without object or calculating a mean image from many different images as a reference, then, an object can be divided from a background environment. But in a dynamical process such as robot navigation, it is very difficult to get such a reference and also almost impossible to take some appropriate images without object because this process is changing with time. Noises are serious problems to any a visual system that works in a real environment. For the solution of this problem, we either explore available algorithm or make the object have its unique features independent from noise as possible as we can.

Our research gives a satisfied result to that kind of problem of line navigation. Noise and uneven illumination analysis are analyzed in section 2. We will discuss our method in section 3 and section 4. The experimental results will be illustrated in section 5.

2. Noise and uneven illumination analysis

A charge-coupling device camera is used in our visual robot system. Image from the camera is in a color device independent bitmap (DIB) format. A pixel data $P(q)$, which is consist of three elements in the well-known red, green, blue (RGB) vector space or channels, $P(q) = [b(q), g(q), r(q)]^T$, is stored in the memory by three bytes, total 24 bits in a 32 bits computer.

In this paper, noise is defined as the effect of environmental conditions on an image. Noise is divided into two sets: statistical noise and the objects except the line guidance. Though statistical noise can be deleted by reference image way or averaging an image sequences in several sampling period because of its zero variance, it is obviously impossible to do such works in a dynamic process. To a bitmap color image, we can found its several formats in different coordinate systems or space transformations that are suitable for special purposes, for example, in HIS (H: hue, I: intensity, S: saturation) space etc..

Color-threshold filter is available to depress noises, but it works in a narrow range and very weak in the chang-

ing illuminant environments and similar color objects. To the problem of robot line navigation, environments should be considered carefully because even a widow in a corridor or in a room could cause changes of sunshine angels and brightness distribution when a robot moves from one place to another one. Therefore, it is necessary to consider some of effective strategies, though it is sometimes also difficult for human beings in fact.

3. Multi-edge detection

Edge detection is an available method to analyze lines in an image. This can be realized by its convolution with widow function. But as well known that the edge detection is sensitive to statistic noise, and hardly get the real useful information if it is used directly. Instead of discussing these principles, we pay attention to the problem of the ways to deal with noises while some of morphological operators are applied. Our tragedies follow the two clues. An edge is detected only when the magnitude of the gradient is above a certain threshold that is related to the level in the image. We definite the edge coefficient

$$C_e = f_c(\|\nabla g\|, r) = \frac{\|\nabla g\|^2}{\|\nabla g\|^2 + \gamma^2} \qquad (1)$$

Here, $\nabla g = [\partial g/\partial x, \partial g/\partial y]^T$ is gradient in horizontal x and vertical y directions, (see Figure 1) and $\|\nabla g\|$ is the magnitude of the gradient ∇g. One of the ways is by mean of selection of edge coefficient C_e. To a noise threshold level V_t:

$$V_t = k_1 V_s f_c(\|\nabla g\|, \gamma) \qquad (2)$$

k_1 is a controllable coefficient, V_s is maximum magnitude of $\|\nabla g\|$, γ is noise coefficient in (1). Because of the effect of illumination, γ should include the changes of gray value in strong, medium, and weak cases and is decided by

$$\gamma = k_2 k_m \|\nabla g\| \qquad (3)$$

where k_m is a dynamic coefficient with respect to the maximum value in a histogram. In a DIB image, a pixel can be regarded as a random variable as any other measured quantity. A discrete histogram function $H(k_i)$
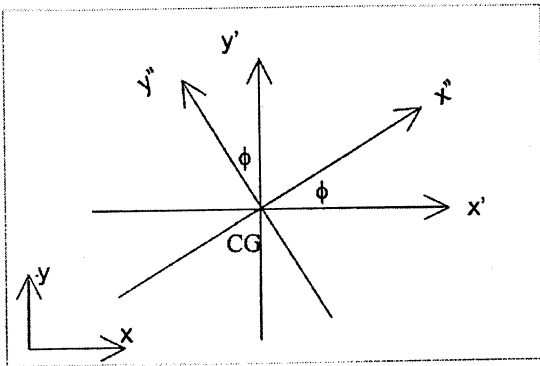
Figure 1: Image coordinate x-y which zero is in the down left, and the angle of orientation to x'-y' is ϕ.

$(k_i \in \mathcal{K} = \{k_i\}, i = 0, 1, 2, \ldots, 2^8)$ has a maximum $H(k_m)$. We notice that though single edge detection is weak in noises, things will become different if there are enough constrains. Increasing constrains also means more cost, weight etc.. So those constrains must be practical. Let the features of a line be presented by a vector $\mathcal{M}_i \in R^{m+2}$, $\mathcal{M}_i = \{d_i, \phi_{ij}, \tau_i\}$, $(i = 1, 2, \ldots, n; j = 1, 2, 3, \ldots, m)$, the elements in $\mathcal{M}_i$ are defined:

$$d_i > V_t \qquad (4)$$

$$\tau_i = p_m \qquad (5)$$

$$\phi_{ij} = \bar{k}_0 \cdot \frac{\nabla g}{\|\nabla g\|} = \left\{ \begin{array}{ll} 1 & (j = 2l + 2) \\ -1 & (j = 2l + 1) \end{array} \right. \quad (l = 0, 1) \qquad (6)$$

where $\bar{k}_0 = \{k_x, k_y\}$ is a unit vector in x and y directions independently. p_m is a constant responding to the pixel width of the line type. We designed a dou-
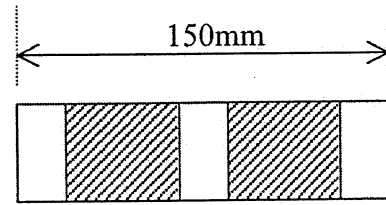
Figure 2: Designed line guidance type, material: clothes.

ble lines guidance.(see Figure 2) Considering the effect of brightness on $\|\nabla g\|$, the contrast of the line guidance are strengthened on purpose. The green channel $g(q) \in P(q)$ is used to detect edges. This either is good enough for image analysis or simply the whole calculating process. Edge detections can be made along both of x and y directions.

4. Two-dimensional tensor method

We describe the line guidance by the following features:

1. In order to give controller [3] correct information, the line's position is necessary. This line is described by its angle of orientation ϕ_i and center of mass CG=$(c_x, c_y)_i$ $(i = 1, 2, 3)$.

2. The image is divided into $i+1$ parts in y direction, in which the curves are approximated by lines. Several such lines compose of a curve too.

The angle of orientation ϕ illustrated in Figure 1 is determined by its structure tensor computed in a way as a combination of linear convolution and nonlinear point operators. If we definite the discrete partial derivative cooperator with respect to the coordinate p, q by $\mathcal{D}_p$, $\mathcal{D}_q$ and the isotropic smoothing operator by $\mathcal{B}$, the local structure tensor of a gray value image can be computed with the following structure tensor operator

$$J = \mathcal{B}(\mathcal{D}_p \cdot \mathcal{D}_q) \qquad (7)$$

The equation is written in an operator notation. In two dimensions, the symmetric structure tensor has 3

components. The three components of the inertia tensor for the rotation of the object around its center of mass is used for image analysis. The tensor J is presented by the eigenvalues and eigenvectors by rotation of coordinate system:

$$J = \begin{bmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix} \begin{bmatrix} cos\phi & sin\phi \\ -sin\phi & cos\phi \end{bmatrix} \tag{8}$$

From the diagonal form of J, we can get the solution of the functions $\phi = \mathcal{F}(J_{xx}, J_{xy}, J_{yy})$ and ϕ is the angle of orientation in Figure (1) (Experimental results are given in section 5.) The three inertia tensors can be formed by the second-order moments. We can transfer symmetric tensor J to second-order moments, and then the orientation is calculated.

5. Experimental results and discussions

In order to describe the feature of the line and limit the useful information in a certain range, three parts of the whole four image ranges are used in y direction (196 pixels, 72dpi) from $y = 0$ to $y = 75$, and every part is 25 pixels based on the basic dividing idea in [3] using the solutions. The i ($\in \{1, 2, 3\}$) in center of mass $(c_x, c_y)_i$ is with respect to the i_{th} divided parts in an image.
Experimental environments are given in Table 1. The results are illustrated in Figure 3, Table 2 and Table 3. Generally, the edge is detected in x direction, but in the following cases, the y direction is needed.

$$p_1 > p_n \quad and \quad (p_2 \vee p_3) < p_n \tag{9}$$

$$p_1 < p_n \quad and \quad (p_2 \vee p_3) > p_n \tag{10}$$

Here, p_n is threshold of the pixel numbers. When $p_i > p_n$, it is regarded that there is line guidance in concerning area. The above conditions ensure enough information in the divided parts. Figure 3-a-1 gives an example of $p_3 < p_n$. From Figure 3-a-2 it is understood that it is very difficult recognize such a line. Figure 3-a-3 and Figure 3-a-4 are the results. It is seen in the third part, there are less information to the detection in x direction, this is also show us this is a possible cross-line, and when this happen, the direction in y direction is put into action. In Figure 3-d-3, there are less information in the third parts because of over-light. But detection in y tell us there are not cross-line. Then we could conclude that this reason from either noise or the

Table 1: Environment analysis: $\triangle$=stong; $\triangledown$=weak; $\otimes$=good; $\times$=not good; C.=cross-line; S.=statistical

No.	S.noise	illumination	C.noise	conclusion
a	$\triangle$	$\times$	$\triangle$	$\times$
b	$\triangle$	$\times$	$\triangle$	$\times$
c	$\triangledown$	$\times$	$\triangle$	$\times$
d	$\triangledown$	$\times$	$\triangle$	$\times$
e	$\triangledown$	$\otimes$	$\triangledown$	$\otimes$

end of line. In such a case, there are other two of centers of mass, so the conclusion is a line with $(c_x, c_y)_1$, $(c_x, c_y)_2$ and no cross-line on the road. This a also what the conditions means in (9)(10).

Table 2: Centers of mass to basics (112, 0) in Figure 3-a-e.

No.	$(c_x, c_y)_1$	$(c_x, c_y)_2$	$(c_x, c_y)_3$
a	(14, 11)	(14, 35)	(05, 66)
b	(07, 13)	(41, 36)	(59, 57)
c	(31, 11)	(28, 37)	(26, 64)
d	(32, 11)	(30, 33)	(32, 60)
e	(-5, 12)	(-5, 35)	(-4, 61)

Table 3: Angles of the orientation in Figure 3-a-e.

No.	ϕ_1 (rad)	ϕ_2 (rad)	ϕ_3 (rad)
a	1.5696	1.4187	1.4216
b	0.6593	0.4385	0.3601
c	1.8171	1.7137	1.8744
d	1.5791	1.2320	1.9326
e	1.4916	1.4187	1.4216

6. Conclusions

Image analysis is the upmost problem in a visual line navigating system. In this paper, we solved following problems for such a visual robot:
1. which kind of line guidance is the suitablest;
2. how to detect the DIB information from a visual sensor;
3. how to describe the features of line guidence;
4. how to deal with the case of cross-line.

We can use fuzzy, neural network, or neural computer controller to drive the motors [3] using the solutions. The results in this paper show us a perfect way to deal with this kind of problems in a rather widely working range in noise and uneven illuminant conditions. To line navigation, one of the key points is that the image information to the controller has to be good enough. So we suggest that most line navigating visual robot should use this method to analyze images. We agree with the viewpoint of *Vision is one of the keys to understanding intelligence*. A machine should understand something, and then it could learn and remember what it has understood. This approach also shows more clues to our research in the future.

References

[1] J. Bovenstein, H. R. Everett, L. Q. Feng, "Navigating Mobile Robot", AK Peter Ltd. Press, Massachusetts, 1996.
[2] B. Jahne, "Iamge Processing for Scientic Application", CRC Press, New York, 1997.
[3] M. Sugisaka, X. Wang, "Intelligent Control Strategy for a Mobile Vehicle", Applied Mathematics and Computation, Elsevier Science vol.91 no. 1, pp. 91-98, 1997.
[4] M. Brown, D. Terzopoulos, "Real-time Computer Version", Cambridge University Press, Cambridge, 1994.
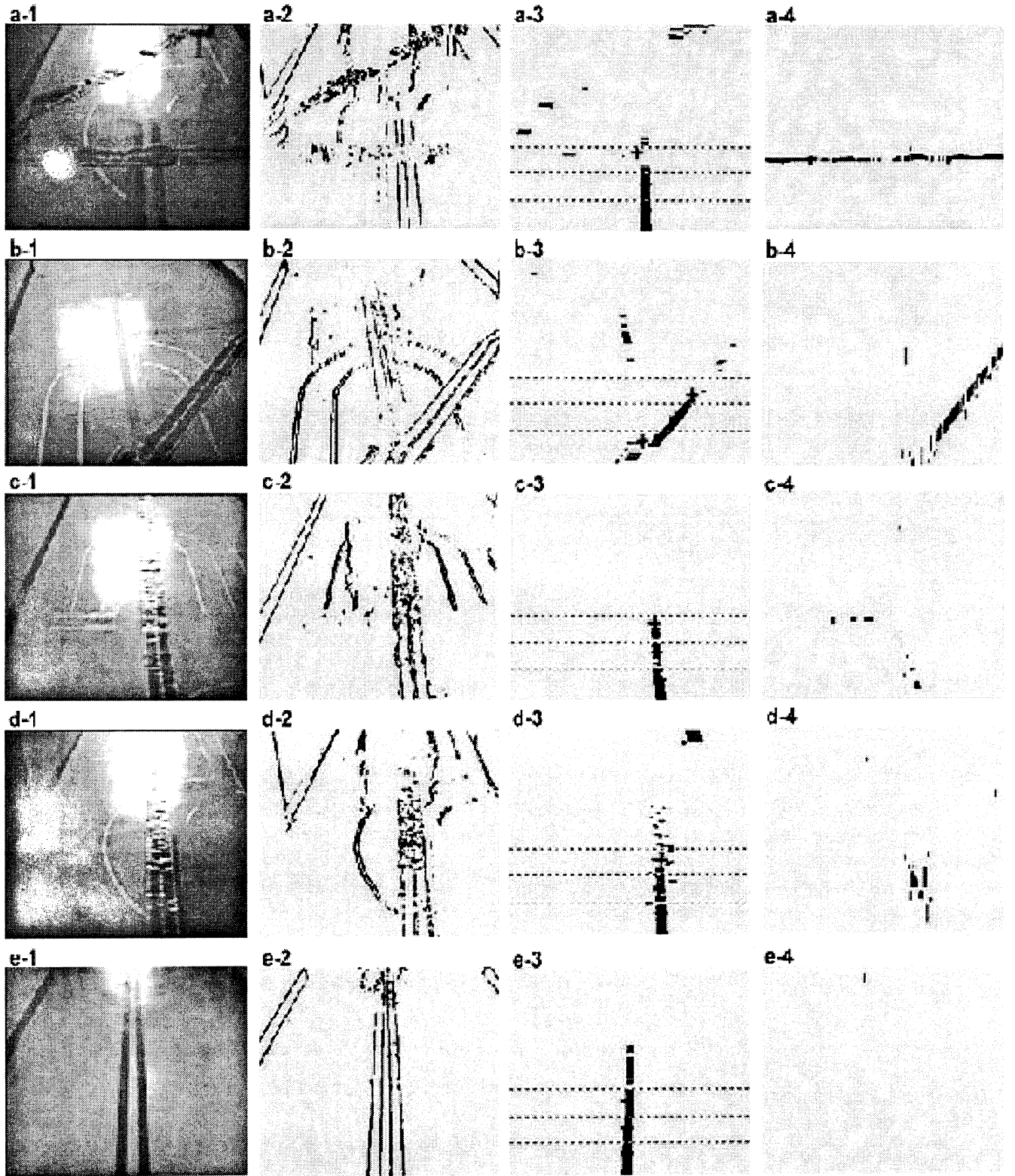
Figure 3: Experimental results: Five typical cases, which cover over-light, reflection of light, contrary light to the camera lens and so on, a, b, c, d, e, are given. The condition analysis are listed in Table 1. In this figure: col. 1) original image; col. 2) the result of edge detection by Prewitt operator in x direction; col. 3) Multi-edge detection result in x direction. The symbol "+" is the positions of the center of mass in corresponding parts. The dash lines are the borders of the parts; col. 4) Multi-edge detection in y direction.

Experimental study of femoral canal shaping by Super-broach, a femur mounted mini robot

HoChul Shin*, Youngbae Park*, Yong-San Yoon*, Anthony Hodgson**
*KAIST, Department of Mechanical Engineering, Korea
**University of British Columbia, Canada
mailto:// taichi@kaist.ac.kr

Abstract

In the cementless total hip arthroplasty, close fit as well as precise alignment of the femoral stem within the femoral cavity is very important for early recovery and long term success. The robot assisted arthroplasty using ROBODOC® system was shown to provide excellent solution for such objects. However, it requires a CT image and complicated registration such as marker based or geometry based registration along with constant monitoring of the femoral motion during robot operation. We proposed a very simple and economic method of robotic operation using mini-robot mounted on the femur along with simple gauge based registration. This new method does not require CT/MRI image nor monitoring of the femoral motion during robot operation while it may achieve the same goal as ROBODOC® system; close fit and good alignment of the implant. We tried our new method on fresh cadaveric femurs and measured the fitness and alignment of the implant within the bones. These results shows significant improvement over the manual operation and comparable with the ROBODOC® system.

1 Introduction

In the traditional total hip replacement arthroplasty, the femoral canal is prepared by the orthopedic surgeons with the reamers and broaches to match the femoral stem. In cementless procedures, this process typically leaves significant gap between the femur and the implanted stem, which may delay or prevent the ingrowth of the bone into the implant surface. Also it is speculated that the long-term success of the femoral stem may depend on the correct alignment of the implant within the femur. Introduction of the robot-assisted femoral canal shaping by Taylor et al [1] has increased the percentage of surface contact between the implant

and the bone from 20.8% using conventional methods to 95.7% with the robotic surgery [2], and also improved the alignment of the implant within the femur [3]. Here, proper registration informing the robot the precise location of the femur of the patient in the operating room is very important. The ROBODOC® system employs a fiducial marker-based registration technique in which markers are implanted into the femur in a preliminary procedure prior to the CT scan used to plan the surgery. Recently, several geometry-based registrations were developed to avoid implanting fiducial markers [4], although these methods still require CT/MRI image to obtain the bone geometry and close monitoring of the femoral motion during robot operation.

2 Gauge Based Registration and femoral shaping
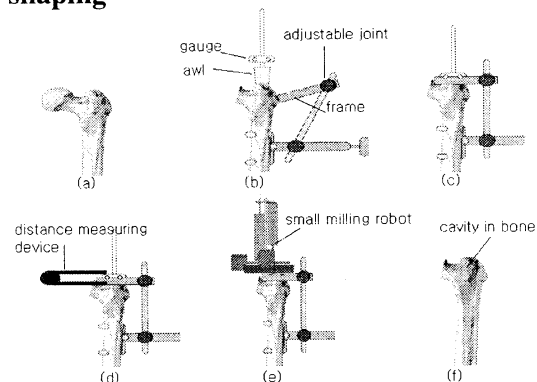
Fig 1 Robot operation procedure

Proposed procedure as shown in Fig 1 is as follows: (a) Resect and expose the hip joint. (b) Remove femoral head, insert reamer shaped gauge and attach the base frame to femur by clamp. (c) Once the base frame is aligned, fasten adjustable joints of the frame. (d) Measure the position of reamer shaped gauge with respect to base frame. (e) Remove gauge and mount mini

robot on the frame. (f) The robot shapes hole for the stem.

We are primarily interested in the location of the femoral canal to prepare for the room of the femoral implant. Thus we used the reamer shaped gauge which may tightly fit into the diaphyseal femoral cavity. Here, the surgeon decides the size and position of the gauge and the femoral stem from the x-ray film. Then, the gauge is inserted into the femoral cavity tightly fitting into the diaphyseal bone while having clearance in the proximal region that the gauge may properly align to the diaphyseal cavity. After aligning the gauge and the base frame, we can get the gauge position with respect to the base frame by measuring 6 distances between the gauge and the base frame. After removing the gauge, robot is mounted on the base frame and performs canal shaping.

As our robot is mounted on the femur and the stem shape has usually simple shape, we need only 3 DOF robot for the cavity shaping. We designed Cartesian type mini robot as shown in Fig.2. The robot has the weight of 1.1kg and size of 230x230x280 mm.

3 Cadaver test results

The femoral canal shaping was done on 5 fresh cadaveric femurs by the mini robot with gauge based registration procedure. Each robot shaping procedure as shown in Fig 3 took about 30 minutes. Before the canal shaping, bi-lateral x-ray images were taken and desired stem size and its position were selected according to conventional procedures. This position is used later to compare with the position of the implant after operation to define the position error. After installing the stem into the femur, we measured two types of errors: One is the alignment error and the other is the fitness error. The alignment error is defined as the difference between the observed position of the installed stem and the desired position determined from the bi-lateral x-ray films. The fitness error is defined as the percentage of the stem surface area which has less than 0.3mm gap with the femur.

Anteversion angle error as defined as the difference of the gauge position and inserted stem position in axial direction was 0.2 degrees. The varus/valgus angle error as defined the angular difference between the planned position from the x-ray images as shown in Fig 4 in anterior-posterior view was 1.1 degrees (S.D. 0.49). Depth error along the longitudinal axis was 0.3mm. Fitness measurement was made on transverse slicing of the stem with the bone. It shows that 95% of contact surface had less than 0.3mm gap. The overall errors were quite comparable with

ROBODOC® system [2, 3, 5] as listed in Table 1.

	Developed system	Robodoc® System
Depth error	0.3mm	0.4~1mm
Alignment error	1.1 degrees	1 degrees
AV error	0.2 degrees	0.4 degrees
Fitness error	95%	95.7%

Table 1 Result compared with Robodoc®
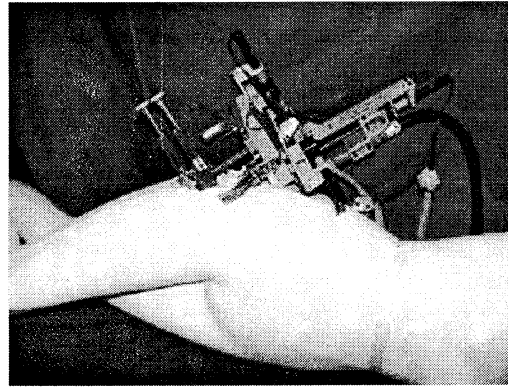
Fig 2 Mini robot in working position

Fig 3 Shaping cadaveric femur
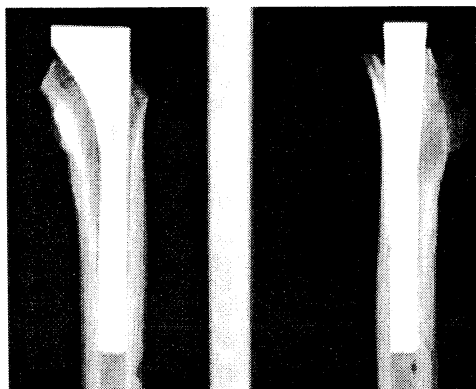
Fig 4 X-ray result

4 Conclusion

In this study, we tested the feasibility of the mini robot's femoral cavity shaping with the gauge based registration. This method does not require CT/MRI image nor femoral motion monitoring during operation as the robot is mounted on the femur. The proposed method was tested for fresh cadaveric femurs and its result shows that the proposed method provides significant improvement over the traditional manual practice and comparable to the ROBODOC® system in terms of correct alignment of the implant and gap between the implant and the femur.

Acknowledgement

References

[1] Taylor RH, Mittelstadt BD, Paul HA, Hanson W, Kazanzides P, Zuhars JF, Williamson B, Musits BL, Glassman E, Bargar WL. "An image-directed robotic system for precise orthopaedic surgery", *IEEE Transactions on Robotics and Automation*, 10(3): 261-275, June 1994

[2] Bargar BL and Carbone EJ. "Robotic femoral canal preparation, Total Hip Revision Surgery", Galante JO, Rosenberg AG, and Callaghan JJ (ed.), 415-423, Raven Press, New York, 1995

[3] Jerosch J, Peuker E, von Hasselbach C, Lahmer A, Filler T, Witzel U. "Computer assisted implantation of the femoral stem in THA - an experimental study", *Int Orthop*, 23: 224-6, 1999.

[4] Simon DA and Kanade T. Geometric constraint analysis and systhesis: "Methods for improving Shape-based registration accuracy", *Proceedings of the First Joint CVRMed / MRCAS Conference*, 181-190, 1997

[5] William L. Bargar, Andre Bauer, Martin Borner, "Primary and Revision Total Hip Using the Robodoc® System", CLINICAL ORTHOPAEDICS AND RELATED RESEARCH, Number 354, pp 82-91, 1998

[6] Yoon YS, Lee JJ, Kwon DS, Won CH, Hodgson AJ, Oxland T. "Accurate femoral canal shaping in total hip arthroplasty using a mini-robot", *Proc. of 2001 International Conference on Robotics and Automation*, 3214-3219, 2001

Improving the Performance of Multi-Objective Genetic Local Search Algorithms

Hisao Ishibuchi and Tadashi Yoshida
Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
{hisaoi, yossy}@ie.osakafu-u.ac.jp

Abstract

We have already proposed a multi-objective genetic local search (MOGLS) algorithm where elitism and local search were introduced to an evolutionary multi-criterion optimization (EMO) algorithm. While our MOGLS algorithm utilizes these two promising concepts, its search ability to find Pareto-optimal solutions is not always comparable to recently developed EMO algorithms. In this paper, we try to improve our MOGLS algorithm by modifying its selection mechanisms of parent solutions for crossover and initial solutions for local search. Roulette wheel selection of parent solutions is replaced with tournament selection. While a local search procedure was applied to all offspring, an appropriate initial solution is chosen from offspring for each local search direction. Poor offspring are not used as initial solutions for decreasing unnecessary calculations. We also examine further modifications of selection mechanisms for keeping the diversity of populations. The best solution with respect to each objective in the current population is likely to be selected as a parent solution. The same bias is also used in the selection of initial solutions. Through computer simulations, we examine the effects of these modifications and compare modified MOGLS algorithms with recently developed EMO algorithms.

Keywords: Hybrid genetic algorithms, evolutionary multi-criterion optimization, multi-objective genetic local search.

1. Introduction

Let us consider the following n-objective minimization problem:

$$\text{Minimize } z = (f_1(x), f_2(x), ..., f_n(x)), \quad (1)$$
$$\text{subject to } x \in X, \quad (2)$$

where z is the objective vector, x is the decision vector, and X is the feasible region. The task of evolutionary multi-criterion optimization (EMO) algorithms is to find a number of solutions that are not dominated by any other solutions. Let a and b are two decision vectors ($a, b \in X$). Then b is said to be dominated by a (i.e., $a \prec b$) if and only if the following two conditions hold:

$$f_i(a) \le f_i(b) \text{ for } \forall i \in \{1, 2, ..., n\}, \quad (3)$$
$$f_i(a) < f_i(b) \text{ for } \exists i \in \{1, 2, ..., n\}. \quad (4)$$

When b is not dominated by any other solutions in X, b is said to be a Pareto-optimal solution. In the case of large problems, it is impractical to try to find true Pareto-optimal solutions. Thus non-dominated solutions among examined ones are presented to decision makers as approximate Pareto-optimal solutions by EMO algorithms.

The main advantage of population-based evolutionary algorithms for multi-objective optimization over point-based search techniques (e.g., local search, simulated annealing, and taboo search) is that a number of non-dominated solutions can be simultaneously obtained by their single run. On the other hand, multiple runs are usually required for finding multiple non-dominated solutions by point-based search techniques. In early studies on EMO algorithms (e.g., [1]-[3]), emphasis was placed on keeping the diversity of populations in order to find uniformly distributed Pareto-optimal solutions. Thus several concepts such as niching, fitness sharing and mating restriction were introduced into EMO algorithms. In recent studies (e.g., [4]-[6]), emphasis was placed on the convergence speed to Pareto-optimal solutions as well as the diversity of populations. The importance of elitism in EMO algorithms was clearly demonstrated in those studies.

Another promising trick for improving the convergence speed is the use of local search in EMO algorithms. This trick was first implemented in our former studies [7,8] as a multiobjective genetic local search (MOGLS) algorithm. The basic idea of our MOGLS algorithm is to use a scalar fitness function with random weights for selection and local search. A pair of parent solutions is selected from a current population using the following scalar fitness function:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x), \quad (5)$$

where w_i is a non-negative weight. The point is to randomly specify the weight values whenever a pair of parent solutions is selected. That is, each selection is governed by a different weight vector. A local search procedure is applied to each offspring using the scalar fitness function with the same weight vector as in the selection of its parent solutions. This weight specification mechanism generates various search directions in the n-dimensional objective space. Our MOGLS algorithm also uses a kind of elitism where a set of non-dominated solutions obtained during its execution is stored as a

secondary population separately from a current population. A few non-dominated solutions are randomly selected from the secondary population and their copies are added to the current population.

While our MOGLS algorithm utilizes two promising concepts (i.e., elitism and local search), its performance is not always comparable to recently developed EMO algorithms such as SPEA [4] and NSGA-II [6]. In this paper, we try to improve our MOGLS algorithm.

2. Multi-Objective Genetic Local Search Algorithm

Our MOGLS algorithm in our former studies [7,8] is written as follows:

Step 0: Randomly generate an initial population of N_{pop} solutions.

Step 1: Calculate the n objectives for each solution in the current population. Then update the tentative set of non-dominated solutions (i.e., secondary population) that is stored separately from the current population.

Step 2: Select ($N_{pop} - N_{elite}$) pairs of parent solutions by repeating the following procedure:.
  (1) Randomly specify the weight values.
  (2) Select a pair of parent solutions using the scalar fitness function. Roulette wheel selection with a linear scaling is used.

Step 3: Apply a crossover operator to each of the selected pairs of parent solutions. A new solution is generated from each pair. Then apply a mutation operator to the generated new solutions.

Step 4: Randomly select N_{elite} solutions from the tentative set of non-dominated solutions. Then add their copies to the ($N_{pop} - N_{elite}$) solutions generated in Step 3 to construct a population of N_{pop} solutions.

Step 5: Apply a local search procedure to each of the N_{pop} solutions in the current population. The local search direction for each solution is specified by the weight values used for the selection of its parent solutions. Local search is terminated if no better solution is found among k neighboring solutions that are randomly selected from the neighborhood of the current solution. The current population is replaced with the N_{pop} solutions improved by local search.

Step 6: If a pre-specified stopping condition is not satisfied, return to Step 1. Otherwise terminate the algorithm and present the secondary population as final solutions.

One characteristic feature of our MOGLS is the inheritance of random weight values from the selection procedure of two parents to the local search procedure for its offspring. Another characteristic feature is to restrict the number of neighboring solutions to be examined in local search. That is, all the neighboring solutions of the current solution are not examined by local search for decreasing computation time spent by local search.

3. Some Modifications for Performance Improvement

3.1 Modification of Selection for Crossover

Tournament selection often outperforms roulette wheel selection. We use binary tournament with replacement instead of roulette wheel for selecting parent solutions in Step 2 (2). The modified MOGLS algorithm is referred to as MOGLS-S.

3.2 Modification of Selection for Local Search

In our MOGLS algorithm, local search was applied to all offspring. The local search direction for each offspring was specified by the random weight values used for the selection of its parent solutions. This implementation of local search leads to the application of local search to poor solutions and the inappropriate specification of the local search direction for each offspring. We modify the local search implementation in Step 5 as follows.

(1) Randomly specify random weight values.
(2) Select a single solution from the current population using tournament selection with replacement. The tournament size was specified as four in our computer simulations. Local search is applied to the selected solution using the weight values in (1).

By iterating this procedure N_{pop} times, we find N_{pop} initial solutions for local search. The local search direction for each initial solution is specified by the weight values used for its selection in (1). Good solutions may be selected several times. For those solutions, local search is applied to several times with different local search direction. Poor solutions will not be selected as initial solutions. The MOGLS algorithm with the modification of both the selection of parent solutions for crossover and the selection of initial solutions for local search is referred to as MOGLS-SL. The MOGLS algorithm with the modification of only the selection of initial solutions for local search is referred to as MOGLS-L.

3.3 Further Modifications

In the MOGLS algorithm and the above-mentioned modified versions, weight values are randomly specified. One trick for keeping the diversity of populations is to bias the selection toward the best solution with respect to each objective. We modify the weight value specification from the totally random specification in the MOGLS algorithm and its modified versions to the following biased one. We use n unit weight vectors $(1,0,...,0)$, $(0,1,...,0)$, ..., $(0,0,...,1)$ for both the selection of parent solutions and the selection of initial solutions for local search. Other weight vectors are randomly specified as in the MOGLS-SL algorithm. The MOGLS-SL algorithm with this modification is referred to as MOGLS-SL*.

A more direct trick is to choose the best solution with respect to each objective. In the selection of parent solutions, a pair of the best two solutions with respect to each objective is selected. In this manner, n pairs of parent

solutions are selected for the n objectives. The other pairs of parent solutions are selected by tournament selection using random weights as in the MOGLS-SL algorithm. On the other hand, the best solution with respect to each objective is selected as an initial solution for local search. Thus n initial solutions are chosen for the n objective. The other initial solutions are selected by tournament selection using random weights. The MOGLS-SL algorithm with this modification is referred to as MOGLS-SL**.

4. Computer Simulations

4.1 Test Problem

In the same manner as in [8], we generated a 40-job and 20-machine flowshop scheduling problem with two objectives: to minimize the makespan and to minimize the maximum tardiness. This problem is a kind of permutation problem. There exist 40! feasible solutions.

4.2 EMO Algorithms

We applied our MOGLS algorithm and its five modified versions to the test problem. The SPEA in [4] and the NSGA-II in [6] were also examined. The following parameters were commonly used in all algorithms:

Population size: 20,
Crossover probability: 0.9,
Mutation probability: 0.3 (for each string),
Stopping condition: Evaluation of 60,000 strings.

In the MOGLS algorithm and its modified versions, the maximum number of neighboring solutions examined by local search for the current solution was specified as $k = 2$ (see Step 5 of the MOGLS algorithm). In the SPEA, the size of the secondary population was specified as 20.

4.3 Simulation Results

Simulation results are summarized in Fig. 1 ~ Fig. 8. Each figure shows non-dominated solutions obtained by 150 trials of the corresponding EMO algorithm. Note that the two objectives of the test problem are to be minimized. From these figures, we can see that all the five modified versions clearly outperformed the original MOGLS algorithm in Fig. 1. The difference in their performance among the five modified versions (i.e., Fig. 2 ~ Fig. 6) is not clear. Thus further computer simulations on various test problems are required for examining their performance further. We can also see from the simulation results in Fig. 2 ~ Fig. 8 that the five modified versions of our MOGLS algorithm are comparable to the recently developed EMO algorithms (i.e., SPEA and NSGA-II).

The average CPU time for a single trial of each algorithm was as follows:

11.66 sec.: MOGLS   12.47 sec.: MOGLS-S
13.21 sec.: MOGLS-L   13.29 sec.: MOGLS-SL
13.34 sec.: MOGLS-SL*   13.28 sec.: MOGLS-SL**
17.47 sec.: SPEA   17.50 sec.: NSGA-II

The average CPU time of the MOGLS algorithm and its modified version was about 3/4 of the SPEA and the NSGA-II. The five modified versions of the MOGLS algorithms outperform the SPEA and the NSGA-II when these algorithms are compared under the same CPU time.

5. Conclusion

In this paper, we proposed some modifications of our MOGLS algorithm for improving its performance. Simulation results clearly showed that the performance of the MOGLS algorithm was significantly improved by the proposed modifications. It was also shown that the modified versions of the MOGLS algorithm were comparable to the recently developed EMO algorithms: SPEA and NSGA-II.

References

[1] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," *Proc. of 5th International Conference on Genetic Algorithms*, pp. 416-423, 1993.

[2] J. Horn, N. Nafpliotis and D. E. Goldberg, "A niched Pareto genetic algorithm for multi-objective optimization," *Proc. of 1st IEEE International Conference on Evolutionary Computation*, pp. 82-87, 1994.

[3] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221-248, 1994.

[4] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation*, vol. 3, pp. 257-271, 1999.

[5] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, pp. 173-195, 2000.

[6] K.Deb, S.Agrawal, A.Pratap, T.Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *Proc. of Parallel Problem Solving from Nature VI*, pp. 849-858, 2000. An extended version is available as *KanGAL Report* 200001, Indian Institute of Technology Kanpur.

[7] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm, "*Proc. of 3rd IEEE International Conference on Evolutionary Computation*, pp. 119-124, 1996.

[8] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling, "*IEEE Trans. on Systems, Man, and Cybernetics –Part C*, vol. 28, pp. 392-403, 1998.

Fig. 1. Simulation results of MOGLS

Fig. 2. Simulation results of MOGLS-S

Fig. 3. Simulation results of MOGLS-L

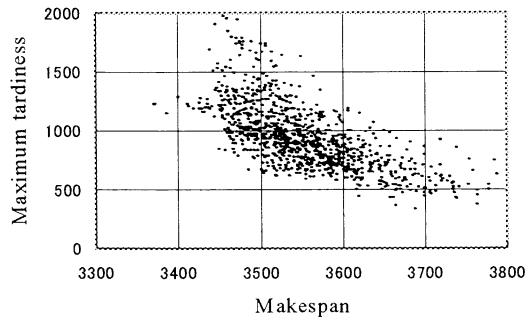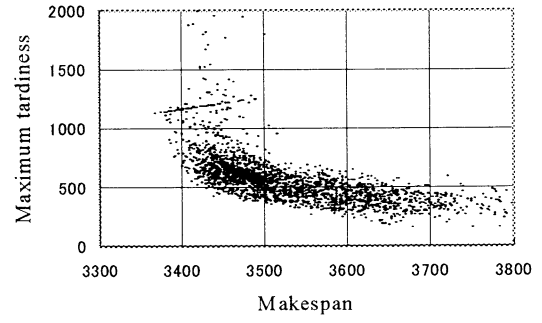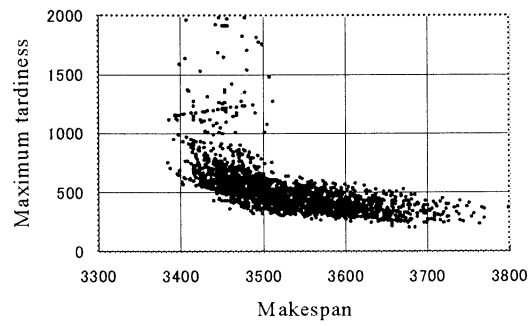**Fig. 4**. Simulation results of MOGLS-SL

Fig. 5. Simulation results of MOGLS-SL*

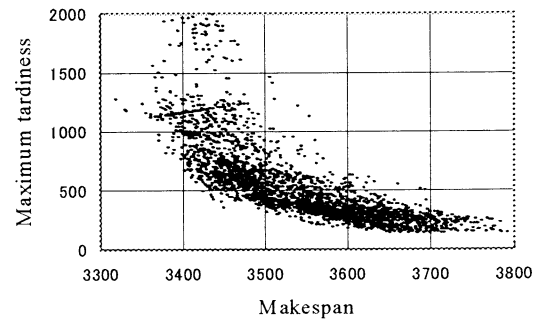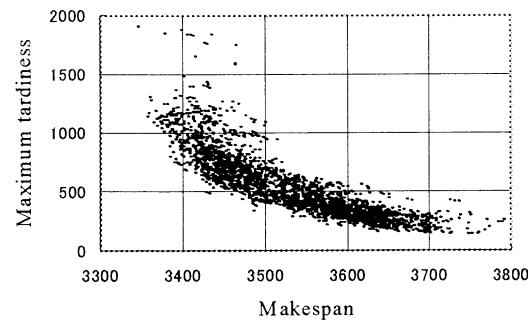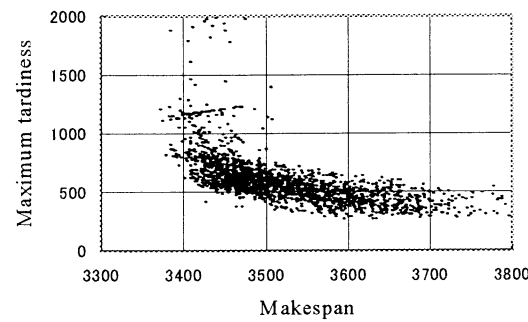**Fig. 6**. Simulation results of MOGLS-SL**
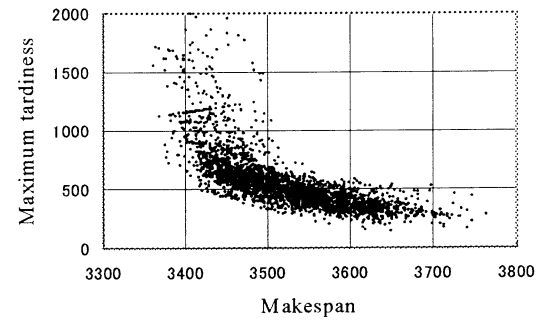
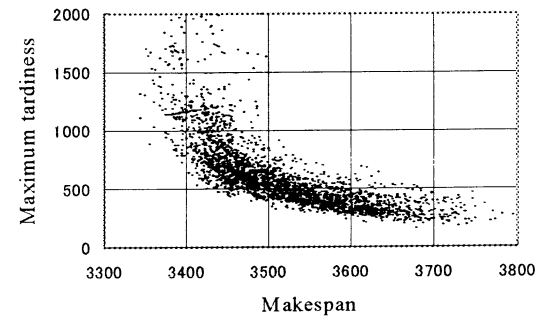**Fig. 7**. Simulation results of SPEA

Fig. 8. Simulation results of NSGA-II

A Study on Several Variants of Cellular Genetic Algorithms

Tomoharu Nakashima
Dept. of Industrial Eng.
Osaka Prefecture University
Gakuen-cho 1-1, Sakai
Osaka 599-8531, Japan
nakashi@ie.osakafu-u.ac.jp

Takanobu Ariyama
Dept. of Industrial Eng.
Osaka Prefecture University
Gakuen-cho 1-1, Sakai
Osaka 599-8531, Japan
ariyama@ie.osakafu-u.ac.jp

Hisao Ishibuchi
Dept. of Industrial Eng.
Osaka Prefecture University
Gakuen-cho 1-1, Sakai
Osaka 599-8531, Japan
hisaoi@ie.osakafu-u.ac.jp

Abstract

In this paper, we examine the performance of various versions of cellular genetic algorithms. First, we examine the effect of sorting individuals on the performance of cellular genetic algorithms. We also examine the effect of the randomized neighborhood size on the performance. Second, we propose a hybrid distributed genetic algorithm where there are four islands and each island interacts with others. Computer simulations show the performance of the proposed method.

Keyword: Cellular genetic algorithms, island model, neighborhood structure, traveling salesman problem

1 Introduction

Genetic algorithms have been successfully applied to various optimization problems [1, 2]. One of the most interesting topics in genetic algorithm community is a distributed version of genetic algorithms [3, 4]. Generally, distributed genetic algorithms fall into two categories. One is a coarse-grained algorithm and the other is a fine-grained algorithm. In this paper, we focus on fine-grained genetic algorithm. In particular, we are interested in cellular genetic algorithms. In cellular genetic algorithms, each individual is located in a cell of the grid-world. It is assumed that each cell has its neighboring cells. That is, when we generate a new individual at a cell, parent individuals are selected from its nearest neighbor individuals including itself. In this manner, we can expect that the cellular algorithms can maintain more diversity in a population than the simple genetic algorithm with no cellular structure. In this paper, we examine several variants of the cellular genetic algorithms. First, we examine the effect of individual sorting where individuals in a row is sorted according to their fitness values. We consider two types of sorting: a uniform

sorting and an alternate sorting. In the uniform sorting, each row is sorted in increasing order. On the other hand, we sort rows alternately in decreasing and increasing order in the alternate sorting. We evaluate the performance of our cellular algorithms for a traveling salesman problem. Computer simulations show the advantage of the sorting over the conventional cellular genetic algorithm without sorting.

Next, we examine the effect of the situation where every cell has different neighborhood sizes. In this case, some individuals are generated from the parents that are selected from the entire population and others are generated from the parents in the neighborhood. In this way, we can readily exploit the information of the good individuals while maintaining the diversity of the population. Computer simulations show the effectiveness of the method.

Finaly, we propose a hybrid genetic algorithm. The proposed genetic algorithm is constructed by combining a coarse-grained genetic algorithm with a fine-grained genetic algorithm.

In computer simulations, we examine the performance of each algorithm for a traveling salesman problem.

2 Cellular GA

Cellular genetic algorithm is first proposed by Whitley [5]. In [5], each cell was updated by using update rules that are frequently used in cellular automata. However, in most of the cellular genetic algorithms a new individual in each cell is generated by a crossover operation between neighboring individuals (see Fig. 1). This algorithm is also refered to as a fine-grained genetic algorithms. In this section, we explain a fine-grained genetic algorithm.

The main difference between the cellular genetic algorithm and the simple genetic algorithm with no cel-
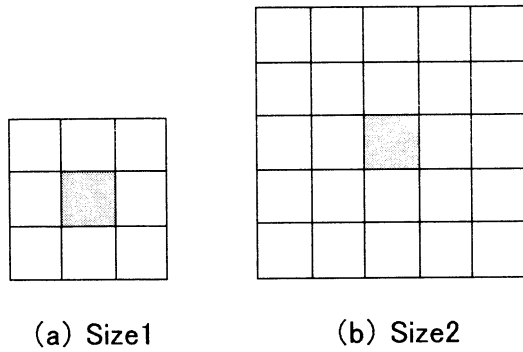
(a) Size1　　　　(b) Size2

Figure 1: Neighborhood structure.

lular structure is that there is a neighborhood structre in the cellular genetic algorithm while there is no neighborhood structure in the simple genetic algorithm. That is, parent individuals for generating a new individual in a cell are selected from its neighboring cells including itself in the cellular genetic algorithms while parent individuals are selected from the entire population in the simple genetic algorithm. The basic procedure of the cellular genetic algorithms is described as follows:

[Cellular genetic algorithm]

1. Generate initial population. A single string is randomly generated for each cell.

2. Evaluate each individual for a particular problem. The evaluation is used for calculating the fitness of an individual.

3. Generate new individuals. For each cell, parent individuals are selected for crossover population. Candidate individuals for the crossover in a cell are located in the neihgborhood cells including itself (See Fig. 1).

4. Mutation. A string for each individual is randomly mutated. The individuals after the mutation is finally used as a child individual.

5. Update the grid-world. A string in each cell is replaced by a new string.

6. If a pre-specified stopping condition is satifsied, the procedure is halted. Otherwise the procedure above is iterated from the evaluation step.

In addition to the above procedure, we employ the elitist strategy where the elite individual with the highest fitness is placed on randomly selected cell in the next generation.

3　Several Variants of Cellular GA

In this section, we introduce several variants of cellular genetic algorithms. Those are indivisual sorting, randomized neighborhood structure, and the hybridization with island model.

3.1　Individual Sorting

Basically, no individual in the grid-world moves from its position during the course of the execution of the algorithm. That is why the convergence speed of the cellular genetic algorithms is usually slower than the simple genetic algorithms without a grid-world structure. In the individual sotring, individuals in each row of the grid-world are sorted according to their fitness values every some pre-specified generations.

The sorting of the row is performed uniformly for the grid-world. That is, all the rows in the grid-world are sorted in the same order. For example, when we sort the first row such that the individual with the lowest fitness values are placed on the leftmost cell in the row, the remaining rows are also sorted in the same manner as in the first row. In the computer simulations in this paper, the individual sorting occurs every 50 generations.

3.2　Randomized Neighborhood Size

In the basic cellular genetic algorithms, the neighborhood size has to be specified before the execution. Moreover, the neighborhood size should be appropriate in order to obtain good results. The best neighborhood size is usually determined by trial and error in preliminary experiments. In this subsection, we explain the randomized neighborhood size method in order to reduce the number of the parameters in the algorithm.

In this method, the neighborhood size for each cell is determined before the execution of the algorithm. The neighborhood size is fixed and never changed during the algorithm.

3.3　Hybridization with Island Model

In this subsection, we propose a hybrid method with island model. The island model is viewed as a coarse-grained genetic algorithm. In our hybrid method, each individual is located on a single cell in a grid-world as in the basic cellular genetic algorithm. The grid world is divided into four subgrid-world as in Fig. 2.

This subgrid-world is viewed as an island where a sub-population is evolved independently from the other islands except some part of the grid-world. In Fig. 2, the marginal cells (shaded area) of each island interact with its neighboring cells including ones from the other islands. This interaction between different islands serves as imigrant operation in coarse-grained genetic algorithm.
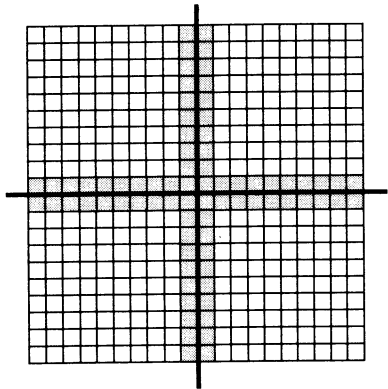
Figure 2: Proposed hybrid method.

The neighborhood size of the marginal cells (shaded area in Fig. 2) is always 1 (see Fig. 1). That is, the marginal cells do not have direct influence only on the marginal cells from the other cells.

4 Computer Simulations

4.1 Problems

In the computer simulations in this paper, we examined the performance of each algorithm in Section 3 for traveling salesman problems. We used Reinelt's TSP library which is available via the internet [6]. Particularly, we examined the performance for two problems in [6]: Eil51 and KroA100. Eil51 is a 51 city problem for which the optimal distance (i.e., the optimal tour with the minimum distance) is 426. The optimal distance for KroA100 with 100 cities is 21282. Fig. 3 shows the optimal tour for each problem.

4.2 Performance Evaluation

First, we evaluated the performance of the basic cellular genetic algorithm with the elitist strategy in Section 2. A 20×20 grid-world was used. That is, the population size was $20 \times 20 = 400$. Partially matched
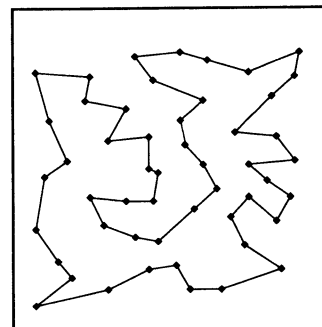
(a) Eil51
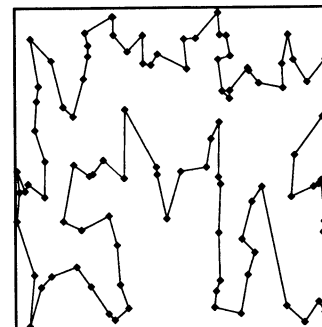
(b) KroA100

Figure 3: Optimal tours.

crossover operator was used for generating new individuals. For mutation we used the inversion operation. The crossover rate P_C and the mutation rate P_M was specified as $P_C = 1.0$ and $P_M = 0.1$, respectively. We terminated the algorithm after the 1000th generation. The neighborhood size N were specified as $N = 1, 3, 5$, and 19. Note that the basic cellular genetic algorithm with the neighborhood size $N = 19$ is the same as a simple genetic algorithm with no cellular structure.

We show the performance of the basic cellular genetic algorithm in Table 1. Table 1 is the average results over 10 trials, the best results, and the worst results.

Next, we show the performance of the individual sorting in Subsection 3.1. Table 2 shows the results of the individual sorting executed in the same manner as for the basic cellular genetic algorithm.

From Table 2, we can see that the performance of the individual sorting is better than that of the basic cellular genetic algorithm. Furthermore, we can see that the best individual in the individual sorting is better than that in the basic cellular genetic algorithm and the worst individual in the individual sorting is worse than the basic cellular genetic algorithm. This

Table 1: Performance of the basic CGA.

Problem	N	Average	Best	Worst
Eil51	1	452.0	437	477
	3	455.2	441	471
	5	459.5	441	480
	19	453.9	446	475
KroA100	1	24311.1	23265	26289
	3	28526.2	24736	31022
	5	28474.9	26441	30631
	19	29841.1	27268	34021

Table 2: Performance of the individual sorting.

Problem	N	Average	Best	Worst
Eil51	1	452.8	441	467
	3	449.4	438	470
	5	458.1	444	476
	19	451.8	436	466
KroA100	1	24185.0	21875	25283
	3	27800.9	26709	30136
	5	28195.3	26022	31322
	19	29009.9	26256	32433

suggests that we can obtain more diverse population by the individual sorting than by the basic cellular genetic algorithm.

We also applied the randomized neighborhood size to the cellular genetic algorithm. Table 3 shows the results.

Table 3: Performance of the randomized neighborhood size.

Problem	Average	Best	Worst
Eil51	458.9	441	503
KroA100	28740.7	26213	30182

Finally, we show the results of the hybridization with an island model. We performed the proposed method in the same manner as the above computer simulations for the basic cellular genetic algorithm, the individual sorting, and the randomized neighborhood size. The only exception is that we examined the neighborhood size $N = 1, 3, 5$, and 9 because in the proposed algorithm each island is composed of a 10×10 grid-world. Table 4 is the results of the proposed algorithm.

From the comparison between Table 1 and Table 4, we can see that the proposed algorithm could obtain better results than the basic cellular genetic algorithm in terms of the average, the best, and the worst indi-

Table 4: Performance of the proposed method.

Problem	N	Average	Best	Worst
Eil51	1	451.6	437	483
	3	450.0	437	470
	5	454.4	445	462
	9	456.6	441	485
KroA100	1	23577.3	22350	25498
	3	25661.6	23703	27596
	5	25249.2	23374	26971
	9	25473.8	23918	27729

viduals.

5 Summary

In this paper, we examined the performance of various versions of cellular genetic algorithms. They were individual sorting, randomized neighborhood size, and the hybridization with an island model. The computer simulations on traveling salesman problems showed that it is possible to improve the performance of cellular genetic algorithms by modifying it in various ways. The future research includes the dynamical analysis of the behavior of each variants. This will help us developing more sophisticated cellular genetic algorithms.

References

[1] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, NY, 1989.

[2] L. Davis, Handbook of Genetic Algorithms, Van Nostrand, NY, 1991.

[3] E. Cantú-Paz, "A survey of parallel genetic algorithms", IlliGAL Report, No. 97003, 1997.

[4] T. C. Belding, "The distributed genetic algorithm revisited", Proc. of the 6th International Conference on Genetic Algorithms, pp. 114–121, 1995.

[5] D. Whitley, "Cellular genetic algorithms", Proc. of the 5th International Conference on Genetic Algorithms, p. 658, 1993.

[6] G. Reinelt, TSPLIB, downloadable from http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

An Analysis on the Efficiency and Complexity of Genetic Algorithm

Shoukun Wang
Department of Automation
Tsinghua University, Beijing, 100084, People's
Republic of China
wangsk@proc.au.tsinghua.edu.cn

Jingchun Wang
Department of Automation
Tsinghua University, Beijing, 100084, People's
Republic of China
wang-jc@proc.au.tsinghua.edu.cn

Abstract

This paper defines a performance measurement of optimization algorithms based on the probability P to find global optimal, and calls it P-performance. The P-performance of random search is deduced in function optimization. Comparisons of the P-performance between genetic algorithm and random search by DeJong's five testing functions and traveling salesman problem are constructed, and computational results show that the complexity difference between genetic algorithm and random search is exponential.

Keywords: Genetic algorithm, Complexity, Global optimization

1. Introduction

Genetic algorithms have most commonly been considered as simple, robust and global optimization methods, and therefore could reach the global optimal more efficiently. But as we all know, random search is more robust, simpler and also could reach global optimal when running time approaches infinite. Actually, there are rarely comparisons of performance between genetic algorithms and random search in literature. Since both methods are based on stochastic models, it is hard to get a formula of computational complexity by the commonly used definition of complexity. DeJong[1] defined two performance measurements of search methods based on stochastic models, one is on-line performance, and the other is off-line performance. Lewchuk[2] presented the performance line between the current best solution of the algorithm and evaluations. However, computational results showed that these performance measurements could not explain the efficiency of genetic algorithms to random search. As we can see in Figure 1-1 of DeJong's comparison between random search and genetic algorithms with different population sizes, the x-axis is off-line performance, the y-axis is function evaluations, and N is population size. The performance curve of random search is over all those of genetic algorithms, but the curves are so close that the difference between random search and genetic algorithms is even smaller than that of genetic algorithms with different population sizes.

In this paper, we define a performance measurement based on the probability P to find global optimal, and call it P-performance. In section 2, we give the definition of P-performance and deduce the formula of random search. In section 3, we compare the P-performance between genetic algorithms and random search by DeJong's five testing functions and a combinatorial optimization problem, traveling salesman problem. Computational results show that the performance difference between genetic algorithms and random search is exponential. Section 4 is conclusion and further research plan.
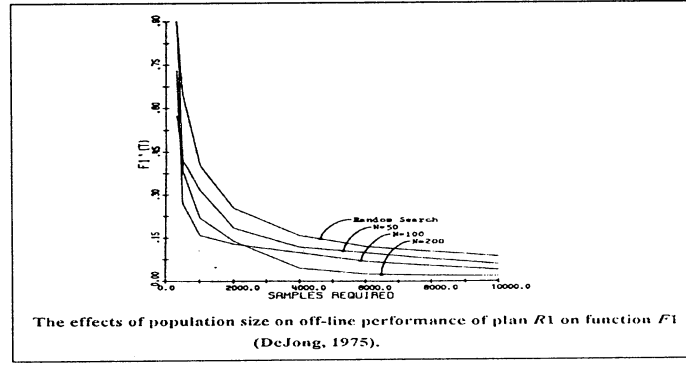
The effects of population size on off-line performance of plan R1 on function F1 (DeJong, 1975).

Figure 1-1 DeJong's off-line performance between random search and genetic algorithms[1]

2. Definition of P-performance

DeJong's on-line performance is defined as the average of all solutions in a run, and off-line performance is defined as the average of best solution in each generation. Since we prefer optimal to average performance in global optimization and considering that the genetic algorithms does not guarantee global optimal in each run, we define P-performance as follows.

Definition 2.1 Given a constant real number p ($0 < p \leq 1$), if algorithm A reaches the global optimal with probability p by precision ε, then the average function evaluations N is called P-performance of algorithm A, noted as $\bar{N}(A_\varepsilon)_{P=p}$.

Without loss of universality, we scale the region of function $f(x)$ to [0, 1] and define the random search method as randomly select and evaluate N points from region [0, 1], and choose the best one as global optimal. Then we get the following theorem.

Theorem 2.1 Suppose $f(x)$ satisfies Lipschitzian continuity condition and given any real number ε, when $\varepsilon \to 0$, the P-performance of random search is $O(1/\varepsilon)$.

Proof: The N selected points X_1, X_2, ... X_N are i.i.d. random variables obeying uniform [0, 1] distribution. X^* is the global optimal of $f(x)$, according to the Lipschitzian

continuity condition $|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|$ and L is Lipschitz constant,

$$p = 1 - P(\bigcap_{i=1}^{N} \{|X_i - X^*| \geq \varepsilon\})$$

$$= 1 - \{ P(\bigcap_{i=1}^{N} \{|X_i - X^*| \geq \varepsilon\}|X^* \leq \varepsilon) + P(\bigcap_{i=1}^{N} \{|X_i - X^*| \geq \varepsilon\}|\varepsilon < X^* \leq 1-\varepsilon)$$

$$+ P(\bigcap_{i=1}^{N} \{|X_i - X^*| \geq \varepsilon\}|1-\varepsilon < X^*) \}$$

$$= 1 - \frac{1-(\varepsilon+1)^{N+1}}{N+1} - (1-\varepsilon)(1-2\varepsilon)^N - \frac{(1-\varepsilon)^{N+1}-(1-2\varepsilon)^{N+1}}{N+1} \qquad (1)$$

When $\varepsilon \to 0$, $N \to \infty$ and $p \to 1 - (1 - 2\varepsilon)^N$, then

$$1 - p = 1 - N \cdot (2\varepsilon) + \frac{N(N-1)}{2}(2\varepsilon)^2 + \cdots$$

Omit the items higher than one-order, then

$$\bar{N}(RS_\varepsilon)_{P=p} = N = p \cdot 2/\varepsilon \sim O(1/\varepsilon) \qquad (2)$$

Omit the items higher than two-order, then

$$\bar{N}(RS_\varepsilon)_{P=p} = N = \frac{(2/\varepsilon+1)-\sqrt{(2/\varepsilon+1)^2-8p/\varepsilon^2}}{2} \sim O(1/\varepsilon).$$

□

Note 2.1: In computer simulation the solution is an n-bit binary string. Thus, $\varepsilon = 1/2^n$, and $\bar{N}(RS_\varepsilon)_{P=p} \sim O(2^n)$, which means that the average evaluations of random search algorithms is exponential times to the dimension of the problem to be optimized.
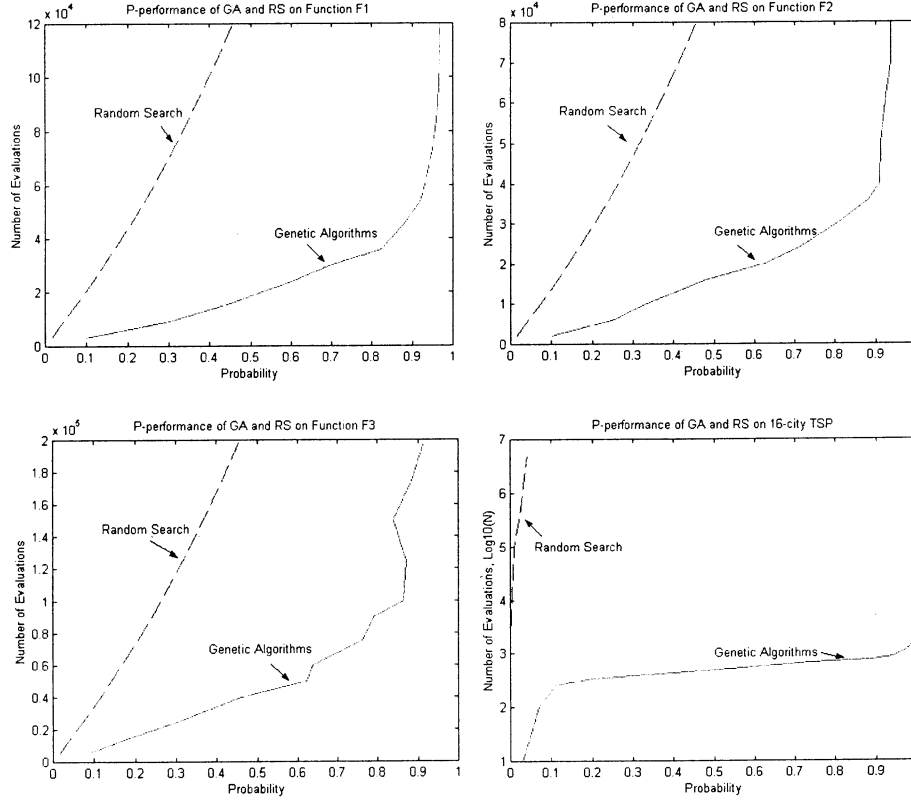
Figure 3-1 P-performance comparisons of genetic algorithms and random search

3. Computational Results

Since it is hard to deduce the exact P-performance formula of genetic algorithms, we compare the efficiency and complexity of genetic algorithms and random search through experiments. The testing problems are three of DeJong's five functions with and a 16-city Euclidean traveling salesman problem (TSP). We use SGA[3] with elitism selection strategy as a standard, and also try other strategies such as fitness linear scaling, adaptive probability of crossover and mutation, and multi-point crossover. For the functions, we use one 16-bit binary string to represent one variable, and compose the n binary strings to form a chromosome of the n-dimensional functions. thus $\varepsilon = 1/2^{16}$. For the TSP, we use one integer permutation to represent a chromosome, and try partial mapping crossover and order crossover. The followings are the three functions to be tested.

F1: $f(X) = \sum_{i=1}^{3} x_i$, $\quad -5.12 \le x_i \le 5.12$

F2: $f_2(X) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$,

$$-2.048 \le x_i \le 2.048$$

F3: $f_3(X) = \sum_{i=1}^{5} \lfloor x_i \rfloor \quad -5.12 \le x_i \le 5.12$

In the Figure 3-1 shown above, the first three figures are P-performance comparisons of genetic algorithms and random search on three functions, and the forth one is P-performance comparison on a 16-city TSP, the city coordinates of which are randomly generated. In the first three figures, x-axis is the probability to reach the global optimal, and y-axis is the total number evaluations. In the forth figure, x-axis is the same, but the y-axis is the logarithm of the total number of evaluations. In all the figures, the dash lines are P-performances of random search, and the solid lines are P-performances of

genetic algorithms.

The four figures clearly show that genetic algorithms can reach the global optimal with higher probability than random search by the same number of evaluations, especially for the combinatorial optimization problems.

From formula (2) in section 2, we know that for any fixed ε, the P-performance N is approximately $1/\varepsilon$ times of the probability, and their relation is linear. As we have introduced above, $\varepsilon = 1/2^n$. Thus, in the first three figures, if two linear lines have different slope, their P-performance complexity difference is exponential. The slope of P-performance lines of genetic algorithms has a sharp change in region (0.8, 0.9), which is a sign of pre-maturity. Before the pre- maturity region, the slope of genetic algorithms is much smaller than that of random search, showing that genetic algorithms are more efficient. Due to the limitation of stochastic models, the slope of genetic algorithms is nearly as high as that of random search after pre-maturity region.

As for the 16-city randomly generated TSP, the slope of P-performance of genetic algorithms is almost zero, while that of random search is nearly infinite. It shows that the genetic algorithms are suitable for that kind of combinatorial optimization problems. Since the valid solutions of an n-city TSP is $O(n!)$, it is very hard for the random search algorithm to reach the global optimal, but the genetic algorithms can acquire useful information from the parents' generation and then evolve more efficiently.

4. Conclusion

In this paper, we present a performance measurement, called P-performance, of stochastic search algorithms based on the probability to reach global optimal, deduce the P-performance of random search, and show the efficiency of genetic algorithms in function and combinatorial optimization through computational results.

P-performance focuses on the ability to find the global optimal, while on-line and off-line index measure the average performance. Since we concern more about the global optimal, P-performance can bring us more information on the efficiency and complexity of genetic algorithms. From the figures in section 3, we think that the P-performance could partly explain that why genetic algorithms are widely in combinatorial optimization problems but seldom in function optimization.

Though the P-performance introduces an initial step in analyzing the efficiency and complexity of stochastic modeling algorithms, the optimization scheme of genetic algorithms is still an open question. Now we are working on the theoretical complexity and convergence of genetic algorithms.

Reference

[1] DeJong, K. A. An analysis of the behavior of a class of genetic adaptive systems. Doctoral dissertation, University of Michigan, 1975.

[2] Lewchuk, M. "Genetic Invariance: A new type of genetic algorithm", Technical Report TR 92-05, Edmonton, Alberta: Department of Computer Science, The University of Alberta.

[3] Holland, J. H. Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1975.

Efficient numerical optimization technique
based on real-coded genetic algorithm for inverse problem

Takanori Ueda[1]   Nobuto Koga[2]   Isao Ono[3]   Masahiro Okamoto[1]
ueda@brs.kyushu-u.ac.jp   nob@brs.kyushu-u.ac.jp   isao@is.tokushima-u.ac.jp   okahon@brs.kyushu-u.ac.jp

[1] Graduate School of Bioresource and Bioenvironmental Sciences, Kyushu University,
6-10-1 Hakozaki Higashi-ku Fukuoka, 812-8581, Japan
[2] Dept. of Biochem. Eng. & Sci., Kyushu Institute of Technology, 680-4 Kawazu Iizuka Fukuoka, 820-8502, Japan
[3] Faculty of Engineering, The University of Tokushima, 2-1, Minami-josanjima, Tokushima, 770-8506, Japan

Abstract

We have developed an efficient algorithm based on real-coded genetic algorithm for optimization of a model of a nonlinear system. Estimation of the interaction mechanisms among system components by using experimentally observed dynamic responses (time-courses) of some of the system components is generally referred to as "inverse problem". The S-system, which belongs to power-law formalism, is one of the best representations to solve such an inverse problem; the S-system is rich enough in structure to capture all relevant dynamics. In this paper, for the purpose of solving the inverse problem, we introduce the real-coded genetic algorithm and propose an efficient procedure for the estimation of large numbers of parameters in the S-system formalism. We applied our method to a simple oscillatory system.

Keywords: numerical optimization, genetic algorithm, inverse problem, S-system

1   Introduction

Real-coded Genetic Algorithms (RCGA) attract attention as numerical optimization methods for nonlinear systems. One of the crossover operators for RCGA called unimodal normal distribution crossover (UNDX) has shown good performance in optimization of various functions including multi-modal ones and benchmark functions with epistasis among parameters [1]. The UNDX generates new population lie on some ponds or along some valleys in order to focus the search on promising areas from a viewpoint of search efficiency. Especially when the function has epistasis among parameters, namely valleys that are not parallel to coordinate axis, the UNDX can efficiently optimize it.

Here, we applied the UNDX to the optimization of real-valued parameters in the form of the S-system that is a type of power-law formalism and is suitable for description of organizationally complex systems such as gene expression networks and metabolic pathways. The S-system is based on a particular type of ordinary differential equation in which such component processes are characterized by power-law functions. When a

Genetic Algorithm (GA) is applied to optimization problems, characteristic preserving in designing coding/crossover and diversity maintaining in designing generation alternation are very important. Generation alternation models are independent of problems, while coding/crossover depends on problems. Simple GA is one of the well-known generation alternation models and Tominaga et al. [2] has developed the numerical optimization technique based on simple GA in the form of the S-system. However, simple GA has two problems. One is early convergence in the fast stage of search and the other is evolutionary stagnation in the last stage of it. Sato et al. [3] has proposed new generation alternation model called minimal generation gap (MGG) to overcome the above problems. The MGG has all advantages of convention models and the ability of avoiding the early convergence and suppressing the evolutionary stagnation.

In this paper we propose efficient numerical optimization technique with the combination of the UNDX and the MGG in order to infer the interactions in gene expression networks.

2   Method

2.1   S-system

The S-system [4] is one of the best formalisms to estimate such interaction mechanisms among system components, and enables us to reconstruct genetic or reaction network architectures with the experimentally observed time-courses of the patterns of gene expression [2][5] or biochemical reactions. The S-system belongs to the type of power-law formalism because it is based on a particular type of ordinary differential equation in which the component processes are characterized by power-law functions;

$$\frac{d}{dt}X_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} \qquad (1)$$

where n is the total number of state variables or reactants (X_i), i, j $(1 \leq i, \quad j \leq n)$ are suffixes of state variables. The

terms g_{ij} and h_{ij} are interactive effectivity of X_j to X_i. The first term represents all influences that increase X_i, whereas the second term represents all influences that decrease X_i. In a reaction network context, the non-negative parameters α_i and β_i are called relative inflow and outflow of reactant X_j, and real-valued exponents g_{ij} and h_{ij} are referred to as the interrelated coefficients between reactants X_j and X_i. The S-system formalism has a major disadvantage in that this formalism includes a large number of parameters that must be estimated (α_i, β_i, g_{ij} and h_{ij}) [5]; the number of estimated parameters in S-system formalism is $2n(n+1)$, where n is the number of state variables (X_i). We describe here an algorithm and procedures for the estimation (optimization) of large numbers of parameters [2][5]. The basic idea is as follows: the Genetic Algorithm (GA) [6] is introduced as a nonlinear numerical optimization method which is much less likely to be stranded in local minima. Furthermore, in order to find the skeletal structure (small-size system) of S-system formalism that matches the experimentally observed responses, some of the parameters (g_{ij} and h_{ij}), absolute values of which are less than a given threshold value, are to be removed (reset to 0) during optimization procedures. By introducing this algorithm referred to as structure skeletalizing [2][5], that optimized essential S-system model that matches the experimentally observed responses should be possible.

2.2 UNDX

When optimizing function has epistasis among parameters the UNDX generates children near the line segment connecting two parents so that the children lie on the valley where the parents locate (Fig. 1). The mathematical description can be written as follows:

$$c_1 = m + z_1 e_1 + \sum_{k=2}^{l} z_k e_k, \quad c_2 = m - z_1 e_1 - \sum_{k=2}^{l} z_k e_k$$

$$m = (p_1 + p_2)/2$$

$$z_1 \sim N(0, \sigma_1^2), z_2 \sim N(0, \sigma_2^2) \quad (k = 2, \cdots, l)$$

$$\sigma_1 = A d_1, \sigma_2 = B d_2 / \sqrt{l}$$

$$e_1 = (p_2 - p_1)/|p_2 - p_1|, \quad e_i \perp e_j$$

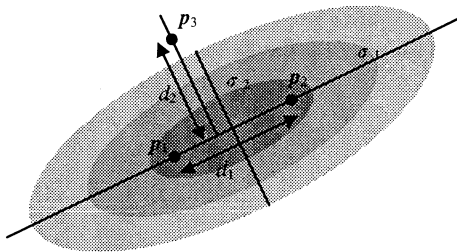$$(i, j = 0, \cdots n_{param}; i \neq j) \tag{2}$$

Figure 1: UNDX

where c_1 and c_2 are children, p_1 and p_2 are parents and m is the middle point of parents. The d_1 is the distance between parents and d_2 is the distance between the third parent p_3 (randomly selected) and the line connecting p_1 to p_2. The $z_1 \sim N(0, \sigma_1^2)$ and $z_k \sim N(0, \sigma_2^2)$ ($k = 2, \cdots, l$: number of estimated parameter) are normal distributed random numbers. The A and B are constants given by the user.

2.3 MGG

The MGG is the most desirable model which can avoid the early convergence and suppress the evolutionary stagnation. The procedure of MGG can be summarized as follows:

begin t = 0; 1) randomly create initial population P(t); 2) evaluate each individual in P(t);
while (not satisfy the terminal condition) **do** t = t + 1; 3) randomly select two parents C(t) from the P(t-1); 4) P(t-1) = P(t-1) - C(t); 5) create children C'(t) from the C(t); 6) evaluate each individual in C'(t); 7) select one elite individual and randomly select one individual among the C(t) and C'(t); 8) add one elite one individual and one randomly selected individual by the above procedure to the P(t-1) and set P(t); **end end.**

2.4 Optimization Task

Since the S-system is a formalism of ordinary nonlinear differential equation, the system can easily be solved numerically by using a numerical calculation program to be customized specifically for this structures [7].

However, when an adequate time-course of relevant state variable is given, a set of parameter values α_i, β_i, g_{ij} and h_{ij}, in many cases, will not be uniquely determined, because it is highly possible that the other sets of parameter values will also show a similar time-course. Therefore, even if one set of parameter values that matches the observed time-courses is obtained, this set is still one of the best candidates that explain the observed time-courses. Our strategy is to explore and exploit these candidates within the immense huge searching space of parameter values.

In this optimization problem, each set of parameter values to be estimated is evaluated using the following procedure: Suppose that $X_{i,cal,t}$ is numerically calculated time-course at time t of state variable X_i and $X_{i,exp,t}$ represents the experimentally observed time-course at time t of X_i. Sum the relative error between $X_{i,cal,t}$ and $X_{i,exp,t}$ to get the total error f

$$f = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(\frac{X_{i,cal,t} - X_{i,exp,t}}{X_{i,exp,t}} \right)^2 \tag{3}$$

where n is the number of experimentally observable state variables, T is the number of sampling points of the experimental data. The problem is to find a set of parameters that minimizes f.

The structure of the genome (design code) of each individual (each set of parameter values) is shown in Fig.2. A genome (corresponds to one individual) contains a set of S-system parameters (n α_is and β_is, and $n \times n$ g_{ij}s and h_{ij}s) which forms an $n \times (2n+2)$ matrix. An individual represents one S-system model. Each small square in Fig.2 corresponds to each parameter that has a real value.
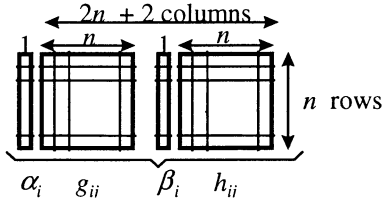

Figure 2: Design code of an individual; two n vectors of α_i and β_i, and two $n \times n$ matrices of $n \times n$ g_{ij} and h_{ij} form $n \times$ (2n+2) matrix. This matrix represents one S-system model.

3 Results

We evaluated the search efficiency of the proposed GA in comparison with the simple GA. As case study, we supposed two dimensional S-system (a simple oscillatory system, $n = 2$ in eq(1)) and estimated the 12 parameters (2 α s, 2 β s, 2×2 g-matrix and 2×2 h-matrix) from the calculated time-course of X_1 and X_2 of the S-system. These time-course shown in Fig. 3 and parameter values of which are shown in Table 1. Given these time-course of X_1 and X_2 (Fig. 3) is presented as experimental data (($X_{i,exp,t}$), $n = 2$, $T = 50$ in eq(3)), we compared three optimizing procedures including simple GA, MGG and MGG+UNDX to evaluate the search efficiency. The major optimizing conditions are as follows: searching ranges are [0.0, 5.0] for α_i and β_i, [-3.0,3.0] for g_{ij} and h_{ij}, population size is 100 for P(t) and C(t) individually, the constant values of A of UNDX is 0.5 and B of UNDX is 0.3 (see eq(2)). The optimization will be terminated when total error f of the elite individual is less than a given threshold value of convergence (10%, 7% and 5% error average per sampling point).

Table 1: Given S-system parameters which provided the dynamics shown in Fig. 3.

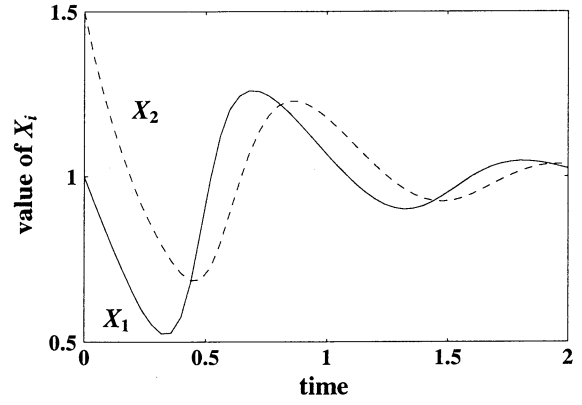| i	α_i	g_{i1}	g_{i2}	β_i	h_{i1}	h_{i2}
1	3.0	0.0	-2.5	3.0	-1.0	0.0
2	3.0	2.5	0.0	3.0	0.0	2.0

We executed optimization 20 times at the each

Figure 3: Given time-course of X_1 and X_2 (parameter values are listed in Table 1)

terminate conditions with three procedures respectively and averaged the explored generations and cpu time for calculation. Table 2 shows the case with 10% error average per sampling point. As for the generation average and time average, simple GA seems to be the best method, however, standard deviation of these values are the largest of the three cases, which means simple GA is not secured and stable method of optimization. Contrary to this, since standard deviation of generation average and time average is the smallest, MGG+UNDX avoids the early convergence, suppresses the evolutionary stagnation and can reach to the terminated conditions with high possibility. Table 3 shows the case with 7% error average per sampling point. In this case, MGG+UNDX indicates the best results of all. Table 4 shows the case with 5% error average per sampling point. This shows superiority of MGG+UNDX in searching efficiency. As shown in Table 4, the searching efficiency of MGG+UNDX is 9.1 times higher than that of simple GA, 2.0 times higher than that of MGG only.

Table 2: Obtained results in the case of 10% error average per sampling point. Time for optimization represents a cpu-time until the optimization was terminated, which was measured on Tempest 5 (Concurrent Systems Inc., Japan (processor:Alpha21264- 667MHz, SPECfp95: 75.3, SPECint95: 37.0)).

	simple GA	MGG	MGG+UNDX
Generation Avg.	131.70	267.35	178.80
Generation S.D.	219.07	97.99	72.50
Time Avg. (sec)	13.31	30.95	20.15
Time S.D.	21.27	12.43	9.12

Table 3: Obtained results in the case of 7% error average per sampling point.

	simple GA	MGG	MGG+UNDX
Generation Avg.	1728.85	670.20	379.25
Generation S.D.	1876.10	174.28	106.02
Time Avg. (sec)	170.52	82.36	46.58
Time S.D.	184.77	21.86	14.2

Table 4: Obtained results in the case of 5% error average per sampling point.

	simple GA	MGG	MGG+UNDX
Generation Avg.	5348.15	1186.45	585.60
Generation S.D.	8041.93	392.32	146.09
Time Avg. (sec)	533.42	150.20	75.07
Time S.D.	793.15	53.14	19.74

Acknowledgement

References

[1] Ono I, Kobayashi S (1997), A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, Proc. 7th ICGA, 246-253

[2] Tominaga D, Koga N, Okamoto M (2000), Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem, Proc. of the Genetic and Evolutionary Computation Conf. (GECCO2000), 251-258

[3] Sato H, Ono I, Kobayashi S (1997), A New Generation Alternation Model of Genetic Algorithms and Its Assessment, J. of Japanese Society for Artificial Intelligence, 12(5):734-744

[4] Savageau M. A (1976), Biochemical Systems analysis: a study of function and design in molecular biology (Addison-Wesley, Reading)

[5] Tominaga D et al. (1999), Nonlinear Numerical optimization technique based on a genetic algorithm for inverse problem: Towoards the inference of genetic networks, Computer Science and Biology (Proc of the German Conf on Bioinfomatics (GCB'99)), 127-140

[6] Davis L (1991), Handbook of genetic algorithms, Van Nostrand Reinhold, NY

[7] Irvine D.H, Savageau M.A (1990), Efficient solution of nonlinear ordinary differential equations expressed in S-system canonical form. SIAM J Numerical Analysis, 27:704-735

Improvement of the efficiency of Distributed GA

Masanori SUGISAKA and Kazuki MORI

Department of Electric and Electronics Engineering, Faculty of Engineering, Oita University

700 dannoharu , Oita 870-1192, JAPAN

Tel: +81-975-54-7841   FAX: +81-975-54-7818

E-mail: msugi@cc.oita-u.ac.jp , kamori@cc.oita-u.ac.jp

Abstract:

In this research, we studied about the improvement of the efficiency in Distributed-Genetic Algorithm (DGA). DGA divides population into more than some sub-population and it does genetic operation every sub-population. Lastly, it does the replacement (migration) of individual among sub-population every constant period (migration interval). In DGA, the division of the population maintains the variety of the solution and various schemata grow every sub-population. The better solution is generated by crossover among those schemata with migration. When comparing with single population GA, it is reported a high quality solution is gotten. This time, it proposes elite sub-population generation type DGA and virus type evolution as the new technique and it is comparing a performance with conventional DGA. But we don't obtained good result to do the method. However, A good result was gotten by using best combination crossover (BCX). Otherwise in virus type DGA, a rarely good result comes out.

Key word: Distributed GA, virus type evolution, elite sub-population.

1. Introduction

The Genetic Algorithm (GA) is the probable optimal algorithm imitating evolution of a living thing. This technique can be adapted for the complicated continuation and a dispersed problem which was difficult to solve by the conventional optimization technique. Furthermore, there is the strong point in which practical use is comparatively easy. However, since GA needs huge repeated calculation, its calculation cost is high.

Then, Distributed GA (DGA) which can calculate a solution in the amount of calculation fewer than the conventional GA according to the distributed effect attracts attention.

Regarding DGA, the research about various crossover and migration has been reported until now. However, we have still desired the improvement of efficiency about DGA. Therefore, we investigate the generation of sub-population, the interval and the time of the migration in order to improve the efficiency of DGA.

2.Distributed GA

2.1 Genetic Algorithm

In GA, a collection of individuals is called population and it is chosen the higher individual of fitness to environment survives by high probability among the population form a certain generation. Furthermore, the next generation is formed of crossover and mutation between individuals. By repeating renewal of such a generation, a better individual (individual near the optimal solution)

increases, and the optimal solution is approached soon.

2.2 Distributed Genetic Algorithm

In DGA, a population is divided into some sub-population, and genetic operation is independently performed for every sub-population. Also, we exchange the individual information, what is called migration, among sub-population every constant interval. There is Migration interval and Migration rate in DGA as a peculiar parameter. Migration interval is defined as the generation interval to migrate and migration is defined as the rate of the individual to migrate. As for featuring DGA, the variety of the solution is maintained by the division of population, and various schemas grow for every sub-population, and a better solution is generated by intersection of those schemas by migration. In case that we want to get the solution with high accuracy, DGA is better than single-population GA.

2.3 The growth of the solution in DGA

In single-population GA, when population has a large number of individual with fairly good schema, growth of other good schemas may be barred and premature convergence may be caused. Therefore, it is necessary not only to save good schemas, but also to destroy them occasionally and to perform large region-search. If we observe sub-population in DGA, since each number of individuals becomes small, it will be easy to start premature convergence. However, as the whole population, the diversity of a solution can be maintained by exchanging the individual which carried out different evolution among some sub-population called migration. In DGA, by migration, the schema generated for every sub-population combines with the schema of other sub-population, and grows. Thereby, DGA can calculate a quality solution at high speed.

3. Experimental method

3.1 Elite sub-population generation type DGA

In the usual DGA, only migration from one sub-population other than itself is received.
And the schema which is growing by other sub-population is not obtained. However, if migration from many sub-population is performed, a possibility of breaking the schema which is growing by the sub-population will arise. Therefore, we thought of what generates an elitist sub population. The schema which is growing in each sub-population in it is combined.

In addition to usual sub-population, as a view, another sub-population called elite sub-population is generated. (Fig.1) And an elite individual is chosen from the inside of the individual group which evolved in each sub-population, and they are collected to elite sub-population. We performed genetic operation to them and we think that a still better individual will be generable. We think that promotion of evolution and quality improvement of a solution are expectation by performing this operation.
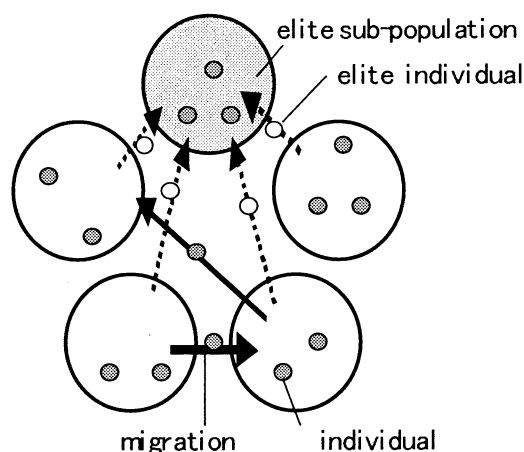

Fig.1 Elite sub-population generation type DGA

3.2 Virus type evolution

The individual in which more various schemas

exist will be generated by generating elite sub-population. It proposes virus type DGA to use it effectively.

One individual is first chosen at random in elite sub-population. A part of gene arrangement of the individual is treated as a virus. And an individual is chosen for every sub-population, and a virus is infected into it. Then, although crossover and mutation are performed ordinarily, the portion of a virus is compulsorily inherited by the child. However, the gene arrangement of a virus does not have a good schema in practice, or even if it combines with the schema which grew by the sub-population, it may not evolve. So, we proceed our study in next step: we set up the parameter of vital power for virus itself. If fitness is better than before, we make virtual power for the same virus or up, but if not, we decrese the number of virus. If virus gives good influences to advances, it can live or spread out in population.

Usually, the difference has arisen in the schema which is growing in each population. For this reason, efficiency may fall by this method of choosing a virus at random. However, we think that those things are stopped to the minimum by using BCX. And when the gene arrangement containing a best schema is chosen as a virus, we think that the effect shows up greatly. Also, since a virus is also changed for every migration, we think that a possibility that the virus which is adapted for sub-population will be born is high.

3.3 Crossover

In this experiment, the usual one-point crossover is mainly used. In GA, there are two-point crossover, and uniform crossover, etc. However, although there is the same thing also by the usual SGA case, a schema may be broken by performing two-point crossover and uniform crossover. Since it sees notably in the tendency in DGA, it is mainly experimenting by one-point crossover this time. And it is experimenting by using BCX in order to develop the efficiency of DGA, and to employ the feature efficiently.

3.3.1 Best Combinatorial Crossover (BCX)

In order to raise the performance of DGA, it becomes important to grow up the schema which is growing in each sub-population. Therefore, one of the optimal intersections is best combinatorial crossover (BCX). In BCX, all the individuals generated by one-point crossover from a parent individual are evaluated, and two individuals with highest fitness are adopted as a child out of parent individuals and those child individuals Therefore, it is possible to generate the high individual of fitness, without being influenced by the probable factor, and the most effective one out of the schema which the parents have can be chosen. However, there is a fault that the number of times of evaluation calculation becomes huge.

4.Simulation

In this time, it is comparing a performance in proposed DGA and conventional dispersion. It uses a standard function in GA which DeJong proposed as the object problem (F1~F3).

$$\text{F 1} : -\sum_{i=1}^{n} x_i^2 \qquad (-5.11 \leqq x_i \leqq 5.12)$$

$$\text{F 2} : -\{100(x_1^2 - x_2)^2 + (1 - x_1)^2\}$$
$$(-2.047 \leqq x_i \leqq 2.048)$$

$$\text{F 3} : -\sum_{i=1}^{n} [x_i] \qquad (-5.11 \leqq x_i \leqq 5.12)$$

5. Experiments and Results

As a result of the experiment, the following graph

was gotten. (Fig.2 Fig.3)

Elite sub-population generation type DGA shows the high rate of convergence in the early convergence situation compared with the usual DGA.Moreover, even when the number of generations progresses, it turns out that high fitness is maintained. However, it doesn't converge when the generation progresses.
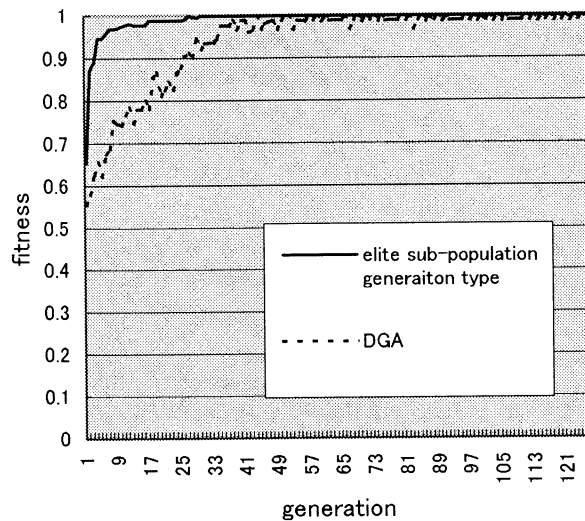

Fig.2 The situation of convergence

Then, the experiment result at the time of using BCX for this is as follows. (Fig.2)

It came to converge more from above result. The good result stabilized as a whole is obtained.

Moreover, elite sub sub-population generation type DGA also affects the number of sub-population. The graph shown in Fig.3 compares the number of generations with the number of sub-population until it finds out the optimal solution in each technique. On this problem, the proposed technique has arrived at the optimal solution with the sub-population number fewer than the usual DGA. The good result comes out about the number of the generations, too. This is considered to be the result with which the schema combines well by elite sub-population. Otherwise a rarely good result comes out in a virus evolution type DGA.

6.Conclusions

In elite sub-population generation type DGA, the good result was able to be obtained by using the BCX. Otherwise in the virus type evolution DGA, we didn't obtained good result because the result influenced by probably element.
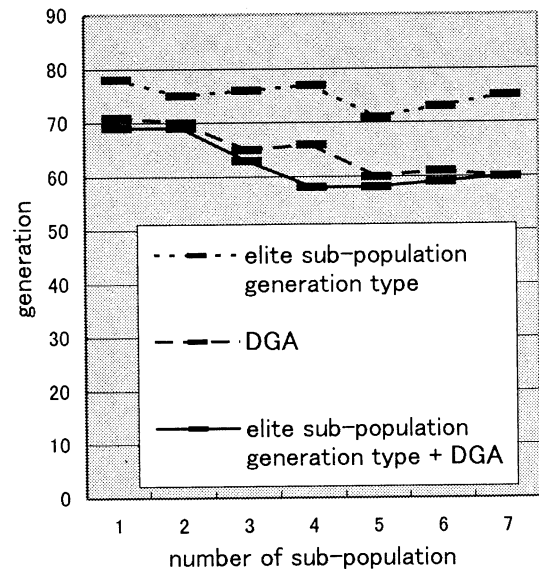
Fig.3 Comparison to the best solution

References

[1]D. whitley , K. Mathias & J. Dzubera "Building Better Test Functions" *International Conference on Genetic Algorithms.* 1995.

[2] Mitsunori MIKI, Tomoyuki HIROYASU, Junichi YOSHIDA and Ikki OMUKAI "Optimal Crossover Schemes for Parallel Distributed Genetic Algorithms"
MPS symposium 2000.03

[3] Mitsunori MIKI, Tomoyuki HIROYASU, Junichi Yoshida and Ikki Omukai "Best Combinatorial Crossover in Genetic Algorithms"
MPS seminar, vol.31, 2000.09

[4] Tomoyuki HIROYASU, Mitsunori MIKI and Masami NEGAMI,"Distributed Genetic Algorithm with Asynchronous", ISCIE, (1999) pp.379-380;

Bridging nonliving and living matter

S. Rasmussen (Los Alomos National Laboratory, USA)

An Origin of Combinatorial Complexity in Replicator Networks

Takashi Ikegami
Department of General Systems Sciences,
The Graduate School of Arts and Sciences,
University of Tokyo, 3-8-1 Komaba, Meguro-ku,
Tokyo 153-8902 Japan
(ikeg@sacral.c.u-tokyo.ac.jp)

A dynamical systems understanding of heritability at the ecosystem level is discussed. Over the years, experimental studies of the higher level group selection have been examined [1]. Many of them have reported that there exist significant responses against group selection with heritable variations. Different from individual selection mechanism, the ecosystem has no genetic bases. Indeed, we see a temporal loss of heritability in the experiments by Swenson et al [2]. In order to explain the higher level heritability, we propose a dynamical systems theory based on the high-dimensional replicator equation with mutation terms [3, 4]:

$$\frac{dx_i}{dt} = x_i \left(\sum_j a_{ij} x_j - \sum_k \sum_l x_k a_{kl} x_l \right) - \mu x_i \frac{1}{N-1} \sum_{k \neq i} x_k$$

(1)

In the equation, the variable x_i denotes a relative frequency of phenotypic variant i with the condition that $\sum_{k=1, k=N} x_k = 1$ We here assume that any variant is generated from any other variant, which is introduced by the mutation term in the above.

Simulating this equation with a random matrix and $N = 100$ possible variants, we report that this system tends to have many attractors of fixed point type to chaotic ones. In addition, their basin boundaries appear to be riddled. With this system, we examine a heritability check experiment. Different from experimental group selection experiments, we don't use any selection criterio n. We sample a small population ($x'_k = int(N_2 x_k)$) of the system to create the next generation of the system when the system reaches an attractor. By the succesive creation of the system, we discuss the heritability of this system. As a conclusion, we report that gene-like traits appear to control the heritability of the system. In particular, gene-like traits control the transition from one attractor to the other in a combinatorial way (See the figure caption
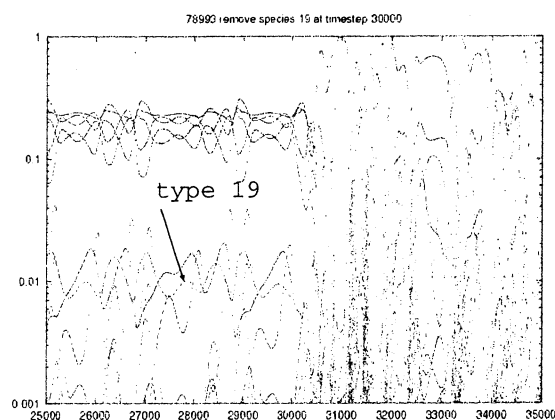
Figure 1: Time-based evolution of population in a log scale, plotted against generation steps. Every phenotype is superimposed. When phenotype 19 (with a wider line) is removed at generation 30000, the entire structure abruptly collapses and switches to a quasi-periodic attractor.

in Fig.2). Based on this finding, a relation between evolvability [5] and heritability will be discussed. Details of this paper will appear in [6].

References

[1] C.J.Goodnight, "Heritability at the ecosystem level" commentary PNAS vol 97 no 17 (2000) pp.9365–9366.

[2] W.Swenson, D.S. Wilson, and R.Elias, "Artificial ecosystem selection" PNAS vol 97 no 16 (2000) pp.9110-9114.

[3] K.Hashimoto and T.Ikegami, "Heteroclinic Chaos, Chaotic Itinerancy and Neutra l Attrac-

Figure 2: A result of controlling neutral phenotypes of the seven lowest reproduction rates. We have used 127 combinatorial removing/non-removing events of those seven neutral types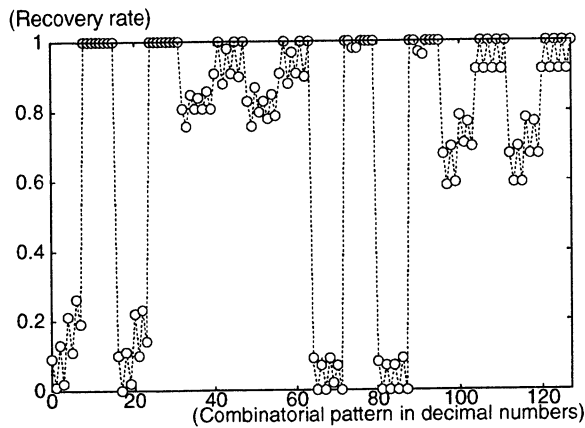. The vertical line denotes the recovery rate of the chaotic attractor after the combinatorial removal of the phenotypes. The combinatorial pattern is decimally encoded on the horizontal line. We have examined 100 events for each combination and the recovery rate has been averaged. See for example number 87, which corresponds to a removal of phenotypes 19 and 76, and the resultant rate is zero: that is, it never recovered the original attractor.

tors in Symmetrical Replicator Equations with Mutation", J.Phys. Soc. Japan Vol.70 (2001) pp.349–352.

[4] T.Ikegami and E.S.Yoshikawa, Chaos and Evolution of Cooperative Behavior in a Host-parasite Game in Towards the Harnessing of Chaos (ed. M.Yamaguchi, Springer)pp.63–72. 1995.

[5] T.Ikegami, "Evolvability of Machines and Tapes", J. Artificial Life and Robo tics vol. 3 No. 4 (1999) pp.242–245.

[6] T.Ikegami and K.Hashimoto, Dynamical Systems Approach to Higher-level Heritability. (to appear in j. Biol. Phys. (2001))

Quantifying the Environment for Adaptation

M. A. Bedau
Department of Philosophy, Reed College, Portland OR 97202, U.S.A.
Department of Zoology, University of Oklahoma, Norman OK 73019, U.S.A.
mab@reed.edu, http://www.reed.edu/~mab

Abstract

We measure the environment that is relevant to a population's adaptation as the information-theoretic uncertainty (Shannon entropy) of the distribution of local environmental states that the adapting population experiences. Then we observe the dynamics of this quantity in simple models of sensory-motor evolution. Although the distribution of resources is static, the agents' evolution creates a dynamic environment for adaptation.

Keywords: environment, adaptation, entropy.

1 The Environment for Adaptation

The process of adaptation is driven by the environment, but it also shapes the environment. If we focus on the *environment for adaptation*—the aspect of the environment that is relevant to adaptive evolution— we would tend to agree with Lewontin [6] that organisms *construct* their environment. But to understand how organisms and their environment affect one another, we must first clarify what the environment for adaptation is. How can it be measured? How is it affected by the behavior of the evolving population? This paper takes a first step toward answering these questions (see [2] for more details). We propose how to measure the environment for adaptation, and we observe its dynamics in evolving systems with fixed resource distributions—what we call *static resource models*. We see that even simple systems with static resource distributions have a dynamic environment for adaptation.

Classifying and quantifying the environment for adaptation is the subject of much recent work in artificial life. The relevance of this work includes both its implications for theoretical insights into the evolutionary process and its practical consequences for engineering and controlling evolving systems. Wilson [11] and Littman [7] provide an abstract categorization of environments, with Wilson focusing on the degree of non-determinism in an environment and Littman characterizing the simplest agent that could optimally exploit an environment. But neither method provides a dynamic quantitative measure of the environment the population experiences. Other recent work experimentally investigates agents adapting in different environments. Todd and Wilson [9] introduce an experimental framework for investigating how adaptation varies in response to different kinds of environments, and Todd et al. [10] demonstrate different adaptations in different kinds of environments. In neither case, though, is environmental structure actually classified or measured. The present project grows out of previous work on how adaptability depends on quantitative measures of environmental structure [1, 4, 5]. What is novel here is the focus on the environment actually experienced, which allows the measures to be dynamic where the previous ones were static.

2 Packard's Static Resource Models

All of our empirical observations are from computer simulations of a certain model—originated by Norman Packard [8, 3]—that is designed to be a very simple model of the evolution of sensory-motor strategies. Packard's model consists of agents sensing the resources in their local environment, moving as a function of what they sense, ingesting the resources they find, and reproducing or dying as a function of their internal resource levels. The model's spatial structure is a grid of sites with periodic boundary conditions, i.e., a toroidal lattice. Resources are immediately replenished at a site whenever they are consumed. The agents constantly extract resources and expend them by living and reproducing. If an agent's internal resource supply drops to zero it dies, and it reproduces if its reservoir exceeds a certain threshold. Here we restrict resources to blocks: a square grid of lattice sites which all have the same fixed quantity of resources. In a resource distribution formed of

blocks, every site that is not inside a block is part of a resource-free desert. We focus on two distributions, one with many small blocks randomly sprinkled across space, the other with one large block. Each agent moves each time step as dictated by a genetically encoded sensory-motor map: a table of behavior rules of the form IF (environment j sensed) THEN (do behavior k). An agent receives sensory information about the existence of resources (but not the other agents) in the von Neumann neighborhood of five sites centered on its present location in the lattice. Thus, each sensory state j corresponds to one of $2^5 = 32$ different detectable local environments. Each behavior k is a jump vector between one and fifteen sites in any one of the eight compass directions, or it is a random walk to the first unoccupied site, so an agent's genome continually instructs it to move somewhere. More details about the model are available elsewhere [2].

3 A Measure of the Environment

We are interested in how best to measure the environment that is relevant to a population's adaptation. For the sake of concreteness, we will develop and apply our methods in the context of the static resource models described above. The quantities we define and observe fit within a family of measures that has been developed and studied in similar contexts previously [1, 4, 5].

The basic idea behind our measure of the environment for adaptation is quite simple. The first step is to define a partition of local environments that are relevant to adaptation. For example, in Packard's model studied here, a natural partition of local relevant environments is the set of different von Neumann neighborhoods that the agents can distinguish (the presence or absence of resources in each of the five detectable neighboring sites). At each time each agent is in exactly one of these local states. The states affect the agents by being their sensory input, and the agents affect the states through their behavior [1]. The states are relevant to the agents' adaptation, for the agents' adaptive task is to evolve a genome that associates each of those states with an appropriate behavioral rule in a sensory-motor strategy and natural selection shapes only the rules for those von Neumann neighborhoods that are actually experienced by the population.

During the course of evolution, the agents experience states in the partition of local relevant environments with different frequencies. The frequency (probability) distribution of the states is a straightforward reflection of the environment that is currently relevant

for the population's adaptation. That is, let $\{e_i\}$ be the partition of local relevant environments, and let $P^E(e_i)$ be the frequency with which agents in the population experience the ith local state e_i (during some time window) in the global environment E. Then the probability distribution P^E reflects the nature of the environment that is *currently relevant* to the population's adaptation. In Packard's model where $\{e_i\}$ is the von Neumann neighborhoods, P^E shows the frequencies with which the population visits different neighborhoods.

The final step in our measure of the environment for adaptation is to quantify the variety in P^E, and information-theoretic uncertainty (Shannon entropy) is the natural tool to use:

$$H(P^E) = -\sum_i P^E(e_i) \log_2 P^E(e_i). \qquad (1)$$

$H(P^E)$ reflects two aspects of P^E: its width (number of different neighborhood indices i) and flatness (constancy of the value of $P^E(e_i)$ for different i). The wider and flatter P^E is, the more uncertainty there is about which neighborhood the agents will experience and the higher $H(P^E)$ will be.

4 Observing the Environment

We observed $H(P^E)$ in 100 3×3 blocks scattered randomly in space, and in one 30×30 block. Both resource distributions pump resources into the environment at the same rate, so they can support comparable maximum population sizes. In this resource-driven and space-limited model, population size is a good reflection of the dynamics of adaptation (fitness). Except for varying the resource distribution, all the parameters in the models observed were the same.

Evolution on an archipelago of 3×3 blocks always exhibits a pattern of period-lengthening adaptations. The population starts with a small period cycle on one of the blocks. This strategy migrates across the archipelago, until it has colonized virtually all the blocks. Eventually the appropriate mutations will create a longer-period cycle on one of the blocks and the agent with this longer cycle will sometimes supplant the agents on that block with the shorter cycle, so the block is now populated by a tiny subpopulation with the longer cycle. Children from this subpopulation migrate to other blocks and supplant the shorter-cycle agents there, until they eventually colonize the entire archipelago. This sequence is repeated again and again with ever longer-period cycles, until the cycle length gets close to its maximum possible value (nine).

The top two graphs in Figure 1 show time series of population level and the environment for adaptation, $H(P^E)$, in a typical simulation in an archipelago of 3×3 blocks. A period-2 cyclic strategy first colonizes the blocks. Then a period-lengthening mutation creates a period-3 cycle, which sweeps across the blocks. Next a combination of mutations creates 5-cycle, which then invades the archipelago. Finally, another period-lengthening mutation creates a 6-cycle, which colonizes the archipelago. These period-lengthening adaptations each cause a rise in $H(P^E)$, as the more complicated strategies make the experienced environment more complex.

Evolution on the 30×30 block tends to exhibit its own distinctive pattern of adaptations. One or two edge strategies in the initial random seed population will live on the block, growing to fill the space available in the niche on the block used by those strategies. From time to time new strategies arise by mutation, but most die off quickly. Some find an unoccupied niche on the block and coexist with their ancestral strategies. Others compete for niche space with ancestral strategies and supplant their ancestors. In this way, the block supports a changing ecosystem of subpopulations surviving through different strategies in different regions on the block. Eventually, though, the Random strategy always gets a foothold in the population and drives all Edge strategies extinct.

The bottom two graphs in Figure 1 show data from a typical illustration of a co-evolving succession of Edge strategies ending with domination by the Random strategy. The first strategy reflects off the bottom edge a six steps into the block, and it fills the entire bottom edge with about 80 agents, and the population experiences two neighborhoods: the inside of the block and the bottom inside edge. Then a handful of agents occupy a second niche with a second strategy along the block's right edge, and $H(P^E)$ shows a slight rise as a subpopulation experiences two new neighborhoods. Then a small subpopulation discovers a third strategy that exploits the hitherto unused left edge with a small jump into the block, and $H(P^E)$ rises again as a new neighborhood (inside left edge) gets experienced more and more frequently. Finally the Random strategy arises inside the block and it quickly sweeps across the block, knocking all the other strategies out of their niches. $H(P^E)$ drops dramatically because the neighborhoods experienced by the population shows much less variety. As the population grows, more agents experience an edge or the desert, so $H(P^E)$ rises slightly.
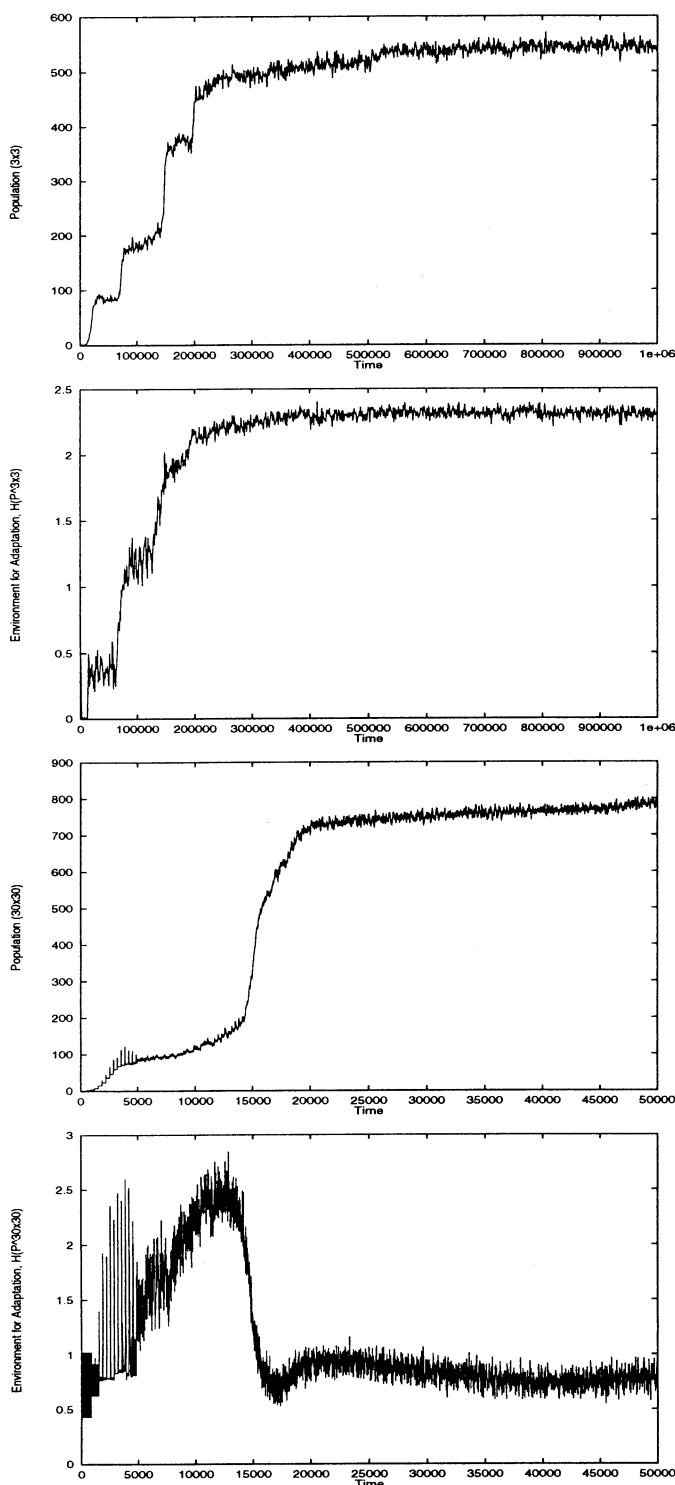
Figure 1: Dynamics of population size and environment for adaptation, $H(P^E)$, in two typical simulations. The top two graphs are from an archipelago of 100 3×3 blocks. The bottom two graphs are from one 30×30 block.

5 Conclusions

Two main conclusions follow from this work, one methodological and the other substantive. The methodological conclusion is that we now have a dynamic and quantitative reflection of the environment that the evolving population actually experiences—the environment for adaptation. This measure of the environment for adaptation is quite general and feasible, and it can straightforwardly be applied to many other evolving systems. Its drawback is that it depends on an *a priori* partition of the environmental states relevant to the population's adaptive evolution. Such a partition is easy in the simple systems studied here but more complex systems can present difficulties, especially if qualitatively new kinds of environmental states emerge unpredictably in the course of evolution. The substantive conclusion is that the environment for adaptation is *dynamic* even in simple static resource models like those studied here. Sometimes a population evolves a more complicated behavioral strategy which enables it to experience and exploit more of the environment's complexity. Subsequent evolution then takes place in the context of this more complicated environment for adaptation. In other cases the population benefits by evolving a simpler behavioral strategy and thus simplifying its environment. These conclusions pertain in the first instance just to the particular systems studied here, but they are likely to hold quite generally.

Acknowledgements

References

[1] Bedau, M. A. (1994). The Evolution of Sensorimotor Functionality. In P. Gaussier and J. -D. Nicoud, eds., *From Perception to Action* (pp. 134-145). New York: IEEE Press.

[2] Bedau, M. A. Dynamics of the Environment for Adaptation in Static Resource Models. In J. Kelemen and P. Sosik, eds., *Advances in Artificial Life, 6th European Conference, ECAL 2001* (pp. 76-85). Berlin: Springer.

[3] Bedau, M. A., Packard, N. H. (1992). Measurement of Evolutionary Activity, Teleology, and Life. In C. G. Langton, C. E. Taylor, J. D. Farmer, S. Rasmussen, eds., *Artificial Life II* (pp. 4431-461). Redwood City, Calif.: Addison-Wesley.

[4] Fletcher, J. A., Zwick, M., Bedau, M. A. (1996). Dependence of Adaptability on Environment Structure in a Simple Evolutionary Model. *Adaptive Behavior* 4, 283-315.

[5] Fletcher, J. A., Bedau, M. A., Zwick, M. (1998). Effect of Environmental Structure on Evolutionary Adaptation. In C. Adami, R. K. Belew, H. Kitano, and C. E. Taylor, eds., *Artificial Life VI* (pp. 189-198). Cambridge, MA: MIT Press.

[6] Lewontin, R. C. (1983). The Organism as the Subject and Object of Evolution. *Scientia* 118, 63-82. Reprinted in R. Levins and R. Lewontin, *The Dialectical Biologist* (pp. 85-106). Cambridge, MA: Harvard University Press.

[7] Littman, M. L. (1993). An Optimization-Based Categorization of Reinforcement Learning Environments. In J. A. Meyer, H. L. Roiblat, and S. W. Wilson (Eds.), *From Animals to Animats 2* (pp. 262-270). Cambridge, MA: Bradford/MIT Press.

[8] Packard, N. H. (1989). Intrinsic Adaptation in a Simple Model for Evolution. In C. G. Langton, ed., *Artificial Life* (pp. 141-155). Redwood City, Calif.: Addison-Wesley.

[9] Todd, P. M., Wilson, S. W. (1993). Environment Structure and Adaptive Behavior From the Ground Up. In J. A. Meyer, H. L. Roiblat, and S. W. Wilson (Eds.), *From Animals to Animats 2* (pp. 11-20). Cambridge, MA: Bradford/MIT Press.

[10] Todd, P. M., Wilson, S. W., Somayaji, A. B., Yanco, H. A. (1994). The Blind Breeding the Blind: Adaptive Behavior Without Looking. In D. Cliff, P. Husbands, J. -A. Meyer, and S. W. Wilson (Eds.), *From Animals to Animats 3* (pp. 228-237). Cambridge, MA: Bradford/MIT Press.

[11] Wilson, S. W. (1991). The Animat Path to AI. In J. -A. Meyer and S. W. Wilson (Eds.), *From Animals to Animats* (pp. 15-21). Cambridge, MA: Bradford/MIT Press.

A Framework of Evolvable Systems and Measures for Intelligent Agents

Seung-Ik Lee
Dept. of Computer Science, Yonsei University
Seoul 120-749, South Korea
cypher@candy.yonsei.ac.kr

Sung-Bae Cho
Dept. of Computer Science, Yonsei University
Seoul 120-749, South Korea
sbcho@cs.yonsei.ac.kr

Abstract

This paper proposes a soft computing framework for intelligent agents. The framework consists of two components: One is for the evolutionary construction of intelligent agents and the other is for the analysis of the evolved agents in both evolutionary and behavioral perspectives. Application of the framework to an agent shows that the framework can be effectively used to the evolutionary construction of intelligent agents. Especially, analysis techniques of the framework can show evolutionary dynamics and behavioral properties of the agent, which have been overlooked by conventional soft computing frameworks.

Keywords: Agent, Soft computing, Evolvability, Emergence, Adaptive Behavior.

1 Introduction

The old ways of interactions between human and computer require more time and effort of users and make untrained users to meet more difficulties. Recent research on agents has opened a new way for solving those more complex works, especially utilizing soft computing techniques.

Two main frameworks are used to apply soft computing techniques to the construction of agents. One is to apply each soft computing technique independently. It has some drawbacks in determining the internal parameters of the technique used. The other one is to combine two or more soft computing techniques to relieve the difficulties of tuning internal parameters. Especially, the combination of an evolutionary algorithm and other techniques has been attempted extensively [1, 2].

Despite the success of the conventional framework, it has at least three shortcomings: First, it cannot guarantee that evolved solutions are from adaptation, since evolutionary phenomena contain not only adaptation but others such as chances, necessity, and random genetic drift [3]. Second, it cannot show what, when, how, and why some genes have successfully evolved into final solutions while the others not. These questions are critical to show causes of adaptation and evolution. Lastly, it is not able to analyze the evolved solutions. Evolution necessarily causes the changes of simple structures into complex ones that may be viewed as emergent behaviors or emergent properties [4]. However, the conventional frameworks cannot tell whether certain behaviors are emergent because they do not have any formal definition on emergence.

This paper presents a soft computing framework for intelligent agents to overcome the shortcomings of the conventional frameworks and applies it to the construction and analysis of a robot agent.

2 The Framework

Fig. 1 shows a framework of evolvable systems and measures for intelligent agents. In the construction part, the combination of rule-based systems and evolutionary algorithms is used to construct an intelligent agent. In the analysis part, evolutionary activity statistics, schema analysis, and observational emergence are applied to the investigation of the evolved agent in both evolutionary and behavioral perspectives.

Rule-based Systems are composed of rules used to make decisions on which behavior to take. The rules take the form of "IF *condition* THEN *action*" as linguistic expressions such that incorporating expert's knowledge into the systems is very easy.

Evolutionary Algorithms cover several variations, such as evolutionary strategies, evolutionary programming, genetic programming, and genetic algorithms. The role of the algorithms is to tune internal parameters of the rule-based systems.

Evolutionary Activity Statistics is to measure the capacity to produce good solutions via evolution, *evolvability*, proposed by Bedau and Packard [5]. It is an objective, empirical measure of the level of evolutionary activity in an artificial or natural system.

A counter, $a_i(t)$, of the ith component at time t is attached to each component of a system. A component's activity changes over time as:

$$a_i(t) = \sum_{k \leq t} \Delta_i(k) \tag{1}$$

where $\Delta_i(k)$ is the activity increment for component i at time k.

The number of components with activity a at time t is the *component activity distribution*, $C(t, a)$, defined as:

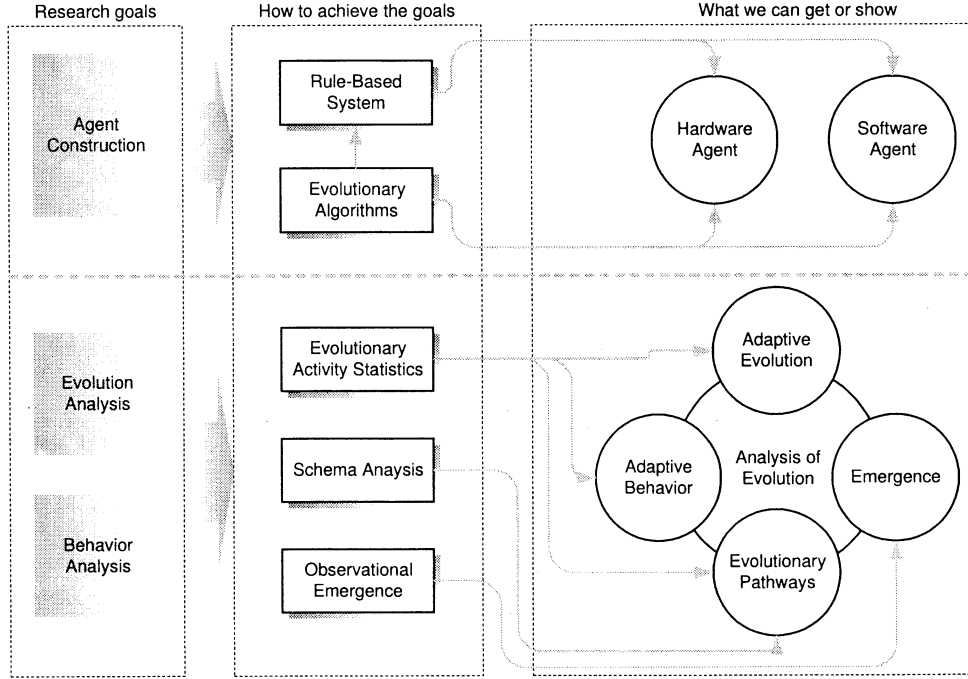$$C(t, a) = \sum_i \delta(a - a_i(t)) \tag{2}$$

Fig. 1: A soft computing framework for intelligent agents

where $\delta(a - a_i(t))$ is one if $a = a_i(t)$ and zero otherwise. *Mean activity*, used to measure the continual adaptive success of the components, is defined as follows:

$$\overline{A}_{cum}(t) = \frac{\sum_i a_i(t)}{D(t)} \qquad (3)$$

where $D(t)$ is the number of components i at time t with $a_i(t) > 0$.

Adaptive innovations correspond to new components flowing into the system and proving their adaptive value through their persistent activity. *New activity*, $A_{new}(t)$, is the reflection of the evolutionary adaptive innovations by summing the activity per component with values between a_0 and a_1 as in (4).

$$A_{new}(t) = \frac{1}{D(t)} \sum_{a=a_0}^{a_1} C(t, a). \qquad (4)$$

Observational Emergence: Emergence is used as a name for creation of new structures and properties as the old philosophical statement, "the whole is more than the sum of its parts." Baas [6] defined a property P is emergent as follows:

P is an emergent property of $S^n \iff P \in Obs^n(S^n)$,

but $P \notin Obs^n(S^{n-1}_{i_{n-1}})$ for all i_{n-1}

$$\qquad (5)$$

where $S^n_{i_n}$ is the ith structure of the nth order, and Obs^n is a kind of observational mechanism to evaluate, observe, and describe S^n.

Let Int^n be a family of interactions among structures $S^n_{i_n}$. Then, emergence is *deducible or computable* if there

is a deductional or computational process or theory D such that $P \in Obs^n(S^n)$ can be determined by D from $(S^{n-1}_{i_{n-1}}, Obs^{n-1}, Int^{n-1})$, and observational if P is an emergent property but cannot be deduced as in *deducible*.

Schema is a similarity template representing a subset of strings with similarity at certain string positions. For example, the schema 1##0 represents all the bit strings with '1' in the first position and '0' at the fourth position. The '#' notation represents "do-not-care" bit and can be either 1 or 0. Bit strings 1000, 1010, 1100, and 1110 are all the instances of the schema 1##0.

3 An Application ot Robot Agents

The framework is applied to the construction of a behavior controller for an agent called Khepera whose eight proximity sensors are used as input to the behavior controller and two control values are produced from the controller for the two motors of the robot. Fuzzy logic is applied for the behavior controller of the robot and a genetic algorithm is used to tune the internal parameters of the fuzzy system (see [2] for more information). A counter is attached to each rule of distinct genotype i. The increment function $\Delta_i(t)$ is defined as the number of rule instances of genotype i at t.

Another model called *neutral shadow* [7] of the fuzzy model was evolved to identify the extent to which a system's evolutionary dynamics depends on adaptation rather than other evolutionary forces. The shadow model is used to screen off the effect of non-adaptive evolutionary forces by comparing the evolutionary dynamics ob-
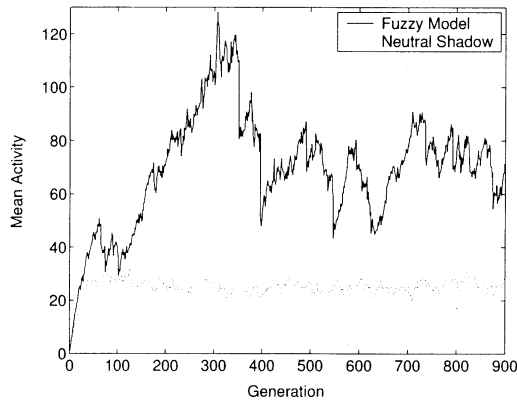
(a) Mean activity $\overline{A}_{cum}(t)$
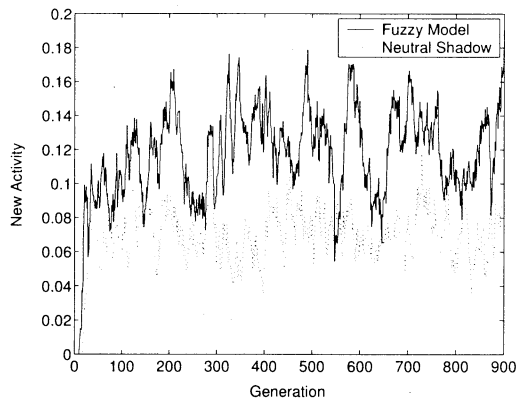
(b) New activity $A_{new}(t)$

Fig. 2: Evolutionary activity statistics

served in target evolutionary systems with those observed in analogous evolutionary systems in which adaptive evolution cannot happen.

3.1 Adaptive Evolution

Fig. 2 (a) shows mean activity, $\overline{A}_{cum}$, of the two models. The mean activity of the fuzzy model is significantly higher than that of the *neutral shadow*. This means that much more adaptive rules are present in the fuzzy model than in its *neutral shadow*.

Fig. 2 (b) shows *new activity*, $A_{new}(t)$, of the models. The new activity values of the fuzzy model is higher than that of the *neutral shadow*, which means that new activity signaling the positive adaptability flows more frequently into the fuzzy model compared to that of the *neutral shadow*.

Therefore, Fig. 2 illustrates that the evolution has maintained high evolvability during the evolutionary process, which implies that the probability of producing good candidate solutions is high.
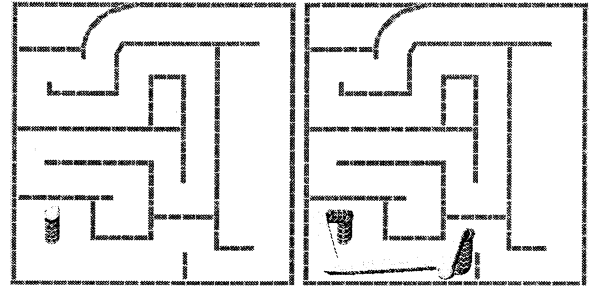
(a) S_1

(b) $S_1 + S_2$
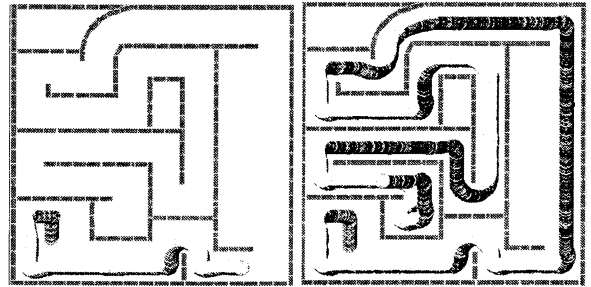
(c) $S_1 + S_2 + S_3$

(d) Best individual

Fig. 3: Behaviors of salient rules and best individual

3.2 Adaptive Behaviors

In order to discover what behavioral properties make a rule have high evolutionary activity value, salient rules that have high evolutionary activity are investigated.

Fig. 3 (a) shows the behaviors of a selected rule S_1; Fig. 3 (b) shows the behaviors of S_1 and S_2; Fig. 3 (c) shows the behaviors of S_1, S_2, and S_3; and Fig. 3 (d) shows the behaviors of the best individual that has all the three rules including others. S_1 drives the robot to move forward, S_2 turns the robot left in front of obstacles, and S_3 keeps the robot from moving far away from right walls.

As these behaviors are important for the robot, they are explored during the evolution.

3.3 Evolutionary Pathways

This section illustrates that the salient rules have contributed to the construction of the best individual by comparing schemata of the rules.

Table 1 shows the gene codes and their corresponding schemata. Each schema is obtained by masking off the alleles that are decoded not to participate in the conditional part of a rule (see [2] for rule encoding).

Comparing the two rows of S_i and B_i shows close relation between the schemata of the two rules. The schema of S_3 (S_4) is the same with the schema of B_3 (B_4), which means that the two salient or other rules of same schemata evolved directly into the rule set of the

Rule	Gene Code	Schema
S_1	0001010001030411441	0#0#0#0#0#0#0#11441
B_1	0102041003010411441	0#0#0#100#0#0#11441
S_2	0400010114000110211	0#0#0#0#140#0#10211
B_2	0400110114000110211	0#0#110#140#0#10211
S_3	0300000410130104031	0#0#0#0#10130#0#031
B_3	0304000410130204031	0#0#0#0#10130#0#031
S_4	0003120304010102401	0#0#120#0#0#0#0#401
B_4	0003120304010103401	0#0#120#0#0#0#0#401
S_5	0411001112140400311	0#110#1112140#0#311
B_5	0111001102140000311	0#110#110#140#0#311
...	...	...
S_{29}	1311000003040411021	13110#0#0#0#0#11021

best individual. In addition, high similarities between S_i and B_i indicate that B_i is a descendant of its corresponding S_i.

3.4 Observational Emergence

According to Baas' definition of emergence [6], the parameters of the fuzzy model for emergence can be set as follows:

- First order structures S_i^1 = the ith fuzzy rule.

- First order interactions Int^1 = the fuzzy inference and defuzzification.

- $Obs^1 = Obs^2$ = the behavioral properties of structure.

- Second order structure S^2 = the evolved fuzzy logic controller.

Fig. 4 shows Obs's of the three first order and a second order structures. $Obs^1(S_{2_1}^1)$ shows that the robot stops after some moves from the start position. $Obs^1(S_{5_1}^1)$ and $Obs^1(S_{7_1}^1)$ show that the robot does not move at all. On the contrary, Obs^2 of the fuzzy controller composed of three interacting structures, $S_{2_1}^1$, $S_{5_1}^1$, and $S_{7_1}^1$ drives the robot to turn around.

(a)
$Obs^1(S_{2_1}^1)$

(b)
$Obs^1(S_{5_1}^1)$
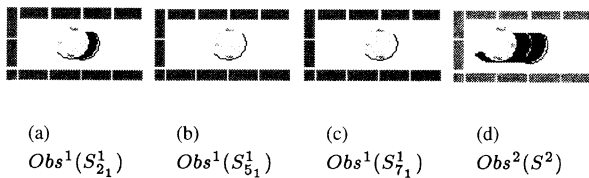
(c)
$Obs^1(S_{7_1}^1)$

(d)
$Obs^2(S^2)$

Fig. 4: Obs of the first and second order structures

In other words, the interactions of the three first order structures, $S_{2_1}^1$, $S_{5_1}^1$, and $S_{7_1}^1$, make $Obs^2(S^2)$ different

from $Obs^1(S_{i_1}^1)$ of first order. This implies that:

$$P \in Obs^2(S^2), \text{but } P \notin Obs^1(S_{i_1}^1) \text{ for all } i_1 \quad (6)$$
$$\text{with } P = \text{``Turn Around''}.$$

Therefore, the "turning around" behavior is an emergent behavior by the definition in (5).

4 Conclusion

This paper proposes a framework for the construction and measurement of intelligent agents and applied it to an agent. Compared with the conventional frameworks, the proposed framework puts an emphasis on the analysis of the behaviors of the agents instead of the evolution itself.

Application of the framework to an agent shows that the framework has shown good results with advantages in analysis than others. In addition, the framework has quantified evolvability and shown that the evolution has maintained high evolvability during the evolution of the agent. It has also revealed that what behavioral properties are regarded as adaptive during the evolution and how the final solutions are constructed from adaptive candidates of earlier generations.

Though we have applied the framework to a kind of hardware agent, it is domain independent and, therefore, can be applied to software agents as well.

References

[1] Hunter A and Chiu KS (2000), Genetic algorithm design of neural network and fuzzy logic controllers. Soft Computing 4(3):186-192

[2] Lee SI and Cho SB (2001), Emergent behaviors of a fuzzy sensor-motor controller evolved by genetic algorithm. IEEE Trans. on System, Man, and Cybernetics: Part B 31(6):919-929

[3] Mayr E (1988), Towards a New Philosophy of Biology. Harvard University Press

[4] Holland JH (1992), Adaptation in Natural and Artificial Systems. MIT Press/Bradford Books

[5] Bedau MA and Packard NH (1992), Measurement of evolutionary activity, teleology, and life. Artificial Life 2, Addision-Wesley, Redwood City, pp. 431-461

[6] Baas NA (1992), Emergence, hierarchies, and hyperstructures. Artificial Life 3:515-537

[7] Rechtsteiner A and Bedau MA (1999), A generic neutral model for measuring excess evolutionary activity of genotypes. Proc. of the Genetic and Evolutionary Computation Conference 2:1366-1373

Evaluation of CCD Camera Arrangement for Positioning System in Intelligent Space

Takashi Akiyama[1], Joo-Ho Lee[1], Hideki Hashimoto[1,2]

[1]Institute of Industrial Science, University of Tokyo, Tokyo 153-8505, Japan

(Tel: +81-3-5452-6258; Fax: +81-3-5452-6259

Email:akiyama@vss.iis.u-tokyo.ac.jp

home page:http://dfs.iis.u-tokyo.ac.jp)

[2]PRESTO, JST

Abstract— The Intelligent Space is a space where we can easily interact with computers and robots, and get useful service from them. In such a space, location information is very important, since the agents cannot provide proper service to a proper person at a proper location without location information. Our positioning system uses CCD cameras. It is important to determine where to arrange the cameras for the best localization depending on the tasks in the space. The purpose of this paper is to consider how to arrange cameras for the best localization in the Intelligent Space.

Keywords— optimal arrangement, CCD cameras, stereovision, positioning system.

I. INTRODUCTION

The Intelligent Space is a space where we can easily interact with computers and robots, and get useful service from them [1]. The space has three fundamental elements; DINDs(Distributed Intelligent Networked Device), agents and network. DINDs have mainly three functions; sensing, processing and networking. DINDs recognize what is happening in the space through sensors such as CCD cameras and produce proper orders for agents. DINDs are connected with other DINDs through network. Agents in the Intelligent Space such as mobile robots, displays and speakers provide proper service to human in the space. Among these three elements, DINDs are the most important elements and location information is especially important among the information. Since the agents cannot provide proper service to proper person at a proper location without location information, it is considered as indispensable information in the Intelligent Space. Moreover precise localization of a human body makes it possible to achieve gesture recognition system. Many researchers keep on researching to develop perfect positioning system, but there are some difficulties to achieve it [2][3].

Our positioning system uses several CCD cameras with which we need no special tags (such as active badge[3]) for localization. CCD cameras can recognize overall pictures and they help making a human

gesture recognition system. To measure the location of target objects, stereovision is applied. When developing stereovision systems, there are many factors causing errors such as camera arrangement, calibration and measurement of a target object [4]. Among these factors, camera arrangement is focused on, since this factor is important in positioning system and this effects overall performance of the system. When the Intelligent Space is applied to a certain environment, it is necessary to make a simulator which determines the best arrangement of cameras with ease.

In this paper, our positioning system called Intelligent Space 3D Tracking System is introduced and the problems of it are discussed. Then, why optimal camera arrangement is important and how to arrange two cameras to perform the best localization under a determined task are discussed. Finally, we will conclude with some comments about our early experimental results of what we have done and future works.

II. INTELLIGENT SPACE 3D TRACKING SYSTEM

In the Intelligent Space, target objects for localization are mainly human and physical agents such as robots. Our positioning system localizes these objects using stereovision [5]. Detecting points of stereovision are the colors of objects.

The localization process is: first, pixels in the two images are clustered using color histograms after subtraction of each background image. The clustered blob is the detecting point for stereo process. Finally, 3D positions of objects corresponding to each blob are reconstructed after data matching of the blobs. The positioning error grows as objects move far away from cameras.

In general, localization accuracy needed in the Intelligent Space depends on the task. For instance, when we want to localize human parts, such as a head and hands, very precisely for gesture recognition, high accuracy is required. On the other hand, when we want to localize only human for human tracking, high accu-

racy is not required generally. However, in this case broad field of view for stereo vision is required. There is optimal arrangement of cameras for the best localization. Thus, how to arrange several cameras more efficiently depending on tasks is one of the most important procedures in the Intelligent Space.

III. The Need of Optimal Camera Arrangement

CCD camera is used as a sensor function of DIND at the moment. The reason for using CCD cameras are that we can get overall pictures of what is happening in the space and the information is useful for the gesture recognition. Moreover, location information of users and agents is obtained by stereovision processing and users do not need to have some special tags.

The demerit is that image processing for stereovision can be complex. In the Intelligent Space, real time processing is required and faster image processing is needed. As described in previous section, our stereo processing method uses blob model made by using color histograms and clustering process. This process is based on pixel-by-pixel processing.

There are a lot of systems using CCD cameras for identification of person and human tracking and gesture recognition and so on. In most of these systems, camera sensors are arranged usually by trial and error depending on tasks. For example, in identification systems, cameras are arranged at the place such as a gate and door where identification is done. In those systems, users have to walk up to the cameras and wait until identification is finished. Moreover, due to improper camera arrangement, working area of system can be small and human motion can be restricted in the systems such as gesture recognition and human tracking systems. This means users have some restriction and burden in the systems.

In our Intelligent Space, it is considered to be favorite for users to move freely and do not have to stop or walk up to determined place for identfication. Identification should be done naturally with no burden to users. To perform this, DIND should be arranged optimally for various tasks. However, how to arrange cameras systematically for each task is not established.

In our Intelligent Space, a lot of DINDs are distributed to a space without changing existing environment. The installation location should be the place which does not interfere with human's daily life such as wall and ceiling. For these reasons, we are not concerned with the cameras which interfere with human's daily life such as ominidirectional camera. Although this kind of cameras are useful for mobile robot, our policy is not make agent such as mobile robot intelligent but to make a space more intelligent. For making

only human localization system, using fish-eye lens on a ceiling can be considered as one way. However, this kind of cameras cannot be used to gesture recognition for the most of the cases. Thus, this paper deals with only single-eyed CCD cameras.

IV. Determination of Optimal Arrangement of Cameras

In general, optimal arrangement of cameras depends mainly on tasks. It is difficult to consider all possible cases because the number of system design parameters becomes huge according to features of cameras used in the system and the size, shape or work space of objects. Furthermore, it is not so easy to deal with three or more cameras at a time. Thus a task and conditions of optimal arrangement of cameras are determined and only the case of two cameras is considered in this paper. A pinhole camera model is considered as a CCD camera model and camera simulator which calculates the field of view from two cameras depending on objects' size and work space is explained.

A. Camera Model

The pinhole camera model is the simplest, and an ideal model of camera function [6]. To make things simple, the image plane is placed between the focal point of the camera and an object, so that the image is not inverted. This mapping of 3D onto 2D, is called a perspective projection shown in Fig.1(the right part), and perspective geometry is fundamental to any understanding of image analysis. The relationship between the world coordinates M (X_w, Y_w, Z_w) and the camera coordinates M (X_c, Y_c, Z_c) is given by,

$$M = RM + t \qquad (1)$$

where R and t is camera extrinsic parameters. R is rotation matrix(3×3) and $t(3 \times 1)$ is translation vector. The relationship between M (X_c, Y_c, Z_c) and virtual image coordinates $m(X_u, Y_u)$ is given by,

$$\begin{cases} X_u = (X_c/X_c)f \\ Y_u = (Y_c/Z_c)f \end{cases} \qquad (2)$$

where f is focal length of a camera. By determining camera parameters and target objects, camera simulator which measures how much area the camera can watch the objects is modeled. We can determine how much area a camera can see the objects if we determine camera parameters such as CCD size and camera position and direction, and objects data. In our simulator, the size of CCD is fixed as 640×480 as our cameras, which we use in the Intelligent Space, have same size of CCD. The cameras in simulator is modeled to our real
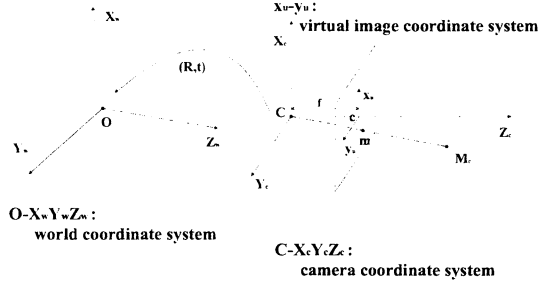
Fig. 1. Pin hole camera model

cameras by changing focal length in simulator. Effects of lens and distortion are not considered currently.

B. Task and Conditions of Optimal Arrangement

The task of localization of human hands precisely is determined, since in the Intelligent Space, we are interested in interacting with the space using some gestures. Positions information with time sequence of human hands, of which motion consists, is a key for recognition. We assume that hand shape is sphere and it exists at a height of between 1.4m and 1.8m. The height bound of work space is determined by possible positions of human hands when a man is doing several gestures in standing erect. We assumed the radius of the sphere to be 5cm. The area where the objects can move is determined by object size and moving area as shown in Fig.2.
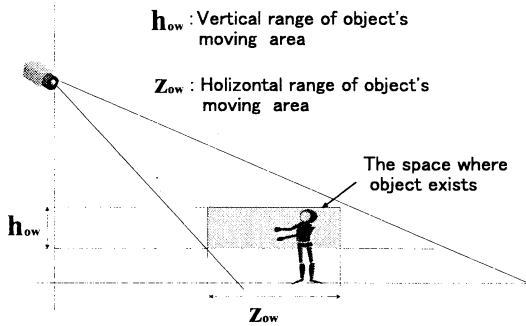

Fig. 2. Area where the object can move

Next, the conditions of optimal arrangement are determined. This condition should be determined by user. Localization error bound is set to 2cm and mean of errors should be as small as possible. The field of view from two cameras should be large as much as possible. Moreover, the arrangement which affords easier stereo matching process is favorable. It becomes difficult to find stereo matching points when distance between cameras(baseline) bl and camera's pan angle θ_p increase. This process also depends on the method of

stereo matching process.

V. FACTORS FOR EVALUATION OF OPTIMAL ARRANGEMENT

A. Maximization of Area

It should be considered to maximize the field which both two cameras can watch an object. If the space where an object can move is determined, the maximum (ul_1, ul_2) and minimum (ll_1, ll_2) perpendicular distance which one camera can watch the object is determined by the camera's tilt angle θ_{t1}, θ_{t2} and camera height h_{c1}, h_{c2}. The positions of the whole area which each camera can watch the object is determined after setting the camera's pan angle θ_{p1}, θ_{p2} and baseline, bl. The area for stereovision is determined as overlapped area of these two areas. We call this area stereovision area as shown in Fig3.
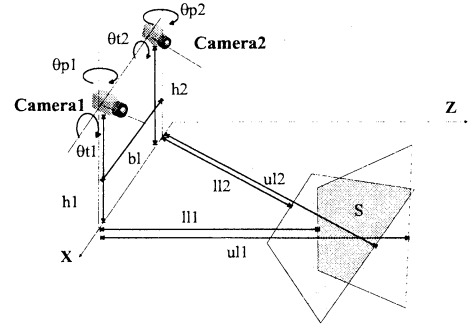

Fig. 3. Camera Parameters

The size of the area S changes depending on θ_{p1}, θ_{p2} and baseline bl when θ_{t1}, θ_{t2} and camera height h_{c1}, h_{c2} are fixed. We want to maximize this area under the condition of maximum error is smaller than allowable error.

B. Localization Errors

There are several errors in the stereo based system as written below.

- **Quantization Errors**
  Due to the characteristic of discrete imaging system, the image coordinates of each pixels suffer from quantization errors from up to ± 1/2 pixels. Therefore, the disparity between estimated position and real position occurs due to this error. Fig.4 shows these situations. To make it simple, both cameras are at the same height and orientation. P is a real position of an object. xL and xR is projected point of P. However, xL and xR is quantized to xL− or xL+ and xR− or xR+. If the pair of xL− and xR− is selected as stereo pair, P− is calculated. As other case, the pair of

xL+ and xR+ is selected. P+ is calculated. Thus ΔZ stands for quantization error.
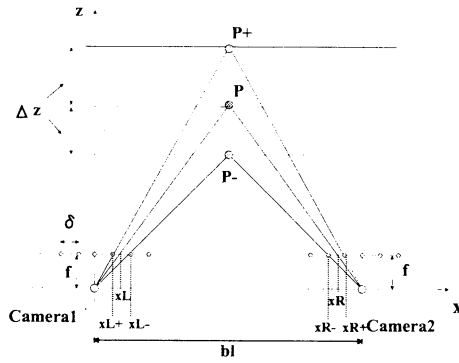

Fig. 4. Quantization Error

- **Image Processing Errors**

  In image processing, the pixels on a target is extracted first and some errors occurs while doing this process. In our image processing, detected points for stereo matching are the center of gravity of the blobs. The shape of the blobs change somehow due to clustering errors. This causes image processing error.

- **Calibration Errors**

  Errors also occurs in calibrating cameras. In stereo vision system, two points of each image are translated to 3D point using perspective camera matrix. Camera calibration means to find this matrix [7]. Several points in image coordinates are needed when calibrating. Errors in calibration occurs mainly due to quantization error of reading the several points in image coordinates.

Among these errors, quantization errors are focused mostly in this paper, because these errors always exist and affect the accuracy of localization and can be reduced by better arrangement of cameras. Pixel extraction errors depends mainly on software and they do not relate with camera arrangement. Calibration errors have few relations with arrangement of cameras.

Quantization errors were measured at the interval of 10cm in each direction of the area which both two cameras can watch an object and maximum errors and mean of errors were evaluated as shown in Fig.5.

C. Stereo Matching and Baseline

In general, it becomes difficult to detect stereo matching point when bl and θ_p increases and target objects close near to cameras. Especially, bl influences more than θ_p. This characteristic should be considered in performance indexes for optimal arrangement as we describe in the next section.
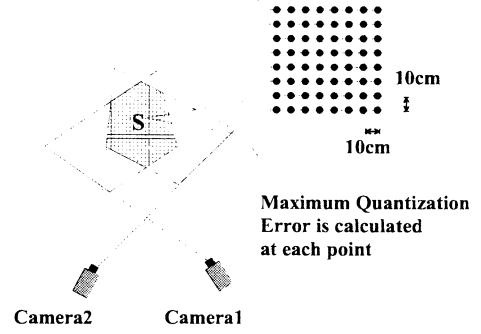

Fig. 5. The Area of Quantization Error Measurement

D. The Number of Pixels

Since we use CCD cameras and image processing is done by pixel-by-pixel, amount of target pixels should be more than certain amount to be clustered. The amount of pixels of a target object decreases as the object moves far away from cameras. Fig.6 shows the relationship between the perpendicular distance of an object from a camera and the amount of pixels in the image from a CCD camera. In this case, we assume that the shape of the object is sphere(radius is 5cm) and camera's tilt angle θ_t is 20 degree.

We have to decide minimum amount of pixels depending on the size of a target object for our image processing. This minimum amount is used as limiting conditions for performance indexes for optimal arrangement.
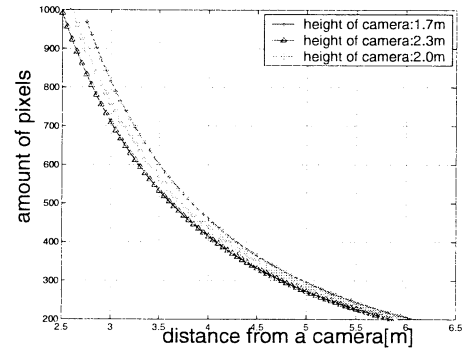

Fig. 6. Amount of Pixels of the Object and Distance from a Camera

E. Symmetric Arrangement

In camera simulator model, system design parameters are θ_{p1}, θ_{p2}, θ_{t1}, θ_{t2}, h_{c1}, h_{c2} and bl. As described in previous section, the area which one camera can watch an objects is determined by θ_{t1}, θ_{t2} and h_{c1}, h_{c2} not by θ_{p1}, θ_{p2}. Moreover each camera model is assumed to have same characteristic. If two cameras are

arranged under the conditions of $\theta_{p1} = \theta_{p2}$, $\theta_{t1} = \theta_{t2}$, $h_{c1} = h_{c2}$. the shape of stereovision area becomes symmetric shape. This would be helpful when more than 4 cameras are arranged in the space. Thus, in this paper. system design parameters keep this symmetry, while they are changed.

VI. CAMERA ARRANGEMENT SIMULATION

In consideration of five factors described in previous section, camera parameters, θ_{p1}, θ_{p2}, θ_{t1}, θ_{t2}, h_{c1}, h_{c2}, bl will be determined for optimal arrangement of cameras in this section.

To determine optimal parameters considering these factors, performance indexes are utilized. In general, performance indexes changes depending on the tasks and need from users. Thus we assumed performance indexes in the light of the tasks, conditions of optimal arrangement as described in section IV, and five factors. Performance indexes are defined as (3).

$$J = \frac{f_S}{f_{qe} f_{sm}} \qquad (3)$$

where f(S), f(e) and f(sm) are performance indexes related to maximum area, quantization error and stereo matching process respectively. These performance indexes depend on the task condition and positioning systems.

f(S) is defined as follows.

$$f_S(S) = (1 + \frac{S}{S_{need}})^2 \qquad (4)$$

where S_{need} is favorable area determined by users. S is the stereovision area as described in the section V. Square suffix is appended because maximum area should be emphasized more than other factors in this case.

f(qe) is defined as follows.

$$f_{qe}(\epsilon_{qmean}, \epsilon_{qmax}) = 1 + \frac{\epsilon_{qmean}}{\epsilon_{qmax}} \qquad (5)$$

where ϵ_{qmean} is the mean of quantization error and ϵ_{qmax} is the maximum value of quantization error. It is favorable to making mean of quantization error ϵ_{qmean} as small as possible, so ϵ_{qmean} is evaluated.

Finally, f(sm) is defined as follows.

$$f_{sm}(bl, d_g) = 1 + \frac{bl/d_g}{bl_{max}/d_{min}} \qquad (6)$$

where d_g is the distance between center of gravity of stereovision area and middle-point of two cameras. bl_{max} and d_{min} are maximum baseline and minimum distance between the object and middle-point of two cameras respectively. As described in Section V-D, the

relation of stereo matching and baseline is evaluated here.

We are interested in making index value J as big as possible. There are a lot of methods of maximizing functions such as genetic algorithm [8], Powell's method, conjugate gradient method [9]. Among these methods, we chose downhill simplex method [9]. The downhill simplex method was originally developed by Nelder and Mead in 1965. The downhill simplex method requires only function evaluations, not derivatives. Since derivatives of our performance index can not be calculated, we use this method for optimization.

Using this method, we simulated under several restrictions given by the task and the performance of image processing. That is,

$$\begin{cases} \epsilon_{qmax} \leq \epsilon_{allow} \\ pn \geq MP \end{cases} \qquad (7)$$

where ϵ_{qmax} is maximum quantization error, ϵ_{allow} is allowable error in the task, pn is the amount of pixels of the object in the image and MP is minimum amount of pixels processed in the image processing. To keep symmetric characteristic, we assumed $\theta_{p1} = -\theta_{p2}$, $\theta_{t1} = \theta_{t2}$, $h_{c1} = h_{c2}$.

A. Current Arrangement

In advance of optimization, we simulated using the position and direction of our current system under the restrictions shown at (7). Camera parameters of our current system is shown in Table.1(Before). Fig.7 is the result of simulation. Both two cameras are positioned at X=0 and X=2.2 respectively. Z-axis represents quantization error. Fig.7 shows stereovision area is $8.57m^2$ and mean of error is 0.94cm.

B. After Optimization

After simulation of our current arrangement, optimal arrangement of cameras are simulated under the same restrictions shown at (7). Fig.8 is the result of simulation and Table.1(After) shows optimal camera parameters. Both two cameras are positioned at X=0 and X=3.73 respectively. Fig.8 shows quantization error is smaller than allowable error 2cm. Stereovision area is $9.24m^2$ and mean of error is 0.67cm. Compared with Fig.7, stereovision area has become larger and mean of error has become smaller.

Although the cameras in simulator is not perfectly modeled to real cameras, this result shows better performance is obtained after optimization.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, the concept of the Intelligent Space and our positioning system using CCD cameras in the space were introduced. The optimal arrangement of

TABLE I
OPTIMAL PARAMETERS AND SIZE OF AREA

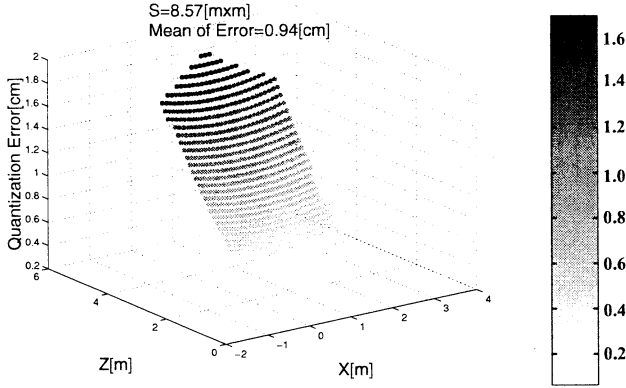|	θ_p[deg]	θ_t [deg]	h_c[m]
Before	21.0	21.9	2.2
After	36.0	29.5	2.75
	bl[m]	S[m^2]	ϵ_{qmean}[cm]
Before	2.2	8.57	0.94
After	3.73	9.24	0.67

Fig. 7. Before Optimization

CCD cameras for the best localization is discussed. Optimal arrangement of cameras depends on tasks and there are many factors to be considered. Thus we made camera simulator to know how much area an object can be watched from two cameras. With the simulator, optimal arrangement of cameras was searched through performance indexes in consideration of the target task. Optimal arrangement of cameras is determined by assuming several restrictions as an early experimental result. The result was compared with our current arrangement of cameras and better performance was obtained. However, the cameras in
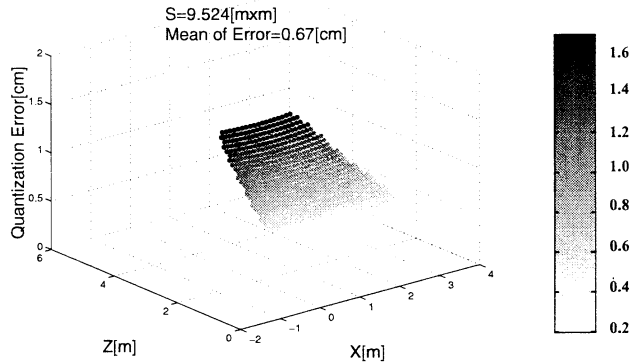

Fig. 8. After Optimization

simulator are not modeled perfectly to real CCD cameras. Our future works are to compare the simulator with our cameras in the Intelligent Space precisely and make further evaluation of the results obtained by simulator. In this paper, only one task is assumed and performance index is determined by the task, so if we assume different tasks such as human tracking or head positioning, different model and performance index are needed. Thus, we will make other task models progressively. After that, we are planning to rearrange cameras in our Intelligent Space. Fig.9 is the image of optimal camera arrangement in our Intelligent Space.
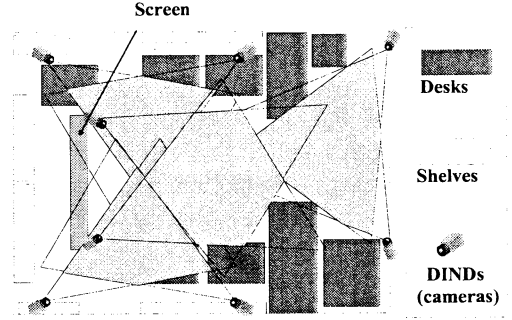

Fig. 9. Image of an optimal arrangement of cameras

REFERENCES

[1] H. Hashimoto, "Intelligent Interactive Space based for Robots", Proceedings of the 2000 International Symposium on Mechatronics and Intelligent Mechanical System for 21 Century, pp26, 2000.
[2] B.Brumitt, B.Meyers, J.Krumm, A.Kern and S.Shafer, "EasyLiving: Technologies for Intelligent Environments", Proceedings of the International Conference on Handheld and Ubiquitous Computing, September 2000.
[3] A. Ward, A. Jones, A. Hopper, "A New Location Technique for the Active Office", IEEE Personal Communications, vol4, No5, pp42-47, October, 1997.
[4] S. Blostein, T. Huang, "Error analysis in stereo determination of 3-D point positions", IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(6), pp752-765, November, 1987.
[5] G. Appenzeller, J. H. Lee and H. Hashimoto, "Building Topological Maps by Looking at People: An Example of Cooperation between Intelligent Space and Robots", Proceedings of IROS97, OO, pp1326-1333, September, 1997.
[6] G. Wei and S. Ma, "Implicit and explicit camera calibration: Theory and experiments", IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(5), pp469-480, 1994.
[7] R. J. Tsai, "A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", IEEE Journal of Robotics and Automation, Vol Vol. 3, No. 4,pp.323-344,1987.
[8] J. Koza, "Genetic Programming", MIT Press, Cambridge, MA, 1992.
[9] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Receipes in C, Second Edition, 1992.
[10] J. A. Nelder, R. Mead, "A Simplex Method for Function Minimization", Computer Journal, vol. 7, pp. 308-313, 1965.

Examination of Change of Stress Reaction by Urinary Tests of Elderly before and after Introduction of Mental Commit Robot to an Elderly Institution

Tomoko SAITO[*1], Takanori SHIBATA [*1,2], Kazuyoshi WADA [*1,3], Kazuo TANIE [*1,3]

[*1] Intelligent Systems Institute, AIST,
1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan
{tomo-saito, shibata-takanori, k-wada, tanie.k}@aist.go.jp

[*2] PRESTO, JST
[*3] University of Tsukuba
1-1-1Tennodai, Tsukuba, Ibaraki, 305-8577 Japan

Abstract

We introduced "mental commit robot" to an elderly institution. The purpose was to clarify relationship between interaction with the robot and change of stress reaction by the level of 17-KS-S and 17-OHCS value in urine of elderly, as well as to investigate potential beneficial effects on health of the elderly. As the results, an excellent adjustment ability to the stress was shown, and mental commit robot had a useful possibility for maintenance of health of elderly.

Keywords: mental commit robot, urinary test, 17-Ketosteroid sulfates (17-KS-S), stress reaction, elderly institution

1 Introduction

After 1970's, the research on the influence from a pet to a human's health came to be performed in various fields, such as psychology and comparison action study, in Europe and America.

There is a research that the existence of the pet has a constant effect for the stress with a large loss of the death of a spouse or an intimate friend[1]. In general, it is said that keeping a pet is good for health. However, there is a lot of elderly who cannot keep a dog or a cat even if wanting to keep them because of the worry of common infectious disease between men and beasts. Moreover, it is difficult to take animals into hospitals and elderly institutions.

By now, authors have researched and developed "mental commit robot" which aims for therapy. We verified the effects of hospitalized children's QOL improvement in the infantile ward. It was confirmed that the introduction of the robot was useful for the improvement of the desire of leaving hospital and for the improvement of symptom as autism[2].

On the other hand, as a method of measuring the stress, it is possible that we evaluate physiology by using urinary tests which are the inspection method

handiness and non-infestation for elderly[3]. We had conducted research on elderly living at home as subjects using the index of urine 17-Ketosteroid sulfates (17-KS-S) and 17-hydroxycorticosteroids (17-OHCS), and had obtained the result that elderly who were keeping the companion animal had excellent adaptability to the stress[4].

In this study, we introduced mental commit robot to an elderly institution, and examined the change of stress reaction of elderly by urine 17-KS-S value and the 17-OHCS value before and after introduction of mental commit robot.

2 Mental commit robot

Mental commit robot (Fig.1) was developed to have physical interaction with human beings. Robot's appearance is from a baby of harp seal, which has white fur for three weeks after its born. As for perception, seal robot has tactile, vision, audition, and posture sensors beneath its soft white artificial fur. In order for the robot to consist of a soft body, an air-bag type tactile sensor was developed and implemented. As for action, it has eight actuators; two for upper and lower eyelids, one for rotation of eyes, two for neck, one for each front fin, and one for two rear fins. Weight of the seal robot is 2.8 kg[5].
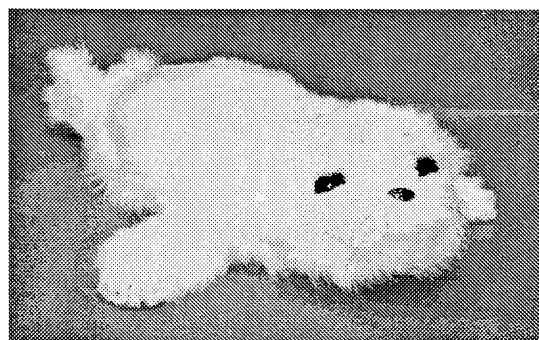

Fig.1 Seal Robot "Paro"

The seal 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:

(1) Proactive Behaviors: The robot has two layers to generate its proactive behaviors: behavior-planning layer and behavior-generation layer. Considering internal states, stimuli, desires, and a rhythm, seal robot generates proactive behaviors.

(a) Behavior-planning layer: This has a state transition network based on internal states of robot and robot's desire produced by its internal rhythm. The robot has internal states that can be named with words of emotions. Each state has numerical level and is changed by stimulation. The state decays by time. Interaction changes internal states and creates character of the robot. The behavior-planning layer sends basic behavioral patterns to behavior-generation layer. The basic behavioral patterns include some poses and some motions. Here, although "proactive" is referred, proactive behaviors are very primitive compared with those of human beings. We implemented similar behaviors of a real seal into the robot.

(b) Behavior generation layer: This layer generates control references for each actuator to perform the determined behavior. The control reference depends on strength of internal states and their variation. For example, parameters change speed of movement, and the number of the same behavior. Therefore, although the number of basic patterns is countable, the number of emerging behaviors is uncountable because numeral parameters are various. This creates living like behaviors. In addition, as for attention, the behavior-generation layer adjusts parameters of priority of reactive behaviors and proactive behaviors based on strength of internal states. This function contributes to situated behavior of robots, and makes it difficult for a subject to predict robot's action.

(c) Long-term memory: The robot has a function of reinforcement learning. It has positive value on preferable stimulation such as stroked. It also has negative value on undesirable stimulation such as beaten. The robot put values on relationship between stimulation and behaviors. Gradually, the robot can be shaped to preferable behaviors of its owner.

(2) Reactive behaviors: Seal robot reacts to sudden stimulation. For example, when it hears big sound suddenly, the robot pays attention to it and looks at the direction. There are some patterns of combination of stimulation and reaction. These patterns are assumed as conditioned and unconscious behaviors.

(3) Physiological behaviors: The robot has a rhythm of a day. It has some spontaneous desires such as sleep based on the rhythm.

3 17-KS-S, 17-OHCS

Selye[6] regarded stress as the rate of wear and tear and 17-OHCS as its indicator. But Nishikaze et al.[7] considered that living organisms, unlike inanimate objects, exist in a dynamic balance between "wear and tear" and "repair and recovery," and sought for a compound related to tissue "repair and recovery," and they discovered 17-KS-S in urine. 17-KS-S value shows the descent by the ageing, progress of the disease or social psychology stresses.

Though 17-OHCS value rises at the stress, shows high level in healthy individuals and decrease with failing health or the progress of disease[8]. In addition, 17-KS-S value shows a sensitive change by psychological and social factor, and relates greatly to person's will, desire, and energy[8]. 17-KS-S/17-OHCS which is the value into 17-KS-S is divided with 17-OHCS, is a method of enabling the grasp of the distortion of the living organisms brought by stressor and understanding the living organisms reaction inclusively[8]. It is reported that 17-KS-S/17-OHCS is indicated clear low value (0.15 or less) under social psychology stress and rises with release of stress cause[7]. Both 17-KS-S value and 17-OHCS value are shown in ratio to Creatinine (mg/g Creatinine). As a result, the influence of the physique (such as racial difference and sex difference) decreases[8].

4 Method

In order to investigate its effects on elderly, we introduced mental commit robot to a day service center for 6 weeks from May to July 2001. The day service center is an institution that aims to decrease nursing load of a family by keeping elderly people in daytime. Services, such as bathing, massage, physical exercise and games, are provided to the elderly people there. Before introduction of the robot, we explained the purposes and ways of the experiment to the elderly people, and obtained their approval from

20 persons among 56 persons in the day service center.

A seal robot was provided to the elderly three days a week for five weeks. Because they didn't come there every day, they interacted with the seal robot one, two or three days a week. We prepared a desk to set the robot in the center of people, and the elderly were arranged up to eight people or less. If there were elderly more than eight, they were divided into two groups randomly. The elderly interacted with the robot about 20 minutes at a time. When the number of people was small, elderly people could interact with the robot about 40 minutes, if they wanted to do.

Urine was gathered in the early morning and was analyzed after frozen preservation by -18 degrees afterwards gathering. Each value was corrected with the creatinine.

We interviewed them concerning medicine which influences urine value, and life events by social readjustment rating scale of Holmes and Rahe[9] which happened in one week before gathering urine and we considered the influence on the urine value.

Besides, we used the questionnaire concerning familiarity with the robot which was introduced etc. as an index of the psychological influence which mental commit robot give the subjects.

We used Wilcoxon signed rank test for the comparison of the urine values before and after the introduction of mental commit robot.

Statistical analyses were performed with the statistical program SPSS10.0J for Windows.

5 Results and discussions

Final analytical subjects were seven persons, because there were the elderly who did not come to the day service center and a dropout person, etc., though the first subjects were 20 persons. All of the subjects were women, and the average age and the standard deviation were 84.3±7.6 years old and highest was 92 years old. There were some subjects who were dementia. Then, the nursing staff judged each dementia's level of subjects in terms of the revised Hasegawa's dementia scale (HDS-R). Their dementia's levels were as follows:

Non-dementia: five people,
Dementia: two people.
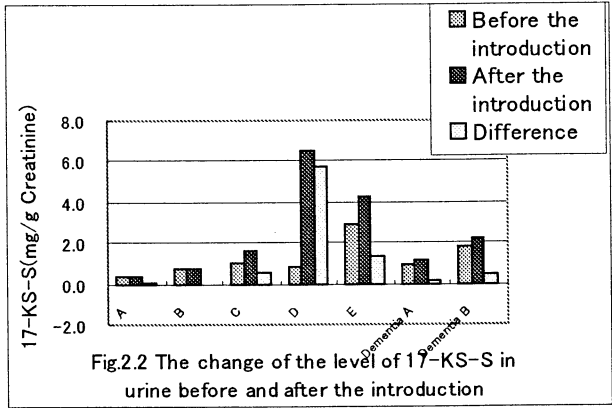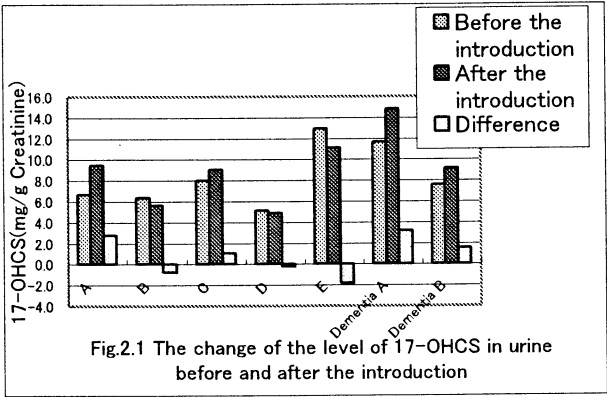
Table 1 Change before and after the introduction

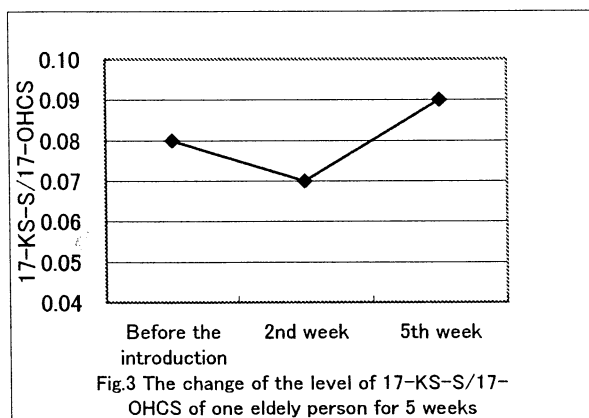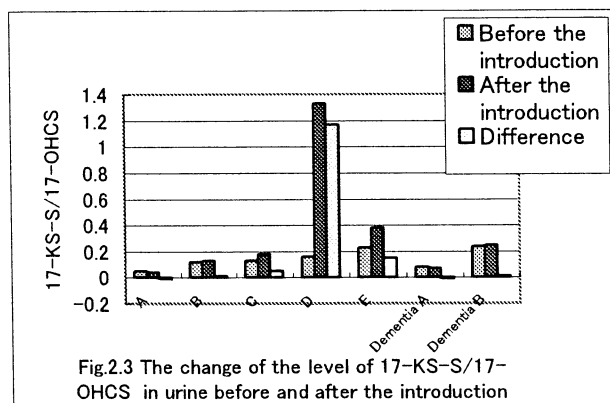|	Before	After	
17-OHCS	8.35±2.87	9.17±3.33	ns
17-KS-S	1.25±0.88	2.41±2.23	*
17-KS-S/17-OHCS	0.14±0.07	0.34±0.45	ns

n=7 Average±SD
# Wilcoxon signed rank test * p<0.05

Table1 shows 17-OHCS value that indicates the stress load degree, 17-KS-S value that indicates the restoration degree to the stress, and ratio of 17-KS-S/17-OHCS that indicates an inclusive living organisms reaction before the introduction of robot and the two weeks after. The restoration degree went up significantly though the whole stress load degree somewhat went up. An inclusive living organisms reaction was in the increasing tendency.

Fig.2.1, Fig.2.2, Fig.2.3 shows the urine value change of the individuals before and after the introduction.

Fig.2.1 The change of the level of 17-OHCS in urine before and after the introduction

Fig.2.2 The change of the level of 17-KS-S in urine before and after the introduction

Fig.2.3 The change of the level of 17-KS-S/17-OHCS in urine before and after the introduction

Fig.3 The change of the level of 17-KS-S/17-OHCS of one eldely person for 5 weeks

The difference of 17- KS-S/17-OHCS of subject A and B were small or the minus (Fig.2.3). They were dropped out of the investigation by five weeks after interaction of robot, because they didn't have much interest from the beginning.

Concerning Dementia A, she didn't like animals and had fear the robot at the beginning. Through physical interaction with the robot, she was gradually accustomed to the robot and finally liked the robot. She often laughed while interacting, and piled up the scene. Fig.3 shows the passage of 17-KS-S/17-OHCS of Dementia A afterwards. The value fell once at the second week of the introduction, but then rose up at the fifth week. This shows that her inclusive living organisms reaction became excellent.

6 Conclusions

We introduced mental commit robot to an elderly institution, and examined the change of stress reaction of elderly by urine 17-KS-S value and the 17-OHCS value before and after introduction of robot. As the results, an excellent adjustment ability to the stress was shown, and mental commit robot had a useful possibility for maintenance of health of elderly. As the further work, longitudinal investigation will be conducted.

References

1) Siegel JM, Stressful life events and use of physician services among the elderly, The moderating role of pet ownership, Journal of Personality and Social Psychology, vol. 58, pp.1081-1086 .1990.

2) T. Shibata, et al., Mental Commit Robot and its Application to Therapy of Children, Proc. of the IEEE/ASME Int'l Conf. On AIM'01, 182, CD-ROM Proc. 2001.

3) O. Nishikaze: 17-KS-Sulfate as a Biomarker in health and Disease, Clinical pathology, 46(6), pp.520-527. 1998.

4) T. Saito, M. Okada, K. kano, Effect of keeping a companion animal to stress reaction seen from 17-KS-S of elderly living at home, Medicine and biology, 141 (6) , pp. 297-302. 2000.

5) T. Shibata, K. Tanie, Application of soft computing to robot, SICE, 39-3, pp.188-193. 2000.

6) Selye H.: Stress and aging, Journal of American Geriatric Society, 18, pp. 669-676. 1970.

7) O. Nishikaze, et al., Distortion of Adaptation (Wear & Tear and Repair & Recovery)-Urine 17-KS-Sulfates and Psychosocial Atressin Humans-Job Stress Res,Vol.3,pp. 55-64. 1995.

8) E. Furuya, et al., 17-KS-Sulfate as a Biomarker in Psychosocial Stress, Clinical pathology, 46 (6), pp. 529-537. 1998.

9) Holmes T.H., Rahe R.H.: The social readjustment rating scale, J psychosom Res,Vol.11 , pp. 213-218.1967.

Human Centered Interface at Cyber-agent system

Toru Yamaguchi
PRESTO, Japan Science and
Technology Corporation (JST)
Department of Electronic Systems
Engineering, Tokyo Metropolitan
Institute of Technology
Asahigaoka 6-6, Hino City, Tokyo
Japan
Email : yamachan@fml.ec.tmit.ac.jp

Takafumi Yoshifuji
Department of Electronic Systems
Engineering, Tokyo Metropolitan
Institute of Technology
Yamaguchi lab, Asahigaoka
6-6, Hino City, Tokyo
Japan
Email : fumi@fml.ec.tmit.ac.jp

Fumio Harashima
President of Tokyo Metropolitan
Institute of Technology
Asahigaoka 6-6, Hino City, Tokyo
Japan
Email : f.harashima@ieee.org

Abstract—**The method which composes the intelligent agent system which recognizes human intention is shown in this paper. The ontology for human of each agent is build by the neural network in this collaborative agent system. Then, a difference of the intention expression by the individual is accepted by holding the ontology in common between the agents. Furthermore, I think that it will apply to the interface at the time of Web perusal.**

Keywords—Ontology, Intelligent agent, Intelligent Interface

I . INTRODUCTION

Recently, the maintenance of the information infrastructure is proceeding rapidly. Though, it has been a great problem for many people using information infrastructure, taking long time to get accustomed to keyboard and mouse.

Especially, Japan became high aged society so radically, which is unprecedent in the world. The aged have time and money for Internet. Therefore, they are being expected to become the biggest user group. A general interface to the user in place of the mouse and the keyboard is indispensable for the people who aren't accustomed to use the computer such as the aged and infants. It is considered to use human movement and voice as a Natural Interface.

The method of intention recognition is indicated in this paper. By using movement of hands and face, an intelligent agent builds the ontology for intention.

And, as the way to accept different expression by the individual in the intention recognition, the technique of holding common ontology between the agents is shown.

II . THE CONCRUTION OF THE AGENT SYSTEM

The composition of the system is shown in Fig.1. This is based on the class model of Rasmussen. Each agent gets movement of face and hands from the picture. And then, it gets an instance from the pattern of an inputted data. The ontology which coped with human intention is made by using Conceptual Fuzzy Set (CFS) composed by Bidirecitional Associative Memory (BAM). Furthermore, build the system for each agent to hold it's ontology in common.

III.INPUT OF INFORMATION

The input information for intention recognition is the human movement. Especially, for input information, the time series data of the face position, face direction and hand position are used.

To detect the movement, first, CCD camera is used to get the color information from the picture. Second, find the position of hand, face and lip from the picture. Third, calculate center gravity to specify each position. Tracking by using the color information is shown in Fig.2. Calculated coordinate value is measured as the data on time series in Fig.3. Then, the extreme value is decided as the characteristics point.

For each other part, two data of the X direction and the Y direction are held. By picking up to position of the lip and the face, it is possible to get the direction of the face. Moreover, it is normalized, that a relative position of the hand and the foot is measured with the position of the face. Therefore, it is possible to extract the character's movement of the person without getting closer to the camera.
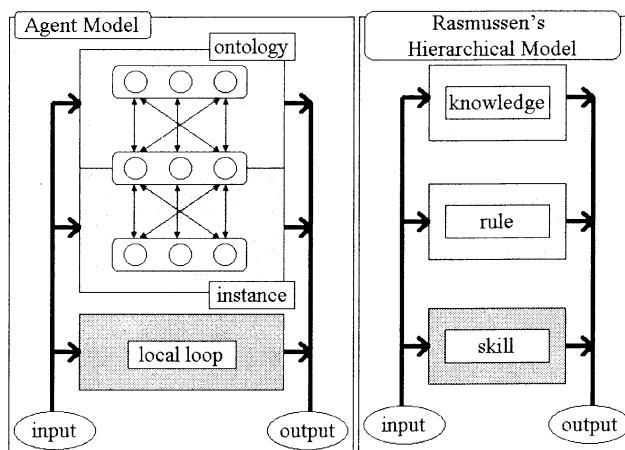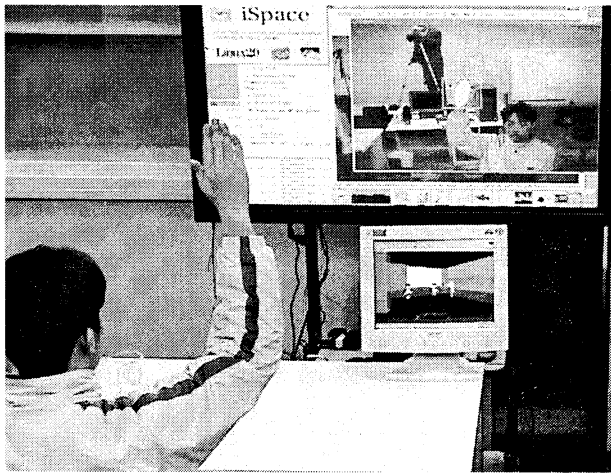

Fig. 1.   Composition of system

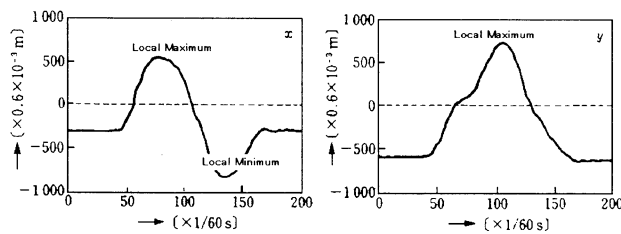Fig. 2. The scenery which does directions for the movement

Fig. 3. Time series data

Fig. 4. Example of CFS

IV. INTENSION RECOGNITION USING ONTOROGY

As for the ontology, abstract general idea structure was defined. The ontology is expressed by the relations with more than one concrete case (instance).

Because speed and size aren't fixed for each person's movement, robust nature becomes necessary to reduce the unstable noise, for the intention recognition. The unstable noise is dissolved by using CFS composed by BAM. An ontology is implemented by this CFS. An easy case is shown in Fig.4.

A human movement is inputted as an instance. An ontology recognize when spread treatment is done in the network which composes CFS. Then, intention is recognized (Fig.5).

When this method was used for recognizing the sign language, the result of the recognition precision was 96%. Also, driver's intention recognition was 86%. As for the intention recognition by the ontology, it is understood that it is fully useful from these results.

V. COOPERATIVE ONTOLOGY

When a certain agent does recognition and an ontology wasn't remembered, the joint ownership of the ontology is done by the method that uses CFS of other agents trying to recognize again. It is judged by the size of the
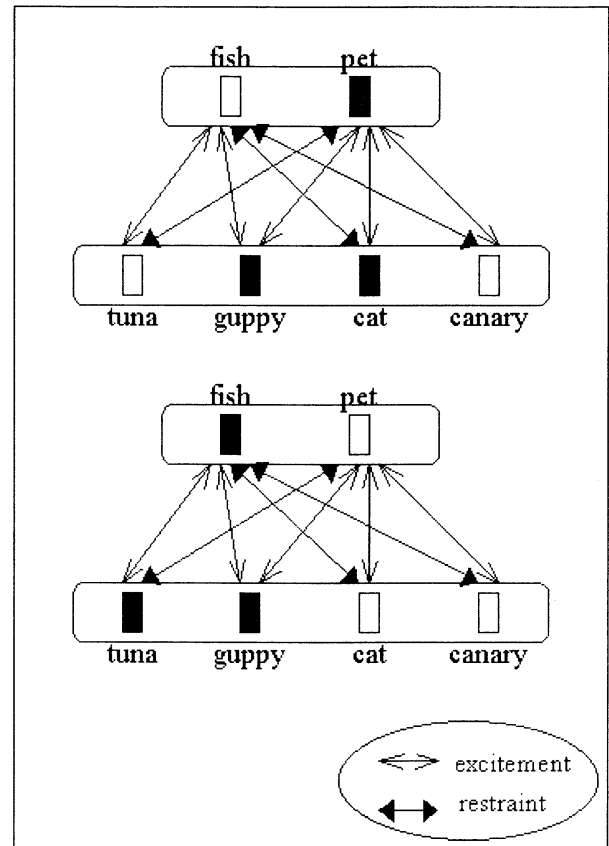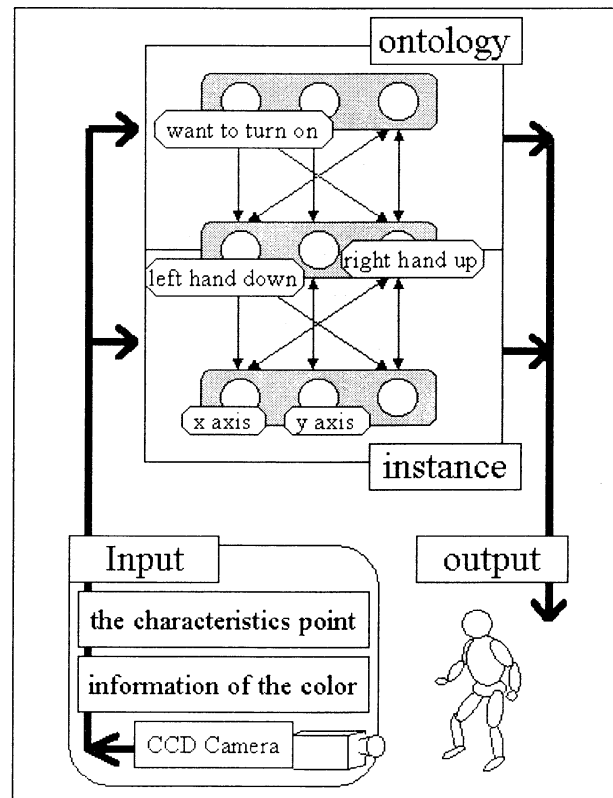

Fig. 5.

active value whether recognition is tried again in the same way as time to remember it. This is thought to be effective from controlling the degree of "fuzzy" by changing the number of times of the spread treatment (the number of the association steps) in the BAM.

A concrete process is shown by flow chart in Fig 6. A certain agent does association reasons by his CFS at first. If an active value is higher than the constant value, and ontology was remembered, intention was decided to be recognized. When active value is small, there is a possibility of error by the difference in the intention expression. Therefore, an active value is looked for by CFS of other agents, and a judgment is done in the method which is the same as former times.

In case that even active value of certain agent is small, the others get big active value existent. Ontology uses bigger value to edit and get new information against learning. Moreover, it is considered making a new ontology, too. When an instance was remembered with more than one agent from the active value of the ontology, the association reasoning is done by using both directions of BAM in a certain instance which could get the same active value.

To show it in Fig.7, human support robot shows validity with the human being by the collaboration of the intellectual agent.

VI. EXPERIMENT RESULT

A. HOW TO MAKE AN EXPERIMENT

An interface was made a human hand, and intention recognition was done in last experiment based on the position coordinate of the hand of the human being inputted from the CCD camera.
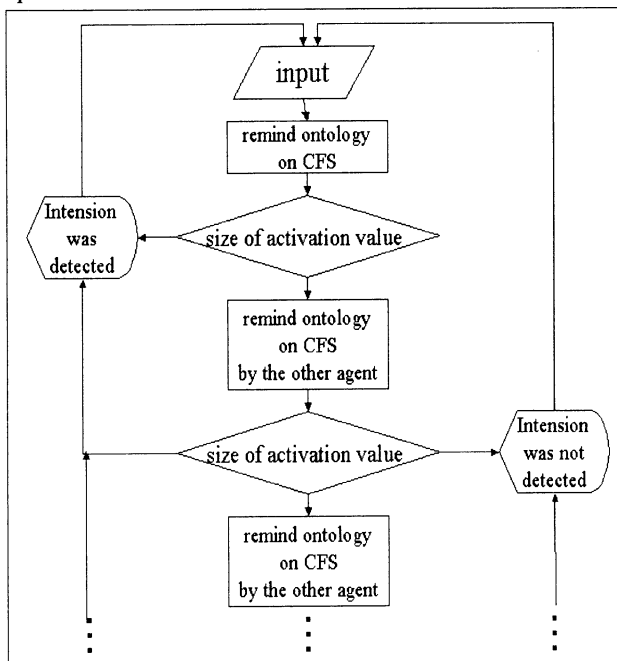

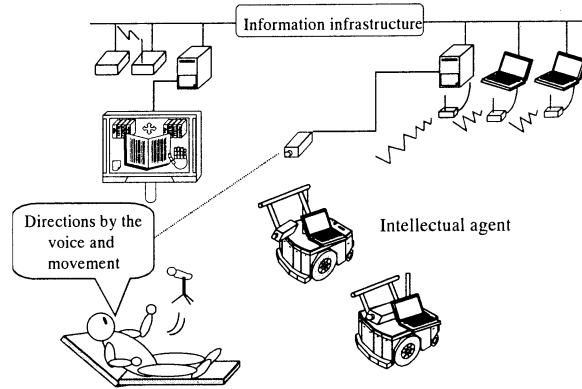Fig.6. The flow of the information joint ownership
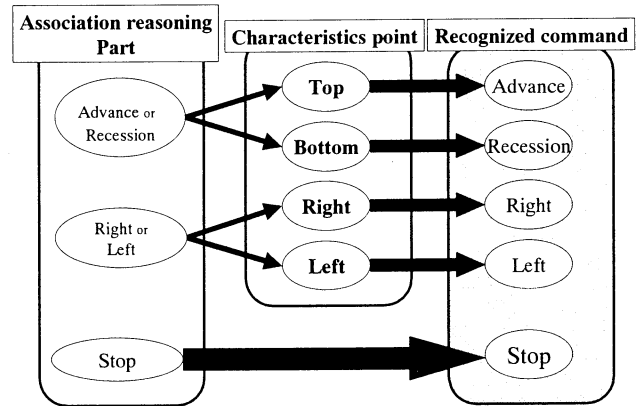

Fig.7.

Fig. 8. Command recognition part

It was done by A, B two testes.

A line says both intention recognition to the agent who has the rule of A first.

Next, intention recognitions is done to be the same toward the agent of A made to share information.

A movement recognition part judges the command of a human movement which is finally from the reasoning result of the coordinate data. The first characteristics point from the movement input part, the association reasoning part, face of a client, and the order which corresponded with the command is sent.

The method of the judgments is shown in Fig .8.

It is limited by three of the forward recession, turning right, turning left or the stopping from the conclusion of the association reasoning part first.

A stop is judged by the command where to be stopped.

When it turns at the forward recession, turning right and left, the coordinate date of the characteristics point of the beginning which the data from the characteristics point extraction next are used.

It judges an advanced or recession whether to be in the bottom if that characteristics point is in the top in comparison with the position of a hand to start a movement.

Fig.9. Experiment scenery

room A   room B

Fig.10. Information joint ownership

TABLE I
THE RECOGNITION RATE OF A IN THE AGENT OF A

Advance	Recession	Right	Left	Stop
86%	87%	95%	96%	99%

TABLE II
THE RECOGNITION RATE OF B IN THE AGENT OF B

Advance	Recession	Right	Left	Stop
88%	86%	92%	96%	96%

A characteristics point is in the right, or in the left, or turning left, turning right are judged in the same way.

It was decided that output in this experiment moved the CG space. Experiment scenery is shown in Fig.9, Fig.10.

B. EXPERIMENT RESULT

An intention recognition rate is shown in Table I , Table II in this experiment.

The rule of the stop in A, B is different.

When the room of A which shows in Fig.9, was replaced.
1. The average of the recognition rate when information joint ownership isn't is 82%.
2. The average of the recognition rate after joint ownership was done was 92%.

What we knows from the upper result is probably the point whose recognition rate of the advance and the recession is bad first.

It is raised that the influence of the light is taken with a command's coming off as a reason from the human fundamental movement.

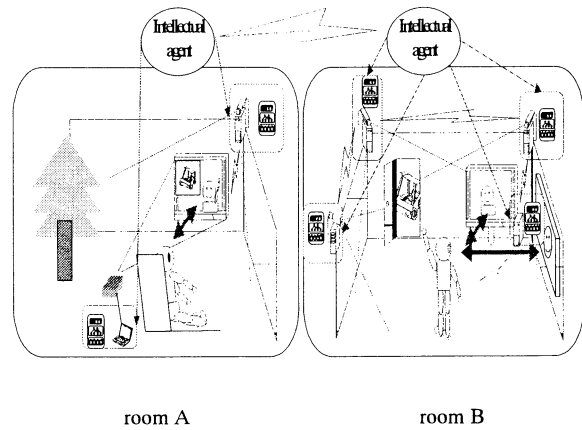A rule except for the stop of A, B was about the same in last experiment. An experiment was done by the setup that the rule of the stop was wrong.

An agent could be able to do the intention recognition of two users at the recognition rate which was same by doing joint ownership.

VII. CONCLUSION

When an agent recognizes human intention, it was shown how to hold to an ontology by the Conceptual Fuzzy Set in common as one technique that the robust enhanced in this paper.

This system is used in the intelligent transport system and welfare robot is applied to the use that a human being is exactly supported.

From now on, the validity of the system will be shown in the concrete conditions, and furthermore building multi modal interface by combining it with the voice and so on will be considered.

References

[1] T.Takagi, A.Imura, H.Ushida, T.Yamaguchi, "Conceptual Fuzzy Sets as a Meaning Representation and their Inductive Construction" *International Journal of Intelligent Systems*, Vol.10, 929-45, 1995.
[2] T.Takagi, A.Imura. H.Ushida, T.Yamaguchi, "Multi Layered Reasoning by Means of Conceptual Fuzzy Sets" *International Journal of Intelligent Systems*, Vol.11, 97-111, 1996.
[3] T.Tanihashi, M.Hagiwara, T.Yamaguchi, "Neural Network and Fuzzy Signal Treatment" *Corona inc*, 1998.
[4] T.Takagi, T.Yamaguchi, M.Sato, "Multi-Modal Information Integration by Conceptual Fuzzy Set for Interactive System" *1998 IEEE World Congress on Computational Intelligence*, 738-743, 1998.
[5] T.Takagi, S.Kasuya, M.Mukaidono, T.Yamaguchi, T.Kokubo, "Realization of Sound-scape Agent by the Fusion of Conceptual Fuzzy Sets and Ontology" *8th International Conference on Fuzzy Systems FUZZ-IEE'99, II*, 801-806, 1999
[6] T.Takagi, S.Kasuya, M.Mukaidono, T.Yamaguchi, "Conceptual Matching and its Application to Selection of TV Programs and BGMs", *IEEE International Conference on Systems, Man and Cybernetics SMC'99, III*, 269-273,*1999*.
[7] T.Takagi, K,Kawase, K.Otsuka, T.Yamaguchi, "Data retieval Using Conceptual Fuzzy Sets" *9th International Conference on Fuzzy Systems FUZZ-IEEE 2000, II*, 94-99, 2000.

Self-rewiring Network based on an Artificial Chemistry

Yasuhiro Suzuki
Hiroshi Tanaka
Bioinformatics, Medical Research Institute, Tokyo Medical and Dental University
1-5-45,Yushima Bunkyo Tokyo
113-8501 JAPAN
suzuki@gentzen.mri.tmd.ac.jp, tanaka@cim.tmd.ac.jp

Abstract

We model a self-rewiring network by using an abstract chemistry and the polya's urn network. Although an abstract chemistry is composed of reaction rules and reactants, the abstract chemistry which we will use does not have any reaction rules. They are generated by the polya's urn network. In the polya's urn network, which we have proposed, each node of the graph remembers the history of visited node. The history gives the probability of transition from the node. Thus transition probability is changed dynamically. Transitions in the network correspond to reaction rules, hence produced reaction rules are also changed dynamically. We confirm that there emerges relationship between the network model and the abstract chemistry.

Keywords: Artificial Chemistries, ARMS, Self-rewiring Network

1 Introduction

Biological organization is more than a self-maintaining closure of chemical transformations. Organization also means control, that is, starting, stopping, synchronizing and sequencing concurrent processes whose coordination underlies behavior, as W.Fontana mentioned[3].

Some signaling components are able to redefine with whom another component of the same network is communicating with. This means that the architecture or topology of the network is dynamic and (at least in part) controlled by the network state (the concentration vector) itself. A signaling component (such as a kinase) is here defined in terms of a repertoire of possible downstream targets (within the same network) that it can interact with. Yet, which of these interactions actually do occur at a given time t, is controlled by the concentration of other signaling components. This generates a feedback loop between the network architecture and the dynamics induced by it: a particular network topology determines the concentration changes of network components which, in turn, rewire the network topology. In a system of this kind, a fixed pool of components represents a variety of possible control networks from which a particular one is induced in response to an external signal[3][1] .

We modeled the network system by combining a random graph model with memory and an artificial chemistry, Abstract Rewriting System on Multisets (ARMS).

2 Random graph model with memory

Each node of the graph remembers the history of visited nodes (the name of nodes that went to the node). The history gives the probability of transition from the node, and the transition probability from node m to i is defined as

$$\frac{\Sigma n_i}{\Sigma n_m}.$$

where Σn_i denotes the number of out degree from node m to i and Σn_m denotes the total number of out degree of node m. And the probability of go out from node i to node m is changed as

$$\frac{\Sigma n_i + 1}{\Sigma n_m + 1},$$

thus transition probability is changed dynamically. This model is closely related to "Polya's urn [1]" which is a famous model of probability theory. Polya's urn is defined as follows;

---

[1] the above part reffered from W.Fontana's home page[3]

Figure 1: Polya's urns Network
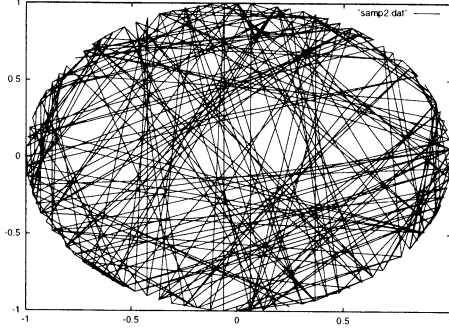

Figure 2: The distribution of thickness of edges (Polya's urn network)

Polya's urn Supposing there is a urn with b black balls and r red balls inside. When we select a ball randomly and put it back to the urn, s and c balls that are painted by the same color of the selected ball and d balls that are painted by the the other color. The procedure is repeated.

Our model allows the condition when there are more than two kind of balls as c = 1 and d = 0. When there are two kinds of colors, it is well investigated [1]. Thus we defined the above model as the network of generalized Polya's urn as follows.

Network of generalized Polya's urns Each node has a "urn" in which balls labeled node number except for own number. First a ball is randomly selected from the urn, and then two balls are returned to the urn, the added ball is labeled the same node number with the selected ball.

Since it is not easy to analyze the problem theoretically, we perform simulations in order to investigate the process of emerging network in this model.

We assume 200 nodes and preformed a simulation until 150,000 edges were linked. The figure 2 displays the network emerged.

Although it is difficult to see the hierarchical structure of edges in the figures 2, figure 2 illustrates there emerges the hierarchical structure in the distribution of thickness of edges. In the figure2 horizontal axes illustrates the thickness of a edge and vertical axes illustrates the each node. At the top of the structure there are highly used edges and based on it there are secondary used edges and the structure is formed like cascade. On the other hand, random graph does not have such the hierarchical structure (figure 2).
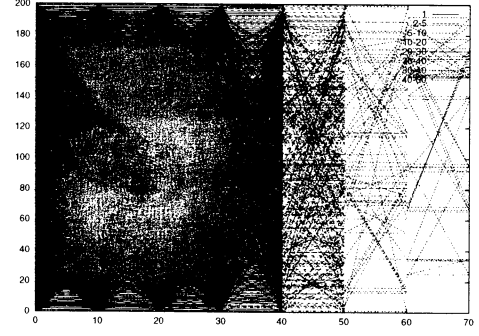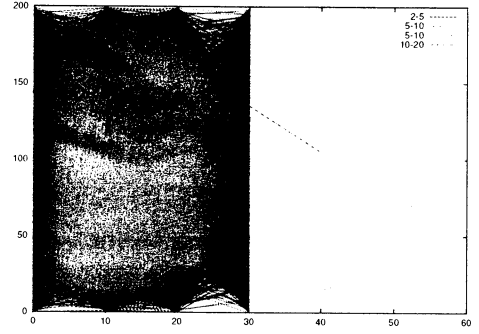

Figure 3: The distribution of thickness of edges (Random graph)

3 ARMS

Intuitively, Abstract Rewriting System on Multisets (ARMS) is like a chemical solution in which floating *molecules* can interact with each other according to reaction rules[3]. Technically, a chemical solution is a finite multiset of elements denoted $A^k = \{a, b, \ldots, \}$; these elements correspond to *molecules*, and reaction rules are specified in terms of rewrite rules. We denote an empty set by ϕ, and the *base number* of a multiset (size of a multiset) by $|S|$ (where S is a multiset), respectively.

Definition 1 (Multiset) *A "multiset" is an element $t \in A^k$ $(1 \leq k \leq n) \in \Sigma$, where n is a finite number, and A^k is a Cartesian product $A_1 \ldots A_k$. $A^k = A_1 \times A_2 \ldots \times A_k$, Σ denotes the set of multisets, and n is called the "maximal multiset size."*

The multisets correspond to possible states of *chemical solution*. The set of multisets corresponds to the space

of transitions of an ARMS.

Definition 2 (Rewriting rule) *A "rewriting rule" is a relation $l \mathcal{R} r$ (l, $r \in \Sigma$), $|l|, |r| \leq$ maximal multiset size, n. A rewriting rule $l \mathcal{R} r$ is denoted as $l \rightarrow r$.*

Definition 3 (ARMS) *An "Abstract Rewriting System on Multisets" (ARMS) is a pair (T, Ru) consisting of a multiset T and a set Ru of rewriting rules.*

Definition 4 (Rewriting on ARMS) *Let (T, Ru) be an ARMS. We rewrite $s \xrightarrow{Ru} t$ if there exists a rewriting rule $l \rightarrow r \in Ru$ such that $l \subseteq s$ and $t = (s - l) \cup r$.*

Definition 5 (Normal Form in ARMS) *If no rule in Ru can be applied to a multiset and no symbols can be inputted to the multiset without the resulting base number exceeding the limit on the multiset, then the multiset is called Normal Form (final state).*

Normal forms correspond to a steady state.

The reader will notice that the method of rewriting of ARMS is different from that of the abstract rewriting system. Since the abstract rewriting system is a string-replacing system, the string ab and the string ba are treated as different strings on rewriting. On the other hand, since ARMS is a multiset replacing system, the system regards ab and ba as multisets of symbols, $\{ab\}$ and $\{ba\}$. Thus, they are treated the same.

An algorithm for rewriting steps in ARMS is described in figure 4.

Example In this example, we assume that a will be inputted on each rewriting step, the maximal multi-set size is 4 and the initial state is given by $\{a, a, f, a\}$. The set of the rewriting rules, Ru_1 is $\{r_1, r_2, r_3, r_4\}$, where each rule is described by the following:

$$aaa \rightarrow b : r_1, \ b \rightarrow a : r_2, \ b \rightarrow c : r_3, \ a \rightarrow bb : r_4.$$

In this example, we assume that rules are selected as following the order $\{r_4 \Rightarrow r_1 \Rightarrow r_3 \Rightarrow r_2\}$. Then, each rule is applied in the following way. First, r_4 is applied. Next, as steps 2 and 3, r_1 and r_3 are applied, respectively. Finally, as step 4, r_2 is applied.

Figure 5 illustrates two rewriting steps of the calculation from the initial state.

As the first step, since the base number of the multi-set is 4, the system can not input a. On the left hand side of r_4, a is included in $\{aafa\}$, however, r_4 can not be used. If a is replaced with bb, the base number of

```
procedure ARMS (Rewriting Step)
begin
    count-step ← 0;
    while count-step ≠ n do
        begin
            if the multiset reached Normal Form then
                count-step:= n;
            else
                Input string(s) to the multiset;
                Select a rule;
                if the rule can rewrite the multiset then
                    Rewrite the multiset;
                    count-step := count-step + 1;
                end if
            end if
        end
    end while
end.
```

Figure 4: An Algorithm for ARMS (until n-th step)

$$
\begin{aligned}
\{aafa\} \quad &\subseteq a \text{ (the left hand side of } r_4) \\
\downarrow \quad &\dots \text{ can not input } a \text{ and can not apply } r_4, \\
\{aafa\} \quad &\subseteq aaa \text{ (the left hand side of } r_1) \\
\downarrow \quad &\dots \text{ can not input } a \text{ but can apply } r_1 \\
\{ba\}
\end{aligned}
$$

Figure 5: Example of rewriting steps of ARMS

the multi-set becomes 5 and it exceeds the maximal multi-set size, 4.

In the next step, the system can not input a, however, r_1 can apply to the multi-set and $\{aafa\}$ is rewritten into $\{ba\}$ (because if aaa is replaced with b, the base number of the multi-set does not exceed the maximal multi-set size, see Figure 5).

In step 3, ARMS inputs a to the multi-set and transforms it to $\{c, a, a\}$ with r_3.

$$\text{Step 3} : \{c, a, a\}.$$

In step 4, the system inputs a, but r_2 can not apply to it. Thus $\{c, a, a\}$ becomes $\{c, a, a, a\}$.

$$\text{Step 4} : \{c, a, a, a\}.$$

4 Self-rewiring network system

In order to compose the network model, we combine the Polya's urn network and the ARMS with no

reaction rules. In this system, the Polya's urn network produces reaction rules for the ARMS.

Since the ARMS dose not have any rules, the Polya's urn network generates rewriting rules; where a link corresponds to a rule, for example, a link from a to b means a rewriting rule, $a \rightarrow b$. The left hand side of a rule is selected probabilistic according to its density, for example, when there are three kind of compound a,b and c, the probability of selecting a, P_a is defined as

$$P_a = \frac{|a|}{|a| + |b| + |c|}.$$

Hence the density of compound is switched by the change of density.

5 Experimental result and Conclusion
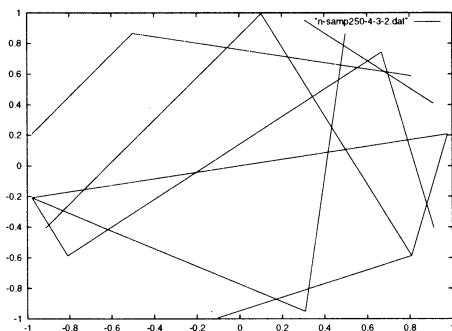
Figure 6: Polya's urn network of Self-rewiring network (no.1)

We confirmed that there emerges the relationship between the ARMS and the Polya's urn network. The figures 6 and 7 illustrate generated the Polya's urn networks where the each node corresponds to a type of compound. The topology of the network is changing according to the reactions in the ARMS. It is interesting that the topology of the network can not be stable, although sometime its topology is fixed to simple pattern (as figure7), eventually it is destroyed (as figure6).

As Fontana mentioned[3], The obvious resemblance of this result is neural networks. Learning is, perhaps, the construction of memory or at least a history-dependent long-term modification of parameters governing a dynamical system. In this experiment, a self-rewiring network do learn, and eventually respond in a
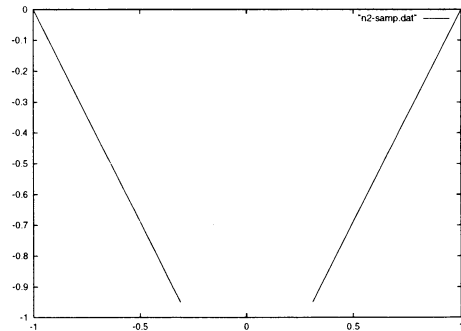
Figure 7: Polya's urn network of Self-rewiring network (no.2)

characteristic fashion to the re-presentation of a signal to which they have been exposed before in the Polya's urn network.

If genes coding for the components of a network are among the targets controlled by that same network, long-term endogenous architectural changes, such as the removal of nodes ("knock-outs") or the addition of nodes ("knock-ins"), can discuss by using this model[3]. They are our future work.

References

[1] Feller W., An Introduction to Probability Theory and Its Applications, I, 1957.

[2] Fontana, W. and L. W. Buss. 1994. The arrival of the fittest: Toward a theory of biological organization. *Bulletin of Mathematical Biology* 56: 1–64.

[3] Fontana, W., Self-rewiring signaling networks: Organization as distributed control, *http://www.santafe.edu/ walter*

[4] Y. Suzuki and H. Tanaka, Symbolic chemical system based on abstract rewriting system and its behavior pattern, *Journal of Artificial Life and Robotics*, 1 (1997), 211–219.

Network Structure of Genetic Codes

Tetsuya Maeshiro
maeshiro@atr.co.jp
Dept.4
ATR Human Information Science Labs.
Kyoto, 619-0288 JAPAN

Abstract

All deviant genetic codes are predicted from the network model of genetic codes. Values of the average path length and clustering coefficient indicate network representation of genetic codes belong to the class of small world networks, and similar properties of network structure are found. However, the role of these properties are different from conventional definition. Weighted link model is also fundamental for the accurate analysis.

1 Introduction

The standard genetic code (SGC) (Table 1) is not universal. With the advance of genome sequencing projects, many deviant genetic codes have been found (Table 2), common in mitochondria and simple organisms, and more deviant genetic codes will probably be discovered in future.

These deviant genetic codes have probably evolved from SGC, and they can be predicted based on the hypothesis of robustness and changeability [1]. Since then, new deviant genetic codes which were predicted from the hypothesis have been found. The prediction is based on the network model of genetic codes, representing them with a graph.

Some properties of network structure of genetic codes conserved during the evolution of genetic codes, which are similar to the properties found in the study of structure of complex networks [2].

Although networks with weighted links have been studied [3], focus of study has been on networks with unweighted links [2]. The links of network model of genetic codes are weighted, for they represent the transition probability among nodes.

The network model of genetic codes is under strong biological constraints. The network size is dictated from the fixed size of genetic codes to 64 codons, and

Table 1: Standard Genetic Code (SGC)

UUU ⎤Phe UUC ⎦ UUA ⎤Leu UUG ⎦	UCU ⎤ UCC ⎟Ser UCA ⎟ UCG ⎦	UAU ⎤Tyr UAC ⎦ UAA *stop* UAG *stop*	UGU ⎤Cys UGC ⎦ UGA *stop* UGG Trp
CUU ⎤ CUC ⎟Leu CUA ⎟ CUG ⎦	CCU ⎤ CCC ⎟Pro CCA ⎟ CCG ⎦	CAU ⎤His CAC ⎦ CAA ⎤Gln CAG ⎦	CGU ⎤ CGC ⎟Arg CGA ⎟ CGG ⎦
AUU ⎤ AUC ⎟Ile AUA ⎦ AUG Met	ACU ⎤ ACC ⎟Thr ACA ⎟ ACG ⎦	AAU ⎤Asn AAC ⎦ AAA ⎤Lys AAG ⎦	AGU ⎤Ser AGC ⎦ AGA ⎤Arg AGG ⎦
GUU ⎤ GUC ⎟Val GUA ⎟ GUG ⎦	GCU ⎤ GCC ⎟Ala GCA ⎟ GCG ⎦	GAU ⎤Asp GAC ⎦ GAA ⎤Glu GAG ⎦	GGU ⎤ GGC ⎟Gly GGA ⎟ GGG ⎦

the mapping of amino acids to codons is clearly constrained [1].

[1] analyzed genetic codes focusing on the evolutionary factors of genetic codes. This paper studies genetic codes from the framework of network structure, and show that some common properties found in so called small world networks are present in genetic codes.

2 Network Model of Genetic Codes

The genetic code is a coding table between 64 codons and 21 phenotypes (20 amino acids and the stop codon). The robustness and changeability of genetic codes are calculated based on their network model (Figure 1). A node represents a set of codons assigned to the same phenotype, where any pair of codons in the set can be changed by a successive mutation in a single base, Therefore, the six codons coding

Ser is divided into two nodes $S2$ and $S4$. Two nodes are connected by an edge when at least one codon in a node differs only in single base from a codon in another node.
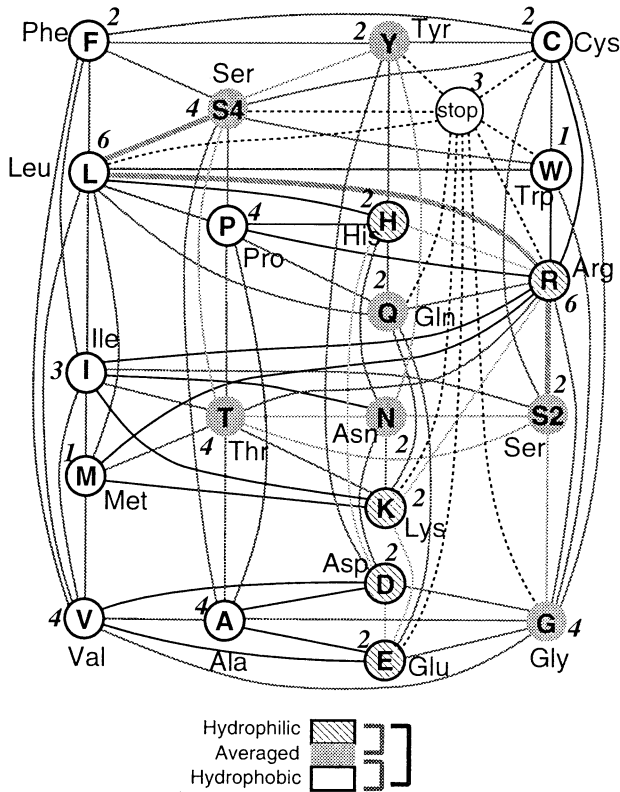

Figure 1: Network model of SGC. Numbers in nodes indicate the number of codons belonging to that node. Node fill pattern represent the polarity of amino acids. Leu, Ser and Arg are respectively the hub nodes for hydrophobic, averaged, and hydrophilic amino acids.

The robustness is the average probability that a codon keeps coding the same phenotype when a single base of codon mutates.

An opposite property to robustness, the changeability measures the average of the transition probabilities along the shortest paths between all of the pairs of phenotypes in the graph representation of the code, because the shortest paths between the nodes practically determine the transition probabilities, and consequently, the changeability of the code. For paths between nodes of amino acids, those paths linked by the node of stop codons are removed, denoted interrupted paths, because they correspond to the nonsense mutations that result in the synthesis of shorter proteins,

and most of them have no biological activity.

The third property, s-robustness, measures the robustness against nonsense mutations, considers the interrupted paths that were excluded from the calculation of the changeability due to the deleterious consequences of nonsense mutations.

Deviant genetic codes have evolved from SGC in order to maximize the variations in one of structural parameters of genetic codes [1]. The robustness, s-robustness, and changeability of genetic codes become relevant when the DNA sequence changes, particularly through replication. These measures are related with the survivability and adaptability of species. With a high robustness, the probability to conserve the protein sequence and its functionality is high. On the other hand, a high changeability gives larger variations of amino acid sequences after replications.

3 Discussions

A conserved property of network structure is the hub node, which has large number of links. During the evolution, genetic codes maintained hub nodes, where the selection of amino acid functioning as the hub is closely related to the structural stability of proteins.

The difference between the conventionally studied networks, whether regular, random or small-world, and the network model of genetic code is clear for the hubs. The conventional concept of network robustness cannot be directly applied to genetic codes. In other words, network robustness is commonly described as the strength of network to resist to the loss of nodes or links. In the network model of genetic codes, loss of nodes or links is impossible. An extreme possibility is that the genome of an organism completely lacks certain amino acid. However, this also is not the case, because a mutation may result in the reappearance of codon coding for missing amino acid. another more extreme possibility is that an organism completely lacks the synthesis mechanism of the missing amino acid. Similar to the former case, mutation may recover the codon of the missing amino acid. This is lethal for the organism, because the codon coding for missing amino acid does not function as the stop codon, and the protein synthesis process simply halts. A third possibility is that an organism lacks tRNA for missing amino acid. This is also lethal as in the second case. There are some primitive organism with very short genome that lacks the tRNA for certain amino acids, but this should be considered as an exception for the reasons listed above. Consequently, none of three mentioned possibilities are viable.

The function of hub in network of genetic codes is to provide robustness of protein structure against mutation. This is opposite from conventionally studied networks, because hub is their weak point against attacks.

The most influencing property of amino acids in protein structure is polarity [4], and SGC presents one hub node for each group of neutral, hydrophobic, and hydrophilic amino acids. Deviant codes also provide hub nodes, with the tendency is to increase the number of hubs of neutral amino acids.

The attack source to network of genetic codes is mutation that may cause transition among nodes, and not the deletion of nodes or links. In the context of robustness against attacks, hubs in networks of genetic codes clearly provides defense, contrary to other networks studied [2]. The protection mechanism is different, and the attack mechanism is also different. Analysis of network tolerance in [2] is limited, and not general as it seems.

The links weight, or transition probability, of network model of genetic codes is governed by the mutation rate of organisms. The mutation among the four nucleic acids is classified in two types, transversion and transition (Figure). The mutation rate of AT and GC are different for each species, thus the link weight distribution is specific for each species. The frequency of codons in genome also affects the link weight distribution. The codon usage frequency is also characteristic for each species. Therefore, species with same genetic code should be analyzed separately for detailed analysis.

It is common to analyze the distribution of node connectivity to test scalability of networks. Such test cannot be applied to the networks of genetic codes because of their small size, with 22 or 23 nodes in most cases. Whether the network of genetic codes is scale free or not is of secondary importance, because the network structure is strongly constrained from biological factors such as codon length, mutation rate, and physico-chemical properties of amino acids. Indeed, Figure 2 proves the insufficient network size.

Figure 3 is the relative variation of clustering coefficient C between SGC and deviant codes. Change in C indicates change in randomness of the network. No unified change is found among deviant codes, suggesting a secondary importance of properties of network structure during the evolution of genetic codes. Figure 3 indicates that widely used simple link model gives wrong results, and weighted link is necessary if the network to be modeled requests so. Relative variation of values should be similar at least qualitatively,
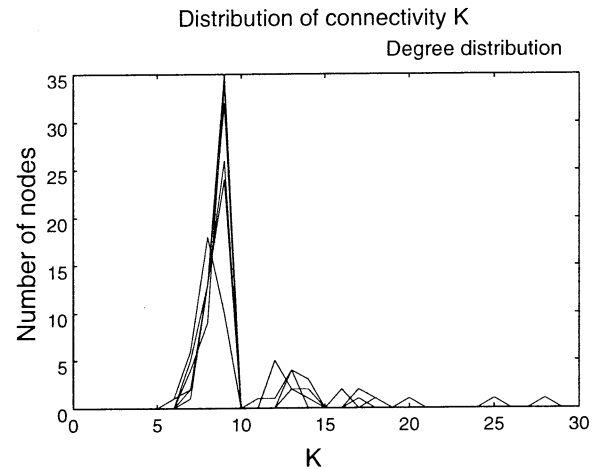

Figure 2: Node degree distribution of SGC and five deviant genetic codes.

for quantitative match is certainly rare. However, no unified change is observed.
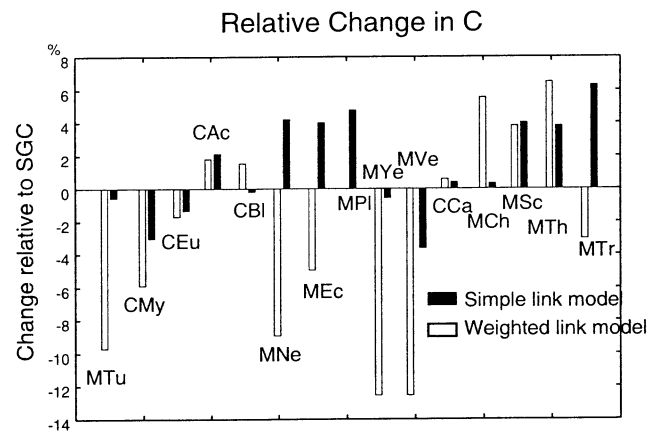

Figure 3: Node degree distribution of SGC and five deviant genetic codes.

The conventional concept of network robustness (attack tolerance) is more appropriate to compare different genetic codes, but also in this case constraining factors are completely different from theoretically studied "constraint-free" networks.

The analysis of error tolerance of networks is also inapplicable to genetic codes, for similar reasons discussed in attack tolerance.

4 Conclusions

Current framework of network analysis is too abstract for the detailed analysis of real networks, and the published qualitative results (some reviewed in [2]) cannot be applied for that reason, at least for the genetic codes. Specific and detailed model is fundamental. For instance, without the knowledge on DNA replication mechanism, one may reach opposite conclusion to the role of hubs on network robustness. Moreover, the analysis of network scalability is meaningless for networks of genetic codes.

The networks of genetic codes are neither regular nor random, and the values of average path length and clustering coefficient indicate they belong to the class of small world networks. However, our analysis suggests conventional methods to analyze network structure should be used with care, as they may give inappropriate or misleading results if the underlying factors of network origin are ignored.

Acknowledgements

References

[1] T. Maeshiro and M. Kimura, "The role of robustness and changeability on the origin and evolution of genetic codes", *Proc. Natl. Acad. Sci. USA*, Vol. 95, pp. 5088-5093, 1998.

[2] R. Albert and A.-L. Barabasi, "Statistical Mechanics of Complex Networks", arXiv:cond-mat/0106096v1, 2001.

[3] M. E. J. Newman, arXiv:cond-mat/0011155, 2001.

[4] W. A. Lim and R. T. Sauer, *Nature* Vol. 339, pp. 31-36, 1989.

Table 2: Known deviant genetic codes. Changes (deviances) involving hub nodes are marked with double circle (◎). In codon specification, N denotes any of A, T, G, C, R denotes any of A, G.

Code	Changes from SGC		
	codon	phenotype	
MYe	UGA	stop ⇒ Trp	
	AUA	Ile ⇒ Met	
	CUN	Leu ⇒ Thr	◎
MPl	UGA	stop ⇒ Trp	
	AAA	Lys ⇒ Asn	
	AGR	Arg ⇒ Ser	◎
	UAA	stop ⇒ Tyr	
MNe	UGA	stop ⇒ Trp	
	AGR	Arg ⇒ Ser	◎
	AUA	Ile ⇒ Met	
MCh	UAG	stop ⇒ Leu	◎
MSo	UAG	stop ⇒ Leu	◎
	UCA	Ser ⇒ stop	◎
MEc	UGA	stop ⇒ Trp	
	AAA	Lys ⇒ Asn	
	AGR	Arg ⇒ Ser	◎
MTu	UGA	stop ⇒ Trp	
	AUA	Ile ⇒ Met	
	AGR	Arg ⇒ Gly	◎
MTr	UGA	stop ⇒ Trp	
	AAA	Lys ⇒ Asn	
	AUA	Ile ⇒ Met	
MVe	UGA	stop ⇒ Trp	
	AUA	Ile ⇒ Met	
	AGR	Arg ⇒ stop	
MEu CMy	UGA	stop ⇒ Trp	
CEu	UGA	stop ⇒ Cys	
CAc	UAR	stop ⇒ Gln	
CBl	UAG	stop ⇒ Gln	
CCa	CUG	Leu ⇒ Ser	◎

Diffusion Property of Dynamically Generated Scale-free Networks

Yukio Hayashi
School of Knowledge Science
Japan Advanced Institute of Science and Technology, Hokuriku
Tatsunokuchi, Ishikawa 923-1292, Japan
yhayashi@jaist.ac.jp

Abstract

We investigate the speeds and rates of information spread over topologically different random (RN) and Web-like networks (WN). By a simulation, we found the spread time is proportional to the size of RN; in contrast, it is independent of that of WN, slightly more edges are needed to be spread over the whole.

Keywords: Information spread, Self-organized network, Growth, Preferential attachment, Power law

1 Introduction

Although self-organized complex networks in social and physical systems are seemed to be disorder, there is a common topological characteristic. For example, the WWW (World Wide Web), the network of movie-actors collaborations, neural networks of the worm C.elegans, and the electric power grid systems, have the following advanced (static) properties [1]:

- The averaged distance between any two vertices is small like a random network (RN). In particular, any two vertices in the WWW are connected by only nineteen clicks.

- The network is well-clustered like a regular network.

- The distributions of connectivity (in- and out-degrees) decays as power law. There are many vertices with sparse connectivity, and a few vertices with very dense connectivity.

Such networks positioned between random and regular ones are called "small-world networks [1]" or "scale-free networks [2] [3]" because the above properties are not dependent on the network size. In the inhomogeneously connected networks, a generic property of the vulnerability to failures or attacks has been shown as the Achilles' heel of the Internet: when the most connected vertices are eliminated, it is drastically isolated into small parts and the ability of communication is rapidly decreased [4].

On the other hand, the dynamic property of diffusion or cascaded information spread to adjacency vertices has been not well-studied for randomly generated self-organized networks. There is only one report based on a computer simulation [5], in which the diffusion on the RN as a rumor model can spread over the whole of it (more than 90 % vertices) from any initial vertex, when the average number of edges are more than three.

In this paper, we investigate dynamic properties of the diffusion speed (speed of the information spread) and rate (percentage of the information spread in the whole network) for the scale-free Web-like networks (WN) with special decay parameters of the in- and out-degree distributions, comparing with ones for the RN.

2 Network Models

Let us consider a directed simple graph without self-loops and multi-edges. Each vertex has binary states with respect to information delivery to the adjacency vertices; 0: undelivered state, 1: delivered state. Thus, updating the states from 0 to 1, the information spread on the network from the origins (vertices with initial 1-state).

2.1 Web-like network

The scale-free distribution of connectivity has been commonly found in many large scale networks [1] [2] [6] [7]. The distribution of vertices with the number of i-edges is proportional to $i^{-\gamma}$, $\gamma > 0$. Surprisingly, a self-organized network with such power law distribution is dynamically generated by only two simple mechanisms [3],

growth: adding new vertices,

preferential attachment: link preference to vertices as rich-get-richer phenomenon.

Both mechanisms are essential to the power law distributions, and common in many self-organized networks. If the one of them is eliminated, different distributions are obtained, while a power law distribution is derived from the mechanisms by a mean-field method [3]. In addition, the Watts-Strogatz model which is constructed by randomly exchanging edges of a regular graph with a small probability ($p \approx 0.01$) known as a small-world network [1] does not follow the distribution. Although it is still open which model is appropriate, both of small-world and scale-free properties may be essential to a self-organized dynamical network.

Artificially, to generate such networks based on the two mechanisms, the (α, β)-model has been proposed [7]. In the model, the probability distributions

$$In: p_{i,t} = i^{\frac{-1}{1-\alpha}}, \quad Out: p'_{i,t} = i^{\frac{-1}{1-\beta}},$$

are approximately derived. In the WWW, the parameters for in-degree and out-degree distributions are $\gamma_{in} = 2.1$ and $\gamma_{out} = 2.38$, $\alpha = 0.52$ and $\beta = 0.58$, respectively. The other networks of movie-actors collaborations, worm C.elegans, and power grid systems have different values [2]. We call the network with the power law distributions of γ_{in} and γ_{out} as WN.

Let us consider examples of several graphs with the total number of vertices: $n = 500, 1000$ and the average number of edges; $\bar{k} = 0.97, 1.6, 2.2, 3.5, 4.8, 6.1$. Note that self-loops and multi-edges are finally eliminated in the probablistic generation of edges par a vertex $k = 1, 2, 3, 5, 7, 9$, corresponded to $\bar{k}$. For the combinations of n and $\bar{k}$, we confirm the networks randomly generated by the (α, β)-model have the power law distributions as shown in Figs. 1 and 2 (vertical axis: the number of vertices, horizontal axis: the number of edges i).

2.2 Random network

On the other hand, a RN is generated by uniformly random connections between any two vertices without the network growth. It is called Erdős-Reńy model [3] with a Poisson distribution differs from the power law. We confirm the distribution in experiment.
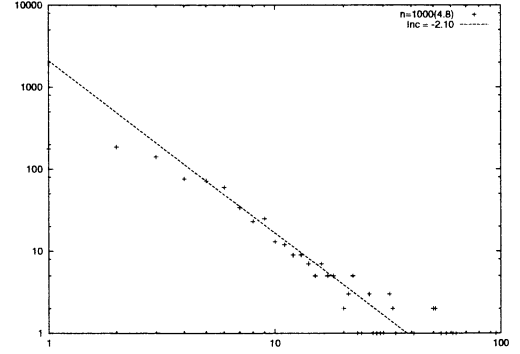
Figure 1: In-degree distribution : $i^{-2.1}$.
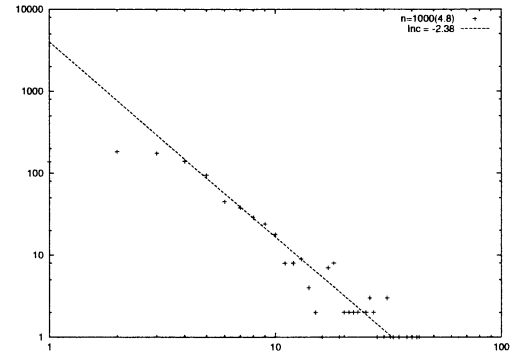
Figure 2: Out-degree distribution : $i^{-2.38}$.

3 The Diffusion Properties

3.1 Dependency on the initial point

We investigate the difference of diffusion properties for the initial point as the origin of information spread. The point is randomly chosen. Figures 3 and 4 show there are little difference in the RN, while the dependency on the initial points in the WN. The spread curves are sigmoid in the both. We obtained the same results for different graphs of the RN and WN generated in ten examinations, respectively, including other combinations of n and $\bar{k}$.

3.2 Relation between the rate of spread and the average number of edges $\bar{k}$

We also investigate the diffusion properties for the network size n and the average number of edges $\bar{k}$. Comparing with Fig. 5 and 6 in the cases of $n = 500, 1000$, we remark that the speed of spread is faster as larger $\bar{k}$ and smaller n in the RN. The graphs are shifted between the two cases of $n = 500, 1000$. However, in the WN, it is almost constant and independent

of the size n (WN/R: randomly chosen initial point in Fig. 7 and 8, WN/M: the maximum out-degree initial point in Fig. 9 and 10).

Therefore, it is supported that the spread time until the convergence by the diffusion and delivery is proportional to the size of RN because of the uniformly random connection structure (uniform delivery). While the effect of size is relaxed in the WN, because a few vertices with much more edges are generated as larger size in the heterogeneous structure mixing with dense (concentrated delivery) and sparse connections.

In both cases of RN and WN, the rate of information spread is higher as larger $\bar{k}$. From the results (Figs. 5 ~ 10), we remark $\bar{k} > 3$ is necessary in the RN [5], however slightly more $\bar{k} > 4.8$ in the WN, to be spread over more than 90 % in the whole networks. Moreover, in each pair of n and $\bar{k}$, the speed until the convergence on the WN is faster than that on the RN. The difference is remarkable as smaller $\bar{k}$.
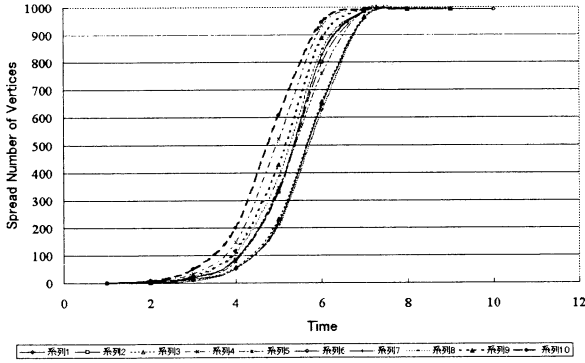
Figure 3: Dependency on initial points (RN: $n = 1000$, $\bar{k} = 4.8$).
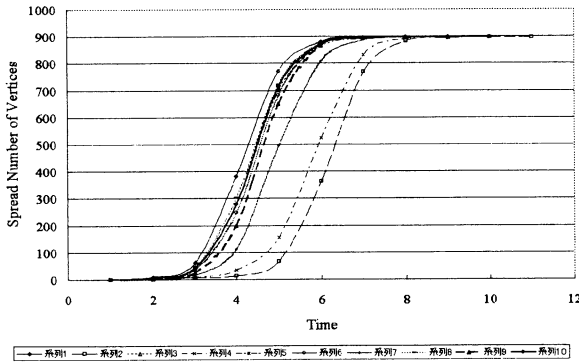
Figure 4: Dependency on initial points (WN: $n = 1000$, $\bar{k} = 4.8$).

4 Conclusion

On the artificially generated networks, we have investigated the dynamic properties of the diffusion speeds and rates of information spread over topologically different RN and WN.

The simulation results show that the speed of spread on the RN is proportional to the network size, however, it is constant on the scale-free WN, while the convergence on the WN is faster than that on the RN within the same network size and average number of edges. In contrast, to be fully spread more than 90 % in the whole network, slightly more edges (averagingly more than 5) on the WN than the averagingly three edges on the RN are necessary.

These results will be useful for many applications to efficient crawling of Web robot servers, rapid message delivery, epidemics or rumor model, estimation or design of networks in self-organized social or physical systems, and so on.

References

[1] Watts DJ, Strogatz SH (1998), Collective dynamics of 'small-world' networks. *nature* 393:440-442.

[2] Barabási A-L, Albert R (1999), Emergence of Scaling in Random Networks. *Science* 286:509-512. http://www.nd.edu/ networks/slide/table.html

[3] Barabási A-L, Albert R, Jeong H (1999), Mean-field theory for scale-free random networks. *Physica A* 272:173-187.

[4] Albert R, Jeong H, Barabási A-L (2000), Error and attack tolerance of complex networks. *nature* 406:378-382.

[5] Ninagawa S, Tsuda N, Hattori S (2000), An Information Diffusion Model (in Japanese). *Trans. on IPSJ*, 41(2):517-520.

[6] Faloutsos M, Faloutsos P, Faloutsos C (1999), On Power-Law Relationships of the Internet Topology. *Proc. of the ACM SIGCOM Conf.* 29(4).

[7] Kumar R et al. (1999), Extracting large-scale knowledge based from the web. *Proc. of the 25tn VLDB Conf.* pp. 7-10.

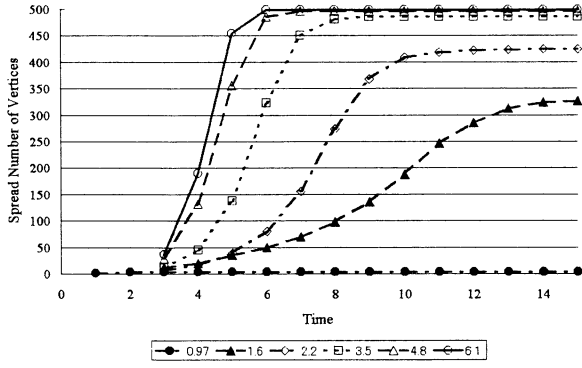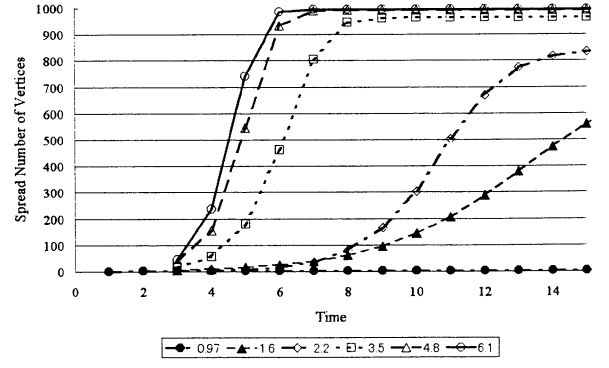Figure 5: Spread curves for $\bar{k}$ (RN: $n = 500$).

Figure 6: Spread curves for $\bar{k}$ (RN: $n = 1000$).
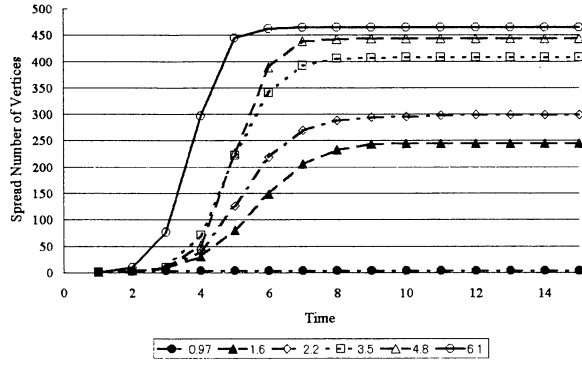

Figure 7: Spread curves for $\bar{k}$ (WN/R: $n = 500$).
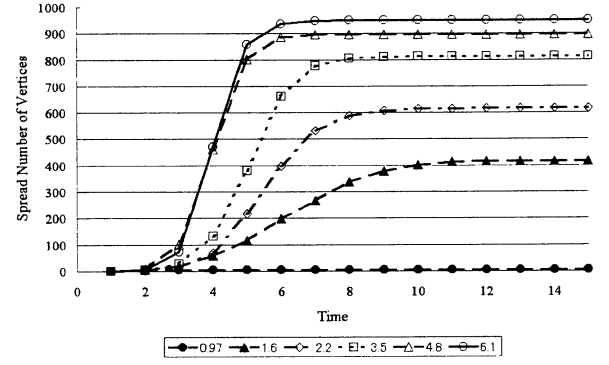

Figure 8: Spread curves for $\bar{k}$ (WN/R: $n = 1000$).
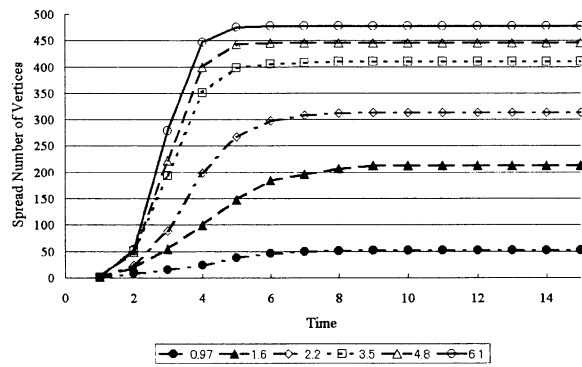

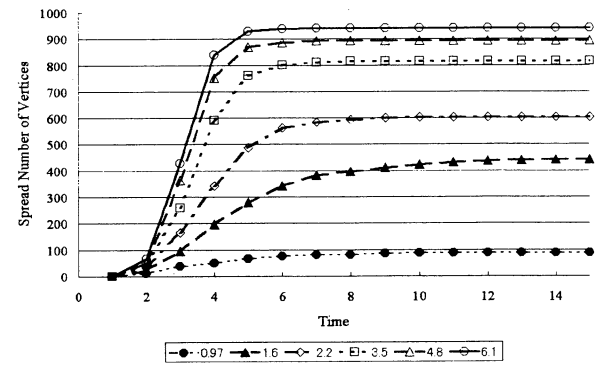Figure 9: Spread curves for $\bar{k}$ (WN/M: $n = 500$).

Figure 10: Spread curves for $\bar{k}$ (WN/M: $n = 1000$).

Wealth Distribution in Complex Networks

Wataru Souma
Human Information Science
Laboratories, ATR International
Kyoto, 619-0288, Japan
souma@atr.co.jp

Yoshi Fujiwara
Keihanna Research Center
CRL
Kyoto 619-0289, Japan
yfujiwar@crl.go.jp

Hideaki Aoyama
Faculty of Integrated Human
Studies, Kyoto University
Kyoto 606-8501, Japan
aoyama@phys.h.kyoto-u.ac.jp

Abstract

We construct a model of wealth distribution, based on an interactive multiplicative stochastic process on static complex networks. Through numerical simulations we show that a decrease in the number of links discourages equality in wealth distribution, while the rewiring of links in small-world networks encourages it. Inequal distributions obey log-normal distributions, which are produced by wealth clustering. The rewiring of links breaks the wealth clustering and makes wealth obey the mean field type (power law) distributions. A mechanism that explains the appearance of log-normal distributions with a power law tail is proposed.

1   Introduction

With the advance of the theory of complex networks (for reviews see Watts[1], Newman[2], Strogatz[3], Albert and Barabási[4], and Dorogovtsev and Mendes[5]), it has become clear that many social networks can be understood as small-world networks (Watts and Strogatz[6]) and/or scale free networks (Albert et al[7]). On the other hand, recent high-precision studies of economic phenomena, helped by the availability of high-quality data in digital format (Aoyama et al[8], and Souma[9]), is starting to reveal the basic characterics of wealth distribution, which form a basis and at the same time a final product of economic activities. In view of these developments, we propose a stochastic model of wealth distribution built on a complex network and present the results of our numerical simulation in this paper.

Several models for wealth distribution have been proposed with some reality and success (Ispolatov et al[10], Solomon[11], Bouchaud and Mézard[12], and Burda et al[13]). They, however, lack one feature we consider important: In some models, each agent (economic body) goes through a stochastic process of increasing and decreasing wealth, quite independently

from other agents. In some others, each agent interacts with either randomly-selected agents or neighboring agents on a fixed lattice. Apparently, such a selection of economic partnership/competition is far from reality, as is evident from complex network studies. This is one reason we propose our model, which features both the stochastic nature of economic activities and underlying network structures that are expected to be close to those of the relevant social activities. We also note that studies of economic phenomena that build upon the underlying complex networks are novel.

2   Model

In considering the mechanisms behind wealth distribution, it is important to include at least two basic natures of the wealth. One is a random change of the value of wealth and the other is the liquidity of it. As a minimal model that contains these features we chose an interactive multiplicative stochastic process (Bouchaud and Mézard[12]), which is defined by the following evolution equation:

$$W_i(t+1) = a_i(t)W_i(t) + \sigma^2 W_i(t) + \sum_{j(\neq i)} J_{ij}(t)W_j(t)$$
$$- \sum_{j(\neq i)} J_{ji}(t)W_i(t), \qquad (1)$$

where $W_i(t)$ is the wealth of the i-th agent at the time t, $a_i(t)$ is a Gaussian random variable with mean m and variance $2\sigma^2$, which describes the spontaneous increase or decrease of the wealth. The rest of the terms in the r.h.s. of Eq. (1) describe transactions on business networks. We note that in Bouchaud and Mézard[12] the process corresponding to Eq. (1) is described by a stochastic differential equation in the Stratonovich sense, which generates the second term proportional to σ^2 when discretized. In addition, although naive discretization brings in a time interval

Δt in Eq.(1), it has been absorbed by the rescaling of the parameters.

The transaction matrix J_{ij} is chosen based on the underlying network structure: If the i-th agent is directly connected to the j-th agent by a link, we take $J_{ij} = J/Z_i$, where Z_i is the number of agents connected to the i-th agent ("neighbors"). Otherwise, $J_{ij} = 0$. The parameter J is a positive number, which represents the ratio of transactions and the wealth. Thus, the third term in the r.h.s. of Eq. (1) describes the incoming wealth, and the fourth term describes the outgoing wealth. Under this selection of transaction matrix, Eq. (1) can be written as follows:

$$W_i(t + 1) = \left\{a_i(t) + \sigma^2 - J\right\} W_i(t) + J\overline{W}_i(t), \quad (2)$$

where $\overline{W}_i(t)$ is the mean wealth of the neighbors of the i-th agent.

We consider one-dimensional and static network of N-agents with periodic boundary conditions (Watts and Strogatz[6]), which are determined by two parameters, i.e., a rewiring probability p and a mean number of links per agent $\overline{Z}$. (We will also use a notation $z \equiv \overline{Z}/N$.) The rewiring probability p is defined by $p \equiv$(the number of rewired links)/ (the total number of links)$\in [0,1]$. At $p = 0$ we have a regular network with $Z_i = \overline{Z}$, in which the i-th agent is connected only with $\{i - \overline{Z}/2, \cdots, i-1, i+1, i + \overline{Z}/2\}$-th agents. For $p \neq 0$, links are rewired to conserve the total number of links. At $p = 1$ we obtain completely random networks.

In the mean field (MF) case, i.e., $z \simeq 1$, the stationary probability density function $p(w)$ is known to be the following in the large N limit (Bouchaud and Mézard[12]),

$$p(w) = \frac{(\mu - 1)^\mu}{\Gamma[\mu]} \frac{\exp\{-(\mu - 1)/w\}}{w^{1+\mu}}, \quad (3)$$

where $w_i(t)$ is the wealth normalized by the mean wealth $\overline{W}(t)$; $w_i(t) = W_i(t)/\overline{W}(t)$. (We will simply call $w_i(t)$ the normalized wealth hereafter.) The exponent is given by $\mu = 1 + J/\sigma^2$. In the large w range Eq. (3) becomes a power law distribution, $p(w) \propto w^{-(1+\mu)}$, and therefore the exponent μ is the Pareto index in economics.

3 Rank distribution

We have carried out numerical simulations for $N = 10^4$, $m = 1$, $J = 0.01$ and $\sigma^2 = 0.01$. Under this parameter set, the Pareto index in the MF case is
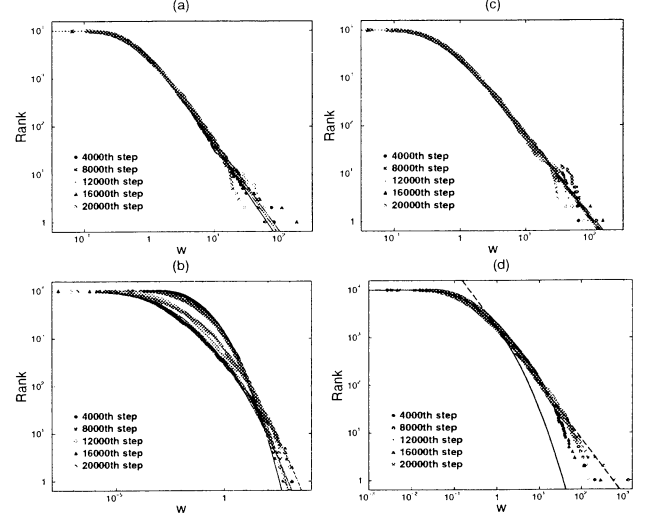

Figure 1: Some of the wealth distributions obtained by numerical simulations. The rank distribution in regular networks is plotted in (a) for $z = 0.1$ and in (b) for $z = 0.001$. The rank distribution in small-world networks is plotted in (c) for $z = 0.001$ and $p = 0.01$ and in (d) for $z = 0.001$ and $p = 0.5$.

$\mu_{\mathrm{mf}} = 2$. Some of the distributions we obtained are shown in Figs. 1 (a)-(d). In these figures the horizontal axis is the logarithm of w and the vertical axis is that of the rank. We have numerically calculated up to 2×10^4 time steps, and drawn distributions at each 4×10^3 step. The rank distributions in the case of regular networks with $z = 0.1$ correspond to Fig. 1 (a). The solid lines show the best fits with the MF result Eq. (3). We see that all of the resulting distributions fit well with the MF result with $\mu \simeq \mu_{\mathrm{mf}}$. Figure 1 (b) is a case of regular networks with $z = 0.001$. We find that the results do not agree with the MF result, but instead fit well with log-normal distributions, which are shown by the solid lines. Note that the best fit log-normal distributions have time-dependent mean value $m_{\mathrm{ln}}(t)$ and time-dependent variance $\sigma_{\mathrm{ln}}^2(t)$. On the other hand, if we rewire links with high probability the MF type distributions reappear. Figure 1 (c) is a case of regular networks with $z = 0.001$ and $p = 0.5$. The solid lines are the fitting by MF type solutions with $\mu = 1.7 \sim 1.8$, which is slightly smaller than μ_{mf}. Hence we observe that the change in wealth distribution occurs at the intermediate value of p. Figure 1 (d) is a case of small-world networks with $z = 0.001$ and $p = 0.05$. We have found that the distributions can not be fitted well by either the log-normal functions or the MF type functions. Interestingly, however, we find that a combination of log-normal and power-law dis-

tributions fits well with the results. One such example at the 2×10^4-th time step is shown with a log-normal function for middle and low wealth ranges (solid line) and a power-law function for the high wealth range (dashed line). Log-normal distributions with a power law tail like this are frequently observed in real world economic phenomena (Aoyama et al[8] and Souma[9]).

4 Wealth clustering

It is natural to question why distribution patterns change. To answer this question, we directly consider the development of the wealth. Examples of numerical simulations are shown in Figs. 2 (a)-(d). In these figures the horizontal axis shows the agents' number and the vertical axis shows time steps. Though we have simulated for $N = 10^4$ agents, wealth developments for 10^3 agents are shown. We have calculated up to 10^4 time steps. If any agent ranks in the top 10%, we mark it with a black dot, and if any agent ranks in the bottom 10% we mark it with a gray dot.

Figure 2 (a) is a case of regular networks with $z = 0.1$. We can see that the black and gray dots are uniformly distributed without notable clustering. Since each agent has 10^3 neighbors, each agent transacts with agents from various ranks, covering a wide range. Figure 2 (b) is a case of regular networks with $z = 0.001$. Contrary to the case of $z = 0.1$, the black and gray dots cluster heavily. This "wealth clustering" phenomenon is supported by the fact that each agent has only 10 neighbors and rich agents transact mainly with rich agents and poor agents transact mainly with poor agents. Hence $W_i(t)$ and $\overline{W}_i(t)$ have strong correlations. Actually, for the case of regular networks with $z = 0.001$, we can obtain $\langle \overline{w}_i(t)w_i(t) \rangle = 0.92 \sim 0.95$. Here $\overline{w}_i(t) \equiv \overline{W}_i(t)/\overline{W}(t)$. On the other hand we can obtain $\langle \overline{w}_i(t)w_i(t) \rangle = 0.02 \sim 0.09$ for the case of regular networks with $z = 0.1$. Figure 2 (c) is a case of small-world networks with $z = 0.001$ and $p = 0.5$. This case is almost similar to Fig. 2 (a). Figure 2 (d) is a case of small-world networks with $z = 0.001$ and $p = 0.05$. Although wealth clustering is observed, the size of each cluster is smaller than the case of regular networks with $z = 0.001$. From these results we find that a decrease in the number of links causes wealth clustering and the rewiring of links disperses it.

If $W_i(t)$ and $\overline{W}_i(t)$ have strong correlations, Eq. (2) becomes a pure multiplicative stochastic process, and the wealth obeys log-normal distributions with mean vt and variance Dt, where $v = \langle \log\{a(t) + \sigma^2\} \rangle$ and $D = \langle [\log\{a(t) + \sigma^2\}]^2 \rangle - \langle \log\{a(t) + \sigma^2\} \rangle^2$. On
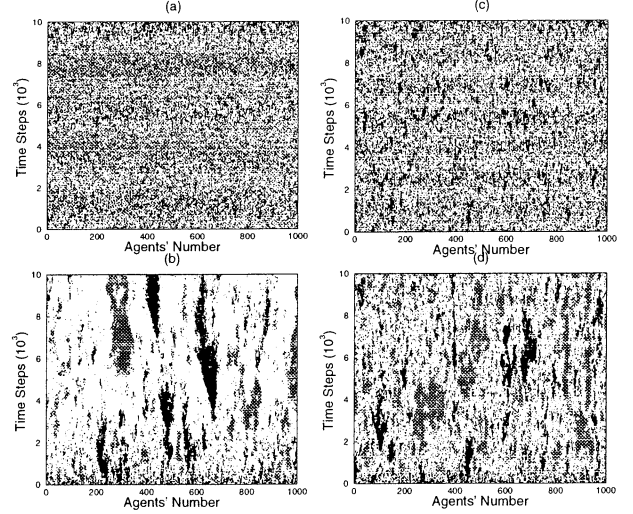

Figure 2: Wealth developments for regular networks with $z = 0.1$ are shown in (a) and with $z = 0.001$ in (b). The case of small-world networks with $z = 0.001$ and $p = 0.05$ is shown in (c), and $p = 0.5$ in (d).

the other hand if $W_i(t)$ and $\overline{W}_i(t)$ have no correlations, Eq. (2) is regarded as a multiplicative stochastic process with additive noise (Kesten[14], Sornette and Cont[15], Takatasu et al[16], and Sornette[17]). It is known that this process induces power law distributions in large $W_i(t)$ ranges, if two conditions are satisfied: One condition is that the stochastic variable $a_i(t)$ and the additive noise $J\overline{W}_i(t)$ must behave independently. The other condition is $\langle \log\{a_i(t)+\sigma^2-J\} \rangle < 0$. Although the first condition does not matter in our model, the parameter ranges of J and σ^2 are restricted by the second condition. If $W_i(t)$ and $\overline{W}_i(t)$ have no correlations in the high wealth ranges and have strong correlations in the middle and low wealth ranges, the wealth will obey log-normal distributions with a power law tail such as those observed in many real world economic phenomena (Souma[9]).

5 Summary

We have carried out a simulation for a wide range of values of p and z, which is summarized by the phase diagram shown in Fig. 3. This diagram has two extreme distributions, the log-normal distributions with $m_{ln}(t)$ and $\sigma_{ln}^2(t)$ and the MF type distributions represented by Eq. (3). These two ranges are interpolated by a decrease in the number of links or the rewiring of the links. In the intermediate region, the wealth distribution may be fitted well with log-normal distri-

p

0   0.001   0.01   0.1   1

Mean Field (Power Law)
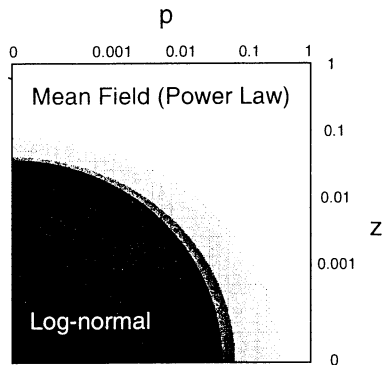
1

0.1

0.01

z

0.001

Log-normal

0

Figure 3: Patterns of wealth distribution in the case of $N = 10^4$.

butions with a power law tail, which are observed in real world economic phenomena (Souma[9]).

Acknowledgements

The authors would like to thank Dr. K. Shimohara (ATR) for his continuous encouragement and warm support.

References

[1] Watts DJ (1999), Small Worlds: The dynamics of networks between order and randomness. Princeton University Press, Princeton, New Jersey.

[2] Newman MEJ (2000), Models of the small world: a review. arXiv:cond-mat/0001118.

[3] Strogatz SH (2001), Exploring complex networks. Nature **410**, 268-276.

[4] Albert R, Barabási AL (2001), Statistical mechanics of complex networks. arXiv:cond-mat/0106096.

[5] Dorogovtsev SN, Mendes JFF (2001), Evolution of random networks. arXiv:cond-mat/0106144.

[6] Watts DJ, Strogatz SH (1998), Collective dynamics of 'small-world' networks. Nature **393**, 440-442.

[7] Albert R, Jeong H, Barabási AL (1999), Diameter of the World Wide Web. Nature 401, 130-131

[8] Aoyama H, Souma W, Nagahara Y, Okazaki HP, Takayasu H, Takayasu M (2000), Pareto's law for income of individuals and debt of bankrupted companies. Fractals **8**, 293-300.

[9] Souma W (2001), Universal structure of the personal income distribution. Fractals **9**, in press.

[10] Ispolatov S, Krapivsky PL, Redner S (1998), Wealth distributions in asset exchange models. Eur. Phys. Jour. B 2, 267-276.

[11] Solomon S (1998), Stochastic Lotka-Volterra systems of competing auto-catalytic agents lead generically to truncated Pareto power wealth distribution, truncated Levy distribution of market returns, clustered volatility, booms and craches. In: Refenes AP, Burgess AN, Moody JE (ed), Proceedings of Computational Finance 97, Kluwer Academic Publications, Dordecht, arXiv:cond-mat/9803367.

[12] Bouchaud JP, Mézard M (2000), Wealth condensation in a simple model of economy. Physica A 282, 536-545.

[13] Burda Z, Johnston D, Jurkiewicz J, Kamiński M, Nowak MA, Papp G, Zahed I (2001), Wealth condensation in pareto macro-economies. arXiv:cond-mat/0101068.

[14] Kesten H (1973), Random difference equations and renewal theory for products of random matrices. Acta Math. 131, 207-248.

[15] Sornette D, Cont R (1997), Convergent multiplicative processes repelled from zero: power laws and truncated power laws. J. Phys. I 7, 431-444.

[16] Takayasu H, Sato AH, Takayasu M (1997), Stable infinite variance fluctuations in randomly amplified Langevin systems. Phys. Rev. Lett. 79, 966-969.

[17] Sornette D (1998), Multiplicative processes and power laws. Phys. Rev. E 57, 4811-4813.

Network of Words

Yoshi Fujiwara*   Yasuhiro Suzuki[†]   Tomohiko Morioka[‡]

* Keihanna Research Center, Communications Research Laboratory
Kyoto 619-0289 Japan

[†] Bio-Informatics, Medical Research Institute, Tokyo Medical and Dental University
Tokyo 113-8501 Japan

[‡] Documentation and Information Center for Chinese Studies
Institute for Research in Humanities, Kyoto University, Kyoto 606-8265

Abstract

Chinese characters form a web of semantic and phonetic links between characters. The links can be observed as structural components in ideographic structure of characters. We examine the structural principles underlying the semantic and phonetic network by studying the statistical properties of the network structure. We report our preliminary results about the distribution of the number of links from components and between characters, and other quantities.

1 Introduction

Human communication is done in various media. Written language is, in its origin, derivative of vocal communications. Any writing system carries information of semantic and phonetic with a system of graphic symbols. How semantic, phonetic and graphic objects are bound and mutually related in a web of characters varies from one writing system to another.

Chinese writing system is one of a few written languages that have been used with so many people and for such a long time in the history of human-kind. In fact, present archeological evidence shows systematic usage of characters at least as early as last quarter of second millennium B.C. in the north-central China. The characters are thought to be mainly used in royal divination in the era of Shang (Yin) dynasty, as is well-known as "oracle bone inscriptions" (characters inscribed on ox scapulas, turtle plastrons). Modern Chinese characters used in wide area of East Asia including China, Vietnam, Korea, and Japan are all direct descendants of the early Chinese scripts in the

period of Shang. (See for example [1][2].) Of course, modern Chinese characters, in graphic appearance, differ from the Shang inscriptions (Fig. 1). Nevertheless the structural principles in the Shang writing system are basically the same as for the descendants of Chinese scripts.

In this paper, we attempt to uncover the underlying structural principles of modern Chinese characters from the viewpoint of network. We regard semantic and phonetic relations between characters as a *network*. Many Chinese characters have complex appearance, but in most cases, are constituted by combination of parts with simpler shapes. Such a combinatorial component is another character or an obsolete symbol which is not used as a stand-alone character today. This graphic structure can be recursive; a component part can be made of others. Since each component carry either semantic and phonetic information, or sometimes both, one can consider a web of relation between characters through their structural parts.

Our goal is to find the structural principles in terms of statistical properties of such semantic and phonetic network for modern Chinese characters. The approach we take here is based on the study of statistical properties of the network structure, partly motivated by the resent study on non-random network in natural science (see [3][4][5] for example). Little is known about such statistical properties in Chinese characters. Indeed, for such a purpose, we need a database of characters as *objects* such that one can access the structural information of each characters and their graphic relation. One of the authors (T.M.) have recently developed such a kind of database, called **UTF-2000** project [6].

In the next section, let us first describe the essential features of the semantic and phonetic web of Chinese

---

*e-mail: yfujiwar@crl.go.jp
[†]e-mail: suzuki@gentzen.mri.tmd.ac.jp
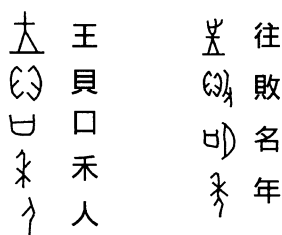[‡]e-mail: tomo@kanji.zinbun.kyoto-u.ac.jp

Figure 1: Examples of Chinese characters. Each column depicts pairs of early scripts ("oracle bone inscriptions") and modern scripts. Meanings are 'king', 'cowrie shell', 'mouth', 'growing grain', 'person', 'go toward', 'defeat', 'call out', and 'year' (from top left to bottom right).

characters. In section 3, we briefly explain about how a character is stored in and extracted from the database of UTF-2000. Section 4 shows our preliminary results about the statistical features of the network of characters. More details and other results are to be presented in the conference. We summarize and discuss several points with our future outlook.

2 Semantic and Phonetic Structures

We first focus on semantic and phonetic threads through web of characters by looking at how the Shang writing system evolved its structural complexity of characters [1].

Chinese spoken language has the persistent principle that each single syllable corresponds to a single word. And the Shang writing system is logographic; i.e. each character stands for a single word (like "$" for "dollar"). So each character carries both information of semantic and phonetic, in contrast with syllabograms and simple ideograms in the world.

Formally one can divide the Chinese characters into those consisting of one graphic element, called *unit characters*, and those constituted of two or more component parts, *compound characters*. Some of the unit characters are pictographic representation of easily depictable things. For obvious examples see the left column of Fig. 1. Many of those graphs were used also for different words in two but closely related ways.

One is the *"rebus"* use of graphs. That is, a graph can stand for a phonetically similar word which is semantically unrelated to it. For example, 太 *wáng* ('king') was used also to write the verb *wǎng* ('go toward')[1]. Words without adequate pictographic representations could be written by this rebus principle.

The other way is that a graph stands for a phonetically dissimilar but semantically related word. For example, 日 *kǒu* ('mouth') was used to stand also for the verb *míng* ('call out'). To put differently, those unit characters are subject to the process of *polyphony*.

These two multiple-usages of characters introduced considerable versatility and flexibility. But the problem is of course significant ambiguity, since the same unit character stands for different words either semantically or phonetically unrelated.

Compound characters solved the problem by adding secondary graphic components, called *determinatives*, to unit characters. In cases of semantic ambiguity, a determinative specifies a meaning. For example, a determinative 屮 representing independently *zhǐ* ('step, stop/go') was added to 太 on its side to specify the word 'go toward' with the graphic shape 往.

On the other hand, phonetic ambiguity was resolved by adding secondary components of phonetic determinatives. For example, a determinative D *míng* was added above the unit character 日 to stand for *míng* ('call out') with the shape 名.

Furthermore, some compounds characters themselves were subject to rebus usage and polyphony, and the eventual ambiguity was resolved by adding more components. Such process brought about hierarchical structures in the appearance of Chinese characters.

It should be emphasized that the above description is not merely one of retrospective interpretations of ancient characters. Of course, since a lot of component parts have been subject to considerable modification and unification in their graphic shapes, it would be a formidable task to reconstruct the evolutionary history of Chinese characters, unless we have sufficient etymological evidence and materials. However, the structural principles we have described above were eventually kept common to the descendants succeeding to the early Shang script.

In fact, the present system of characters is affected by lexicographic interpretations over centuries in the following history, most notably by orthographic lexicon, *Shuō wén jiě zì* 説文解字 in 100 A.D, a lexicon of about 9500 characters. The lexicon was not merely orthographic standardization but also provided explicit semantic and phonetic relations of characters. First, the distinction between unit (*wén*) and compound (*zì*) characters was made referring to older literature at the time. Second, 540 graphic elements were selected, one of which appears in the graphic structure of every character and was considered to indicate the semantic information of the character. (For characters with more than one of such basic component parts, only one

---

[1]In what follows, *Italic* letters are (modern) Chinese pronunciations. 'Single quotations' are meanings in English.
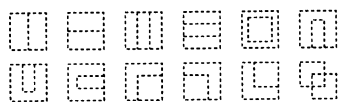
Figure 2: Deposition of component parts in the ideographic structure of a character, which is decomposed recursively.
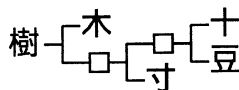

Figure 3: Ideographic structure for the word "tree". Terminal nodes of the tree-like structure are used as structural components of this character.
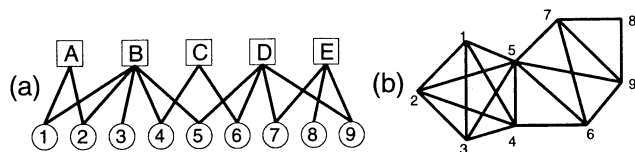

Figure 4: (a) Bipartite graph of components (alphabetically indexed row) and characters (numerically indexed row). (b) Reduced graph from (a) consisting only of characters.

of them were singled out in the lexicon.) Thirdly, it classified the structural principles of characters. This lexicon, and others later including *Kāngxī zìdiǎn* 康熙字典 in 1716 (about 47000 characters with 214 semantically basic elements), affected great influence on the usage, interpretation, orthography, literacy education in modern systems of characters until today.

3 UTF-2000 Project

UTF-2000 is a project to study and develop character processing based on a database of characters. One of the aims in the project is to represent characters in the world with more flexibility and in a way unbounded to standard coded character sets such as unicode. Each character is indicated by a set of properties including phonetic and ideographic information, reference to other characters with slightly modified graphic shapes in alternative usage, and bibliographic reference. Code points in coded character sets are also included so that the database can easily access to them.

For Chinese characters, ideographic structure is given as a property showing decomposition of a character into a few component parts. Deposition of components is listed in Fig. 2. The decomposition can be recursive; a component part can be a combination of other parts. One can extract such hierarchical ideographic structures for each character from UTF-2000 implemented in xemacs and lisp environement at present. Thus a hierachical ideographic structure looks like Fig. 3.

4 Network Structure

We used the database of UTF-2000 which contains totally about 20000 Chinese characters (used in China,

Japan and Korea today) at the time of writing, and excluded non-characters. In our first analysis, which is reported here, we extracted only Chinese characters used in Japan. Technically, the set of characters is the JIS (Japanese Industrial Standard) X 0208 (1983), which contains about 6500 Chinese characters. This selection is done simply for the purpose of selecting relatively frequently used characters in a random way[2].

Each character is decomposed into component parts in a hierarchical fashion as shown in Fig. 3, and then gather all the obtained components together, which appeared as leaf nodes in the tree-like decomposition. The resulting two sets, one for components and the other for characters, form a bipartite graph as depicted in Fig. 4(a). This may be viewed as an extension of the orthographic lexicon *Shuō wén jiě zì* 説文解字. Then one can reduce the bipartite graph to that only for the nodes of characters by connecting all the characters that have a common component Fig. 4(b). However, it should be noted that connectivity information present in the bipartite graph will be partly lost in the reduced one.

The numbers of components and characters are respectively 659 and 5565. First let us take a look at the statistical properties of the bipartite graph. Fig. 5 shows the distribution of the number of links from a component to characters in the form of cumulative frequency. There is obviously a range for which the distribution is of exponential form, or Poisson-like, corresponding to the number of links between 30 and 200. Left to this intermediate range is present a deviation from the exponential behavior due to the fact that a fraction of the components has only a few links to characters in the representation of bipartite graph. In the tail, on the other hand, one can see a dozen of points displaying more frequent occurance of compo-

---

[2]It should be pointed out that the usage of Chinese characters in Japan experienced unfortunate educational policy around 1940. A fraction of Chinese charaters were subject to modification in their graphic shapes without reason but for simplicity in writing. See [7]. The range of our analysis contains traditional graphic shapes in addition to the corresponding modified ones, so we need careful examination in future.
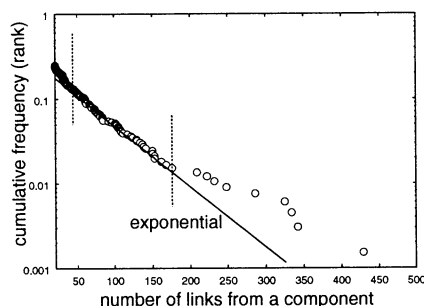
Figure 5: The cumulative frequency of the number of links from a "component" node in the bipartite graph. The vertical value can be regarded as a rank (divided by the total number of components).

Figure 6: The frequency distribution of the number of links between characters in the reduced graph.

nents than what is expected from the Poisson-like behavior. Such a component means a large cluster of characters with that structural component.

Next from the bipartite graph we constructed a graph composing of only characters. The number of characters is $N = 5565$. The average of the number of links from a characters to others was calculated to be $k = 350$. The distribution of the links between the characters is shown in Fig. 6. Corresponding to the large clusters shown above is a bump in the distribution around the number 700 of links.

For the reduced graph we also calculated two quantities. One is the average length L of the shortest path between a randomly chosen pair of characters. Our analysis shows that $L = 2.0$. The other quantity is cliqueness C. Suppose that a character has "friends" to which one can jump by a single link from it. All those friends are not linked to each other in general. C is the fraction of links between the friends which actually exist. The result is $C = 0.64$. If the graph is a random network, typical values for these two quantities are given by $L \sim \log N / \log k \sim 1.5$ and $C \sim k/N \sim 0.06$. This result implies that the world of characters is "small", but the connectivity involving more than two characters is much higher than that for a random graph. The latter is not very surprising since the above observation of distributions shows the existence of large clusters which are not isolated but linked with other semantic and phonetic component parts.

5 Summary

In summary, we have reported our preliminary results on statistical properties of a web of semantic and phonetic links between Chinese characters. The links can be observed as structural components in

ideographic structure of characters. Components and characters compose a bipartite graph, for which the distribution of the number of links was examined. From its reduced graph, we constructed a network between characters, and analyzed the distribution of the number of links and the average shortest path and cliqueness. Further results and how they are related to studies of characters and practical applications are to be presented in the conference.

References

[1] W. G. Boltz, "Early Chinese Writing" in ed. P.T. Daniels and W. Bright *The World's Writing Systems*, Oxford University, 1996.

[2] T. Atsuji, *History of Chinese characters (Kanji no Rekishi)*, Taishūkan, 1989 (in Japanese).

[3] D. J. Watts, *Small Worlds*, Princeton University, 1999.

[4] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications", Phys. Rev. **E 64**, (2001) 026118.

[5] ed. S. Bornholdt and K. Klemm, *International Conference on Dynamical Networks in Complex Systems*, (July 2001, Kiel, Germany).

[6] T. Morioka, "A Short Introduction to UTF-2000 Project", in the First TEI Character Set Issues Working Group Meeting, (October 2001, University of California, Berkeley, USA).

[7] T. Takashima, "Chinese Characters and Japanese Language", Japanese Book News, **23** (1998) pp.1-3, **24** (1998) pp.4-5, **25** (1999) pp.3-5, (in English) available from http://www.jpf.go.jp/e/media/publist.html

Memory Dynamics of Associative Conditioning in Terrestrial Mollusk

Masafumi YANO, Yoshinari MAKINO, Haruki MIURA and Hisanori Makinae
Research Institute of Elevtrical Communication, Tohoku University, Sendai, 9808577, Japan

Abstract

It is one of the goals of science to realize how animals process flexibly semantic information in the real world in real time. All animals seem to use available input for information retrieval by integrating different information and using flexible inference function. To understand the function of brain as information processing organ, it is necessary to investigate the whole brain simultaneously. Because the brain works as a whole, that is composed of various modules. However, higher animals including human beings are too complex to analyze the information processing mechanism. Because of their relative simplicity, we pick up the terrestrial slug, which is well known to have ability of some logical operations similar to those in vertebrates. They show first- and second-order conditioning, blocking or extinction of memory. Even in an unpredictably changing environment, it needs to process the information including ambiguity and uncertainty, which might be an ill-posed problem. They, however, solve the ill-posed problems flexibly, while the processes can not be explained by inductive or deductive logic. Here we propose an abductive process to solve the ill-posed problem and a mechanism to acquire new knowledge.

Key words; abduction, ill-posed problem, memory, associative learning

1. Introduction

When in an unpredictably changing environment, creatures need to process the information including ambiguity and uncertainty. It might be an ill-posed problem. The differences in information processing between the present highly advanced information systems and creatures might be caused from whether the two systems have the methodology for the ill-posed problems or not. In order to process or avoid the ill-posed problem, it is necessary to add some constraints until to make it well-posed problem. In the real world, however, it is impossible to define such an objective function and to set constraints in advance, because the environment changes unpredictably. It indicates that the traditional method based on the causality can not apply to these problems. That is, only inductive and deductive methods can not explain flexible information processing. It also means that the brain should autonomously create necessary information depending on the unpredictably changing environment.

It is great ability of creatures that they evaluate the input information including ambiguity or uncertainty. Even in an unpredictably changing environment, they need to acquire necessary information to make predictions and motion plans to survive in a flexible. Since creatures seem to use available input for information retrieval by integrating different information and using flexible inference function, it is great important to clarify this novel information processing mechanism.

Brain is composed of various interacting function modules. But we don't know how to integrate and how to control these various functions. To understand the function of brain as information processing organ, it is necessary to investigate the whole brain simultaneously, because the brain works as a whole. However, higher animals including human beings are too complex to analyze the information processing mechanism.

The highly sophisticated function in animal memory is the one of the most important key to flexible information processing. Especially, associative memory and its efficient use for the self-operation of cognitive process are essential. Because of their relative simplicity, we take the case of the terrestrial slug, which is well known to have ability of some logical operations similar to those in vertebrates. They show first- and second-order

conditioning, blocking or extinction of memory [1-4].

So we firstly discuss 'the information principle' of the terrestrial slug. Then we propose an abductive process to solve the ill-posed problem and a mechanism to acquire new knowledge, which is a new paradigm suitable for the complex surroundings.

2. Characteristics of associative learning of terrestrial mollusk

Associative learning is widely observed in various animals, in which they learn the relationship among stimuli. It has been clearly demonstrated by Gelperin and his colleague that the terrestrial slug shows some logical operations similar to those in vertebrates, such as first- and second-order conditioning, blocking or extinction of memory [1-2]. These associations in various higher-order conditioning, especially the relationships between stimuli, have been extensively studied. Cooling-induced retrograde amnesia has been used to analyze these relationships [3-4], because the induction of retrograde amnesia is closely related to the recollecting process. The state of memory is crucially depended on the sequence of the experience as shown in Table 1. It revealed that the behaviors to the same stimulus are the same, but the memory states are different depending on the sequence of the experiences. For example, in the case of second order conditionings, the memory states of the sequential conditioning are different from those of the simultaneous conditioning. That is, the relationship between the memory of the two stimuli in the former case is unidirectional, but the latter case is bi-directional. This relationship seems to be followed the temporal sequence, which can be explained by Hebb rule. The difference of the memory states between the sequential second order conditioning and the sensory preconditioning becomes conspicuous. When the sequence of the conditionings is reversed, the relationship between the stimuli is the same but the meanings of the stimuli are different. The two stimuli are associated according to the temporal sequence in a similar way. The meaning of CS2 in the case of the sensory preconditioning is still appetitive while the CS2

induces an aversive behavior as long as the two stimuli are associated. It is great important that the formation of the relationship and the acquisition of the meaning are achieved separately. Another characteristic of the memory formation can be seen in the blocking. Comparing the second order conditioning of simultaneous type and the blocking, the only difference in the sequence of conditionings between the two is whether the unconditioned stimulus is added or not. This difference may be caused from the mechanism of figure-back ground separation.

Conditioning Procedures		Results CS1 CS2		CS1+F CS1 CS2		CS2+F CS1 CS2	
first-order conditioning	CS1 + Q CS2 + Q	—	—	+	—	—	+
2 nd order conditioning (sequential)	CS1 + Q CS2 + CS1	—	—	+	—	+	+
2 nd order conditioning (simultaneous)	CS1 + Q (CS2 /CS1)	—	—	+	+	+	+
Sensory Pre-Conditioning	CS2 + CS1 CS1 + Q	—	—	+	+	+	+
blocking	CS1 + Q (CS2 + CS1) + Q	—	+				

From YAMADA et al; Gelperin et al.

Table 1 Memory states of classical conditionings

Superior Tentacle   Cerebral ganglion

Inferior Tentacle   lip
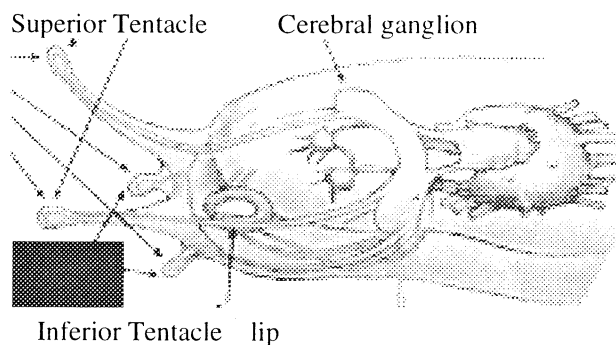
Fig.1 Illustration of a brain of terrestrial slug

The recollection of memory is induced by the input through the inferior tentacle. The structure of the brain of terrestrial slug is briefly illustrated in Fig.1. Usually, an aversive-conditioned odor induces aversive behavior to the odor source, and an appetitive-conditioned odor induces eating behavior. However, when an aversive-conditioned odor which is originally appetitive, is presented to the slug at far

distance, the slug approaches up to the odor and then turns back [5]. This result strongly suggests that the inputs through two kinds of tentacles play quite different roles. At first stage, the input through the superior tentacle is processed to be appetitive and the slug approaches close to the odor. Then the input through the inferior tentacle is recognized as the aversive one and turns back.

3. Mechanism of memory formation and recognition

The mechanism of memory formation has been proposed by Rescola-Wagner, which is well known as Rescola-Wagner rule. It is a simple rule for the formation of the relationship between the stimulus and unconditioned stimulus described the following equation,

$$\Delta Wi = \beta (\lambda - \Sigma Wi * CSi),$$

where ΔWi is the change of the associative strength of conditioned stimulus of CS1, _is the value of the inputted unconditioned stimulus and Wi is the associative strength of CSi. However, this rule can only explain the first order conditioning and the blocking. As pointed by Mazer, there is essential difference between the first order conditioning and the higher order conditioning. So, quite different rule is required to explain the higher order conditionings.

In order to study the mechanism of memory formation, it is necessary to clarify the information flows in the brain of the terrestrial slug. We have investigated the information flow in the slug brain by using an optical recording technique as shown in

0.25%   -ΔF/F   -0.25%
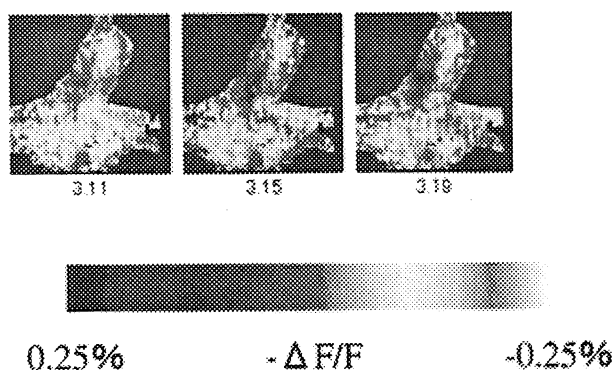
Fig.2 An example of optical recording of the activities of the whole brain of slug.

Fig.2 (6). The input from the superior tentacle nerve is rapidly propagated into the cerebral ganglion (CG) and after short delay inputted into the terminal mass of the procerebral lobe. In the procerebral lobe oscillation of a local field potential was observed. Similarly, the input through the inferior tentacle is fast propagated into the cerebral ganglion, in which specific types of neurotransmitter or neuromodulators. These biogenic amines mediate the behavior the slug, indicating that evaluation of the input information and making predictions and motion plan. According to these behavioral, physiological and anatomical studies, the evaluation of the input through superior tentacle is achieved independently from its experience and memory. In this sense, this evaluation of the information might be innately achieved. Since the slug can not predict what kind of odors it will meet, we call this evaluation process " first abduction process". Odor presentation to the inferior tentacle induced eating or turning back behavior depending on the learned preference. This evaluation might be memory mediated. We call this memory-based prediction as "second abduction process". The evaluation of the stimuli is achieved in a hierarchical way. That is, an innate evaluation of olfaction through the superior tentacle is over-evaluated by the memory–mediated evaluation through the inferior tentacle. And it is necessary that the inputs through the two pathways are consistent, which means that the temporal identification of the stimuli is essential. The taste evaluation is superior to the evaluation of the olfactory system. The change of the evaluation of the stimuli is occurred only for the present stimulus associated with the unconditioned stimulus. While the relationship between the associated stimuli remains unchanged. This information processing and the memory formation are summarized in Fig.3. These processes are achieved dynamically utilizing synchronization phenomena. The oscillation of the local field potential (LFP) in the procerebral lobe, which is considered as memory storage site, is changed depending on the learned preference for odors [5]. The aversive-conditioned odors reduced the frequencies of the LFP, while the appetitive odor increased them. Neutral or unpaired odors did not affect. The slug can acquire new memory only after the eating behaviors according to the predictions.

Further mechanism and modeling of the abduction processes will be discussed utilizing the synchronization phenomena.
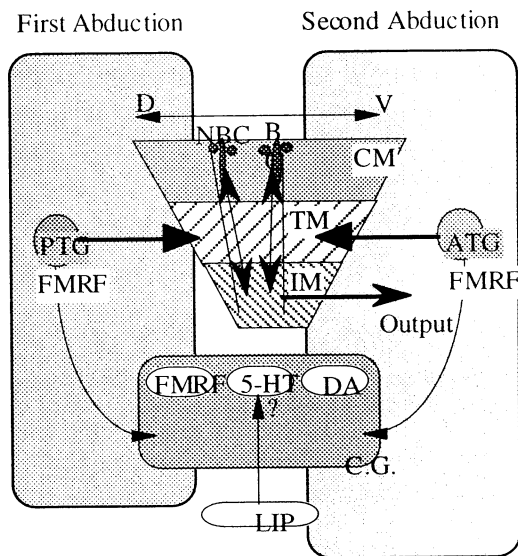

Fig.3 Scheme of information flow of the slug brain; PTG; superior tentacle ganglion, ATG; inferior tentacle ganglion, CM; cell mass of procerebral lobe, TM; terminal mass of procerebral lobe, IM; internal mass of procerebral lobe, NBC; non-bursting cell, BC; bursting cell, D; dorsal site, V; ventral site, DA; dopamine, 5-HT; serotonine, CG; cerebral ganglion

References

[1] Gelperin A, "Rapid food-aversion learning by a terrestrial mollusk", Science vol.189, 1975, pp. 567-570.

[2] Shaley CL., Rudy JW. & Gelperin A., "An analysis of associative learning in a terrestrial mollusk. I. Higher order conditioning, blocking and a transient US pre-exposure effect." J. Comp. Physiol. Vol 144, 1981, pp. 1-8.

[3] Yamada A., Sekiguchi, T., H. Suzuki & M. Mizukami, "Behavioral analysis of internal memory states using cooling-induced retrograde AMNESIA in LIMAX FLAVUS", J. Neurosci., vol. 12, 1992, pp. 729-735.

[4] Sekiguchi, T., H. Suzuki, A. Yamada & M. Mizukami, Neurosci. Res., "Cooling-induced retrograde amnesia reflexes Pavlonian conditioning associations in Limax flavus",

Neurosci. Res. vol. 18, 1994, pp. 267-275.

[5] Kimura T., S. Toda, T. Sekiguchi & Y. Kirino. "Behavioral modulation induced by food odor aversive conditioning and its influence on the olfactory responses of an oscillatory brain network in the slug Limax marginatus.", Learning & Memory, vol 4, 1998, pp. 365-375.

[6] Makinae H., Y. Makino & M. Yano, "Optical recording of the network response induced local electric stimuli to tentacle nerve in the slug", Biophys. vol. 40, 2000, pp.s184.

Generalization and Spatio-temporal Perception in Dynamical Recognizers

Takashi Ikegami
Department of General Systems Sciences,
The Graduate School of Arts and Sciences,
University of Tokyo, 3-8-1 Komaba, Meguro-ku,
Tokyo 153-8902 Japan
(ikeg@sacral.c.u-tokyo.ac.jp)

Dynamical Recognizer (DR) is a second-order recurrent network, which is also known as a recursive iterative functional system. This network consists of two different kinds of nodes; those being directly coupled with the external sensory inputs/outputs and those only being circulated internally. The latter nodes are called "context nodes".

As is first pointed out by J.Pollack [1], DR can recognize more complex language than regular and context free ones. Interestingly, when DR learns a finite state machine, the section pattern of the context node space can be mapped onto the finite state machine. On the other hand, some DR has distributed more complex section patterns. In this study, we mainly focus on those complex patterns to interpret the complexity in terms of perception and memory structures.

We have been studying coupled dynamical recognizers to see its mutual learning behaviors [2, 3, 4, 5]. There we have discussed non-convergent learning behaviors are relevant to discourse or cooperative behaviors. Here, we study a single learning dynamics as an issue of a dynamical system.

Any definition of learning cannot be complete and it always faces with Kripke's paradox [7]. An issue of learning as a dynamical system is how to internalize the given sensory flow into the dynamics of the context flow. Namely, a sensory image $(n + 1)$ which follows a sensory image (n) is expressed as a causal sequence that determines an organizing mapping of context nodes.

We tend to think that conditional learnings like Pavlov's as a primitive and prototypical learning behavior. However, as a dynamical systems, given input/output relationship can't always be mapped stably onto internal dynamics. Sometimes, a dog may generate very different learning images than we expect. For example, a dog may think that there exists some relationship between his prediction of food and color of the shirt who rings a bell. We thus have to care more about diversity of extensions of learning behavior than whether a subject accidentally learns it or not. From dynamical systems points of views, we make correspondence between diversity of extensions and that of hierarchically ordered attractors in this talk.

The easiest learning problem is to learn a given sensory flow by constructing a finite state machine (FSA). When DR succeeds in having a FSA representation, it is said that data flow is successfully learned by the DR. However, a FSA is a special learning image in the sense that it doesn't generate any sense of time direction. FSA's behavior is only dependent on its current sensory flow and the entire behavior of the DR falls into periodic motion pattern. We name this behavior **space perception**. On the other hand, a general DR has **time perception** in the sense that its current state is sensitive to its current and past sensory flow (see Fig.1).

We show that an experiment of DR learning dynamics has two kinds of attractors; ones with time perception and ones with space perception. However, basin boundaries of those two attractors become highly complex so that we can't even decide the final attractor whether DR will organize space or time perception. It is also shown that some attractors are highly sensitive to noises on the weight connections.

We thus have to deal with significant unstable behavior in modeling cognitive functions. At the same time, we can show that such instability can function positively in the cases of games [2, 3, 4] discourse [5] and turn-taking [6] dynamics. Details of this talk will be reported elsewhere [8].
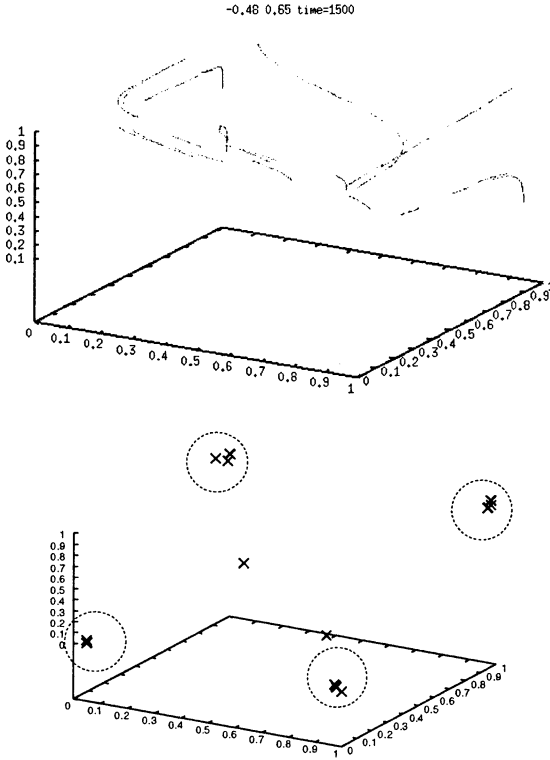
Figure 1: Two kinds of attractors are found in a parity checker learning experiments. Return maps of the attractors are depicted. Two axis denote the context nodes and one the output node. In the below figure, points with circle can be taken as nodes of corresponding finite state machine. We call the above figure as a dynamical recognizer having time perception and the below as that with space perception.

References

[1] J.B. Pollack, The induction of dynamical recognizers. *Machine Learning*, 7:227–252, 1991.

[2] M.Taiji, and T.Ikegami, Dynamics of internal models in game players Physica D, 134, 253–266. 1999.

[3] T.Ikegami and M.Taiji Structures of Possible Worlds in a Game of Players with Internal Models (ps.gzip file) Acta Polytechnica Scandinavica Ma. 91(1998)pp.283-292.

[4] T.Ikegami, and M. Taiji, Imitation and Cooperation in Coupled Dynamical Recognizers (ps.gzip file) 545–554. Advances in Artificial Life eds. Floreano,D. et al., Springer-Verlag, 1999).

[5] I.Igari and T.Ikegami, Coevolution of Mind and Language. presented at the int'l conf. on Evolution of Language (Paris, April, 2000).

[6] H.Iizuka and T.Ikegami, Simulating Turn-taking dynamics, (to be submitted).

[7] S.Kripke, "Wittgenstein on Rules and Private Language " (Harvard Univ. Press, 1984).

[8] T.Ikegami, Generalization and Memory Structures of Dynamical Recognizers. (to be submitted).

Fractals in Neurodynamics
-from Hybrid Dynamical Systems point of view-

K. Gohara
Department of Applied Physics
Hokkaido University
Sapporo, 060-8628, Japan
gohara@nd-ap.eng.hokudai.ac.jp

Abstract

Recently, "Hybrid Dynamical Systems" have attracted considerable attention in automatic control domain. In this paper a theory for recurrent neural networks are presented from hybrid dynamical systems point of view. The hybrid dynamical system is defined by a continuous dynamical system discretely switched by external temporal inputs. The theory suggests that the dynamics of continuous-time recurrent neural networks, which is stochastically excited by external temporal inputs, is generally characterized by a set of continuous trajectories with a fractal-like structure in hype-cylindrical phase space.

Key words: recurrent neural networks, neurodynamics, fractals, hybrid dynamical systems

1 Introduction

A *Hybrid Dynamical System* (HDS) [1] is a dynamical system that involves an interaction of discrete and continuous dynamics, e.g., a continuous-time device controlled by a *sliding mode* controller [2]. Control strategies introduced for HDS have been applied to various design problems for realizing desired behavior in power plants [3], chemical plants [4], etc.

The dynamical evolution of HDS can be described by a differential equation involving a number of vector fields that are switched one after another [5]. It has been shown that HDS displays very complex behaviors such as chaotic behavior [6]. Branicky has presented a numerical experiment for HDS described by a simple linear equation and shown that the state of the system moves around on the Sierpinski gasket, a very well-known fractal set. This result suggests that the fractals may universally appear in some classes of HDS [7]. In this paper a theory for continuous recurrent neural networks with temporal inputs are presented from hybrid dynamical systems point of view.

We focus on dissipative, continuos, and non-autonomous recurrent neural networks defined by the following ordinary differential equations:

$$\dot{x} = f(x, t), \tag{1}$$

$$x \in R^N,$$

where x, t and f are state, time, and vector field, respectively. Equation (1) implies that the vector field depend on time. In general, this suggests that a network is influenced by other systems. To emphasize that the vector fields depend on time throughout the input $I(t)$, we rewrite Eq. (1) as follows:

$$\dot{x} = f(x, I(t)), \tag{2}$$

$$x, I \in R^N.$$

2 Dynamics with Periodic Inputs

We will begin by considering a dynamics with a periodic input:

$$I(t) = I(t + T),$$

where T is the period of the input. The vector field f is also periodic with the same period T:

$$f(t) = f(t + T).$$

Introducing the angular variable $\theta = \frac{2\pi}{T}t \mod 2\pi$ and new state variable $y = (x, \theta)$, we can transform the non-autonomous system expressed by Eq. (2) into the following autonomous system:

$$\dot{y} = f_I(y), \tag{3}$$

$$y \in R^N \times S^1.$$

The vector field f_I is defined on a manifold $\mathcal{M} : R^N \times S^1$ that is a hyper-cylindrical space. In other words,

Eq. (3) expresses a continuous dynamical system D_c defined by the manifold $\mathcal{M}$ and the vector field f_I:

$$D_c = (\mathcal{M}, f_I). \qquad (4)$$

In the hyper-cylindrical space $\mathcal{M}$, we can define the Poincaré section:

$$\Sigma = \left\{ (x, \theta) \in R^N \times S^1 | \theta = 2\pi \right\},$$

where a trajectory starting from an initial state at $\theta = 0$ returns at $\theta = 2\pi$. On the section Σ, a mapping can be defined which transforms a state x_τ to another state $x_{\tau+1}$ after interval T:

$$x_{\tau+1} = g_I(x_\tau), \qquad (5)$$

$$x_\tau \in R^N,$$

where g_I is an iterated function. In other words, Eq. (5) expresses a discrete dynamical system D_d defined by the manifold Σ and the iterated function g_I:

$$D_d = (\Sigma, g_I). \qquad (6)$$

We can summarize the dynamics with a periodic input as follows. The periodic input I defines two dynamical systems, a continuous one D_c and a discrete one D_d defined by Eqs. (4) and (6), respectively. In order to emphasize the relation among I, D_c and D_d, we use the following schematic expression:

$$I \to D_c \to D_d. \qquad (7)$$

3 Dynamics with Switching Inputs

3.1 A Set of Inputs

In this section, we consider a dynamics in which plural input patterns are stochastically fed into the system one after the other. Let us suppose that *each input is one period of a periodic function*. For example, we can define the periodic function by the following Fourier series:

$$I(t) = \frac{a_0}{2} + \sum_{m=1}^{M} (a_m \cos \frac{2\pi m}{T} t + b_m \sin \frac{2\pi m}{T} t), \quad (8)$$

where a_0, a_m, $b_m \in R^N$ are vectors for Fourier coefficients, and T is the period. The set of these parameters defines the *input space*:

$$\mathcal{I} = \left\{ a_0, \{a_m, b_m\}_{m=1}^{M}, T \right\},$$

$$\mathcal{I} : R^{N+2 \times N \times M + 1}.$$

Within this space, an arbitrary point represents an external temporal input. We consider the input as a set $\{I_l\}_{l=1}^{L}$ of time functions I_l sampled on the parameterized space $\mathcal{I}$. In the following sections, we abbreviate subscripts and express individual sets as $\{\cdot\}$ for simplicity.

3.2 Two Sets of Dynamical Systems

Much as in the case of periodic input, we can define two sets of dynamical systems corresponding to the set $\{I_l\}$. One is the set of continuous dynamical systems:

$$\{D_{cl}\} = (\mathcal{M}, \{f_l\}), \qquad (9)$$

where $\{f_l\}$ is the set of vector fields on the hyper-cylindrical space $\mathcal{M}$. The other is the set of discrete ones:

$$\{D_{dl}\} = (\Sigma, \{g_l\}), \qquad (10)$$

where $\{g_l\}$ is the set of iterated functions on the global Poincaré section Σ. We also use the following schematic expression, which is similar to expression (7):

$$\{I_l\} \to \{D_{cl}\} \to \{D_{dl}\}. \qquad (11)$$

3.3 Excited Attractor

In this paper, we are considering a continuous dynamical system that is dissipative and has an attractor for a periodic input. When an input pattern is fed into the system repeatedly, i.e, in the case of periodic input, a trajectory converges to an attractor. But how do we describe the dynamics when the inputs are switched stochastically? Even for an input with finite interval, we can assume an attractor corresponding to a periodic input with infinite interval. We call this an *excited attractor* in order to emphasize that the attractor is excited by the external input. Although a trajectory tends to converge to a corresponding excited attractor, the trajectory cannot reach the excited attractor due to the finite time interval. If the next input is the same as the previous one, the trajectory again goes toward the same excited attractor. If the next input is different from the previous one, the trajectory changes its direction and goes toward an excited attractor distinct from the previous one. Continuing this process, the trajectory takes a transient path to the excited attractors. Intuitively, the trajectory will be spread out around excited attractors in the hyper-cylindrical phase space $\mathcal{M}$. How, then, do we characterize the properties of the transient trajectory?

4 Fractal Transition

4.1 Iterated Function System

In the following two sections, we focus on the set $\{g_l\}$ of iterated functions on the global Poincaré section Σ.

4.1.1 Hutchinson's Theory

Hutchinson [8] has proved that a set $\{h_l\}$ of iterated functions, which are not limited on the Poincaé section, defines a unique and invariant set C that satisfies the following equation:

$$C = \bigcup_{l=1}^{L} h_l(C), \qquad (12)$$

where

$$\bigcup_{l=1}^{L} h_l(C) = h_1(C) \cup h_2(C) \cup \cdots \cup h_L(C),$$

and

$$h_l(C) = \bigcup_{x \in C} h_l(x).$$

Such an invariant set C is often a fractal or sometimes used as a mathematical definition of various fractals.

A sufficient condition for satisfying Eq. (12) is the *contraction* property of h_l for all $l = 1, 2, ..., L$. The contraction for h_l is definitely defined by the Lipschitz constant $Lip(h_l)$:

$$Lip(h_l) = \sup_{x_i \neq x_j} \frac{d(h_l(x_i), h_l(x_j))}{d(x_i, x_j)}, \qquad (13)$$

where d is a distance on a metric space. When

$$Lip(h_l) < 1,$$

the map $h_l : x \to x$ is called the *contraction* or the *contraction map*.

4.1.2 Iterated Function System with Probabilities

Barnsley has named a set $\{h_l\}$ as the IFS (*Iterated Function System*) [9]. He introduced the *IFS with probabilities* as follows:

$$(\{h_l\}, \{p_l\}), \qquad (14)$$

where $\{p_l\}$ is a set of probabilities corresponding to $\{h_l\}$.

Based on the IFS with probabilities, he proposed an alternative method for constructing the invariant set C that satisfies Eq. (12). The iterated functions h_l are switched with probabilities p_l for $l = 1, 2, ..., L$ as follows. Choose an initial point and then choose recursively and independently $x_{\tau+1} \in \{h_1(x_\tau), h_2(x_\tau), ..., h_L(x_\tau)\}$ for $\tau = 0, 1, 2, ..., \infty$, where the probability of the event $x_{\tau+1} = h_l(x_\tau)$ is p_l. Thus a sequence constructs a set $\{x_n\}_{n=0}^{\infty}$. Using Hutchinson's theory, Barnsley has shown that the set $\{x_n\}_{n=0}^{\infty}$ constructed by random sequence, and here assumed to have equal probability, "converges to" the set C defined by Eq. (12) when all iterated functions are the contractions. The set $\{x_n\}_{n=0}^{\infty}$ is thus an approximation of C.

4.2 Vector Field System

We are now ready to consider the trajectory of continuous dissipative dynamical systems excited by the temporal inputs. When the inputs I_l are stochastically fed into the system one after another, the vector fields f_l and the iterated functions g_l are also stochastically switched as explained in Sec. 3. To emphasize the relation among the set $\{I_l\}$, $\{f_l\}$ and $\{g_l\}$, we use the following schematic expression instead of expression (11):

$$\{I_l\} \to \{f_l\} \to \{g_l\}. \qquad (15)$$

We call the set $\{f_l\}$ the *Vector Field System* (VFS), which is similar to the *Iterated Function System* (IFS) for the set $\{g_l\}$. The discrete dynamics on the Poincaré section Σ correspond to the random iteration algorithm using the IFS with probabilities. That is, when all iterated functions g_l are the contractions, the state x_τ on the Poincaré section approximately changes on the invariant set C defined by Eq. (12) after sufficient random iterations. The property of the set C having the fractal structure affects the trajectory in the hyper-cylindrical phase space $\mathcal{M}$.

The trajectory set $\Gamma(C)$ corresponding to the input set $\{I_l\}$ is obtained by the union of the trajectory set $\gamma_l(C)$ for each input I_l:

$$\Gamma(C) = \bigcup_{l=1}^{L} \gamma_l(C),$$
$$= \gamma_1(C) \cup \gamma_2(C) \cup \cdots \cup \gamma_L(C). \qquad (16)$$

We can conclude that the dissipative dynamical systems excited by plural temporal inputs are characterized by the trajectory set $\Gamma(C)$ starting from the initial set C defined by Eq. (12). All of the trajectories are considered to represent the transition between

the excited attractors. We call this the *fractal transition between the excited attractors*. At this point, we should emphasize that the contraction property of iterated functions defined on the Poincaré section is a *sufficient*, but not *necessary*, condition for the fractal transition.

5 Discussion

In this paper a theoretical framework is presented for a recurrent neural networks stochastically excited by external temporal inputs. In this section we discuss some related works. More general theory has been presented in order to model complex systems that interact strongly with other systems. It has been revealed that these dynamics are generally characterized by fractals when the iterated functions are not the contractions [10]. The hierarchical structure of fractals and the noise effect of inputs have been investigated [11]. The fractals generated by switching vector fields have been observed in different domains such as a forced damped oscillator [12], an electronic circuit [13], artificial neural networks [14], and human behavior [15]. Closure of the fractals in both linear [16] and non-linear systems [17] has been also presented. These works show that fractals are indispensable for understanding of dynamics observed in the recurrent neural networks.

References

[1] A.S. Matvee and A.V. Savkin, Qualitative theory of hybrid dynamical systems, Birkhauser, Boston, 2000.

[2] V.I. Utikin, Sliding Modes in Control Optimaization, Springer-Verlag, Berlin, 1992.

[3] H.E. Garcia, A.Ray, and R.M. Edwards, "A reconfigurable hybrid systems and its application to power plant control," *IEEE Trans. Control Systems Technology*, Vol.3, pp.157-170, 1982.

[4] B. Lennartson, B. Egardt, and M. Tittus, "Hybrid systems in process control", *Proc. 33rd CDC*, pp.3587-3592, 1994.

[5] J. Guckenheimer and S.D. Johnson, "Planar hybrid systems", *in Hybrid Systems II: Lecture Notes in Computer Science*, vol.999, pp.202-225, Spronger-Verlag, 1995.

[6] S.D. Johnson, "Simple hybrid systems", *Int. J. Bifurcation Chaos*, vol.4, pp.1655-1665, 1994.

[7] M.S. Branicky, "Multiple lyapunov functiojns and other analysis tools for switched and hybrid systems", *IEEE Trans. Autmatic Control*, vol.43, pp.475-482, 1998.

[8] J. Hutchinson, "Fractals and self-similarity", *Indiana Journal of Mathematics*, vol.30, pp.713-747, 1981.

[9] M.F. Barnsley, Fractals Everywhere, Academic Press, Boston, 1988.

[10] K.Gohara and A.Okuyama, "Dynamical Systems Excited by Temporal Inputs: Fractal Transition between Excited Attractors", *Fractals*, vol.7(2), pp.205-220, 1999.

[11] K.Gohara and A.Okuyama, "Fractal Transition - Hierarchical Structure and Noise Effect", *Fractals*, vol.7(3), pp.313-326, 1999.

[12] K.Gohara, H.Sakurai, and S.Sato, "Experimental Verification for Fractal Transition Using a Forced Damped Oscillator", *Fractals*, vol.8(1), pp.67-72, 2000

[13] J.Nishikawa and K.Gohara, "Fractals in an Electronic Circuit Driven by Switching Inputs", *Int. J. Bifurcation and Chaos*, in press.

[14] S.Sato and K.Gohara, "Fractal Transition in Continuous Recurrent Neural Networks", *Int. J. Bifurcation and Chaos*, vol.11(2), pp.421-434, 2001.

[15] Y.Yamamoto and K.Gohara, "Continuous Hitting Movements Modeled from the Perspective of Dynamical Systems with Temporal Input", *Human Movement Science*, vol.19(3), pp.341-371, 2000.

[16] R.Wada and K.Gohara, "Fractals and Closures of Linear Dynamical Systems Stochastically Excited by Temporal Inputs", *Int. J. Bifurcation and Chaos*, vol.11(3), pp.755-779, 2001.

[17] R.Wada and K.Gohara, "Closures of Fractal Sets in Non-linear Dynamical Systems with Switched Inputs", *Int. J. Bifurcation and Chaos*, vol.11(8), pp.2205-2215, 2001.

Speech Recognition using Pulse Coupled Neural Networks with a Radial Basis Function

T. Sugiyama	N. Homma	K. Abe
Graduate School of Eng.	College of Medical Sciences	Graduate School of Eng.
Tohoku University	Tohoku University	Tohoku University
Aoba-yama 05, Aoba-ku	2-1 Seiryo-machi Aoba-ku,	Aoba-yama 05, Aoba-ku
Sendai, 980-8579, JAPAN	Sendai. 980-8575, JAPAN	Sendai, 980-8579, JAPAN

({sugiyama, homma, abe}@abe.ecei.tohoku.ac.jp)

Abstract

In this paper, we develop a novel pulse coupled neural network (PCNN) for speech recognition. One of advantages of the PCNN is in its *biological-based neural dynamic structure* using feedback connections. To recall the memorized pattern, a radial basis function (RBF) is incorporated into the proposed PCNN. Simulation results show that the PCNN with RBF can be useful for phoneme recognition.

1 Introduction

A medium of communication is analog-value signals for the most of conventional neural network models, such as multilayered feedforward neural networks (MFNNs) and radial basis function (RBF) networks. On the other hand, in central nerve systems, neural information is transfered by neural impulse signals. Since pulse signals can be used in digital communication systems, neural information on the neural impulse has a high tolerance for noise and thus it is appropriate for long-distance transmission [1].

As an another alternative, pulse coupled neural networks (PCNNs) have been proposed [2], where impulse signals are used as the medium of information transfer as in the biological brains. For example, a PCNN can be used to realize neural synchronization effects that are observed in visual cortex of monkey and cat [2].

Conventional PCNNs have been mainly used for image processing and storing memories using their dynamic synchronization property [3, 4]. However, these works were only for static processing.

In this paper, we propose a novel PCNN for speech recognition. The proposed PCNN can store patterns dynamically using its feedback connections in the synaptic and somatic neural operations. In order

to recall the memorized patterns, a radial basis function (RBF) is incorporated into the neurons of the proposed PCNN as a neural confluence function. Simulation studies show that the proposed PCNN with a RBF can be useful for recognizing phoneme patterns.

2 Pulse Coupled Neural Networks

Figure 1 shows a neural unit in a pulse coupled neural network (PCNN). Basically, a neuron receives amplitude inputs as synaptic potentials and provides impulse outputs through the axon. The first important component is the dynamic synapse that is represented by leaky integrator. Through the ith synapse, the input impulse series $I_i(t)$ from pre-synaptic neuron is changed into amplitude potential $v_i(t)$ (synaptic potential) given as

$$v_i(t) = K_v e^{-t/\tau_v} \int_0^t e^{t'/\tau_v} I_i(t') dt' \qquad (1)$$

where, $i \in 1, 2, \cdots, n$, n is the number of inputs. K_v is a constant, and τ_v is the leakage time constant of $v_i(t)$. When a synapse gets an input impulse the integrator is charged and its amplitude (synaptic potential) rises steeply. This is followed by an exponential decay with τ_v. Because of the decay dynamics, the influences of old information become weaker gradually.

A neuron in the PCNN receives synaptic potentials $\boldsymbol{v}(t) = [v_1(t) \cdots v_n(t)]^T \in \Re^n$ at the dendrite and has the internal potential $u(t)$ given as

$$u(t) = g(\boldsymbol{v}(t)) \qquad (2)$$

$g(.)$ is a confluence function of $\boldsymbol{v}(t)$. For example, in a conventional PCNN, $g(.)$ is the following summation.

$$g(\boldsymbol{v}(t)) = \sum_{i=1}^{n} v_i(t) \qquad (3)$$
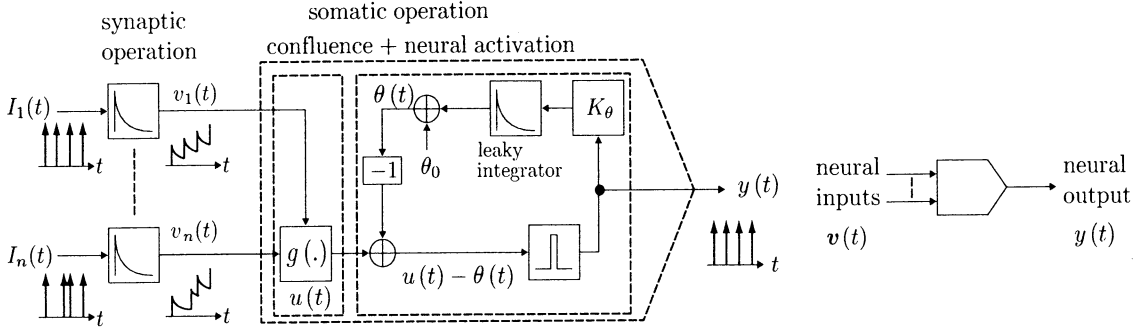
Figure 1: A neural unit in a PCNN.
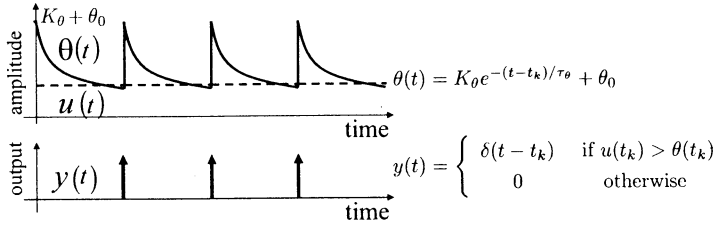
Figure 2: The dynamic threshold $\theta(t)$ and impulse generation for a constant-value input series $u(t)$.

The internal potential is the important value to generate impulse in somatic operation.

We assume that, at time t_k, $k = 1, 2, \cdots$, the internal potential $u(t_k)$ is greater than its dynamic threshold $\theta(t_k)$ given as

$$\theta(t) = K_\theta e^{-(t-t_k)/\tau_\theta} + \theta_0 \ , t_k < t \le t_{k+1} \qquad (4)$$

where K_θ is a constant, τ_θ is the leakage time constant, and θ_0 is a bias. Then the neural impulse output is generated as

$$y(t) = \left\{ \begin{array}{ll} \delta(t - t_k) & \text{if } u(t_k) > \theta(t_k) \\ 0 & \text{otherwise} \end{array} \right. \qquad (5)$$

A typical example of the action in the soma is shown in Fig. 2. That is, the output series of neural impulse is given by

$$y(t) = \sum_k \delta(t - t_k) \qquad (6)$$

Once an output impulse of the neuron is generated, by using Eq.(4), the impulse at time t_k charges the leaky integrator to such a high value of $\theta(t_k) = K_\theta + \theta_0$ that $u(t)$ cannot exceed $\theta(t)$ for a duration $t_k < t \le t_{k+1}$. This transitory elevation of $\theta(t)$ produces absolute and relative refractory periods [2].

PCNNs have been used for image processing like object detection because they are based on visual nervous system, in particular on its synchronization property
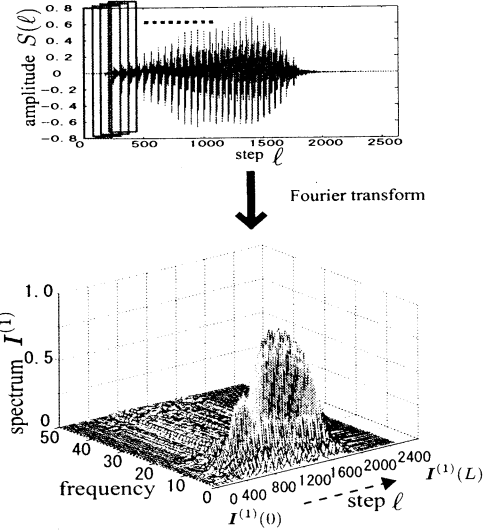
Figure 3: The transformation into frequency spectrum sequence $I^{(1)}(t)$.

[3]. In this study, however, the damping property is focused to deal with time variant sound inputs.

3 A Recognition Network using PCNN

3.1 Pattern recognition

We consider a phoneme time series $\{S(\ell T)\} = \{S(\ell)\} = \{S(1), \cdots, S(L)\}$, where T is a sampling period and L is the total number of steps and $S(\ell T) = S(\ell)$ represents sound amplitude at the ℓth step. As in the conventional pattern recognition problem, the entire time series can be regarded as the phoneme pattern. Let m be the number of phoneme patterns. For the ith phoneme pattern, $i \in \{1, \cdots, m\}$, the desired output vector is defined a

$$y_i^d = [y_{1i}^d \ \cdots \ y_{mi}^d]^T$$

$$y_{ji} = \begin{cases} 1 & (j = i) \\ 0 & (j \neq i) \end{cases}$$

where $j = 1, 2, \cdots, m$. Here, inspired by biological auditory organs and nerve [1], speech input series $\{S(l)\}$ is changed into a sequence of frequency spectrum for partial input series in stead of entire time series (Fig. 3). In this paper, the frequency spectrum sequence $I^{(1)}$ is given by using the following discrete Fourier transform (DFT) with overlapping window as shown in Fig. 3.

$$I_i^{(1)}(\ell) = \left| \sum_{s=\ell}^{\ell+N-1} S(s)e^{-j\frac{2\pi i}{N}s} \right| \qquad (7)$$

Where N is the width of the window of DFT, and i is the frequency number.

3.2 Network structure

For pattern recognition, we propose a PCNN as illustrated in Fig. 4. The network consists of two layers with interlayer feedforward connections. The ith neuron of the first layer, called the *input layer*, receives a single input $I_i^{(1)}(\ell)$ that corresponds to the ith element of the spectrum vector $I^{(1)}(\ell)$ at step ℓ and then gets the internal potential directly as

$$u_i^{(1)}(\ell) = I_i^{(1)}(\ell) \qquad (8)$$

If the internal potential $u_i^{(1)}(\ell)$ is greater than the dynamic threshold $\theta_i^{(1)}(\ell)$ given by the same formulation in Eq. (4), then this neuron fires and provides an impulse output, otherwise the neuron doesn't fire and provides 0 output.

$$y_i^{(1)}(t) = \begin{cases} \delta(t - t_k) & \text{if } u^{(1)}(t_k) > \theta^{(1)}(t_k) \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

The output $\boldsymbol{y}^{(1)}(t) = [y_1^{(1)}(t) \cdots y_n^{(1)}(t)]^T \in \Re^n$ of the input neurons is provided as the input $\boldsymbol{I}^{(2)}(t)$ to the second layer, called the *output layer*. Through the synapse, $v(t)$ is calculated by Eq.(1) and then the final outputs of the PCNN , $\boldsymbol{y}^{(2)}(t) = [y_1^{(2)}(t) \cdots y_m^{(2)}(t)]^T \in \Re^m$ are calculated by the Eqs.(2), (4) and (5). Here m is the number of output neurons.

3.3 Recognition mechanism

We consider the following error index

$$E_i = \|\boldsymbol{y}_i^d - \boldsymbol{y}_i^{(2)}(t)\|^2$$
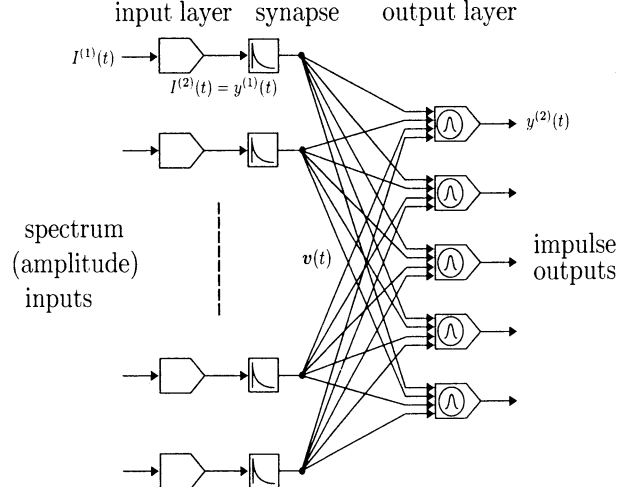
Figure 4: The recognition network.

$$= \sum_{t=T_1}^{T_2} \sum_{j=1}^{m} (y_{ji}^d - y_{ji}^{(2)}(t))^2$$

where $y_{ji}^{(2)}(t)$ is the output for the ith phoneme input and T_1 and T_2 are start and end time of evaluation. Then the total error index is defined as

$$E = \sum_{i=1}^{m} E_i \qquad (10)$$

The recognition task requires the networks to minimize the total error. However, there haven't been any models which can have memories in the conventional PCNN proposed by Eckhorn.

In this paper, we propose a memory mechanism using a radial basis function (RBF) [5] to get the corresponding desired output. RBF is used in place of simple summation of synaptic potentials in the conventional PCNN. The internal potential of the jth neuron of the output layer is given as the following RBF.

$$u_j(t) = g_j(\boldsymbol{v}_j(t)) = \exp\left(-\frac{\|\boldsymbol{v}_j(t) - \boldsymbol{w}_j\|^2}{2\sigma_j^2}\right) \qquad (11)$$

where vector $\boldsymbol{v}_j(t)$ is the synaptic potentials, vector $\boldsymbol{w}_j$ is the center of RBF which is equivalent for a memory stored in the jth neuron, and σ_j^2 is the regularization parameter of the jth neuron. When the Euclidean distance between vector $\boldsymbol{v}_j(t)$ and vector $\boldsymbol{w}_j$ is small, the internal potential $u_j(t)$ becomes large and the jth neuron of the output layer fires. The problem is how to get the memory vector $\boldsymbol{w}$ of RBF in the neuron. For a duration from T_1 to T_2, a successive neural firing of the corresponding output neuron can decrease the total error E given by Eq. (10). In order to get such a
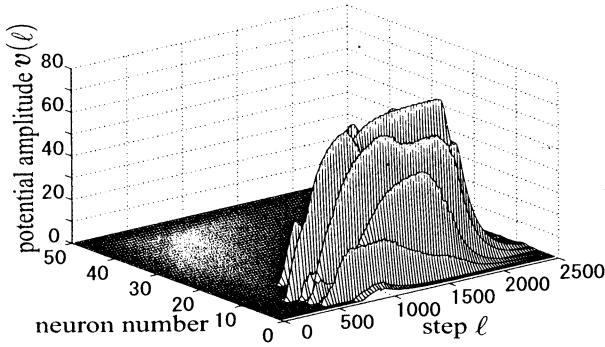
Figure 5: The synaptic potentials $v(t)$ for phoneme input "お".

Figure 6: The recognition result for phoneme input "お".

successive impulse sequence, we need almost constant sequence of synaptic potential $v_j(t)$ as the memory w_j of the RBF. As shown in Fig. 5, all the elements of $v_j(t)$ form *mountain-like shape* and thus for a appropriate duration from T_1 to T_2, the $v_j(t)$ are almost constant; that is $v_j(t) \simeq v_j(T_1)$, $T_1 \leq t \leq T_2$. Then we can choose a synaptic potential $v_j(t^*)$, $T_1 \leq t^* \leq T_2$, as the memory w_j, and $w_j = v_j(t^*)$.

4 Simulation Results

Five Japanese phoneme patterns 「あ」 ("a"), 「い」 ("i"), 「う」 ("u"), 「え」 ("e"), 「お」 ("o") were used for evaluating the recognition performance of the proposed network. Each phoneme pattern consists of 1250 steps time series with sampling frequency 8 [kHz]. In this simulation, the network has five neurons in the output layer. All the synaptic potentials are the same; that is $v_1 = \cdots = v_5 \equiv v$. In this simulation, the parameters were chosen as $K_{\theta_1} = 5$, $\tau_{\theta_1} = 0.5$, $\theta_{01} = 0.1$, $K_v = 1$, $\tau_s = 150$, $K_{\theta_2} = 5$, $\tau_{\theta_2} = 2$, $\theta_{02} = 0.5$, $N = 128$, $n = 50$ (from 0.06 [kHz] to 2.94[kHz]), $\sigma^2 = 0.4$, $T_1 = 1300$ and $T_2 = 1600$. A typical result of recognition of a phoneme input "お" is shown in Fig. 6. Only one neuron which corresponds to the input pattern fires for the duration from T_1 to T_2 when the synaptic potential vector $v(t)$ approaches to the memory vector w and thus E is decreased. The results of recognition for other phoneme inputs were good as same as that of Fig. 6.

5 Summary

The recognition network using PCNN has been proposed. In this network, it is possible for PCNN to store memories by means of a RBF as the neural confluence function of the neuron. The key mechanism in the network is a dynamic storage in the synapse and the soma. Furthermore, the decaying function may realize the natural period of recognition which is impossible in discrete network systems.

Since the PCNN is a good architecture for implementing pulse-based physiologically observed phenomenon, the pulse dynamics with feedback connections can be applied for other recognition problems. These are further research interests.

References

[1] F. Delcomyn, *Foundations of Neurobiology*, Nankodo, 2000.

[2] R. Eckhorn, H. J. Reitboeck, M. Arndt, P. Dicke, "Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex," *Neural Comput.*, Vol. 2, pp. 293-307, 1990.

[3] R. D. Broussard, Steven K. Rogers, Mark E. Oxley, and Gregory L. Tarr "Physiologically Motivated Image Fusion for Object Detection using a Pulse Coupled Neural Network," *IEEE trans. on Neural Networks*, Vol. 10, No. 3, pp. 554-563, May, 1999.

[4] Eugene. M. Izhikevich, "Weakly Pulse-Coupled Oscillators, FM Interactions, Synchronization, and Oscillatory Associative Memory," *IEEE trans. on Neural Networks*, Vol. 10, No. 3, pp. 508-526, May, 1999.

[5] T. Possio, F. Girosi "Networks for approximation and learning," *IEEE Proc.*, Vol. 78, No. 9, pp. 1481-1497, 1990.

The Level Organization by "Forwarding Forward Models": From Robot Experiments

Jun Tani
The Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama, 351-0198 Japan

Abstract

This paper reviews our recently propsed idea of the "Forwarding Forward Model" that explains how behavior primitives and their level structures can be self-organized in the context of the robot imitation learning. The model illustrates that (1) the sensory-motor sequences are generated and recognized as hierarchically articulated, (2) behaviors are generated both robustly and flexiblely by means of the bootom-up and the top-down interactions between levels. These claims are verified through the experiments using a 4-degrees of freedom arm robot with a vision system.

1 introduction

It is generally understood that higher-order cognition involves structural information processing which deals with the level of abstraction. For motor systems, it is generally assumed that the lower level system stores motor primitives and the higher level manipulates them for generating complex behaviors [1, 2]. Previously, we proposed a neural network model [7] which is characterized with its modular and level structures. Through the learning processes, each primitive for the sensory-motor representation and their abstract representation is self-organized in a local module in the lower and the higher levels, respectively.

In contrast with this localist scheme, the current study introduces a novel scheme, namely "Forwarding Forward model" (FF-model), which emphasizes its distributed representation nature. The FF-model is characterized with two levels of forward models [5] those interact each other. The lower level forward model learns to generate a specific sensory-motor sequence with self-associating values of the so-called parametric bias. The lower level forward dynamics can be bifurcated as the parametric bias is modulated by which various behavior patterns can be genetrated. The higher level forward model, on the other hand, learns to generate the sequential changes of the parametric bias by which the desired sensory-motor profile can
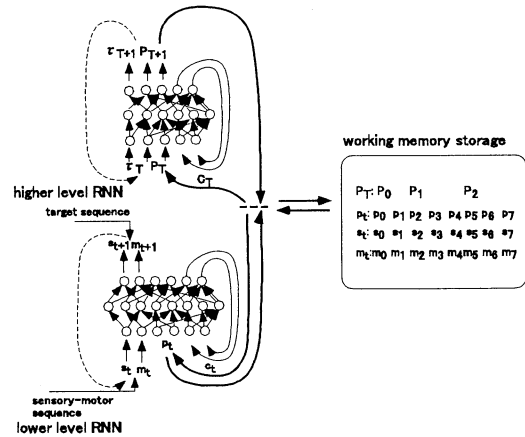
Figure 1: The FF-model utilizing two levels of RNNs.

be generated in the lower level. Since the higher level forward model learns to predict how characteristics of the lower level forward model changes in terms of the parametric bias, the higher level is said to play a role of the 2nd order forward model.

In the following, the FF-model will be described briefly. Then, we introduce our experimental results for the robot imitation learning. Those results will clarify how primitives are generated in a distributed manner and how they are utilized in articulating complex behavioral sequences. Furthermore, it will be clarified that the bottom-up and the top-down interactions are essential to generate both robust and flexible behaviors.

2 Model

We explain our model briefly. Figure 1 shows our proposed neural network architecture. The main architecture on the left-hand side consists of two Jordan type RNNs [4] which correspond to the lower and the higher level networks. These RNN networks are operated through utilizing the working memory storage shown on the right-hand side of the figure. The work-

ing memory stores the sequences of the parametric bias and the sensory-motor inputs/outputs where the regression as well as planning processes take place. In the main architecture, the lower level RNN receives two types of input. One type is the vector of the current sensory-motor values (s_t, m_t); the other is the vector of the current parametric bias p_t. This RNN outputs the prediction of the sensory-motor values at the next time step $(\hat{s_{t+1}}, \hat{m_{t+1}})$. On the other hand, the higher level RNN receives p_t as inputs, then outputs its prediction at the time step $t + 1$. It is noted that p_t can change only smoothly from the constraints mentioned later. It is also noted that the connection of p_t between the lower and the higher levels is bi-directional depending on the operational processes.

In the top-down process, the sequence of p_t is generated in the higher level RNN by means of its forward dynamics of which sequence is fed into the parametric bias units in the lower level RNN. Then, the lower level RNN generates the sensory-motor sequence as corresponding to the inputs of p_t. If p_t changes stepwisely, the sensory-motor profiles could be generated differently. With closing the loop between the outputs of the sensory-motor state and its inputs, lookahead prediction of future sequences are obtained. This mechanism is utilized for motor planning processes.

The bottom-up processes are utilized in the case of recognizing its own sensory-motor experiences. For example, let us consider that the system experiences a sensory-motor sequence while its arm is moved with a specific patten through manual guidances. If the system learned this pattern previously, the lower level RNN can re-generate this target sequence with adapting the parametric bias sequence to adequate one. The sequence of p_t is serached by means of the inverse problem of minimizing the errors between the target and output sequences with the smoothness constraints on the p_t sequence. Actually, p_t is obtained by backpropagating the error obtained during the regression and the delta error in the parametric bias node is utilized to update p_t.

When the robot actually behaves, the motor plan is generated in the top-down process while the past sensory-motor sequence is regressed. Since this regression of the past can re-interpretate and update p_t sequence, the on-line planning for future, which depends on the p_t sequence, can be generated as contextually dependig on the past.

Finally, the learning is a process to serach for optimal synaptic weights of networks and p_t sequences by which the target sensory-motor sequences (given through the manual guidances) can be regenerated
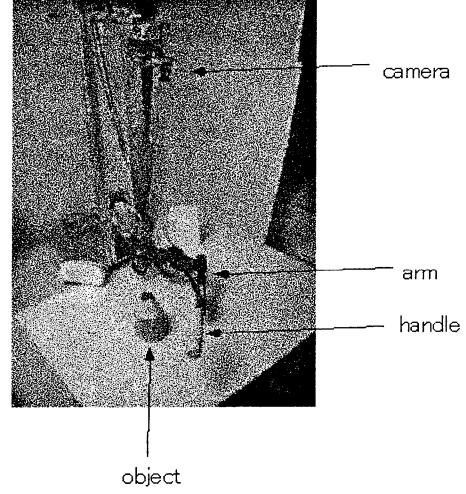
Figure 2: The arm robot.
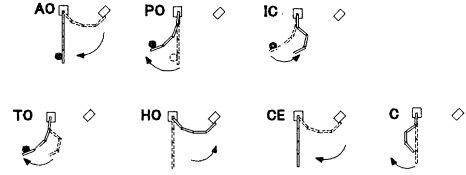
Figure 3: A set of primitive behaviors.

with minimum errors. Firstly, the lower level RNN is trained with a set of target sequences. After determination of corresponding p_t sequences, the higher level RNN is trained to be able to regenerate these p_t sequences. The update of the synaptic weights are conducted by utilizing the backpropagation through time algorithm [6].

3 Experiments

The proposed model was examined in the context of imitation learning using an arm robot with 4 degrees of freedoms equipped with a vision system as shown in Figure 2.

The arm sweeps horisontally on the surface of the task table on which a colored object is located. The positions of the object as well as the arm hand are perceived in a real time by the vision system. A handle is attached in the arm top by which the trainer can guide the arm for specific behaviors.

Before experiments, the trainer prepares a set of primitive behaviors , as shown in in Figure 3, for which he himself practices to guide the robot arm until his manual guidances become stable enough. Then, the robot is guided with a set of sequences each of which is

Adaptive Mediation based Modular Q-Learning(AMMQL) for Agents Dynamic Positioning in Robot Soccer Simulation

Mr.   Kwon   Ki-Duk

Kyonggi Univ.   Korea

kdkwon@hanmail.net

Mr.   Kim   Ha-Bin

Kyonggi Univ.   Korea

talkable@nownuri.net

Mr.   Nam   Duk-Wo

Kyonggi Univ.   Korea

carrotlover@hanmir.com

Dr.   Kim   In-Cheol

Kyonggi Univ.   Korea

kic@kyonggi.ac.kr

The Robot Soccer Simulation Game is a complex, dynamic, adversarial, multi-agent environment. What an agent should do when it is far away to the ball is one of the most difficult tactics to plan. An agent using the fixed position strategy doesn t move from its home position. In real soccer game, however, a player moves around in the field incessantly because its optimal position is changed dynamically. Reinforcement Learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. It differs from the more widely studied problem of supervised learning in several ways. The most important difference is that there is no presentation of input/output pairs. It requires on-line performance of the problem, but Robot Soccer Simulation game problem satisfies the request. This paper addresses applying reinforcement learning to agent s positioning in Robot Soccer Simulation game. We assume that the field represented by 16«11 grid-world. Each cell indicates position of teammates and opponents. An agent determines one of the empty cells as the position to move toward next. The position evaluates the safety of pass and the convenience of attack. But there are difficult problems, because there are important points that determine effect of real agents behaviors. Our studied agents choose their best possible positions based on the AMMQL based Q function values