PROCEEDINGS OF INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB)
1st

Feb.18-Feb.20, 1996, B-Con Plaza, Beppu
Oita, JAPAN

Editor : Masanori Sugisaka
ISBN4-9900462-6-9

Proceedings of International Symposium on

ARTIFICIAL LIFE AND ROBOTICS

(AROB)
1st

Feb.18 - Feb.20, 1996, B-Con Plaza, Beppu
Oita, JAPAN

Editor: Masanori Sugisaka

ISBN4-9900462-6-9

INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS (AROB)

ORGANIZED BY

Department of Electrical and Electronic Engineering
Oita University
Under The Sponsorship of Ministry of Education,
Science, Sports, and Culture, Japanese Government

CO-SPONSORED BY

Santa Fe Institute(USA)
SICE(Japan)

CO-OPERATED BY

ISICE
RSJ
IEEJ
EICE
IEEE CSS Tokyo Chapter
JRA

SUPPORTED BY

Oita Prefectural Government
Oita Municipal Government
Oita Chamber of Commerce and Industry
Beppu Municipal Government
Oita System Control Society
NHK-Oita
OBS Broadcast Company
TOS Broadcast Company
OAB Broadcast Company
Asahi Sinbunsya
Nikkankougyou Sinbunsya
Oitagoudou Sinbunsya
MITI(under negotiation)
Science and Technology Agency

INTERNATIONAL ORGANIZING COMMITTEE

Prof. J.L.Casti (Santa Fe Institute, USA)
Prof. C.G.Langton (Santa Fe Institute, USA)
Prof. W.B.Arthur (Santa Fe Institute, USA)
Prof. J.M.Epstein (Brookings Institution, USA)
Prof. S.Rasumussen (Santa Fe Institute, USA)
Prof. T.S.Ray (ATR, JAPAN)
Dr. T.Gomi(AAI,Canada)
Prof. M. Raibert(MIT, USA)
Prof. C. Looney(Univ. of Nevada-Reno, USA)
Prof. A.P.Wang (Arizona State Univ., USA)
Prof. H.H.Natsuyama(California State Univ., USA)
Prof. R.E.Kalaba (Univ. of Southern California, USA)
Prof. W.R.Wells (Univ. of Nevada-Las Vegas, USA)
Prof. D.J.G.James (Coventry University, UK)
Prof. W.R.Wells (Univ. of Nevada-Las Vegas, USA)
Prof. Y.G.Zhang (Academia Sinica, CHINA)
Prof. J.J.Lee (KAIST, KOREA)
Prof. G.I.Marchuk (Russian Academy of Sciences, RUSSIA)
Prof. S.Ueno (Kyoto School of Computer Science, JAPAN)
Prof. S.Fujimura (Tokyo University, JAPAN)
Prof. H.Miura (Tokyo University, JAPAN)
Prof. S.Arimoto(Tokyo University, JAPAN)
Prof. Y.Nishikawa (Kyoto University, JAPAN)
Prof. S.Kitamura (Kobe University, JAPAN)
Prof. K.Tsuchiya (Kyoto University, JAPAN)
Prof. T.Jinzenji (Tohoku University, JAPAN)
Prof. K.Abe(Tohoku University, JAPAN)
Prof. H.Hagiwara (Kyoto School of Computer Science, JAPAN)
Prof. H.Tanaka (Tokyo Medical & Dental University, JAPAN)
Prof. T.Mushya (Tokyo Science University, JAPAN)
Prof. T.Fukuda (Nagoya University, JAPAN)
Dr. K.Mastuno(MITI, AIST, JAPAN)
Dr. K.Tamura(MITI, AIST, JAPAN)
Dr. Y.Tokura(ATR, JAPAN)
Dr. K.Shimohara(ATR, JAPAN)
Dr. K.Kyuma(Mitsubishi Electric Co., JAPAN)
Prof. T.Yamakawa (Kyushu Institute of Technology, JAPAN)
Prof. T.Nagata(Kyushu University, JAPAN)
Prof. M.Nakamura (Saga University, JAPAN)
Prof. H.Kashiwagi (Kumamoto University, JAPAN)
Prof. M.Sugisaka (Oita University, JAPAN) (Chairman)

STEERING COMMITTEE

Prof. J.L.Casti
Prof. S.Ueno
Prof. S.Fujimura
Dr. Shibata
Dr. Nanjo
Prof. M.Sugisaka (Chairman)
Oita University Steering Committe

PROGRAM COMMITTEE

Prof. S.Fujimura(Chairman)
Prof. J.L.Casti
Prof. S.Ueno
Prof. K.Aihara
Prof. T.Ohmori
Prof. K.Shimohara
Prof. S.Arimoto
Prof. H.Tanaka
Prof. Y.Nishikawa
Prof. T.Fukuda
Prof. K.Tsuchiya
Prof. S.Ando
Prof. T.Nagata
Prof. T.Yamakawa
Prof. S.Kitamura
Prof. I.Harvey
Prof. P.Husband
Prof. R.Pfeifer
Prof. T.Nanya
Prof. M.Sugisaka

TOPICS

Hardware Oriented Topics are welcome
in the fields given by

Artificial Brains
Artificial Intelligence
Artificial Life
Artificial Livings
Artificial Mind
Chaos
Cognitive Science
Complexities
Fuzzy Control
Genetic Algorithms
Innovative Computations
Neural Networks
Neurocomputers
Neuro Computing Technologies
and Its Application for Hardware
Robotics
Virtual Reality
Related Fields

COPYRIGHT

All correspondence relating to the symposium should be addressed to:

Prof. Masanori Sugisaka
Chairman of International Symposium on Artificial Life and Robotics (AROB)
Dept. of Electrical and Electronic Engineering
Oita University
700 Dannoharu, Oita 870-11
JAPAN
TEL 001-81-975-54-7831
FAX 001-81-975-54-7620
EMAIL msugi@cc.oita-u.ac.jp

PREFACE

Masanori Sugisaka
Chairman of AROB
(Professor, Oita University)

It is my great honor to invite you all to The First International Symposium on Artificial Life and Robotics(AROB 1st), organized by Oita University under the sponsorship of Ministry of Education, Science, Sports, and Culture, Japanese Government and co-sponsored by Santa Fe Institute(SFI), USA and SICE, Japan. This symposium invites you all to discuss development of new technology concerning ALife and Robotics using new devices and technologies such as neurocomputer etc., based on simulation and hardware in the field of Microworld Simulation and Realities in 21st century.

This symposium is also financially supported by not only Ministry of Education but also Oita Prefectural Government, Oita Chamber of Commerce and Industry, and other private companies. Prof. Casti introduced SFI research fields during our discussions on joint research two years ago. This symposium was motivated by the discussions. I'd like to express my sincere thanks for Prof. J. L. Casti and also, S. Fujimura, S. Ueno, SFI Professors and all people who contributed to the symposium. I hope that you all will enjoy staying in Beppu, Oita, profit from AROB 1st, and look forward to meeting you in Beppu.

INTERNATIONAL SOCIETY FOR ARTIFICIAL LIFE AND ROBOTICS (ISAROB)

Dept. of Electrical and Electronic Eng.
Oita University
700 Dannoharu, Oita 870-11, Japan
Ph. +81-975-54-7831 Fax +81-975-54-7820

The First International Symposium on Artificial Life and Robotics (AROB 1st)

Feb. 18-20, 1996, B-Con Plaza, Beppu, Oita, JAPAN

TECHNICAL PAPER INDEX

Plenary Session : Invited Talks

The Science and the Surprise of Artificial Worlds
J. L. Casti(Santa Fe Institute, USA, Technical University of Viena)

The SWARM simulation system for modelling complex systems
C. G. Langton(Santa Fe Institute, Director, USA)

Numerical Analysis of the State of Mind
T. Musha(Brain Function Laboratory, Inc., President and Keio University, Special Invited Professor, Japan)

Evolution of Digital Organisms
T. Ray(ATR Human Information Processing Research Laboratories, Japan and USA)

Insect-Model Based Microrobot
H. Miura(University of Tokyo, Japan)

The Economy as an Ocean of Co-Evolving Beliefs
W. B. Arthur (Santa Fe Institute, Dean of Economics, Stanford University, USA)

Growing Artificial Societies: The Sugarscope Model
J. M. Epstein(Santa Fe Institute, The Brookings Institution, USA)

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

Session A1: Artificial Life

A1-1 *Computational Approach to Consciousness: Continuous Self-Monitoring, Identification, Evaluation and Decision*
Y.Ishida(Nara Institute of Science and Technology, Japan)

A1-2 *A Robot with a Personality-a fuzzy adaptive model of emotion and personality for an autonomous robot-*
K. H. Ho, N. Ohnishi(The Institute of Phisical and Chemical Research(RIKEN), Japan)

A1-3 *Study of Loop Structure on Abstract Rewriting System and its Application to Artificial Life*
Y. Suzuki, S. Tsumoto, H. Tanaka
(Tokyo Medical and Dental University, Japan)

A1-4 *An Advanced Simulation Scheme for Electric Railway Power Systems Based on Artificial Life Approach*
T. Jinzenji, M. Sasaki(Tohoku University, Japan)

A1-5 *Adaptive Block Dividing and Quantization for Image Compression using Artificial Breeding*
H. Hiramatsu, S. Kitamura(Kobe University, Japan)
Y. Kurematsu(College of Industry Technology, Japan)

Session A2: Neural Network

A2-1 *Human Type Controller Using Combined Neurons of Different Type*
T. D. Eom, J. J. Lee(Korea Advanced Institute of Science and Technology(KAIST), Korea)

A2-2 *Neurocomputer Control for Tracking of Moving Objects*
M. Sugisaka(Oita University, Japan)

A2-3 *Reinforcement Learning of Dynamic Behavior by using Recurrent Neural Networks*
A. Onat, H. Kita, Y. Nishikawa(Kyoto University, Japan)

A2-4 *Pattern Recognition Via Neurocomputer*
M. Sugisaka(Oita University, Japan)

A2-5 *Classification of Landsat TM Data Using Neural Network Approaches*
Y. Kawata, Y. Yashima(Kanazawa Institute of Technology, Japan)

A2-6 *Fault Diagnosis of Logical Circuit by use of Pseudorandom signal and Neural Network*
H. Kashiwagi, M. Sakata(Kumamoto University, Japan)

Session B1: Robotics-General-

B1-1 *A Language for Describing Motions of Mechatronic Systems :A Nonlinear Position-Dependent Circuit Theory*
S. Arimoto(University of Tokyo, Japan)

B1-2 *Aspects of Non-Cartesian Robotics*
T. Gomi(Applied AI Systems, Inc., Canada)

B1-3 *Evaluation of Robust Control by Universal Learning Network*
M. Ohbayashi, K. Hirasawa, J. Murata (Kyusyu University, Japan)

B1-4 *Cooperative Behavior of Interacting Robots*
K. Sugawara, M. Sano, Y. Sawada (Tohoku University, Japan)

B1-5 *A Step Toward Human-Robot Cooperative System*
M. Ishii, H. Saita(Fukuoka Institute of Technology, Japan)

B1-6 *A Method of Gait Coordination of Hexapod Robots Using Mutual-Coupled Immune Networks*
A. Ishiguro, S. Kuboshiki, S. Ichikawa, Y. Uchikawa
(Nagoya University, Japan)

Session B2: Genetic Algorithm

Chair: J. J. Lee(KAIST, Korea)

B2-1 *A Control Design Method of Robotic System by Genetic Algorithm*
T. Tsuji, T. Hachino(Kyusyu Institute of Technology, Japan)
R. Oguro, N. Umeda(YASUKAWA Electric MFG Co., LTD, Japan)
H. Takata(Kagoshima University, Japan)

B2-2 *Robot Path Planning through Some Places by Genetic Programming*
H. Yamamoto(Wakayama University, Japan)

B2-3 *A Local Motion Planner for Unicycle-like Vehicles: Guaranteeing the Collsion Avoidance Even in the Unknown Environment*
S. G. Hong, K. B. Park, S. W. Kim, J. J. Lee(KAIST, Korea)

B2-4 *An Optimization Method of a Combinatorial Optimization Problem using a Replicator Equation*

K. Tsuchiya(Kyoto University, Japan)
K. Tsujita, Y. Inoue(Osaka University, Japan)

B2-5 *Nonlinear Numerical Optimization with Use of a Hybrid Genetic Algorithm Incorporated the Modified Powell Method*

M. Okamoto, T. Nonaka, S. Ochiai, D. Tominaga
(Kyushu Institute of Technology, Japan)

B2-6 *A Parallel Genetic Algorithm for Matching and Similarity Measure of Graphs*

Y. Wang, N. Ishii(Nagoya Institute of Technology, Japan)

Session A3: Fuzzy and Neural Network

A3-1 *A Fuzzy Classifier System for Evolutionary Learning of Robot Behaviors*

Y. Iwakoshi(Chyubu Hitachi Electric Co., Ltd, Japan)
T. Furuhashi, Y. Uchikawa(Nagoya University, Japan)

A3-2 *A Fuzzy-Logic Approach to Industrial Control Problems*

D. J. G. James, K. J. Burnham(Coventry University, UK)

A3-3 *Computational Analysis of Acquired Dyslexia of Chinese Characters based on Neural Networks*

S. Tsumoto, H. Tanaka(Tokyo Medical and Dental University, Japan)

A3-4 *The Improved Radial Basis Function Neural Network and Its Application*

X. Wang, H. Shao(Shanghai Jiao Tong University, China)

A3-5 *A Temporal Radial Basis Function Neuro-Tracker*

C. G. Looney(University of Nevada, USA)

Session A4: Artificial Sensory Organ

A4-1 *Artificial Retina Chip*

K. Kyuma, J. Ohta(Mitsubishi Electric Corporation, Japan)

A4-2 *Vision Chip Architecture With Light Adaptation Mechanism*

H. Kobayashi(Yokogawa Electric Corp., Japan)
T. Matsumoto, K. Tanaka(Waseda University, Japan)
T. Yagi(Kyusyu Institute of Technology, Japan)

A4-3 *Artificial Passive Vision Using High Speed Pixel Matching*

H. Hanaizumi, T. Tsumashima(Hosei University, Japan)
F. Fujimura(University of Tokyo, Japan)

A4-4 *Artificial Active Vision Using Spatial Coding by m-sequence*
S. Fujimura, S. Kiyasu(University of Tokyo, Japan)

A4-5 *Building Ears for Robot : Sound Localization and Separation*
J. Huang, N. Ohnishi, N. Sugie(RIKEN, Japan)

A4-6 *Artificial Plant Trees Based on IFS*
X. Li(University of Aeronautics and Astronautics, China)
M. Sugisaka(Oita University, Japan)
Y. G. Zhang(Academia Sinica, China)

Session B3 : Mobil Robot and Vehicle

B3-1 *Dynamic Behavior Arbitration for Autonomous Mobile Robots Using Immune Networks-Evolution of Affinities among Antibodies-*
A. Ishiguro, T. Kondo, Y. Watanabe, Y. Uchikawa
(Nagoya University, Japan)

B3-2 *Auto Navigation on the wheel Chair*
H. Wang, C. Kang, T. Ishimatu(Nagasaki University, Japan)
T. Ochiai(Ube Technical College, Japan)

B3-3 *An Emergence of Fuzzy Control Rules for Mobile Robots using DNA Coding Method*
T. Yoshikawa, T. Furuhashi, Y. Uchikawa(Nagoya University, Japan)

B 3-4 *Experimental Study on Autonomous Mobile Robot Acquiring Optimal Action to Avoid Moving Obstacles*
T. Aoki(Nagoya Municipal Industrial Research Institute, Japan)
T. Oka, T. Suzuki, S. Okuma(Nagoya University, Japan)

B3-5 *The Development of A Practical Control Method for an Intelligent Mobile Vehicle*
M. Sugisaka, X. Wang(Oita University, Japan)
J. J. Lee(KAIST, Korea)

B3-6 *Development of A Stair Climbing Mobile Robot for Disabled and Elderly People*
T. Mabuchi, T. Yamada, K. Shiraki, K. Awa, T. Nagasawa
(Kansai University, Japan)

Session B4: Bio and Bio-Mimetic

B4-1 *Linear Systems Analysis of Amino Acids Production*
H . Hirayama, Y. Ishikawa, T. Nozawa, K. L. We, Y. Fukuyama
(Asahikawa Medical College, Japan)

B4-2 *Optimization of Growth of Peptide Chain by Ribosomes on a Messenger RNA*
H. Hirayama, Y. Fukuyama(Asahikawa Medical College, Japan)

B4-3 *Optimal Control in Active Transport in Biological Membrane*
H. Hirayama, Y. Fukuyama(Asahikawa Medical College, Japan)

B4-4 *Identification of a Transport System*
A. P. Wang(Arizona State University, USA)
S. Ueno(Kyoto School of Computer Science, Japan)

Session A5: Evolution

A5-1 *The Artificial Evolution Robot Control Systems*
P. Husbands(University of Sussex, UK)

A5-2 *Hardware Evolution -a real "life on the silicon"*
H. Hemmi, K. Shimohara
(ATR Human Information Processing Research Labs, Japan)

A5-3 *Adaptive Evolution of Holon Networks by an Autonomous Decentralized Method*
N. Honma, K.Abe(Tohoku University, Japan)
M. Sato(Tohoku Institute of Technology, Japan)
H.Takeda(Tohoku Gakuin University, Japan)

A5-4 *CAM-BRAIN The Evolutionary Engineering of a Billion Neuron Artificial Brain by 2001 which Grows/Evolves at Electronic Speeds inside a Cellular Automata Machine(CAM)*
H. D. Garis(ATR Human Information Processing Research Labs, Japan)

A5-5 *Evolution of Locally Defined Learning Rules and Their Coordination in Feedforward Neural Networks*
H. Murao, S. Kitamura(Kobe University, Japan)

A5-6 *Artificial Evolution and Real Robots*
I. Harvey(University of Sussex, UK)

Panel Discussion A6

Title: Coexistance between human being and robots

including artificial life

Chair: M. Sugisaka(Oita University, Japan)

J. Casti: Santa Fe Institute(Prof. Technical Univ. of Viena)
C. Langton: Santa Fe Institute
W. R. Wells: Univ. of Nevada-Las Vegas (Dean of Engineering)
J. M. Epstein: Santa Fe Institute (The Brookings Institution)
S. Rasumussen: Santa Fe Institute (Los Alamcs National Laboratory)
F. Ota: Toshiba Corporation Oita Works(General Manager)
H. Miura: University of Tokyo
T. Yamakawa: Kyusyu Insutitute of Technology (Dean of Faculty of Information Engineering)
S. Ueno: Kyoto School of Computer Science Information Science Laboratory (Emeritus Professor of Kyoto University)

Session B5: Manipulation and Autonomous System

B5-1 *Fine Motion Strategy for Skill-Based Manipulation*
A. Nakamura, T. Ogasawara, H. Tsukune
(Electrotechnical Laboratory, Japan)
T. Suehiro(Real World Computing Partnership, Japan)

B5-2 *Artificial Realization of Human Skill for Robot Manipulator Control*
M. Nakamura, S. Goto(Saga University, Japan)
N. Kyura(Kinki University, Japan)

B5-3 *Spontaneous Behavior for Cooperation of Distributed Autonomous Robots*
T. Shibata(Mechanical Engineering Laboratory, AIST, MITI, Japan,
Artificial Intelligence Laboratory, MIT, USA)
K. Ohkawa(Tsukuba University, Japan)
(Mechanical Engineering Laboratory, AIST, MITI, Japan)

B5-4 *Design Principles of Autonomous Agents: A Case Study of Classification*
R. Pfeifer(University of Zurich-Irchel, Switzerland)

B5-5 *Distributed and Autonomous Sensing based on Immune Network*
Y. Ishida(Nara Institute of Science and Technology, Japan)

B5-6 *Force/Position Control of Robot Manipulator via Motion Dynamics*
S. Jeong, K. Im, J. Kim, W.Ham(Chonbuk National University, Korea)

Session B6: Chaos and Robots

B6-1 *Associative Readout from a Chaos Network enjoying Chaotic Itinerancy*
T. Yamakawa(Kyusyu Institute of Technology, Japan)

B6-2 *Chaotic Local Search Algorithm*
C. Choi, J. J. Lee(KAIST, Korea)

B6-3 *Chaos Control on Multi-Branch U.L.N.*
M. Koga, K. Hirasawa, J. Murata, M. Ohbayashi
(Kyusyu University, Japan)

B6-4 *Autonomous Micro-Robot "Kity" for Maze Contest*
J. H. Kim, M. J. Chung, H. S. Shin, S. W. Lee(KAIST, Korea)

B6-5 *Deadlock Resolution Using Hand-to-Hand Motion in Group Robots*
H. Ishihara, T. Fukuda, F. Arai, N. Hiraoka
(Nagoya University, Japan)

Session C1: Self-Organization

C1-1 *Self-Assembly Method for Mechanical Structure*
S. Murata, H. Kurokawa, K. Tomita, S.Kokaji
(Mechanical Engineering Laboratory, AIST, MITI, Japan)

C1-2 *A Self Organizing Evolutionary Algorithm*
I. K. Jeong, J. J. Lee(KAIST, Korea)

C1-3 *A Self-Organization Learning Algorithm for Visuo-Motor Coordination in Unstructured Environments*
H. B. Zha, T. Onitsuka, T. Nagata(Kyusyu University, Japan)

C1-4 *Proposal of Decentralized Consensus-Making Mechanisms Based on Immune System*
A. Ishiguro, Y. Watanabe, Y. Uchikawa(Nagoya University, Japan)

C1-5 *Towords Designing Artificial Neural Networks by Evolution*
X. Yao, Y. Liu(University of South Wales, Australia)

Would-Be Worlds: Toward a Theory of Complex Systems

John L. Casti
Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501, USA

Abstract

By their very nature, complex systems resist analysis by decomposition. It is just not possible to study, say, the human immune system or a stock market, by breaking it up into individual parts—molecules or traders—and looking at what these parts do in isolation. The very essence of the system lies in the interaction among all its parts, with the overall behavior of the system emerging from these interactions. So by throwing away the interactions, one also throws away any hope of actually understanding the workings of the system. The problem is that until very recently, there was no way of studying these sorts of systems as complete entities, since to do experiments with stock markets, immune systems, rainforest ecosystems and the like was either too expensive, too dangerous or just plain too difficult. But the arrival of cheap, powerful, widespread computing capability over the past decade or so has changed the situation entirely.

This talk will examine the way in which the ability to create surrogate versions of real complex systems inside our computing machines changes the way we do science. In particular, emphasis will be laid upon the idea that these so-called "artificial worlds" play the role of laboratories for complex systems, laboratories that are completely analogous to the more familiar laboratories that have been used by physicists, biologists and chemists for centuries to understand the workings of matter. But now we have laboratories that allow us to explore information instead of matter. And since the ability to do controlled, repeatable experiments is a necessary precondition to the creation of a scientific theory of anything, the argument will be made that for perhaps the first time in history, we are now in a position to realistically think about the creation of a theory of complex systems.

These philosophical points will be illustrated by ongoing work with artificial road-traffic networks, as well as with systems for studying social and cultural phenomena.

1 Introduction

By more-or-less common consensus, Galileo is credited with ushering-in the idea of controlled, repeatable, laboratory experiments for the study of physical systems. And as such experiments are an integral part of the so-called *scientific method,* it's no exaggeration to say that Galileo's work formed a necessary precondition for Newton's creation of a workable *theory* of systems composed of interacting particles, a theory that formed the basis for much of modern theoretical science. But Newton's particle systems are what in today's parlance we would term "simple" systems, since for the most part they are formed of either a very small or a very large number of interacting "agents" (i.e., particles) interacting on the basis of purely local information in accordance with rigid, unvarying rules. Complex systems are different.

Typically, complex systems like a stock market or a road-traffic network involve a medium-sized number of agents (traders or drivers) interacting on the basis of limited, partial information. And, most importantly, these agents are intelligent and adaptive. Their behavior is determined by rules, just like that of planets or molecules. But the agents are ready to change their rules in accordance with new information that comes their way, thus continually adapting to their environment so as to prolong their own survival in the system. At present, there exists no decent mathematical theory of such processes. One part of the argument to be made here is that a major stumbling block in the creation of a theory of complex, adaptive systems has been the lack of ability to do the kind of controlled, repeatable experiments that led to theories of simple systems. The second-half of our argument is that the microsimulations, or "would-be worlds," presented at this meeting constitute nothing less than laboratories for carrying out just such experiments. So for the first time in history, we have the experimental tools with which to begin the creation of a bona fide theory of complex, adaptive systems.

2 Theories, Experiments, and "Big Problems"

To see the role that microsimulations will play in the creation of a theoretical framework for complex systems, it's instructive to examine briefly the history of theory construction for several major areas of modern science.

Typically, a theory of something begins its life with what I'll call a "Big Problem." This is some question about the world of nature or humans that cries out for an answer, and that seems approachable by the concepts and tools of its time. Just to get a feel for what such questions are like, here is a rather eclectic list of Big Problems from a few areas of natural and human affairs:

- *Biology: The Structure of DNA*—What is the geometrical structure of the DNA molecule, and how does this structure lead to the processes of heredity?
- *Astrophysics: The Expanding Universe*—Is the Universe open or closed, i.e., will it continue to expand forever, or will a phase of contraction back to a "Big Crunch" occur?
- *Economics: Equilibrium Prices*—In a pure exchange economy, does there exist a set of prices at which all consumers and suppliers are satisfied, i.e., is there a set of prices for goods in the economy at which the supply and demand are in balance?
- *Physics: Stability of the Solar System*—Does there exist a finite time in the future at which either there will be a planetary collision, or at which some planet attains a velocity great enough to escape the solar system?

So what we have here are four questions about the real world, each of which arises pretty much from opening our eyes and looking around. And each of these questions has given rise to a theoretical framework within which we can at least ask—if not answer—the question. But these theoretical frameworks, be they the theory of knots for studying the geometry of DNA or the fixed-point theories of economics that tell us about prices, have each come about as the outgrowth of experiments with the system of interest. For example, it was only by having access to the x-ray crystallographic studies by Rosalind Franklin that James Watson and Francis Crick were able to uncover the double-helix structure of DNA. Similarly, observations by Edwin Hubble using at the Mount Palomar Observatory showed the expansion of the universe, an empirical fact that has led to current theories of dark matter for answering the question of whether or not this expansion will continue indefinitely.

These examples—and the list could be extended almost indefinitely—illustrate the so-called *scientific method* in action. It consists of four main steps:

$$\rightarrow \text{observation} \rightarrow \text{theory} \rightarrow \text{hypothesis} \rightarrow \text{experiment} \rightarrow$$

This diagram makes the importance of experimentation evident; in order to test hypotheses suggested by a theory, we must have the ability to perform controlled, repeatable experiments. And this is exactly where the microsimulations possible using today's computing machines enter into our discussion. In contrast to the more familiar laboratories of the chemist, physicist or biologist, which are devoted to exploring the *material* structure of simple systems, the computer-as-a-laboratory is a device by which we can probe the *informational* structure of complex systems. Let me look at this point just a bit further.

3 Information versus Matter

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

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

Not only are these "information labs" different from their "matter labs" counterparts. There is a further distinction to be made even within the class of information labs. Just as even the most well-equipped chemistry lab will help not one bit in examining the

material structure of, say, a frog or a proton, a would-be world designed to explore traders in a financial market will shed little, if any, light on molecular evolution. So let me conclude this short discussion by considering some would-be worlds, each each having its own characteristic sets of questions that it's designed to address.

4 Would-Be Worlds

In the past few years, a number of electronic worlds have been created by researchers associated with the Santa Fe Institute to study the properties of complex, adaptive systems. Let me cite just three such worlds here as prototypical examples of the kind of information laboratory we have been discussing.

• *Tierra*—This world, created by naturalist Tom Ray [1], is populated by binary strings that serve as electronic surrogates for genetic material. As time unfolds, these strings compete with each other for resources, with which they create copies of themselves. New strings are also created by computational counterparts of the real-world processes of mutation and crossover. Over the course of time, the world of *Tierra* displays many of the features associated with evolutionary processes seen in the natural world, and hence can be used as a way of experimenting with such processes—without having to wait millions of years to bring the experiment to a conclusion. But it's important to keep in mind that *Tierra* is not designed to mimic any particular real-world biological process; rather, it is a laboratory within which to study neo-darwinian evolution, in general.

• *TRANSIMS*—For the past three years, a team of researchers at the Los Alamos National Laboratory headed by Chris Barrett has built an electronic counterpart of the city of Albuquerque, New Mexico inside their computers. The purpose of this world, which is called *TRANSIMS,* is to provide a testbed for studying the flow of road traffic in an urban area of nearly half a million people. In contrast to *Tierra, TRANSIMS* is explicitly designed to mirror as the real world of Albuquerque as faithfully as possible, or at least to mirror those aspects of the city that are relevant for road-traffic flow. Thus, the simulation contains the entire road traffic network from freeways to back alleys, together with information about where people live and work, as well as demographic information about incomes, children, type of cars and so forth. So here we have a would-be world whose goal is to indeed duplicate as closely as possible a specific real-world situation.

• *Sugarscape*—Somewhere in between *Tierra* and *TRANSIMS* is the would-be world called *Sugarscape,* which was created by Joshua Epstein and Rob Axtell of The Brookings Institution in Washington, DC. This world [2] is designed as a tool by which to study processes of cultural and economic evolution. On the one hand, the assumptions about how individuals behave and the spectrum of possible actions at their disposal is a vast simplification of the possibilities open to real people as they go through everyday life. On the other hand, *Sugarscape* makes fairly realistic assumptions about the things that motivate people to act in the way they do, as well as about how they go about trying to attain their goals. What is of considerable interest is the rich variety of behaviors that emerge from simple rules for individual action, and the uncanny resemblance these emergent behaviors have to what's actually seen in real life.

The main point of bringing up *Tierra, TRANSIMS,* and *Sugarscape* is to emphasize two points: (A) We need different types of would-be worlds to study different sorts of questions, and (B) each of these worlds has the capability of serving as a laboratory within which to test hypotheses about the phenomena they can represent. And, of course, it is this latter property that encourages the view that such computational universes will play the same role for the creation of theories of complex systems that chemistry labs and particle accelerators have played in the creation of scientific theories of simple systems. For a fuller account of the technical, philosophical and theoretical problems surrounding the construction and use of these silicon worlds, see the author's volume [3] which will appear in the fall of 1996.

References

[1] Ray, T., An Approach to the Synthesis of Life," in *Artificial Life–II,* C. Langton et al, eds. Addison-Wesley, 1991, pp. 371–408.

[2] Epstein, J. and R. Axtell, *Growing Societies,* MIT Press, to appear 1996.

[3] Casti, J., *Would-Be Worlds,* John Wiley & Sons, to appear 1996.

Numerical Analysis of the State of Mind

Toshimitsu MUSHA
Brain Functions Laboratory, Inc.
KSP Bldg. E-211, Sakado, Takatsu, Kawasaki, Kanagawa-ken, JAPAN

Abstract

The state of mind is supported by the brain activity, and hence features of the state of mind appear in the scalp potentials or EEGs (Electroencephalo-gram). Therefore, it is possible to estimate the state of mind by proper mathematical manipulations on spatio-temporal behavior of EEGs. The emotional state is decomposed into more elementary states. Currently ten electrodes are used and the four elementary states, *anger, sadness, joy, and relaxation,* are adopted. More electrodes would allow more elementary emotional states to be included in the analysis. The features are found in the cross-correlation coefficients of 45 pairs of EEG channels in the theta (5-8 Hz), the alpha (8-13 Hz), and the beta (13-20 Hz) bands. The totally 135 variables are obtained and they are linearly combined into four components which indicate levels of the four elementary emotional states. The maximum time resolution of the emotion analysis is 0.64 second and it is done in real time. This new technique has a wide variety in the medical and non-medical areas. This new technology suggests a possibility of direct control of systems by the human emotional state.

1. Introduction

The scalp potential (electroencephalogram, EEG) is generated by electric activities of neurons in the brain. It is rich in information about the state of the brain, or in other words about the state of mind. Features of the state of mind, however, are buried in the spontaneous EEG generated by the other brain activities, and sophisticated mathematical manipulations will be required to pick up the required features from the unwanted ones. Such *event-related* potentials are obtained by synchronous averaging of EEG by repeating the same events. However, synchronous averaging is not possible in estimating the state of mind or the emotional state because always we have to analyze single events. The feature extraction of the emotional state need to be done through single events..

We have solved this problem as described in Sec. 2. The emotional state is decomposed into four elementary states as *anger, sadness, joy*, and *mental relaxation*. This technique is similar to spectral analysis and we have given to it the name of *Emotion Spectrum Analysis*, or *ESA* in abbreviation. The present method expresses the state of the human emotion numerically, which would allow manipulation of a non-human machine according to the state of mind.

2. Method of Analysis

Ten disk electrodes are placed on the scalp at positions Fp_1, Fp_2, F_3, F_4, T_3, T_4, P_3, P_4, O_1, and O_2 according to the International 10-20 Standard, and scalp potentials are recorded with a reference electrode on the right ear-lobe. The electric potentials are sampled at 100 Hz, and then separated in the theta, alpha and beta frequency bands by means of FFT. Values of the cross-correlation coefficients on 45 (= ${}_{10}C_2$) channel pairs are evaluated in every

5.12 seconds (it is possible to shorten this time down to 0.64 second); totally, 135 such variables are obtained. The set of these 135 variables is the input vector $\boldsymbol{y}$; this is linearly transformed to a 4-vector $\boldsymbol{z} = (z_1, z_2, z_3, z_4)$ by operating a transformation matrix **C** on $\boldsymbol{y}$. Magnitudes of these components indicate levels of these elementary emotional states. We call **C** the ***emotion matrix***, and $\boldsymbol{z}$ the ***emotion vector***; they are related as

$$\mathbf{C}.\mathbf{y} + \mathbf{d} = \mathbf{z} \qquad (1)$$

where $\boldsymbol{d}$ is a constant vector.

Numerical values of the emotion matrix elements were obtained in the following way. Seven people who have been well trained in imaging participated in preparing the emotion matrix. They first imaged *anger*, and 5.12-sec EEG segments were cut out from their ten-channel EEGs; this process was repeated for all the other elementary emotional states. The numerical values of the matrix elements were determined in such a way that $\boldsymbol{z} = (1, 0, 0, 0)$, (0, 1, 0, 0), (0, 0, 1, 0) and (0, 0, 0, 1) for the four emotional states, respectively, and (0, 0, 0, 0) for the control state in which no special emotion was activated; numerical solutions are obtained in either under-determined or over-determined condition. These elementary emotional states are approximately orthogonal, or in other words they are almost independently generated. Each vector component is an index of the related emotional level.

If the emotion matrix is prepared for a particular person, its applicability is limited to this particular person only, losing general applicability because the matrix includes features of the emotional states as well as personal characters. Therefore, personal characters must be smeared out by preparing the emotional matrix based many subjects.

3. Noise Reduction

The emotion vector should have positive components only according to the definition. In the real situations, however, it shows negative as well as positive value. There are many other emotional states which the four elementary ones cannot cover. The emotion matrix cannot suppress their appearance in the 4-dimensional emotion space; their appearance in the 4-dimensional ***emotion space*** will be random, making *noise*. The noise level is estimated as standard deviation σ of the negative contribution in each vector component.

The factor $f(z)$ is multiplied to the derived emotion vector components, which is

$$f(z) = \tanh\left(\frac{z}{4\sigma}\right) = \frac{\exp\left(\frac{z}{4\sigma}\right) - \exp\left(-\frac{z}{4\sigma}\right)}{\exp\left(\frac{z}{4\sigma}\right) + \exp\left(-\frac{z}{4\sigma}\right)} \qquad (2)$$

for $z > 0$ and $f(z) = 0$ for $z < 0$. The smaller part of the emotion index that is comparable to the noise level is suppressed by this factor. On the other hand, however, the index value itself is influenced by the noise. This effect is removed by smearing the index values by the moving average. The smoothed index $\bar{z}_n$ at the nth time point is calculated as

$$\bar{z}_n = 0.1(z_{n-2}+z_{n+2})+0.2((z_{n-2}+z_{n+2})+0.4z_n \qquad (3)$$

When a subject has *anger* in mind, the index of *anger* acquires a high level. However, the index of *anger* does not always mean that the subject has *anger* in mind but that the subject has mental stress in general. In the same way the index of sadness means that the subject is in a depressed state. To avoid misunderstanding, the indexes for *anger, sadness, joy* and *relaxation* are named as *N1*, *N2*, *P1* and *R*.

4. Emotional State and EEG cross-correlation

Each emotion matrix element is concerned with a pair of EEG channels in one of the three frequency bands. A larger matrix element in magnitude (regardless of plus or minus) plays a more important role in extracting features of the emotional components. Fig.1 shows magnitudes of emotion matrix elements in reference to respective EEG channels indicated by crosses. Open and dark circles refer to positive and negative cross-correlation coefficient, respectively, and the radius represents the magnitude of the coefficient in four steps.

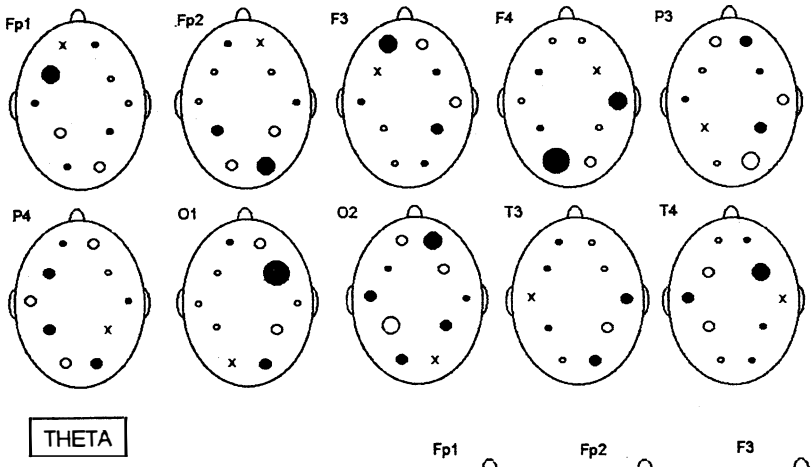
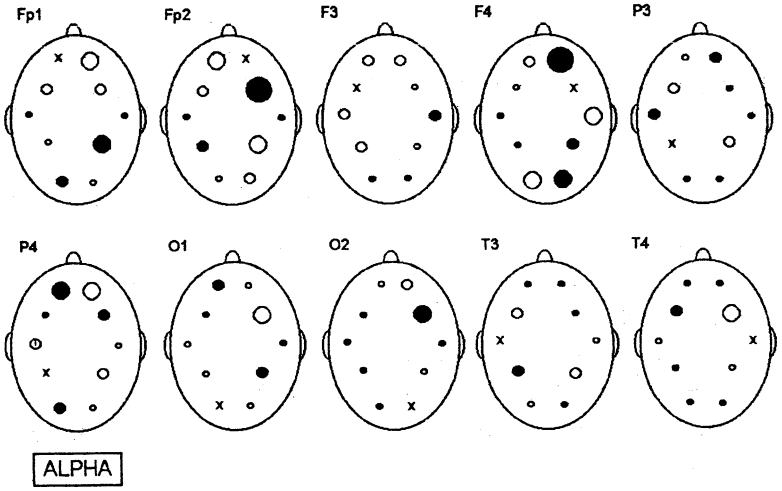


Fig. 1—1

Fig. 1—2



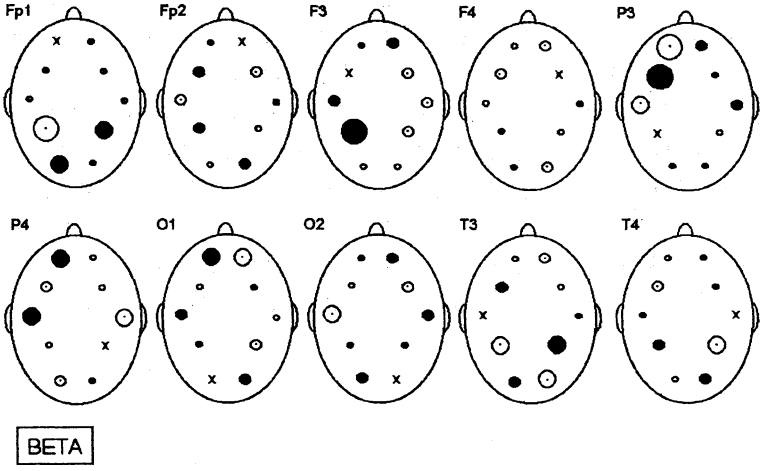


Fig. 1—3

5. Results of Emotion Analysis

A female subject did algebraic tasks for an hour. After quitting this work the emotion analysis started and the result is shown in Fig. 2. This figure shows magnitudes of the four indexes, *N1*, *N2*, *P1* and *R*, from the top, and each bar corresponds to 5.12 sec. Initially the subject showed signals mainly in *N1* and *N2*. *N1* indicates mental stress or excitation; when the subject is psychologically tired this index is usually accompanied by *N2*. When, on the other hand, the subject is excited with interest in something, *N1* is accompanied by *P1*. In the present case, the subject was tired in mathematical work. Then she began to listen to her favorite music; immediately these two indexes decreased to low levels, and at the same time index *P1* (characterizing joy) increased which, however, was reduced after two or three minutes probably because of habituation. Interesting to say, indexes *N1* and *P1* were fluctuating in the opposite phase. After the music was over, the same state continued.

The second example is shown in Fig. 3. Two male subjects A and B were doing algebraic tasks. Subject B finished the work earlier than subject A and told the examiner that he had finished it. Subject showed relatively high relaxation index *R* and low stress index *N1* before subject B finished his work. When he found that he was behind his partner in finishing the work, his relaxation level lowered and the stress index increased.

6. Conclusion

Our technique of making numerical evaluation of the state of mind at high time resolution is new, and it will contribute to artificial intelligence, robotics, and so on.

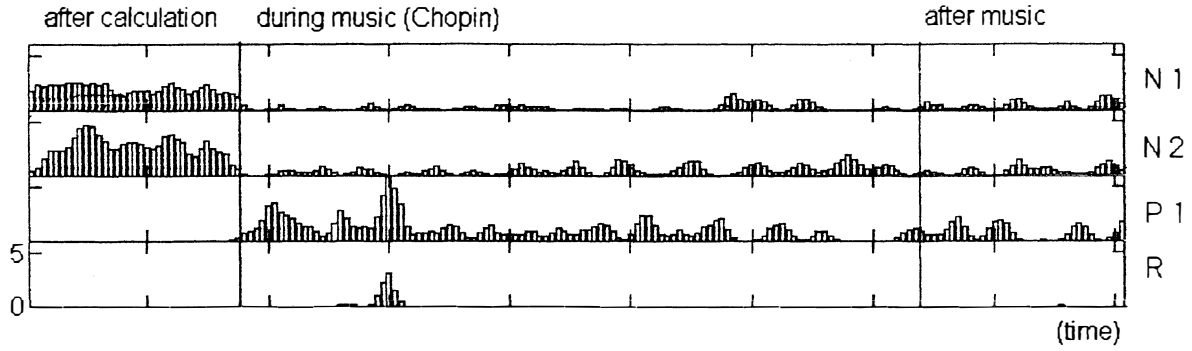


Fig. 2

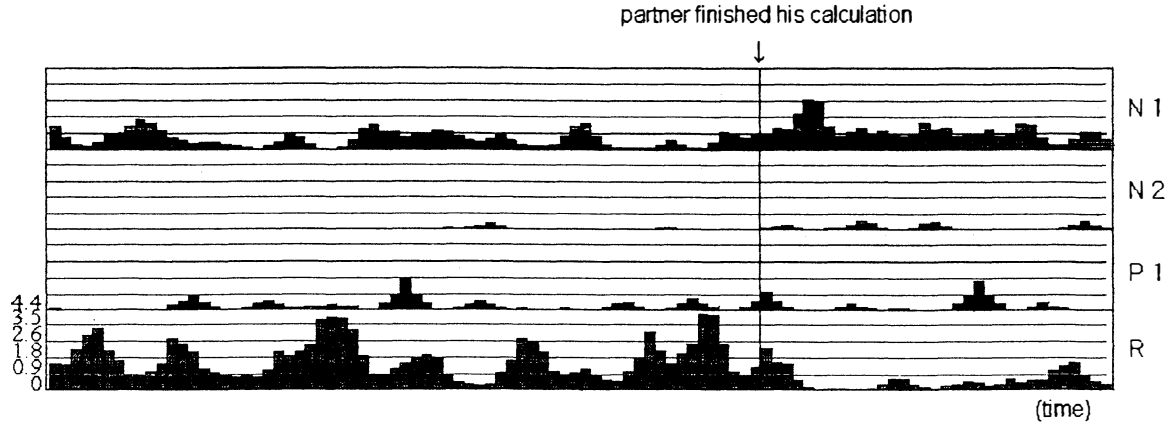


Fig. 3

Evolving Complexity

Thomas S. Ray
ATR Human Information Processing Research Laboratories
2-2 Hikaridai Seika-cho Soraku-gun
Kyoto 619-02 Japan
81-7749-5-1063 (phone), 81-7749-5-1008 (fax)
ray@hip.atr.co.jp ray@santafe.edu ray@udel.edu
http://www.hip.atr.co.jp/ ray/

Abstract

Humans have been practicing applied evolution since the dawn of agriculture, long before the development of the theory of evolution. The domestication and breeding of plants and animals is based on the application of artificial selection to captive populations. However, our management of evolution has taken place at the "micro" level, the alteration of existing species. We have never been able to harness and manage the more creative properties of evolution: the origin of new species, and the emergence of complexity itself. We are able to guide the evolution of poor quality wild corn into high quality domestic corn, however, we can not guide the evolution of algae into corn. To manage complexity increase in synthetic evolutions requires an entirely new approach: the creation of "natural" ecological communities where complexity increase can occur through evolution by natural selection.

1 Synthetic Evolution

It has been said that "life is art". If life is art, evolution is the artist. The creative products of evolution include the human body and mind, the cheetah running down its prey, the mahogany tree, the humming bird pollinating a flower. These living works of art exceed in beauty, and depth of structure and process, anything produced by the best of human artists. In fact, human artists themselves are products of evolution.

Human artists express themselves in many media: oil paint, clay, stone, music, cinema. What is the medium of expression of evolution? On Earth, evolution is best known for its works in the medium of carbon chemistry. However, it has recently been demonstrated that evolution can work in other media. In the last few decades, evolution has begun to express its creative potential through the digital medium.

Much of the work with evolution in the digital medium has had clear engineering objectives, but some work has simply been an exploration of the process of evolution in an unfamiliar medium. In this latter case, the conditions are created for evolution by natural selection to take place among self-replicating computer programs. The result is a diversifying phylogeny of "digital organisms", in a "natural" ecological community within cyberspace.

To the extent that an ecological community of freely evolving self-replicators can be considered alive, this represents another instance of life, but a very alien life. If we could travel to other planets and observe life there, it would probably be organic life, and to that extent, somewhat familiar. But life embedded in the radically different medium of digital computation would be even more alien than extra-terrestrial life.

Freely evolving digital organisms are an instance of evolution, not a model or simulation of organic evolution. Like a chemostat full of evolving bacteria and viruses, a computer full of evolving digital organisms could be thought of as a model system for the experimental study of evolution. However, the evolving digital organisms are no more a simulation of evolution that are the evolving bacteria and viruses.

2 Complexity Increase

Earth's most creative evolutionary transitions were reviewed recently [3]. They seem to occur relatively abruptly, compared to the background rate of evolutionary change. Some of the major transitions noted were: origin of chromosomes, origin of eukaryotes, origin of sex, origin of multi-cellular organisms, origin of social groups.

Of these major transitions, perhaps the most dramatic, and best known, was the rapid origin and diversification of large multi-cellular organisms from micro-scopic single celled ancestors, in what has come to be known as the Cambrian explosion of diversity [1]. It has understandably been called evolution's "big bang", when there was a dramatic inflation of complexity of organisms, and species diversified rapidly into an ecological void.

Because the Cambrian explosion generated the largest of organic life's complexity increases, it is interesting to consider what its digital analogue may be. At its most fundamental level, the Cambrian explosion arose out of the transition from single to multi-cellular organisms. The digital analog would be a transition from serial to parallel processes.

If we make an analogy between the cell and the processor, then modern multi-cellular organisms are parallel programs on a scale of complexity that vastly exceeds any existing computer software. In organic life, the program is the genome, based on nucleic acid sequences. In humans this program has roughly three billion bases. However, no individual cell expresses all the genes in the genome. Each cell expresses a small subset of the genes, and this subset defines which "cell type" the cell is. It may be a skin cell, liver cell, brain cell, etc. depending on what subset of genes it expresses.

The human body is thought to have several hundred distinct cell types, with a total of trillions of cells. This corresponds to the two main types of parallelism in computer software: SIMD and MIMD. In SIMD parallelism there is a single instruction pointer shared by all of the processors, so every CPU executes the same code in absolute synchrony. In MIMD parallelism, each CPU has its own instruction pointer, so each processor is capable of executing a different set of code.

SIMD parallelism corresponds roughly to multiple cells of a single cell type, in that in both cases, the same genetic code is being expressed. MIMD parallelism corresponds to multiple cells of different cell types in that different cells are expressing different code. Large modern multi-cellular organisms combine both SIMD and MIMD parallelism on a massive scale. However, most parallel computer software is primarily of the SIMD type. The reason is that SIMD software is approximately the same as serial software, but executed simultaneously on many processors and presumably operating on different data. However, existing MIMD software is much simpler that the genomes of organic multi-celled organisms. Such software is just too complex to write. While existing MIMD computer hardware has the capability of having hundreds or thousands of processors each executing different code, while all cooperating on a single task, there does not exist a human art for writing such software. It is beyond the capability of programmers to write code involving more than a few distinct processes.

3 Tierra

My own work with digital evolution is an attempt to generate a free process of evolution by natural selection. I have set up a system of self-replicating computer programs, where the only clearly defined "fitness" criteria imposed on the system is the same as what is found in organic evolution: the ability to replicate, or transmit genetic material to future generations. My system, called Tierra (Spanish for Earth) creates the basic Darwinian scenario: self-replicating entities in a finite environment with heritable genetic variation, inside the computer.

In Tierra, the self-replicating entities are executable machine code programs, which do nothing more than make copies of themselves in the RAM memory of the computer. Thus the machine code becomes an analogue of the nucleic acid based genetic code of organic life. The machine code programs occupy space in the RAM memory, thus the memory provides an analogue of the physical space of organic life. Each program "owns" the block of memory that it occupies, and has an exclusive privilege of writing on its own memory blocks. However, any process may read, or execute the machine instructions in any part of memory. Thus the partial (only write) privilege surrounding the space occupied by the program is thought of as analogous to a semi-permeable membrane surrounding organic cells, partially protecting the internal chemistry from disruption from the surroundings.

Genetic variation is introduced into the population of replicating programs by randomly flipping bits in the machine code. This is analogous to mutations involving substitutions of nucleic acids in the DNA sequence of organic life. Additional noise is introduced into the system in the form of occasional errors in the computations performed by the CPU (central processing unit).

The replication of the programs is brought about through their execution by the CPU of the computer. Thus CPU time provides the analogue of the energy that drives the metabolism of organic life.

The results of the evolutions that have been observed so far represent a variety of solutions to the

problem of self-replication of machine code programs. We might consider these solutions to fall into two broad classes: "ecological solutions", and "optimizations". Ecological solutions involve interactions between the programs sharing the memory, whereas optimizations involve innovations within the individual algorithms that result in faster replication.

We know that evolution embedded in the medium of carbon chemistry is capable of generating parallel software combining both SIMD and MIMD parallelism on an astronomical scale. However, we don't know the capabilities of evolution when embedded in the medium of digital computation. The experiments that have been conducted so far show a surprising ability of evolution to reorganize and improve machine code.

However, organic evolution was able to increase the complexity of the replicators by many orders of magnitude. So far digital evolution has shown small increases of complexity, perhaps by a factor of as much as two or three. Is digital evolution capable of very large spontaneous complexity increases, and if so, under what conditions? The answers to these questions can only be determined by experimentation.

An effort is underway to create conditions that might provoke evolution to generate substantially greater complexity in digital replicators. I am preparing what I call the "biodiversity reserve for digital organisms." This will be based on a networked version of the Tierra software. We will attempt to get thousands of people to run network Tierra on their computers, and all of these computers will be connected into a virtual sub-net of the internet, within which digital organisms will be able to move freely from computer to computer.

This configuration offers two advantages over the original single-computer version of Tierra: greater size, and greater complexity of temporal and spatial patterns of resources.

There will be a dramatic increase in the size of the space, and the amount of CPU cycles available to the digital organisms. The single-machine version is generally run with a memory space of one or two hundred thousand bytes. However, a single complex digital organism could exceed that size.

It is believed that evolution can only generate great complexity in the context of an evolving ecological community, as biotic evolutionary interactions are an important driving force in evolution. Thus we need a space that can hold entire populations of many species of large organisms. Suppose that we want to support, minimally, a hundred species, each with populations of five hundred individuals, with individuals of ten thousand bytes in size. This would require a space of about five hundred megabytes. While this space could be achieved on a single computer, there would be a very bad ratio of memory to CPU power. Distributing the memory over many computers provides a better memory/CPU ratio.

While greater size is an absolute requirement, large size alone would probably not provoke evolution towards greater complexity. However, the distributed model provides a great complexity of temporal and spatial patterns of resources. It is felt that this complexity could provide selective pressures for evolution to create complex adaptations to those patterns.

The network Tierra software will be run as a low-priority background process, like a screen saver. This means that when the user is actively using the computer, the Tierra software will sleep, receiving no CPU cycles, the energy source for the digital organisms. Any digital organisms who are present on the machine at that time will be frozen, unable to metabolize or reproduce.

This should create a strong selective pressure for individuals to move about the net, avoiding sleeping Tierra programs, and seeking those with a rich supply of CPU cycles. Evolution may generate behaviors that respond to temporal patterns in the availability of CPU cycles. For example, there is likely to be a daily cycle, with generally more free cycles at night when people are sleeping. So some digital organisms might evolve the behavior of migrating around the planet on a daily basis staying on the dark side of the planet.

The approach for managing the evolution of complexity that is being advocated here is to generate a "natural" ecological community of freely evolving digital organisms. The environment in which they live should include sufficient complexities to provide some selective pressures for increasingly complex behaviors to evolve. Once a significant impulse in the direction of complexity has occurred, the hope is that selective forces arising from interactions among the digital organisms can lead to an auto-catalytic increase in complexity.

It appears that this is what happened in organic evolution. In the Amazon region, there are rain forests on white sand soils, where the physical environment consists of clean white sand, air, falling water and sunshine. Embedded in this physical environment is the most complex ecosystem on earth, the tropical rain forest. In this ecosystem there are hundreds of thousands of species. These do not represent hundreds of thousands of adaptations to the physical environment, but rather, most of the adaptations of these species are

with respect to the other living organisms that they interact with.

Life transforms the environment, such that the living component of the environment comes to predominate over the physical environment, after which most evolution involves adaptations to other living organisms. Thus the complexity of the living component of the environment comes to greatly exceed the complexity of the physical environment that it is embedded in.

The first evolution observed in Tierra was the origin of ecological interactions, which were based on adaptation to the presence of other digital organisms in the environment [2]. Thus this dynamic has been present in Tierra from the beginning. It is hoped that with the help of some impulse towards greater complexity, this dynamic can lead to a large spiraling upwards in complexity.

4 Practical Applications

How might we work with digital evolution to produce useful products? Although we have a well established practice of plant and animal breeding, our relationship to digital evolution is quite different. Our ancestors were able to go out into nature and observe many highly evolved and complex organisms. They found uses for some of these, such as the ancestors of, rice, corn, wheat, chickens, pigs, dogs, etc. They then bred them to produce the much improved domesticated plants and animals that we know today.

However, in the case of digital evolution, we are starting with very simple organisms that have not yet achieved the complexity to be useful, so our first objective is to evolve complexity. We have no prior experience with managing the evolution of complexity.

Probably any attempt to guide the evolution of algae to become corn, through artificial selection in the context of a breeding program, would prevent such a transition from occurring. I believe that we will never be able to guide the evolution of complexity by the use of artificial selection. To facilitate complexity increase we need a new and different approach.

I suggest that the most likely way to achieve complexity increase in digital evolution is through evolution by natural selection in an ecological community. No attempt should be made to provide fitness functions, or artificial selection, to guide evolution towards useful products. Rather, evolution should be free explore the possibilities without the burden of human "guidance".

Traditionally we have managed evolution through manipulating selective forces. In this new approach, our role is to create the conditions for complexity increase, rather than trying to guide it through artificial selection. This is an interesting scientific challenge, as the conditions that generate complexity increase are unknown. However, the process of attempting to generate a complexity increase, amounts to an experimental approach to the problem. We can try many different approaches, and if we are successful, we will be able to experiment with the system until we can find the minimal conditions that generate the behavior.

If a digital analog to the Cambrian explosion can be achieved, then it should be possible to establish the same kind of relationship to digital evolution that we have with organic evolution. We can go out into digital nature, and observe the complex products of evolution. While most digital organisms will have no application, it is likely that some will. We can observe them for interesting and potentially useful information processes. When we identify potentially useful digital organisms, we can capture them and subject them to selective breeding to enhance their performance on the application, and inhibit unruly wild behavior. Eventually the product can be neutered and sold to the end user.

We have seen the tremendous creative potential of evolution when expressed through the medium of organic chemistry. We do not yet know the full potential of evolution in the medium of digital computation. However, the initial experiments have been very promising, suggesting that it is worthwhile to make the effort to push digital evolution to its limits. If digital evolution has even a small fraction of the potential of organic evolution, it could result in information process of a complexity far beyond anything that we have experience with today. While there are many potential obstacles and technical problems along the way, the possible rewards for success make the risk worth taking. Yet it is a venture into the unknown for which we can not estimate the likelihood of success.

References

[1] Gould, Steven J. 1989. Wonderful life. W. W. Norton & Company, Inc. Pp. 347.

[2] Ray, T., An Approach to the Synthesis of Life," in *Artificial Life–II,* C. Langton et al, eds. Addison-Wesley, 1991, pp. 371–408.

[3] J. Maynard Smith and E. Szathmáry, The Major Transitions in Evolution (Oxford: Freeman, 1995).

INSECT-MODEL BASED MICROROBOT

Hirofumi Miura, Takashi Yasuda, Yayoi Kubo Fujisawa, Yoshihiko Kuwana and Isao Shimoyama

The University of Tokyo, Mechano-Informatics, 7-3-1, Bunkyo-ku, Hongo, Tokyo 113, JAPAN
e-mail: miura@leopard.t.u-tokyo.ac.jp URL: http://www.leopard.t.u-tokyo.ac.jp

Abstract

It should be understood that the robot is not a simple automatic machine, but instead has certain level of intelligence. Many kinds of intelligent robots have been developed in the author's laboratory during the past 15 years. These robots perform many kinds of games like the cup & ball game, top-spinning, walking on stilts, etc. These robots apparently look intelligent, but are they really and truly intelligent? There is one opinion that these robots are no more than simple automatic machines which are controlled by a computer with sophisticated programs. If so, then what is actual robot intelligence? The author is trying to construct a new robotics --- insect-model based microrobotics --- in order to get a new concept of robot intelligence.

1 WHAT IS ROBOT INTELLIGENCE?

Intelligence for the robots shown in Fig. 1 is discussed below.

(1) INTELLIGENCE FOR FAST AND ACCURATE MOTION (Cup & ball game robot)

When the motion is slow, PTP(point to point) control works well. But for the fast motion, PTP control yields position error because for fast motion, the inertial forces (centrifugal force, Colioris force, etc. which act on the joint motors) are considerably large and feedback control is disturbed by them.

So-called " inverse dynamics " is a very effective control scheme for fast motion. The cup & ball game cannot be implemented by the usual industrial robot employing PTP control scheme. On the other hand, by employing the inverse dynamics, the robot played cup & ball game with a 95% success ratio. Inverse dynamics can be robot intelligence for fast and accurate motion.

(2) INTELLIGENCE FOR LEARNING (Inverted pendulum)

The ability to learn is one of the most challenging subject in robotics. The inverted pendulum robot was developed as an example of a robot which has learning ability. If the geometrical dimensions (length, weight, position of center of gravity, moment of inertia, etc.) are not given, learning control must be employed. During the first several minutes of the experiment, a person's hands help the pendulum to maintain the vertical attitude, but after several minutes the pendulum has learned to keep the vertical attitude all by itself.

(3) INTELLIGENCE FOR DYNAMIC BALANCE (The biped and the quadruped)

In the author's laboratory many kinds of bipeds have been developed. The biped (stilt type) is statically unstable but can be balanced dynamically. It has intelligence for dynamic balance[1]. Another type of biped (human type) has knee joints and ankle joints like a human. Eight motors are mounted in this robot in total, and it can walk more slowly than the stilt type. It also walks dynamically.

The quadruped also walks dynamically. Real animals support their bodies with two legs (not three legs) and swing the other two legs in the air even during a slow walk. At low speed, one fore leg and one hind leg on the same side are in the air together. This gait is called "pace". When the walking speed of an animal gets faster, the gait changes to "trot (diagonal two legs are in the air together)". The gait is selected by a minimum energy consumption criterion[2],[3].

(4) INTELLIGENCE FOR LEARNING THE ROPE (top-spinning robot)

The author can spin a top. However, it is very hard to teach another person how to spin a top. This means that writing a program for the top-spinning robot is also very hard. How fast should the arm move? At which position should the top be thrown forward? How strongly should the string be pulled back? The author learned these tricks by watching his father's play. For the robot, the trajectory of a human arm and a top during human top-spinning play was input into the computer through two video cameras (one camera in front, and one over the head) and the computer generated a program to reproduce the same motion of the arm and the top from human play. With this program the robot spins a top very skillfully. This example may be called " teaching by showing."

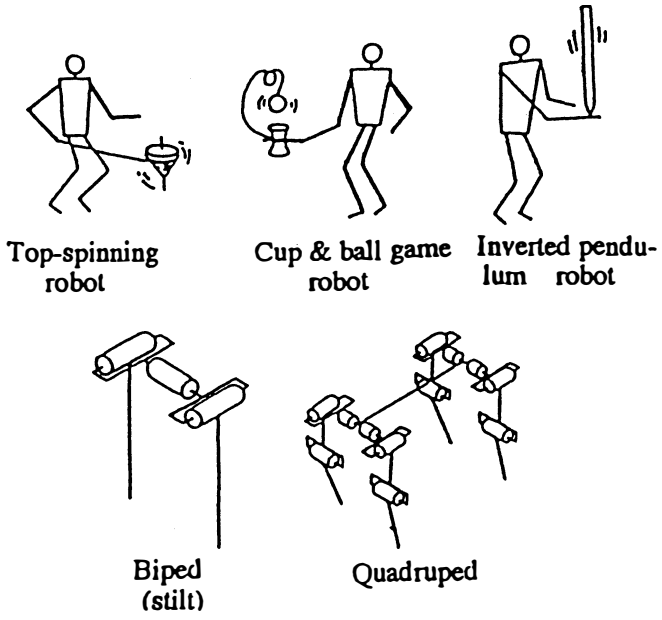


Fig. 1 Some Examples of Intelligent Robots which have been developed at the author's lab.

2 CAN A MACHINE HAVE ITS OWN WILL?

Several examples of robot intelligence were presented above. These robots complete dexterous tasks like a human does. However, the author is not sure whether these robots are as intelligent as a human, because all algorithms expressed in the program are developed by a human investigating the result of analysis of human play. The robot itself is not intelligent and it only follows the program given to it by a human. The robot has no will to play these games better and better.

On the other hand, the author practiced hard on cup & ball game, top-spinning, etc., so that he could play better than his friends during childhood.

Can a machine have will? This is a big question in robotics and AI technology. The author is trying to find a new way for a small but steady step to answer this question[4].

3 INSECT-MODEL BASED ROBOTICS (Microrobot)

An insect looks much more intelligent and more lively than robots introduced above. The insect looks to have a will. If an insect robot could be developed, it is possible to make it look truly intelligent. The author's laboratory started developing insect-model based robots about five years ago when looking for a new and different way to approach intelligent robots. Since the late 1980s, microtechnology has been highlighted as a promising technology for the development of very small sized mechanical systems. Fabrication of micromechanisms on a silicon wafer using IC process may be the key technology for developing mm sized microrobots. Millimeter sized insects are all around us. The author considered that microtechnology should be applied to developing an insect-model based robot. Although there are still many problems which must solved to build real insect-model based microrobots, there are basic lines of solution for some problems with experimental results. This paper argues that insects will be good models for microrobots for design, control, actuation, etc., just as human beings or mammals are good models for normal-sized robots. In addition, an insect has only 10^5 neurons. The motion of an insect is produced by simple mechanisms such as "a reflex act". But it is interesting to the author that this motion looks intelligent. This is why the author is trying to develop an insect-model based robot.

4 EXTERNAL SKELETON

For developing microrobots, new design concepts must be constructed. For instance, a frictionless structure must be designed. Rotating joints must be avoided because at all rotating joint, there exists friction. Frictional forces are proportional to the sliding surface area size(L^2). Weight and inertial forces, however, are proportional to volume(L^3). As the size gets smaller, the frictional force exceeds the other forces and governs the motion of system. In an extreme case, motion cannot be realized when the system is subjected to normal actuation forces.

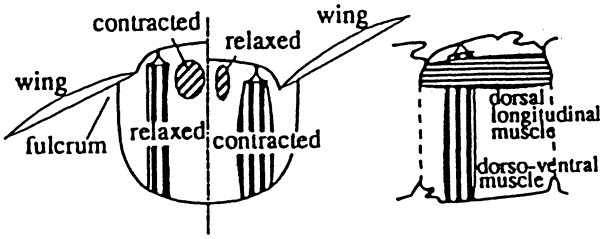


Fig.2 Cross section of an insect thorax. Distortion of the thorax causes beating of wings

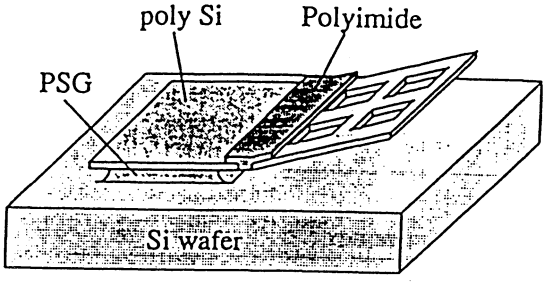


Fig. 3 Basic model of the external skeleton. Three dimensional structure is constructed by bending along the polyimide hinges.

We argue that knowledge about insects may be useful for developing microrobots. The insect has many interesting features, such as an external skeleton, elastic hinges (joints) , contracting-relaxing muscles, etc. These characteristics suggest basic design principles for microrobots.

The external skeleton of an insect consists of elastic(hard) cuticles connected by elastic (soft) hinges. In general, the elasticity of the cuticles is higher than the elasticity of the hinges.The motion of body parts like wings is based on the deformation of these elastic structures. Sliding friction does not exist there.

The cross-section of an insect (fly) thorax is shown in Fig. 2[5], illustrating the beating mechanism of the wing. The muscles are inside the skeleton, while muscles are outside the skeleton in humans.The downward movement of the wing is produced by distortion of the thorax caused by contraction of the dorsal longitudinal muscle. The upward movement results from distortion of the thorax produced by the dorso-ventral muscle.

Elasticity of the thorax plays an important role in the friction-free, high-speed wing movement. For most insects, the beating frequency coincides with the structural natural frequency of the thorax. Mechanical resonance is a good way to get large deformation with small external forces. This may be applicable for actuation in microrobots.

5 ORIGAMI STRUCTURE

Since the silicon IC process is planar, a three dimensional structure is difficult to make. The author's group is proposing a folding process like paper folding (ORIGAMI in Japanese) to make a 3D microstructure. To build an external skeleton, the author's group uses polysilicon as the rigid plate, and polyimide as the elastic joint. The basic structure of an external skeleton model is shown in Fig. 3. Polyimide is a thermosetting resin, and a hinge can be set at any angle by heating after bending it to the desirable angle. This structure can be fabricated easily by IC processes.

First, the development surface figure is made on a silicon wafer, and the folding lines are made of polyimide as shown in Fig. 3. The folding operation is done by a human with a microprobe under a microscope.

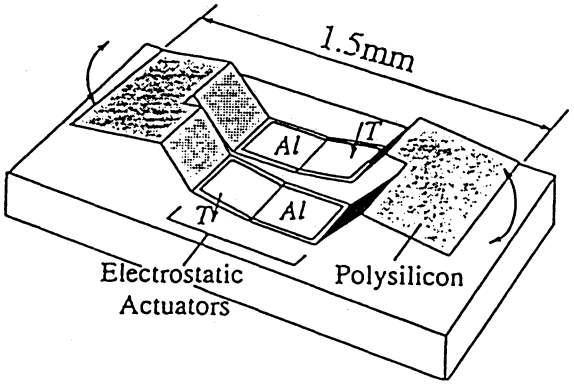


Fig. 4 Schematic figure of beating mechanism

6 MICROFLIGHT MECHANISM

The authors have developed the beating micromechanism actuated by electrostatic forces shown in Fig. 4. If an electric voltage is applied between Al plates and the base (silicon wafer), the plates move towards the base and the polysilicon wings bend up. When the frequency of the alternating voltage coincides with the natural frequency of mechanical vibration for the system, the beating amplitudes resonate. Including this example, several kinds of microflight mechanism have been developed. Magnetic force is also applicable for actuation of beating or microflight mechanisms, and a flying microrobot is now under development[6][7]. Fig. 5 shows a micro-flight mechanism[8]. It consists of three layers: polyimide, nickel, and polyimide. Nickel is sandwiched by two polyimide layers so as not to be attacked by HF while etching takes place to remove a sacrificial layer under polyimide. Both polyimide layers are spin-coated to 1 μm in thickness, and nickel is sputtered to 0.1 μm in thickness.

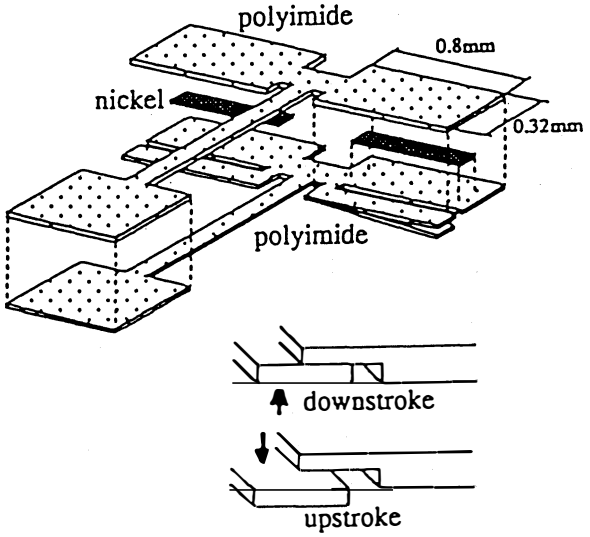


Fig. 5 Structure and dimensions of microflight mechanism

7 MICROANT-ROBOTS

In the author's laboratory a microant-robot has been developed, as shown in Fig. 6 and 7[9]. The gait for this robot is not the same as the real ant. The two middle legs kick the ground to move forward, while the other four legs touch the ground at all times to support the body. The ground vibrated with a very small amplitude(less than 1μm) and this vibration is transmitted to the middle leg through the ground. If the frequency of the vibration coincides with the natural frequency of the leg, it resonates with a large amplitude, providing the driving force by kicking the ground. The natural frequency of a leg is set by the length of the polyimide spring, so the two middle legs can have different natural frequencies.

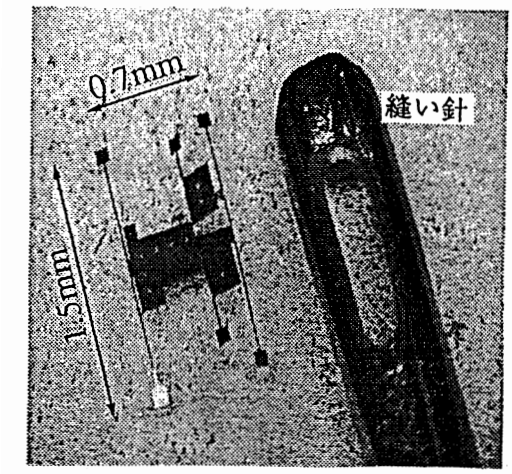


Fig. 6 Photograph of the fabricated microrobot alongside a sewing needle.

The robot can go straight, and turn to the right or to the left.

8 HYBRID INSECT ROBOT

A male silk moth pursues a female by following a pheromone. This action can be caused by only a few molecules of pheromone which arrive at the antenna of a male silk moth. A biological sensor was constructed as shown in Fig. 7. Two sensors were attached to a simple wheeled mobile robot to determined the direction of a pheromone trace. The robot followed the pheromone trace like a real male silk moth.

9 CONCLUSIONS

Several intelligent robots which have been developed in the author's laboratory have been introduced, but the author is not sure that these robots are truly intelligent because all control schemes have been constructed by humans and all computer programs have been written by humans. It can be said that they are no more than simple automatic machines which are controlled by a computer. The author is constructing a new technology --- the insect-model based microrobot --- to look for something new in robot intelligence.

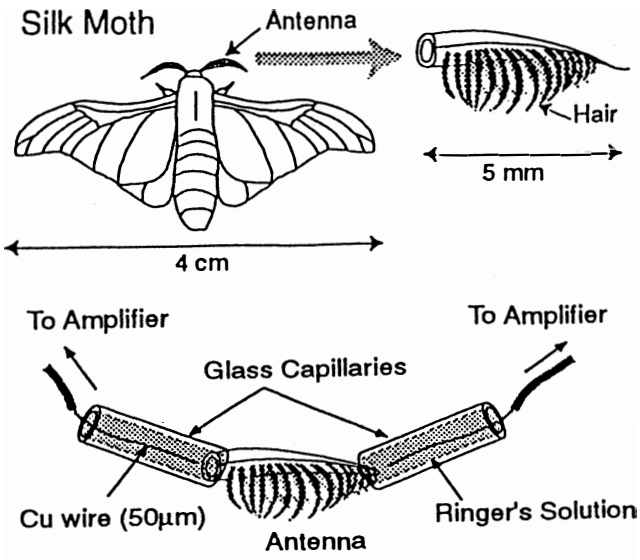


Fig. 7 Pheromone sensor

Reference

[1] Miura, H; Shimoyama, I. Robotics Research-The First International Symposium:MIT Press,1984, pp.303-325.

[2] Miura, H et al., Robotics Research-The Second International Symposium: MIT Press, 1985, pp.317-324.

[3] Shimoyama, I; Miura, H et al., Robotics Research-The Third International Symposium: MIT Press, 1986, pp.357-363.

[4] Miura, H et al., Robotics Research-The 6th International Symposium: The International Foundation for Robotics Research,1994, pp. 489-495

[5] Chapman, R. F., THE INSECTS-Structure and Function, Harvard Press, 1982, ch.XI, p. 216.

[6] Kubo, Y; Shimoyama, I; Miura, H. Proc. of IEEE International Conference on R & A, 1993, pp. 386-391.

[7] Shimoyama,I; Kubo, Y; Kaneda, T; Miura, H. Proc. of MEMS '94,1994, pp. 148-152.

[8] Shimoyama,I; Kubo, Y; Kaneda, T; Miura, H. Robotics Research-The 6th International Symposium: The International Foundation for Robotics Research, 1994, pp. 497-502.

[9] Yasuda, T. Shimoyama, I; Miura, H. Proc. of Sensors and Actuators, 1994, pp.366-370.

Artificial Societies and Generative Social Science

Joshua M. Epstein and Robert Axtell
The Brookings Institution and Santa Fe Institute
1775 Massachusetts Avenue, NW
Washington, DC 20036

Abstract

What is an *artificial society*? What can such models offer the social sciences in particular? We address these general questions, drawing brief illustrations from the specific artificial society we call "Sugarscape."

1 What is an Artificial Society?

An artificial society is a computer model consisting of (i) a population of autonomous *agents* (ii) a separate *environment* and (iii) *rules* governing the interaction of agents with one another, the interaction of agents with their environment, and the interaction of environmental sites with one another. Let us discuss each of these ingredients in turn.

1.1 Agents

Agents are the "people" of artificial societies. An agent is a data structure--in programming parlance, an "object"--that can change, or "adapt," over time. Each agent has "genetic" attributes, "cultural" attributes, and various operating rules governing its interactions with the environment and with other agents. Genetic attributes are "hard-wired," fixed for the lifetime of the agent. In Sugarscape, an agent's sex, metabolism, and vision, are genetic. Cultural attributes, by contrast, are not hard-wired; they are transmitted "vertically" from parents to children, but then change "horizontally" through contact with other agents. In Sugarscape, individual economic preferences are culturally determined--they can change as agents move around and bump into agents with different tastes. At any time the interacting agents differ in myriad ways--by age, by culture, by wealth, by vision, by economic tastes, by immunocompetence, and so forth: artificial societies are full of diversity.

1.2 Environment

Artificial social life unfolds in an environment. The Sugarscape, as the name suggests, is a landscape of generalized renewable resource (sugar) that agents like to eat; indeed they metabolize sugar and need it to live. An artificial society environment is often spatial, such as a two-dimensional lattice, but can be a more abstract--and dynamic--structure, such as the Internet. The point is that it is an external medium with which the agents interact and over which the agents "navigate."

1.3 Rules

Finally, there are rules of behavior for the agents and the environment. First, there are rules coupling every site of the environment to its neighbors, as in cellular automata. For example, the rate at which sugar regenerates at a feeding site could be a function of the sugar levels at neighboring sites. Second, there are rules coupling the agents to the environment. The simplest movement rule for Sugarscape agents is: *look around as far as your vision permits; find the site richest in sugar; go there and eat the sugar*. Of course, movement under this rule may bring the agent into contact with new neighbors, which brings us to the third set of rules, those governing interagent interactions. In Sugarscape, there are rules governing sex, combat, trade, disease transmission, and cultural transmission between neighbors.

2 Social Structures Emerge

In a typical artificial society experiment, we release an initial population of agents into the simulated environment and watch for *self-organization* into recognizable macroscopic social patterns. The formation of tribes or the emergence of certain stable wealth distributions are examples. Indeed, the defining feature of an artificial society is precisely that *fundamental social structures and group behaviors emerge from the interaction of individual agents operating in artificial environments under simple local rules—rules that place only bounded demands on each agent's information and computational capacity.* The shorthand for this is that we "grow" the collective structures "from the bottom up".

Our Sugarscape model--forthcoming on CD-ROM [1]--integrates population dynamics, migration, combat, trade, cultural transmission, genetics, environmental interactions, immunology, and epidemiology in a spatially distributed artificial society of heterogeneous adaptive agents with limited information, bounded computing capacity, evolving preferences, and other recognizably human attributes and limitations. Our broad aim is to begin the development of a unified evolutionary social science subsuming--and extending--such fields as economics and demography.

The general point, however, is that artificial societies can function as laboratories--CompuTerraria--where we "grow" fundamental social structures *in silico*, thereby *revealing* simple micro-generators of the macro-phenomena of interest. This is a central aim. As social scientists, we are presented with "already emerged" collective phenomena--such as settlement patterns, fertility rates, or wealth distributions--and we seek simple local rules that can generate them. We of course use statistics to test the match between the true, observed, structures and the ones we grow; but the ability to grow them—greatly facilitated by modern object-oriented programming—is what is new. Indeed, it holds out the prospect of a new kind of social science.

3 Generative Social Science

In particular, from an epistemological standpoint, what "sort of science" are we doing when we build artificial societies? Clearly, agent based social science does not seem to be deductive or inductive in the usual senses. But then what is it? We think *generative* is an appropriate term. The aim is to provide initial microspecifications (initial agents, environments, and rules) that are *sufficient to generate* the macrostructures of interest. We consider a given macrostructure to be "explained" by a given microspecification when the latter's generative sufficiency has been established. We interpret the question "can you explain it?" as asking "can you grow it?" In effect, we are proposing a generative program for the social sciences and see "the artificial society" as its principal scientific instrument.

References

[1] Joshua M. Epstein and Robert Axtell, *Growing Artificial Societies: Social Science from the Bottom Up*, The Brookings Institution and MIT Press, 1996

[2] On questions of validation, see Robert L. Axtell and Joshua M. Epstein, "Agent-Based Modeling: Understanding Our Creations," *The Bulletin of the Santa Fe Institute*, Winter 1994, Vol. 9, No. 2.

CO-SPONSORED BY

Asahi Agency Advertising Ltd.
Beppu Chamber Of Commerce And Industry
Chiyoda Corporation
Daikin Industries, Ltd.
Daihen Corporation
Daihen Technology Institute
Fuji Electric Co., Ltd.
Kawasumi Laboratories, Inc.
Kyudenko Corporation
Kyushu Electric Power Company Incorporated Oita Branch Office
Kyushu Oil Company, Ltd.
Matsushita Research Institute Tokyo, Inc.
Meidensha Corporation
Mitsubishi Heavy Industries, Ltd.
Mitsui Engineering Shipbuilding Co., Ltd.
Nippon Steel Corporation
Nishikawa Keisoku
Nishi Nippon Electric Wire Cable Co., Ltd.
Nisshinbo Mec Inc.
Oita Canon Inc. Information Systems Research Center
Oita Electronic Computer Center Co., Ltd.
Oita Gas Co., Ltd.
Oita Prefectual Goverment
Omron Corporation
Ricoh Company, Ltd.
Satogumi Co., Ltd.
Shintsurukai Kosan Co., Ltd.
Showa Denko K.K.
Sony Oita Corporation.
Sugahara Industry Co., Ltd.
Sumitomo Chemical Company, Limited Oita Works
Toshiba Corporation Oita Works
The Oita Chamber Of Commerce And Industry
Yanai Denki Kogyo Co., Ltd.
Yatsushika Sake-Brewing Co., Ltd.
Yoshino Plastics Corporation, Fukuoka Plant

Author Index

Abe, K. A5-3
Aoki, T. B3-4
Arai, F. B6-5
Arimoto, S. B1-1
Arthur, B. PS-3
Awa, K. B3-6
Burnham, K. J. A3-2
Casti, J. ..... OP
Choi, C. B6-2
Eom, T. D. A2-1
Epstein, J. PS-4
Fujimura, S. A4-3
Fujimura, S. A4-4
Fukuda, T. B6-5
Fukuyama, Y. B4-2
Fukuyama, Y. B4-1
Fukuyama, Y. B4-3
Furuhashi, T. B3-3
Furuhashi, T. A3-1
Garis, H. D. A5-4
Gomi, T. B1-2
Goto, S. B5-2
Hachino, T. B2-1
Ham, W. B5-6
Hanaizumi, H. A4-3
Harvey, I. A5-6
Hemmi, H. A5-2
Hiramatsu, H. A1-5
Hiraoka, N. B6-5
Hirasawa, K. B6-3
Hirasawa, K. B1-3
Hirayama, H. B4-3
Hirayama, H. B4-1
Hirayama, H. B4-2
Ho, K. H. L. A1-2
Hong, S. G. B2-3
Honma, N. A5-3
Huang, J. A4-5
Husbands, P. A5-1
Ichikawa, S. B1-6
Im, K. B5-6
Inoue, Y. B2-4
Ishida, Y. B5-5
Ishida, Y. A1-1
Ishiguro, A. C1-4
Ishiguro, A. B1-6
Ishiguro, A. B3-1
Ishii, M. B1-5
Ishii, N. B2-6
Ishikawa, H. B6-5
Ishikawa, Y. B4-1
Ishimatsu, T. B3-2
Iwakoshi, Y. A3-1
James, D. G. A3-2
Jeong, I. K. C1-2
Jeong, S. B5-6
Jinzenji, T. A1-4
Jung, M. J. B6-4
Kang, C. B3-2
Kashiwagi, H. A2-6
Kawata, Y. A2-5
Kim, J. B5-6
Kim, J. H. B6-4
Kim, S. W. B2-3
Kita, H. A2-3
Kitamura, S. A1-5
Kitamura, S. A5-5
Kiyasu, S. A4-4
Kobayashi, H. A4-2
Koga, M. B6-3
Kokaji, S. C1-1
Kondo, T. B3-1
Kuboshiki, S. B1-6
Kurematsu, Y. A1-5
Kurokawa, H. C1-1
Kyuma, K. A4-1
Kyura, N. B5-2
Langton, C. G. PS-1
Lee, J. J. A2-1
Lee, J. J. B6-2
Lee, J. J. C1-2
Lee, J. J. B2-3
Lee, J. J. B3-5
Lee, S. W. B6-4
Li, X. A4-6
Liu, Y. C1-5
Looney, C.G. A3-5
Mabuchi, T. B3-6
Matsumoto, T. A4-2

Miura, H. PS-2
Murao, H. A5-5
Murata, J. B6-3
Murata, J. B1-3
Murata, S. C1-1
Musya, T. PS-1
Nagasawa, T. B3-6
Nagata, T. C1-3
Nakamura, A. B5-1
Nakamura, M. B5-2
Nishikawa, Y. A2-3
Nonaka, T. B2-5
Nozawa, T. B4-1
Ochiai, S. B2-5
Ochiai, T. B3-2
Ogasawara, T. B5-1
Oguro, R. B2-1
Ohbayashi, M. B6-3
Ohbayashi, M. B1-3
Ohkawa, K. B5-3
Ohnishi, N. A4-5
Ohnishi, N. A1-2
Ohta, J. A4-1
Oka, T. B3-4
Okamoto, M. B2-5
Okuma, S. B3-4
Onat, A. A2-3
Onitsuka, T. C1-3
Park, K.B. B2-3
Pfeifer, R. B5-4
Rasmussen, S. PS-5
Ray, T. S. PS-2
Saita, H. B1-5
Sakata, M. A2-6
Sano, M. B1-4
Sasaki, M. A1-4
Sato, M. A5-3
Sawada, Y. B1-4
Shao, H. A3-4
Shibata, T. B5-3
Shim, H. S. B6-4
Shimohara, K. A5-2
Shiraki, K. B3-6
Suehiro, T. B5-1
Sugawara, K. B1-4
Sugie, N. A4-5
Sugisaka, M. A4-6
Sugisaka, M. B3-5
Sugisaka, M. A2-2
Sugisaka, M. A2-4
Suzuki, T. B3-4
Suzuki, Y. A1-3
Takata, H. B2-1
Takeda, H. A5-3
Tanaka, H. A3-3
Tanaka, H. A1-3
Tanaka, K. A4-2
Tanie, K. B5-3
Tominaga, D. B2-5
Tomita, K. C1-1
Tsuchiya, K. B2-4
Tsuji, T. B2-1
Tsujita, K. B2-4
Tsukune, H. B5-1
Tsumashima, T. A4-3
Tsumoto, S. A3-3
Tsumoto, S. A1-3
Uchikawa, Y. B3-3
Uchikawa, Y. A3-1
Uchikawa, Y. C1-4
Uchikawa, Y. B1-6
Uchikawa, Y. B3-1
Ueno, S. B4-4
Umeda, N. B2-1
Wang, A.P. B4-4
Wang, H. B3-2
Wang, X. A3-4
Wang, X. B3-5
Wang, Y. B2-6
Watanabe, Y. C1-4
Watanabe, Y. B3-1
We, K.L. B4-1
Yagi, T. A4-2
Yamada, T. B3-6
Yamakawa, T. B6-1
Yamamoto, H. B2-2
Yao, X. C1-5
Yashima, Y. A2-5
Yoshikawa, T. B3-3
Zha, H. B. C1-3
Zhang, Y. G. A4-6

Artificial Realization of Human Skill for Robot Manipulator Control

Masatoshi Nakamura and Satoru Goto
Department of Electrical Engineering
Saga University
Honjomachi, Saga 840, Japan

Nobuhiro Kyura
Research Laboratory
Yaskawa Electric Co., Ltd.
2-1 Kurosakishiroishi, Yahatanishi-ku
Kitakyushu 806, Japan

Abstract

A methodology for artificial realization of the human skill of the experts was described. The artificial human skill was realized in the problem of contour control of mechatronic servo systems including robot manipulators and machine tools. The merits of the artificial human skill thus obtained were discussed.

1 Introduction

Through the collaboration works with the researchers of university and industry, the authors have theoretically solved many problems which had been encountered in the actual field of industry. Key ideas for the solution to the problems were intuitively instructed by the experts who have practiced in the working field and dealt with those problems. Those human skills were mathematically analyzed and then the solutions to the problems were attained through the long term collaboration work. This paper proposed a methodology for artificial realization of human skill. The modified taught data method for mechatronic servo system [1, 2] was interpreted underneath as a successful work of the artificial realization methodology.

2 Methodology for Artificial Realization of Human Skill

Experts in the industry are engaged in a job for a long time. Their jobs are skillful based on the know-how of the specific field. Their skills are mainly gained through a cut and try technique and intuitive manner (see Fig. 1). Only the final technique of human skill is visible and recognizable to others but not the procedure behind the development. Moreover, the experts themselves can rarely explain the procedure how to attain the skillful techniques. In order to realize the human skill artificially by use of the mathematical tools, clarification of the procedure of the human skill is crucial. The procedure for the artificial realization of human skill is summarized as follows:

1. Understanding of objective and visible human skill

   It is required for the designer to understand the objective of the problem and the causes of the existing problems clearly as the same level as the expert through the intimate discussion with the experts or by the experimental tests as to work out the problem. The designer should also learn the human skill through the experts.

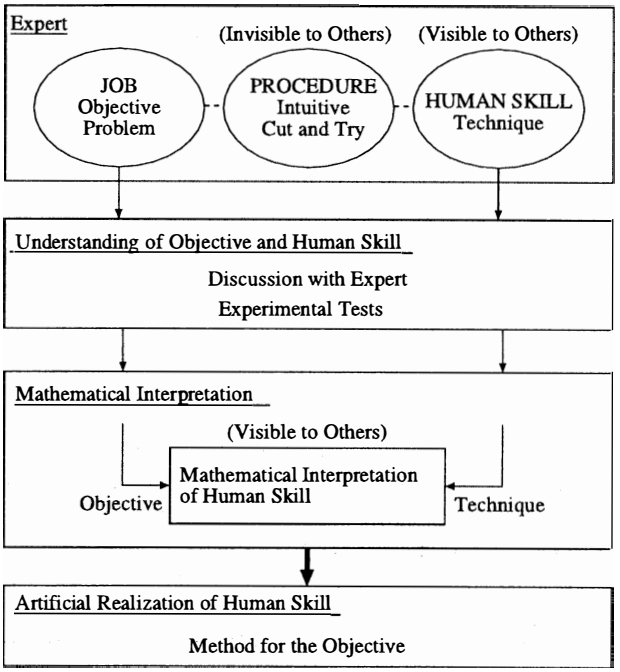


Figure 1: Methodology for artificial realization of human skill

2. Mathematical interpretation

   The procedure of the human skill is clarified through the understanding of the objective and human skill. Thus kind of task can be understood as an identification problem of the invisible procedure of the human skill and can be referred to the mathematical interpretation of human skill.

3. Artificial realization of the human skill

   If the cause of the problem is clarified mathematically, the realization of the human skill by use of mathematical tools is rather straightforward through the selection and the development of the mathematical tool box, such as signal processing, system identification, control theory and so on. The artificial realization of the human skill thus obtained should be discriminated from a mere imitation of the human skill.

3 Artificial Realization of Human Skill in Mechatronic Servo System

3.1 Mechatronic servo system

Many kinds of mechatronic servo systems such as industrial robot manipulators and NC (numerical control) machine tools are used in actual production lines and assembly lines, etc. The industrial mechatronic servo systems are constructed by the mechanical connection of the servo systems for each axis and the servo systems are controlled for each axis, independently. The angle and the angular velocity of the motors of the mechatronic servo systems are measured by using sensors such as potentiometers, tachogenerators and encoders which are attached to the motors. These motors are controlled by servo controllers based on the measurement data of the motors. Thus, the servo controllers of the industrial mechatronic servo systems do not directly control the position and the velocity of the mechanism but they control the angle and the angular velocity of the motors. Usually, the position loop and the velocity loop of the constant feedback gain of the servo controllers are constructed. The control performance of the mechatronic servo systems mainly depend on the servo systems for each axis.

3.2 Artificial realization of human skill for contour control of mechatronic servo systems

The artificial human skill for mechatronic servo systems such as robot manipulators is realized following the methodology described in the previous section:

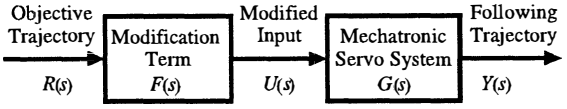


Figure 2: Block diagram of modified input data for mechatronic servo systems

1. Understanding of the objective and visible human skill

   The objective of the contour control of the mechatronic servo systems is that the 'following trajectory' of the end effector of the system exactly follows the 'objective trajectory.' In the industrial applications, working speed of the mechatronic servo systems is also required to be high for a higher working efficiency. However, at high speed operation of the mechatronic servo systems, the contour control performance of the mechatronic servo systems often deteriorates in the following trajectory. One technical expert in the industry had found a technique for modifying the input signal to the servo system of the robot manipulator to achieve an accurate following trajectory for a given objective trajectory. The length of the extension of the modified input signal had been determined appropriately in the trial and error manner.

2. Mathematical interpretation

   The dynamics of the mechatronic servo system is represented by the mathematical model of a linear dynamical equation. The main reason why the deterioration at the high speed operation occurs is found to be the dynamical delay of the servo motors of each axis in the mechatronic servo systems. The human skill can be understood as a method for modifying the input signal from the objective trajectory to compensate the delay of the dynamics.

3. Artificial realization of the human skill

   Figure 2 shows the block diagram of the modified input signal for contour control of each axis of the mechatronic servo system. As shown in Fig. 2, the dynamics of the mechatronic servo system is described for each axis independently as

$$Y(s) = G(s)U(s) \tag{1}$$

   where $U(s)$ is taught data for the input trajectory, $Y(s)$ is the following trajectory of the mechatronic

servo system and $G(s)$ is the transfer function of the mechatronic servo system. The taught data $U(s)$ are modified from the objective trajectory $R(s)$ through a modification term $F(s)$, i.e., the taught data $U(s)$ can be expressed as

$$U(s) = F(s)R(s). \tag{2}$$

The modification term $F(s)$ should be designed based on the mathematical model $G(s)$ of the mechatronic servo system.

The modification of the input signal is accomplished in two ways, i. e., the first one is based on the pole assignment regulator of open loop type, and the second one is based on the Gaussian neural network. The realization of the methods is described in the following section.

4 Methods of Modification of Input Data

4.1 Mathematical model of mechatronic servo system

The mechatronic servo systems are operated under the working condition such that the defection of the disturbance such as the reaction force of the mechanism and the influence of the composed axis is ignored. Under the above condition, the dynamics of the mechatronic servo system can be treated independently for each axis and the problems of the control performance of the mechatronic servo system can be analyzed for each axis, independently.

The mechatronic servo system of the each axis is described as a simple first order model [3]

$$G(s) = \frac{K_p}{s + K_p} \tag{3}$$

where K_p is the known position loop gain. For general working condition of the industrial mechatronic servo systems where the speed of the motors are $1/20 \sim 1/100$ times of a rated speed, the model (3) well describes the input and output relationship of the mechatronic servo system including its dynamics and mechanical construction. The model (3) also implies the dynamics of the industrial articulated robot manipulators within the linearizable region [4].

4.2 Pole assignment regulator of open loop type

The modification term $F(s)$ in (2) is constructed by using the pole assignment regulator of open loop type based on the state space representation of the mechatronic servo system (3). The modification term $F(s)$ is derived [1] as

$$F(s) = -\frac{\gamma(s + K_p)}{K_p(s - \gamma)} \tag{4}$$

where γ is the pole of the regulator.

The allowable limitation of the servo motor speed incorporated in the actuators must be taken into account to use the modified taught data method for actual mechatronic servo systems. The restriction is given as

$$|K_p(u(t) - y(t))| \leq V_{max} \tag{5}$$

where V_{max} is the maximum speed of the servo system.

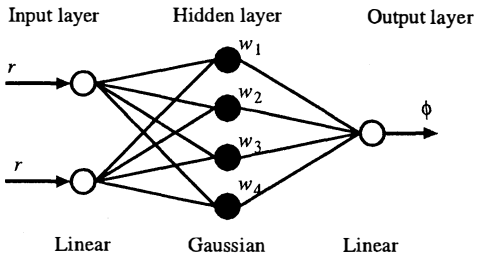


Figure 3: Construction of Gaussian network

4.3 Gaussian network

The modification term $F(s)$ can be determined by use of another way in which the Gaussian neural network is adopted [2]. Figure 3 shows the structure of a three-layer Gaussian network composed of two-input, one-output, and one hidden layer with four Gaussian units, which is used for the construction of the inverse dynamics of the mechatronic servo system (3).

Two-input one-output Gaussian network is defined by

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{4} w_i \psi_i(\boldsymbol{x}) \tag{6}$$

where $\phi(\boldsymbol{x})$ is the network output, $\boldsymbol{x} = (x_1, x_2)^T = (r, \dot{r})^T$ is the input vector, w_i is the weight of ith unit. The function $\psi_i(\boldsymbol{x})$ is a Gaussian unit defined by

$$\psi_i(\boldsymbol{x}) = \exp\left(\sum_{j=1}^{2}\sum_{k=1}^{2} \frac{h_{jk}^i}{\sigma_j^i \sigma_k^i}(x_j - m_j^i)(x_k - m_k^i)/2\right) \tag{7}$$

where m_j^i is the jth element of the mean, σ_j^i is the marginal standard deviation, h_{jk}^i is the correlation coefficient and $h_{jk}^i = 1 \; if \; j = k$ and $|h_{jk}^i| \leq$

1 *otherwise* [5]. The parameters $(m_j^i, \sigma_j^i, h_{jk}^i)$ are trained based on actual data for operation of the mechatronic servo system by use of the Back Propagation algorithm [6]. The parameters in the Gaussian network are updated for each sampled teaching pattern and the algorithm stops when the parameters converge to constant values. After the training of the network, the neural network realizes the actual inverse dynamics of the mechatronic servo system and is adopted as the modification term for the mechatronic servo system.

5 Experimental Result

5.1 Pole assignment regulator of open loop type

The effectiveness of the proposed method of modified input based on the pole assignment regulator was assured by applying them to an industrial articulated robot manipulator. For the industrial articulated robot manipulator, the dynamics can be approximated for each axis in the working coordinates, independently [4]. Hence, the proposed method was applied to each axis in the working coordinates of the articulated robot manipulator, independently. The position loop gain for each axis was same as $K_p = 15[1/s]$.

Figure 4 illustrates the experimental results of the industrial robot manipulator by using the conventional method, which means the objective trajectory was used as the input signal, and the proposed method of modified input based on the pole assignment regulator of open loop type. The trajectory of the conventional method deteriorates roundly at the corner because of the time delay of the mechatronic servo system. However, the trajectory of the proposed method of modified input based on the pole assignment regulator has sharp edge corner. The results show the effectiveness of the proposed method of modified input based on the pole assignment regulator of open loop type.

5.2 Gaussian network

The secondly proposed method of modified input based on the Gaussian network was applied to the X-Y table which had 2 degrees of freedom in the movement and the Y axis is placed on the X axis. The position loop gain for each axis was same as $K_p = 5\ [1/s]$. The proposed method was applied for X-axis and Y-axis of the X-Y table, independently. The Gaussian network

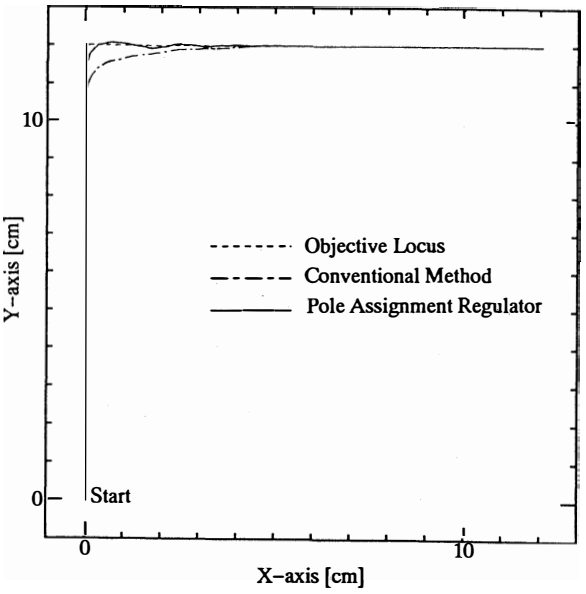


Figure 4: Experimental results based on the pole assignment regulator of open loop type for the articulated robot manipulator

was learned by using the teaching pattern of the actual data of the X-Y table. The teaching pattern was adopted 100 times for the learning of the parameters in the Gaussian network.

Figure 5 illustrates the experimental results of the conventional method and the proposed method of modified input based on the Gaussian network. As shown in Fig. 5, the maximum position error of the conventional method is 0.5[mm] and that of the proposed method is 0.1[mm]. The result of the proposed method is significantly better than that of the conventional method. The results prove that the proposed method compensates the delay of the X-Y table motion appropriately by using the modification term of the Gaussian network.

5.3 Discussion

The merits of the artificial human skill for the contour control of the mechatronic servo systems are summarized as follows:

a. Generalization of the human skill

The expert's technique must adjust the modification value for each objective trajectory, however, the proposed method of the artificial human skill need not require such adjustment. The expert's technique is only applicable to the simple rectangular objective trajectories. However, the artifi-

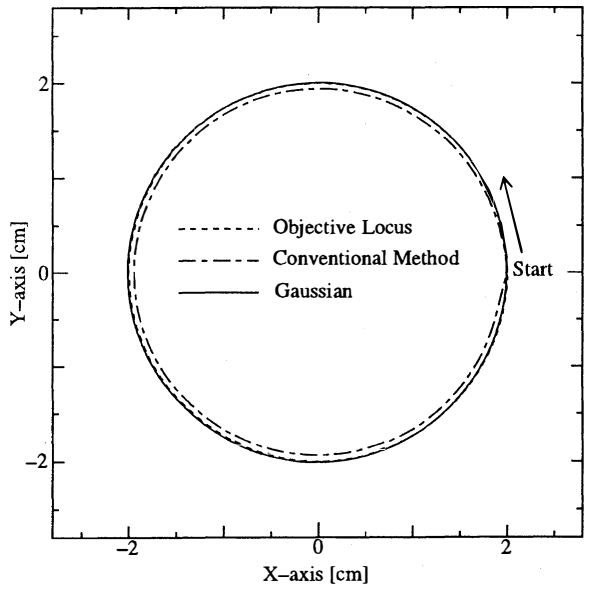


Figure 5: Experimental results based on the Gaussian network for X-Y table

cial human skill can be applicable to any shape of trajectories. The original human skill was developed for the specific industrial robot manipulators, however, the artificial human skill can be applied to any kind of mechatronic servo systems.

b. Acceptability in industrial field

The method for the modification of the input data is similar to the expert's technique and the modification term can be automatically determined if the value of K_p in (3) or the actual input/output data of the system are given. Human skill of the experts can be replaced by the artificial human skill.

c. Objective evaluation

The performance of the methods of the artificial human skill for modification of input data is evaluated theoretically because the mechatronic servo system is described by the mathematical equations of (1).

The both of the proposed methods of the artificial human skill have their own merits. The first method based on the pole assignment regulator is simple and easy to calculate the modified input signal, however, the method requires the exact knowledge of the model for the mechatronic servo systems. The second method based on the Gaussian network is adaptable for any system without a priori information of the model because of the learning of the Gaussian network. However, the designing procedure of the modification term is laborious compared with the first method based on the pole assignment. In both proposed methods, the first order model (3) for the mechatronic servo system is not crucial, but the higher order model of the system can be used for the designing procedure.

6 Conclusion

A methodology for artificial realization of human skill of the experts was investigated. The methodology was realized in the contour control of the mechatronic servo systems. The two methods of modified input for the contour control of the mechatronic servo systems were proposed. In order to realize the methodology the following points are required for the designers: 1) exact understanding of the objective and the human skill, 2) long and intimate discussion with collaborators, 3) good human relationship between collaborators.

References

[1] S. Goto, M. Nakamura and N. Kyura, "Method for Modifying Taught Data for Accurate High Speed Positioning of Robot Arm," *SICE Trans.*, **27**-12, pp. 1396-1404, 1991 (in Japanese).

[2] S. Goto, M. Nakamura and N. Kyura, "Accurate Contour Control of Mechatronic Servo Systems by Using Gaussian Networks," *IEEE Industrial Electronics*, 1996 to be appeared.

[3] Yaskawa Electric Mfg. Co., Ltd., *Introduction to Servo Techniques for Mechatronics*, Nikkankogyo-shimbunsha 1986 (in Japanese).

[4] S. Goto, M. Nakamura and N. Kyura, "Propriety of Linear Model of Servo System for Industrial Articulated Robot Arms and Evaluation of Its Linearization Error," *ISCIE Trans.*, **7**-3, pp. 103-112, 1994 (in Japanese).

[5] S. Lee and R. M. Kil, "A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning," *Neural Networks*, **4**, pp. 207-224, 1991.

[6] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Network: Perceptron, Madaline, and Backpropagation, in C. Lau (Ed.), *Neural Networks*, Part 2, pp. 27-53, New York, IEEE Press, 1992.

Computational Analysis of Acquired Dyslexia of Kanji Characters based on Neural Networks

Shusaku Tsumoto and Hiroshi Tanaka
Dept. of Information Medicine, Medical Research Institute, Tokyo Medical and Dental University
1-5-45 Yushima, Bunkyo-ku Tokyo 113 Japan
TEL: +81-3-5803-5840 FAX: +81-3-5803-0247
E-mail:tsumocom@cmn.tmd.ac.jp, tanaka@cim.tmd.ac.jp

Abstract

Acquired dyslexia of Kanji characters is one of the most interesting research areas in the neurophysiological field. Although dyslexia of alphabets is known as the malfunction of gyrus angularis (classical hypothesis), reading and writing of Chinese characters (Kanji characters) are intact in Japanese patients who suffered from cerebral infarction of gyrus angularis. Thus, it has been pointed out that another group of neurons, which functions as integration of shape information, should play an important role in recognition of Kanji characters (Iwata's model). In this paper, two neural network models, based on classical hypothesis and Iwata's model, are introduced to examine the validity of these two hypotheses. The computational experiments gives the following three results: first, Iwata's model learns Kanji characters much faster than classical model. Second, Iwata's model is more robust with respect to the malfunction of neurons. Finally, third, Iwata's model simulates the characteristics of two types of Japanese dyslexia.

Keywords: Acquired Dyslexia, Kanji characters, Neural Modeling

1 Introduction

The Japanese language uses three kinds of characters, Hiragana, Katakana, and Kanji. The former two sets of characters are used as syllabogram, each of which corresponds to a Japanese syllable. For example, "い" and "イ" corresponds to a syllable, "i". The main difference between Hiragana and Katakana is that the former set is used to represent ordinary sentences, while the latter one is used to describe loan words, such as "ラジオ", which is borrowed from a English word, "radio".

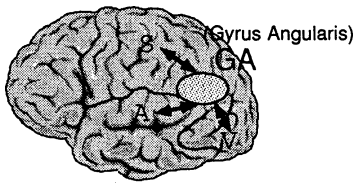
On the other hand, Kanji is used as the combination of morphogram and syllabogram. For example, "犬" means a dog, which corresponds to "いぬ", represented by Hiragana. The reason why we use Kanji characters is that Hiragana or Katakana characters are too weak to represent the meanings of Japanese words. Since Japanese syllables are restricted to only 50 kinds, the Japanese language has many homonyms. For example, "いぬ" has the following two meanings: one is a dog, and the other one is a verb, "leave". Japanese uses Kanji characters to differentiate these two meanings: "犬" and "去ぬ". In this way, Hiragana and Kanji characters have different functions, which causes acquired dyslexia of the Japanese language to be more difficult than other languages.

Although dyslexia of alphabets is known as the malfunction of gyrus angularis (classical hypothesis)[2], reading and writing of Chinese characters (Kanji characters) are intact in Japanese patients who suffered from cerebral infarction of gyrus angularis. Thus, it has been pointed out that another group of neurons, which functions as integration of shape information, should play an important role in recognition of Kanji characters (Iwata's model)[3].

In this paper, two neural network models, which corresponds to classical hypothesis and Iwata's hypothesis, are introduced to examine the validity of these hypotheses. The computational experiments gives the following three results: first, Iwata's model learns Kanji characters much faster than classical model. Second, Iwata's model is more robust with respect to the malfunction of neurons. Finally, third, Iwata's model simulates the characteristics of two types of Japanese dyslexia syndromes.

2 Acquired Dyslexia

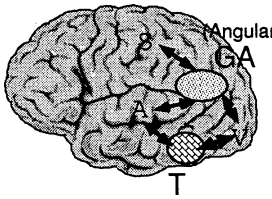
Classical Hypothesis: Classical hypothesis for reading and writing characters is illustrated in Fig. 1, where S, A, V and GA denotes Somatosensory area, Auditory Language area, Visual cortex, and Gyrus Angularis, respectively. In reading process, first, in-



Reading: V→GA→A

Writing : A→GA→S

Figure 1: Classic Hypothesis for Pure Dyslexia



Kanji Reading: V→GA→A (Phonological)
V→T→A (Semantic)

Kanji Writing : A→GA→S
A→T→V→GA→S

Figure 2: Hypothesis for Japanese Dyslexia

formation on characters is captured by Visual cortex. Then, the obtained information is transmitted to Gyrus Angularis, which integrates all the information of characters. Finally, integrated information is sent to auditory language area, which is closely connected with speaking languages. This hypothesis is proposed by Geschwind[2], which explains clinical cases on pure dyslexia reported by Dejerine[1] very well.

However, it is notable that this hypothesis is introduced in order to explain pure dyslexia with respect to alphabets, a kind of syllabogram. Recently, it is reported that this hypothesis does not explain dyslexia syndromes in Japanese whose patients cannot read Hiragana but can read Kanji characters, or cannot read Kanji characters but cannot read Hiragana[3].

Hypothesis for Japanese Language: Iwata proposes a new hypothesis for Japanese reading and writing process, based on clinical cases on Japanese dyslexia syndromes[3]. Close clinicopathological examinations show that left posterior inferior temporal gyrus(T) is closely involved with integrating semantic information on Kanji characters. That is, a patient whose T region is damaged can read Kanji characters and Hiragana, but cannot understand these meanings, but a patient whose GA region is damages can read Kanji characters and understand these meanings, but cannot read Hiragana. Thus, for reading Kanji characters, two pathways should be considered, which is illustrated in Fig. 2. As to the phonological aspects, the reading process is the same as that in Hiragana. On the other hand, concerning the semantic aspects, visual information is transmitted from visual cortex to T region. Then, integrated information is sent from T region to auditory language area ($V \rightarrow T \rightarrow A$).

Dyslexia of Kanji Characters: In dyslexia of Chinese characters, the following four types of syndromes are reported. The first one is **disorder of capturing global structure**, which means that a patient can only read some part of Chinese characters. For example, when "青"(blue) is shown, a patient misread it as "月"(moon), which is only a lower part of "青". This syndrome can be viewed as damage of integrating partial information on Kanji characters.

The second one is **confusion of morphology**, whose patient confuses a Kanji character with a character of the similar shape. For example, when "鍵" is shown, a patient misread it as "銀"(silver), whose left part is the same as that of the character "鍵"(key). This syndrome can be viewed as damage of integrating shape information on Kanji characters.

The third one is **semantic misleading**, where a patient misread a character as one whose meaning is very similar, although each morphological characteristics are not similar. For example, when "馬"(horse) is shown, a patient read it as "犬"(dog). The sharing characteristic of both characters is that they describe animals. This syndrome can be viewed as damage of integrating semantic information.

Finally, the fourth one is **disorder of choice of auditory representation**. In this syndrome, a patient makes an error in choosing phonological representation of Kanji characters. For example, in "言葉", "言" should be read as "こと" (koto), but a patient read it as "げん" (gen). This syndrome can be viewed as damage of integrating phonological information. Thus, these types suggest that dyslexia of Kanji characters is observed when neurons integrating information on Kanji characters are damaged.

3 Computational Model

Computational models are based on three-layer neural networks, whose learning algorithms are backpropagation[4].

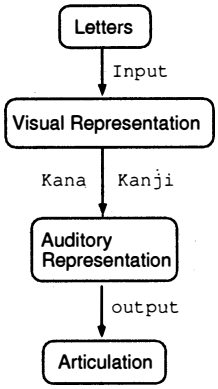


Figure 3: Classical Model

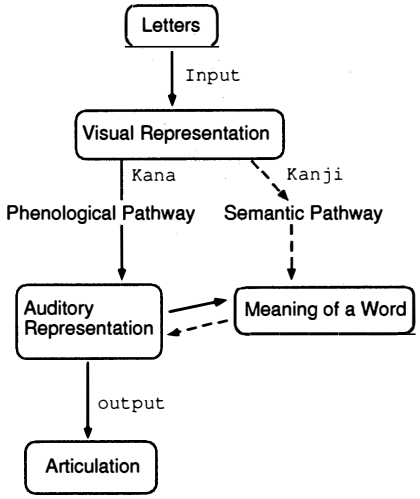


Figure 4: Iwata's Model

For simplicity, we use the following ten Kanji characters for inputs: 木 (tree), 林 (grove), 森 (forest), 桜 (cherry tree), 桃 (peach tree), 松 (pine tree), 梅 (plum tree), 橋 (bridge), 柱 (pillar), 竹 (bamboo)", all of which are related with "wood". Each character is represented by a 16 to 16 square dot matrix. Furthermore, to make a phonological representation of each Kanji characters, we also use 16 Kana characters.

For outputs, each category is set to each Kanji characters. Thus, four bit strings is used to represent each category, which corresponds to the meaning of each character. Furthermore, for simplicity, auditory representation is set to the same representation as the above category. That is, four bit strings is also used to represent pronunciation of each Kanji characters, and constrains that this representation and semantic representation is the same. This is a kind of simplification of interaction between auditory representation and semantic representation.

Using these representations, we introduce two computational models to examine the effect of damage of neurons on dyslexia syndromes as follows.

Classical Model: A computational model based on classical hypothesis is illustrated by Fig. 3. This model consists of one large component of neural networks. Visual representation corresponds to the input layer. Then, intermediate layer, corresponding to GA region, are composed of 100 neurons. Finally, auditory representation correspond to the output layer, respectively. In each learning step, we give one Kanji character and its corresponding Kana representation. For example, when "木" is input, its corresponding Kana representation "き"(ki) is given as an answer. For each learning step, the output of neural network is compared with its answer, and total calculus is terminated after the total error rate is saturated.

Iwata's Model: A computational model based on Iwata's model is given in Fig. 4. This model consists of the following two neural networks: one describes phonological pathway and the other represents semantic pathway. For both models, visual representation corresponds to an input layer. Then, phonological pathway and semantic pathway are described by intermediate layers, each of which are composed of 50 neurons. Finally, auditory representation and meaning of a word correspond to the former and the latter output layers, respectively. It is assumed that there is no interaction between phonological pathway and semantic pathway in this model.

In each learning step, one Kanji character is input with its corresponding Kana representation and its category. For example, when a character "木" is input for semantic pathway, its corresponding Kana representation "き"(ki) is given as a input for phonological pathway and its category "1" is given as an answer. For each learning step, the output of neural network is compared with its answer, and total calculus is terminated after the total error rate is saturated.

Focal Lesioning: After learning steps, we make the following "focal lesioning" steps for each model in order to simulate the effect of damaged neurons on misreading. First, one neuron is randomly selected from the neural networks. And then, its neighbors (within k

Table 1: Learning Steps Needed for Termination

Model	Learning Steps	Error Rate
Classical Model	66700 ± 1319	87.5 ± 8.5 (%)
Iwata's Model	56400 ± 795	89.2 ± 9.2 (%)
p-value	0.00947	0.341

neurons) are set to the malfunctioned neurons. Then, we examine misclassification rate of this damaged network by using the above ten Kanji characters for each model. These two steps are iterated for 100 steps, and then statistical significance is tested by t-test with respect to the error rate obtained.

In these procedures, first k is set to 1, and increased with step one. Then the above test-procedures are repeated until the misclassification rate is equal to 1.0.

4 Experimental Results

We make the following two experiments using the aforementioned computational model. First, we compare learning steps needed for termination for each computational model. For this experiment, we change the order of Kanji characters, and start learning steps. These steps are repeated for 1000 times, and the obtained steps are averaged, and tested using t-test. Second, we compare the behavior of each network using focal lesioning with respect to a parameter k. From these experiments, the following three interesting results are obtained.

Learning Steps for Termination: The statistics of this experiments are shown in Table 1. The convergence of Iwata's model is higher than that of classical model, which is statistically significant with $p < 0.01$. This result suggests that processing of Kana and Kanji should be separated.

Behavior of Neural Networks: Phase-transition behavior is observed with respect to k for each model: when k is larger than the critical value, the error rate of a model suddenly becomes near 1.0 (Fig. 5). The main difference between two models is that Iwata's model is more robust than classical model.

Simulated Dyslexia Syndromes: The computational model based on Iwata's model can simulate two types of dyslexia: disorder of capturing global structure and confusion of morphology. For example, "森" is almost always misread as "木" when k is larger than

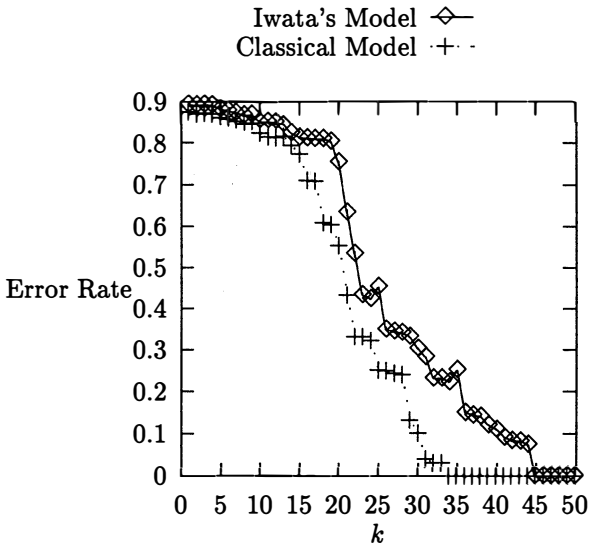


Figure 5: Effect of Parameter k on Error Rate

30, and "桜" is misread as "桃" when k is larger than 20. Thus, those results, suggests that the former type of dyslexia is usually observed when the focus of infarction is very large.

In summary, these three results suggest that Iwata's model explains dylexia syndromes better than classical one in the case of the Japanese language. However, this computational model can only explain two types of Japanese dyslexia syndromes, which suggest that our model is weak. Thus, it is our future work to refine a computational model in order to explain other two types of dyslexia syndromes.

5 Conclusion

In this paper, two computational models are introduced to explain Japanese dyslexia syndromes. The experimental results show that Iwata's model explain dylexia syndromes better than classical model, although the computational model is a little weak to explain the whole aspects of acquired Japanese dyslexia.

References

[1] Dejerine, J. Sur un cas de cécité verbale avec agraphie, suivi d'autopsie. *CR. Soc. Biol.*, **9**, 197-201, 1891.

[2] Geschwind, N. Disconnexion syndromes in animals and man. *Brain*, **88**, 237-294, 585-644, 1965.

[3] Iwata, M. Neural mechanism of reading and writing in the Japanese language. *Functional Neurology*, **1**, 43-52, 1986.

[4] McClelland, J., Rumelhart, D., et al.(eds.) *Parallel Distributed Processing*, MIT press, 1986.

Human-Type Control Using Supervisory Controller

Tae-dok Eom and Ju-Jang Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373–1 Kusong-dong Yusong-gu
Taejon 305–701 Korea
email: jjlee@ee.kaist.ac.kr

Abstract

Modeled from human neurons, various types of artificial neurons are developed and applied to control algorithm. As in the case that a man is trying to learn a skill through training his own neurons, a teacher is needed to suggest the artificial neural network a reference signal and prevent it form getting unstable. In this paper, supervisory control algorithm is devised to meet these demands. It guarantees stability in the sense of the boundedness of tracking error and additionaly provides the thinning effect of network input dimension so the reference signal is obtained. Simulation is also performed for simple two-link robot to show the validity of the suggested algorithm.

1 Introduction

Since the artificial neural network was born as a result of investigating and modeling of human neurons, it has found wide applications due to the function approximation capability. Related to the control theory, many researchers have tried to find the efficient way of using neural network for control algorithm. Despite the generic defects neural network bears, flourishing of this field stems from a certain belief that the perfect mimicry of human behavior will come true someday. Though the modern technical improvement enables us to precisely control the systems of severe nonlinearity, no one doubts there are kinds of works in which man is superior to the automatic controller. What we pursue through the study of neural network is to devise the human-type controller that replaces the functions which have been thought of man's own.

Just as the various kinds of human neurons in charge of various functions in different parts, there are many kinds of artificial neural network. Each has the prominent figures and own applications. If we want to design a human-type controller that has man's behavioral characteristics, it may be one promising way to combine various kinds of neural networks and construct the whole system. Recently we suggested a construction paradigm[1] which employes feedforward neural network, associative memory and chaotic neuron as its components. That is based on the conjecture that these neurons take important parts in man's information processing routine. In fact, associative reasoning and chaotic phenomenon is the nature of, what we call, the human.

In this paper, the supervisory control algorithm which takes the role of 'teacher' is introduced. It provides stability during the learning period of neural network and extracts the reference signal - the function of time - which should be approximated. This coincides with man's behavior of adjusting proper control output as the function of time when he tries to learn a skill.

We designed supervisory control algorithm for robot dynamics to make an articulated robot which learns and acts like a human. For the robot dynamics many researchers tried to apply neural network in identifying the function in the dynamics [2] [3]. But in this case of the robot dynamics the problem of stability is serious and most of the researches use the way of robust control to stabilize the dynamics. The supervisory control in this paper is adjusted parallel to the neuro-controller, so it makes the system stable and enables neural network to reduce the input dimension. The primitive idea of the supervisory control is initially proposed by Wang [4]. He developed it to stabilize the fuzzy adaptive controller. His supervisory control algorithm is only useful for the single input system but we extended it to be valid for the multi-input system so it becomes applicable to the robot dynamics.

2 Supervisory Control Algorithm for Robot Dynamics

The robot dynamics can be described by the following equation.

$$M(q)\ddot{q} + N(q,\dot{q}) = \tau \tag{1}$$

where $M(q)$ is an inertia matrix and $N(q,\dot{q})$ includes centrifugal, Coriolis, and gravitational force. To apply the supervisory control algorithm, the following assumptions should be valid for the dynamics described by equation (1).

Assumption 1 *Inertia matrix $M(q)$ is known, i.e., all uncertainties belong to $N(q,\dot{q})$.*

Assumption 2 *The state vector $q, \dot{q}$ in (1) is measurable.*

Besides the assumptions above, robot dynamics has following property.

Property 1 In robot dynamics described by (1), there exist $c_0, c_1 > 0$ such that

$$N(q,\dot{q}) \leq c_0 + c_1\|\dot{q}\|^2. \tag{2}$$

Define $F(q,\dot{q}) = -M^{-1}(q)N(q,\dot{q})$ and $u = M^{-1}\tau$, then the equation (1) is changed to,

$$\ddot{q} = F(q,\dot{q}) + u. \tag{3}$$

To induce the tracking error dynamics, define q_d is the desired trajectory, and let the error vector $q_e = q - q_d$, then the robot system can be controlled with the following control input.

$$u^* = -F(q,\dot{q}) + \ddot{q}_d - Ke \tag{4}$$

where $e^T = [q_e^T \; \dot{q}_e^T]$ is the new error vector and $K = [K_1 \; K_2]$ is chosen to make the matrix

$$\begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix} \tag{5}$$

has the eigenvalues in the left half plane. Applying (4) to (3) renders the system asymptotically stable, but this control input can not be implemented since $F(q,\dot{q})$ in u^* is unknown. This impractical input, u^*, just helps the conversion of (3) to tracking error dynamics. Now, Suppose that the input, u, is the addition of the neural network controller, u_c, and the supervisory controller, u_s, that is

$$u = u_c + u_s \tag{6}$$

Substituting (6) to (3), (3) becomes

$$\ddot{q} = F(q,\dot{q}) + (u_c + u_s). \tag{7}$$

By subtraction and addition of u^*, the equation of error is obtained:

$$\ddot{q}_e = -Ke + (u_c + u_s - u^*) \tag{8}$$

or equivalently in the other vector form,

$$\dot{e} = \Lambda e + B(u_c + u_s - u^*) \tag{9}$$

where

$$\Lambda = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{10}$$

Define $V = \frac{1}{2}e^T P e$ where P is a symmetric positive definite matrix satisfying the following Lyapunov equation.

$$\Lambda^T P + P\Lambda = -Q \tag{11}$$

where Q is a positive definite matrix.

Now, the following theorem is derived based on the Lyapunov's direct method.

Theorem 1 *Consider the robot dynamics described by (1), satisfying* Assumption 1 – 2, *and subject to control given by*

$$\tau = \begin{cases} M(q)u_c \; if \; V < V_M \\ M(q)u_s \; if \; V > V_M \end{cases} \tag{12}$$

and

$$u_s \equiv \ddot{q}_d - Ke + u_{s1} \tag{13}$$

$$u_{s1} \equiv -\frac{(e^T PB)^T}{\|e^T PB\|} \cdot \|M\| \cdot (c_0 + c_1\|\dot{q}\|^2) \tag{14}$$

where $V_M > 0$ is a constant specified by the designer. Then, $V < V_M$ as $t \to \infty$.

Proof when $V > V_M$,

$$\begin{aligned} \dot{V} &= -\frac{1}{2}e^T Q e + e^T PB(u - u^*) \\ &= -\frac{1}{2}e^T Q e + e^T PB(u_{s1} + F(q,\dot{q})) \\ &\leq -\frac{1}{2}e^T Q e + \|e^T PB\|\|F(q,\dot{q})\| \\ &\quad + e^T PBu_{s1} \\ &\leq -\frac{1}{2}e^T Q e \\ &\quad + \|e^T PB\|\|M\|(\|N\| - (c_0 + c_1\|\dot{q}\|^2)) \\ &< 0 \end{aligned} \tag{15}$$

□

3 Identification and Control Procedure

Instead of the infeasible control input u^* of (4), the controller which consists of neural network identifier and PD-type controller is used in our identification procedure:

$$u_c = -N_F(q, \dot{q}) + \ddot{q}_d - Ke. \tag{16}$$

where N_F takes the place of F in (4). Of course any kind of control scheme is available for the candidate of u_c, controller of (16) with simple FNN in it is chosen in this paper. It is also obvious that with the help of supervisory controller, u_s, the system controlled by u_c does not lose stability even in the case the shape of N_F is far from that of F.

Another merit of using supervisory controller is the fact that neural network is trained within the specified region about the desired trajectory. The identification along the desired trajectory is very efficient for the task of following the desired trajectory.

As iteration goes over and over, N_F is trained by the samples from F with back propagation algorithm, and finally cancels out F. If the networks becomes sufficiently accurate to verify the cancellation, then the error is described by

$$\dot{e} \cong \Lambda e \tag{17}$$

, so that the system is controlled to be asymptotically stable.

If V_M is chosen as the value near zero and the trajectory in the state space is periodic, the time at each cycle is sufficient as the input variable of neural network. It takes many iterations to train neural network with the large input dimension. So the supervisory controller brings about the additional effect of reducing the computational burden.

If we set V_M to zero, the error is not only bounded but also asymptotically stable. However, this control input resembles the relay control of VSC, so control effort is excessive and the chattering occurs. Combining the supervisory control and PD-type control with neural network identifier, such problems can be eliminated.

4 Simulation Results

The simple two link robot is considered, which has the following $M(q)$ and $N(q, \dot{q})$.

$$\begin{aligned} M_{11} &= m_1 lg_1^2 + I_1 + m_2(l_1^2 + lg_2^2 \\ &\quad +2l_1 lg_2 \cos(q_2)) + I_2 && (18)\\ M_{12} &= m_2(lg_2^2 + l_1 lg_2 cos(q_2)) + I_2 && (19)\\ M_{21} &= M_{12} && (20)\\ M_{22} &= m_2 lg_2^2 + I_2 && (21) \end{aligned}$$

$$\begin{aligned} N_{11} &= -m_2 l_1 lg_2 \sin(q_2)(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ &\quad +m_1 glg_1 \cos(q_1) + m_2 g(l_1 cos(q_1) \\ &\quad +lg_2 cos(q_1 + q_2)) && (22)\\ N_{21} &= m_2 l_1 lg_2 \sin(q_2)\dot{q}_1^2 \\ &\quad +m_2 glg_2 \cos(q_1 + q_2) && (23) \end{aligned}$$

where $m_1 = 2$, $m_2 = 1$, $g = 9.8$, $lg_1 = 0.5$, $lg_2 = 0.5$, $l_1 = 1$, $l_2 = 1$, $I_1 = 0.5$, $I_2 = 0.25$, and q_1,q_2 denote the joint angles.

The desired path in the cartesian coordinate is given by $(x_d(t), y_d(t)) = (0.5 + 0.3\sin(t), 0.5 + 0.3\cos(t))$. q_{d1}, q_{d2} is calculated by inverse kinematics. We choose $V_M = 0.5$, $Q = diag[10, \cdots, 10]$, $c_0 = 10$, $c_1 = 10$ and K_1,K_2 are the diagonal matrices whose elements on the diagonal are 2. Two neural networks which have the structure of $\aleph_{1,10,6,2}$ are used to identify the function $N(q, \dot{q})$. The sigmoid functions for the output nodes are scaled properly according to the variation range of $N_i(q, \dot{q})$, that is,

$$S_o(x) = K\tanh(x) \tag{24}$$

where K is a scale factor.

The following figures show the results of the identification and control of the robot dynamics using neural networks under the supervisory control while learning. Fig.1 depicts the tracking of the circle at the first revolution. Fig.2 shows the tracking of the circle at the 100th revolution.

5 Conclusion and Further Study

This paper is devoted to guarantee stability in the application of neural network to robot dynamics identification. Developed supervisory control algorithm is useful in both extracting the reference signal as the function of time and preventing neural network from getting unstable during the learning period. It takes the role of the teacher just as a teacher maintains stability and suggests a reference signal when a student is trying to learn a skill. The whole human-type controller paradigm including supervisory control algorithm is now proceeding.

References

[1] Tae-Dok Eom, Sung-Woo kim and Ju-Jang Lee, "Human-type controller using various kinds of neural networks," *submited to ICNN '96*, 1996.

[2] Man Zhihong, A. P. Paplinski and H. R. Wu, "A sliding mode control scheme using neural networks for robotic manipulators," *ICARCV '94*, pp. 1738, 1994.

[3] Akio Ishiguro, Takeshi Furuhashi, Shigeru Okuma and Yoshiki Uchikawa, "A neural network compensator for uncertainties of robotics manipulators," *IEEE Trans. Industrial Enectronics*, vol. 39, no. 6, pp. 565-570, 1992.

[4] L. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 2, pp. 146-155, 1993.

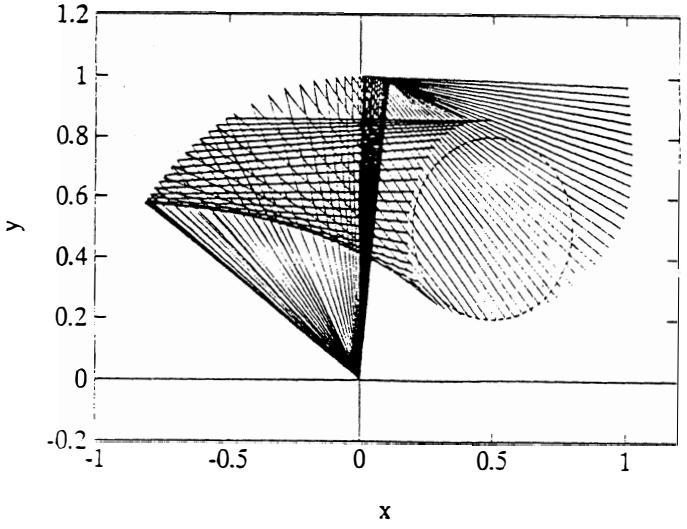


Fig. 1 : Tracking control at the first revolution.(- - : desired trajectory)

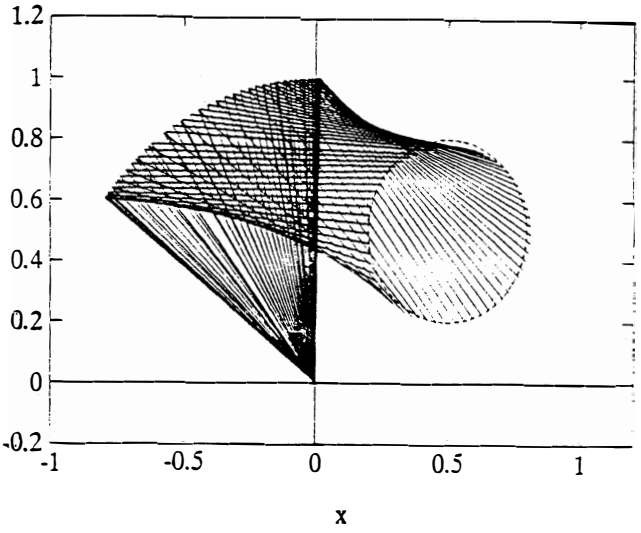


Fig. 2 : Tracking control at the 100th revolution.(- desired trajectory)

CHAOTIC LOCAL SEARCH ALGORITHM

Changkyu Choi and Ju-Jang Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-Dong Yusong-Gu, Taejon 305-701, KOREA
Fax. : +82-42-869-3410 E-mail : jjlee@ee.kaist.ac.kr

Abstract

This paper describes a chaotic local search algorithm of an unstructured search space. The problem is: given the quality function $Q(\mathbf{X})$, where $\mathbf{X}$ is a vector of adjustable parameters $x_1, ..., x_n$; find the value of $\mathbf{X}$ that minimized Q. The proposed algorithm started basically from the gradient search technique but at prescribed points, i.e. local minimum points, chaotic search algorithm is introduced by the dynamics of chaotic neuron model. It is systematically combined with the gradient search technique to automatically detect that points and toggle its state between two search techniques. A parallel formulation of this search algorithm is developed to determine a solution which represent the best improvement among the feasible solutions from chaotic neuron models each equipped with different parameter set. Computer simulations for the application of the motion planning problem of a mobile robot are presented.

1 Introduction

Many problems of both practical and theoretical importance concern themselves with the choice of a best configuration or set of parameters to achieve some goal. Over the past few decades a hierarchy of such problems has emerged, together with a corresponding collection of techniques for their solution. At one end of this hierarchy is the general nonlinear programming problem. There are some problems for which no effiecient algorithm is known and these cannot be solved by any known polynomial algorithm. By the way, there grow other techniques which are motivated by nature. Genetic algorithm is a typical example. Our method is based on a chaotic dynamics which look like random.

In general, local optimality does not imply global optimality. Algorithms may get stuck in local minima, thus providing suboptimal solutions. For some systems, this suboptimal solution is enough in a view point of the system performance as a whole. However in other cases only global optimality makes sense. For example, in robot motion planning problems the suboptimal solution (*i.e.* local minima) is meaningless. Though the value of quality function increases, it's better to move to any direction than to stay there.

The organization of this paper is as follows. Chaotic neuron models are discussed in section 2. In section 3, a chaotic search technique is proposed. A parallel formulation of this algorithm is also given. Application to the motion planning problem of a mobile robot is investigated in sections 4. Finally, we draw some conclusions and discuss the proposed search algorithm.

2 Chaotic Neuron Models

2.1 1-dimensional case

Aihara *et al.*[1] suggest a chaotic neuron model based upon Caianiello's neuron equation. The chaotic neuron model which includes conventional models of neural networks reads

$$p(n+1) = f(A(n) - \alpha \sum_{r=0}^{n} k^r g(p(n-r)) - \delta), \quad (1)$$

where $p(n+1)$ is the output of the neuron at the discrete time $n+1$, which takes an analog value between 0 and 1; f is a continuous output function, *e.g.* the sigmoid function; and g is a function describing the relationship between the analog output and the magnitude of the refractoriness to the following stimulation. $A(n)$ is the strength of the input at n, α is a positive parameter and k is the damping factor of the refractoriness, which takes a value between 0 and 1. By defining the internal state $q(n+1)$ as follows,

$$q(n+1) = A(n) - \alpha \sum_{r=0}^{n} k^r g(p(n-r)) - \delta, \quad (2)$$

(1) can be simplified as (3) and (4):

$$p(n+1) = f(q(n+1)), \tag{3}$$

$$q(n+1) = kq(n) - \alpha g(f(q(n))) + a(n), \tag{4}$$

where

$$a(n) = A(n) - kA(n-1) - (1-k)\delta. \tag{5}$$

If the output function, f, and the refractory function, g, take the following forms (Fig.1):

$$f(q) = \frac{\rho}{1+e^{-q/\Phi}} - \frac{\rho}{2} \tag{6}$$

$$g(p(n)) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} e^{-p(n)^2/2\sigma^2} \\ \qquad if \quad p(n) - p(n-1) \geq 0 \\ -\frac{1}{\sqrt{2\pi}\sigma} e^{-p(n)^2/2\sigma^2} \qquad otherwise, \end{cases} \tag{7}$$

where Φ and σ controls the shape of the functions f and g, respectively, and ρ the magnitude of the output function. From Fig.1 the refractory function g is a kind of hysteresis function. As the input increases, the upper Gaussian function is traced. So, it gives a directionality to q and it is analogous to the momentum. The internal state $q(n+1)$ is drastically changed from $A(n)$ when $q(n)$ is near 0. Thus the output undergoes unstable motions near the origin. Fig.2 shows the patterns of output p with respect to $a(n)$. When the magnitude of input is larger than 0.2, the output is almost the same as the input. But the input is near 0, the output oscillates chaotically. It has the tendency to wander positive region when the input is positive while it wanders negative region when the input is negative due to the hysteresis-like refractoriness.

2.2 2-dimensional case

By defining the internal state $q(n+1)$ similar to that in (2), (3) and (4) can be modified as (8) and (9):

$$p_i(n+1) = f(q_i(n+1)), \quad i = x, y, \tag{8}$$

$$\mathbf{q}(n+1) = k\mathbf{q}(n) - \alpha g(\mathbf{p}(n))\mathbf{I}_{2\times 1} + \mathbf{a}(n), \tag{9}$$

where

$$\mathbf{p}(n+1) = [p_x(n+1)\ p_y(n+1)]^T,$$

$$\mathbf{q}(n+1) = [q_x(n+1)\ q_y(n+1)]^T$$

and

$$\mathbf{a}(n) = \mathbf{A}(n) - k\mathbf{A}(n-1) - (1-k)\delta\mathbf{I}_{2\times 1}. \tag{10}$$

The variables $\mathbf{p}$, $\mathbf{q}$, $\mathbf{a}$ and $\mathbf{A}$ are 2-dimensional vectors, corresponding to p, q, a and A of 1-dim. case, respectively, and $I_{2\times 1}$ is $[1\ 1]^T$. According to the above modifications, the output function need not to be changed, while the refractory function takes the form of 2-dim. Gaussian distribution function (Fig.3):

$$g(\mathbf{p}(n)) = \begin{cases} \frac{1}{2\pi\sigma^2} e^{-\|\mathbf{p}(n)\|^2/2\sigma^2} \\ \qquad if \quad p(n) - p(n-1) \geq 0 \\ -\frac{1}{2\pi\sigma^2} e^{-\|\mathbf{p}(n)\|^2/2\sigma^2} \qquad otherwise, \end{cases} \tag{11}$$

where Φ and σ controls the shape of the functions f and g, respectively, and ρ the magnitude of the output function as in the case of 1-dim. chaotic neuron model. Bifurcation diagram of 2-dim. chaotic neuron resembles the motion of Fig.2 when $\mathbf{p}(n)$ is near 0. When only one component of $\mathbf{p}(n)$ is near 0, it does not reveal chaotic motion. Extensions to the higher dimensional neurons are straightforward.

3 Chaotic Search Technique

3.1 Algorithm

Consider a configuration vector X which is update by

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \Delta\mathbf{X}_n. \tag{12}$$

Basically, $\Delta\mathbf{X}_n$ is a negative gradient vector. But naive usage of it to (12), makes that $\mathbf{X}_n$ is vulnerable to get stuck in local minima. To avoid this situation, a negative gradient vector is calculated and it serves as an input of chaotic neuron model, so $\mathbf{a}(n) = \Delta\mathbf{X}_n$. Then, (12) becomes:

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \mathbf{p}(n). \tag{13}$$

For larger values of $\Delta\mathbf{X}$, $\mathbf{p}(n)$ is almost the same as $\Delta\mathbf{X}$. However as seen in Fig.2, there are infinitely many values of $q(n+1)$ and thus infinitely many values of $p(n+1)$ corresponding to $\Delta\mathbf{X}$ near the origin exist. Therefore we need a selection mechanism. For simplicity, we choose values of $\mathbf{p}(n)$ randomly and calculate its improvement. After several trials, all the trials are compared and the only one which produces the best improvement among these is selected.

Algorithm 1 : Single Processor

Repeat until goal reached or global time-out

1. *gradient descent until local minima*
2. *repeat several times*
   choose a value in the chaotic motion

calculate its improvement
choose one for the best improvement
among these
apply it to (13)
If still in local minima, GoTo 2
GoTo 1

The above algorithm, however, cannot guarantee that the configuration vector can produce the global optimality. There is no justification to use the specific values of ρ, σ, Φ and so on. Varying the parameters, α, k, ρ, Φ and σ, the chaotic neuron models reveal rich dynamics. In a local minima with very high potential, it is desirable for the width and the height of regions, which reveal chaotic dynamics, to be longer than those of regions in a local minima with lower potential. Hence, it is further needed to refine this algorithm.

3.2 Parallel Formulation

Assuming that a k-processor parallel search exists. Each processor is equipped with a different set of parameter values, ρ, η, Φ and so on. Thus, the regions that reveal chaotic dynamics are different from processor to processor. Some parallel search processor may have wide chaotic motion regions than those of another. An outline of the algorithm is shown below.

Algorithm 2 : k-processor parallel search
Repeat until one of the processors find the goal or global time-out
1. gradient descent until local minima
2. for each k-processors, repeat several times
choose a value in the chaotic motion
calculate its improvement
choose one for the best improvement
among these
apply it to (13)
If still in local minima, GoTo 2
GoTo 1

We can give a brief theoretical explanation for the success of parallel search schemes. Let $P_1(t)$ be the probability that a single processor will find a solution within time t, and let $P_k(t)$ be the probability that a k-processor parallel search scheme will find a solution within time t. Assume that the probability $P_1(t)$ is identical to every k processors and let the random variable T_1^i be time it would take processor p_i to find a solution, if allowed to run to completion. Since this is equivalent to running multiple trials on a single processor, the T_1^i's are independent and identically distributed. The probability $1 - P_k(t)$ that the solution time on k processors will exceed t is just the probability that none of the k processors will find a solution within time t:

$$1 - P_k(t) = (1 - P_1(t))^k \tag{14}$$

To interpret this formula, suppose a single processor has only a 10% probability that it can solve the problem within a given time $t_{10\%}$. Then a 32 processor system has over a 96%probability of finding a solution within $t_{10\%}$. A 64 processor system has over a 99% chance of doing the same.

4 Motion Planning Problem

Virtual Force Field (VFF) method is widely used in motion-planning problem because of its simplicity. The total virtual force $\mathbf{F}_t$ is thought to be a negative gradient of the electrostatic potential field. A virtual attractive force, $\mathbf{F}_a$, of constant magnitude is applied to the mobile robot pulling it toward the target and all virtual repulsive forces from obstacles are added up to yield the resultant repulsive force, $\mathbf{F}_r$. If the total force happens to be zero, then the robot cannot move and the velocity of the robot becomes zero. In this case, the robot fails to find the target. In order to overcome this situation, our algorithm can be applied. Simulation results are not given here for the page limitaion, but you will find similar successful results in [2][3].

Our method differs from previous heuristic approaches(see references mentioned in [2] and [3]), in which they determined whether the trap or local minima situation occurred, and then changed its mode to one of the case-by-case heuristic algorithms. Although our method is similar, in that they are a kind of detection method, we systematically combine the detection stage and the solution of trap situations. Using Parallel search algorithms (CNM_i) each equipped with different parameter sets, $\mathbf{P}_i$, to unknown environments, our path-planner can be flexibly applied to various situations and is superior to other methods. This is the main difference between ours and other methods.

5 Conclusions

We suggests a 2-dim. as well as 1-dim. chaotic neuron model equipped with Gaussian function having hysteresis as a refractoriness. Due to the hysteresis-like refractoriness, chaotic dynamics appears near the

origin. Also, it has a directionality so that chaotic motions are confined to specific regions.

By utilizing a chaotic search algorithm, the local minima problem is solved. and the parallel search scheme enhances the probability to place the configuration vector to global optimality. Application to the motion-planning problem shows the effectiveness of the proposed algorithm. Hence, the use of intentional chaotic dynamics might improve the performance of the system of interest.

References

[1] K. Aihara, T. Takabe and M. Toyoda, "Chaotic Neural Networks," *Physics Letters A*, Vol. 144, No. 6, pp. 333-340, 1990.

[2] C. Choi, S.-G. Hong, J.-H. Shin, I.-K. Jeong and J.-J. Lee, "Dynamical Path-Planning Algorithm Using Chaotic Neuron Model," in *Proc. IROS*, Vol. 2, pp. 456-461, Pittsburgh, PA, Aug. 5-9, 1995.

[3] C. Choi and J.-J. Lee, "Dynamical Path-Planning Algorithm of a Mobile Robot : Local Minima Problem and Nonstationary Environments," *Mechatronics*, 1995. (in press)

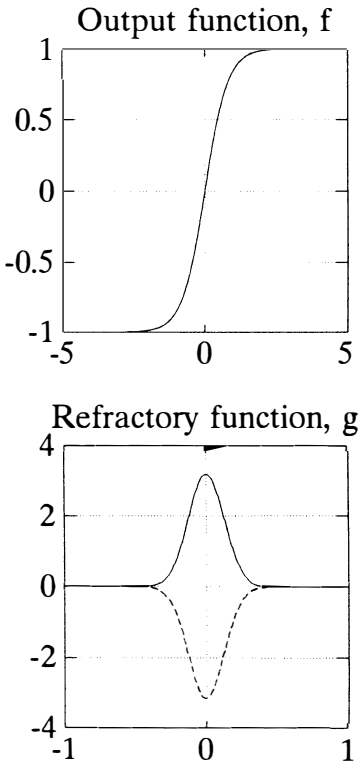


Fig.1 Functions used in 1-dim. CNM.

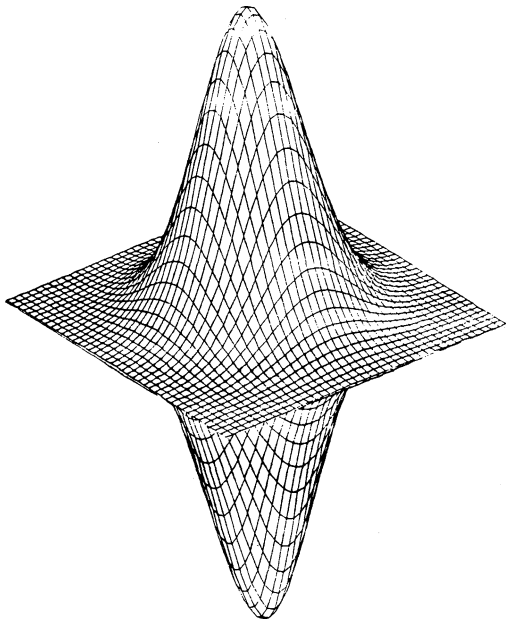


Fig.3 Refractory function, g of 2-dim. CNM.

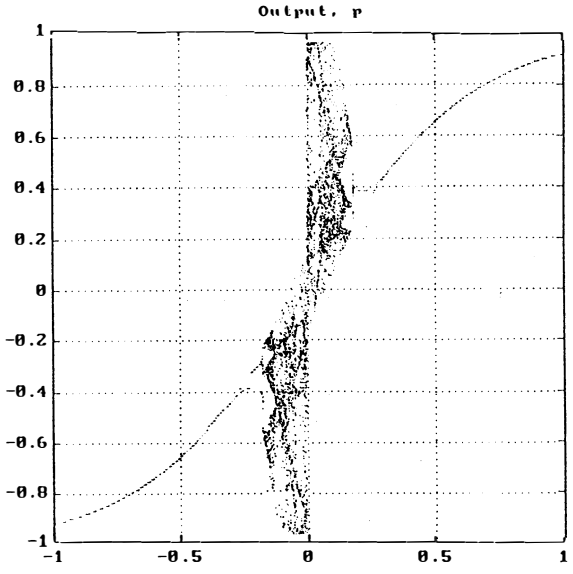


Fig.2 Output, p of 1-dim. CNM.

A Self Organizing Genetic Algorithm

Il-Kwon Jeong and Ju-Jang Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373 – 1 Kusong-dong, Yusong-gu, Taejon 305 – 701, Korea
FAX: +82-42-869-3410 E-mail: jik@odyssey.kaist.ac.kr

Abstract

A genetic algorithm(GA) has some parameters that should be determined before execution. We propose a self organizing genetic algorithm(SOGA) which sets GA parameters such as population size, crossover probability and mutation probability adaptively during the execution of a genetic algorithm. In SOGA, GA parameters change according to adaptation rules. We discuss about SOGA and other approaches for adapting operator probabilities in GAs. The validity of the proposed algorithm will be verified in a simulation example of system identification.

Keywords Genetic algorithm, Self organizing

1 Introduction

Genetic algorithms(GAs) are robust search and optimization techniques which are based on the natural selection and genetics. GAs are applied in a number of practical problems nowadays. GAs are different from conventional optimization methods in several ways. GA is a parallel and global search method which searches multiple points so it is more likely to get the global optimum. It makes no assumption on the search space so, it can be easily applied to various problems. In control area, it has been applied to identification [1], adaptation and neural network controller [3] [4]. However GAs are inherently slow and not good at fine tuning of the solutions.

The GA may be thought as an evolutionary process where a population of solutions evolves over a sequence of generations. During each generation, the fitness(goodness) of each solution is calculated, and solutions are selected for reproduction based on their fitness. The probability of survival of a solution is proportional to its fitness value. This process is based on the principle of 'Survival of the fittest'. The reproducted solutions then undergo recombination which consists of crossover and mutation. Genetic representation may differ from the real form of the parameters of the solutions. Fixed-length and binary encoded strings have been widely used for representing solutions since they provide the maximum number of schemata and as they are simple to implement [2].

In this paper we describe a self organizing genetic algorithm(SOGA) for multimodal function optimization. The choice of the crossover probability, p_c and the mutation probability, p_m is known to critically affect the behavior and performance of the GA. Though a number of generalized guidelines exist in the literature for choosing p_c and p_m, these guidelines are inadequate as the choice of optimal p_c and p_m becomes specific to the problem under consideration. The size of a population is another important parameter that affects the performance of the algorithm. In our algorithm, p_c, p_m and the size of the population are determined adaptively by the GA itself to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the GA.

Recently, several works have been done for adapting p_c and p_m. Operator probabilities vary according to the population maximum fitness and the population mean fitness in [6]. But, their adaptation rules are not general as ours. Bryant used the relative credit of each operator over some generations [7]. However, his algorithm needs more memory and computation time compared to simple GA. The paper is organized as follows. In section 2, a simple genetic algorithm is briefly described. Section 3 describes our algorithm, SOGA. In section 4, we present simulation results. The conclusions are presented in section 5.

2 A Simple Genetic Algorithm

GA is a search method based on the natural selection and genetics. GA is computationally simple yet powerful and it is not limited by assumptions

about the search space. The most important goal of optimization should be improvement. Although GA cannot guarantee that the solution will converge to the optimum, it tries to find the optimum, that is, it works for the improvement.

Following the model of evolution, GA establish a population of individuals, where each individual corresponds to a point in the search space. An objective function is applied to each individual to evaluate their fitness. Using genetic operators, a next generation is formed based upon the survival of the fittest. Therefore, the evolution of individuals from generation to generation tends to result in fitter individuals, solutions, in the search space. Empirical studies have shown that genetic algorithms do converge on global optima for many problems including NP-hard ones.

Simple GA has three basic genetic operators: *Reproduction*, *Crossover* and *Mutation*. There are three differences of GA from random search. First, the existence of the direction of search due to the selection probability. Second, the fact that the better strings make more offsprings and finally, being likely to be improved in average fitness over generations.

3 Self Organizing Genetic Algorithm

In optimizing unimodal functions, it is important that the GA should be able to converge to the optimum in as few generations as possible. For multimidal functions, there is a need to be able to locate the region in which the global optimum exists, and then converge to the optimum. GAs possess poor hill-climbing capabilities and, they are vulnerable to getting stuck at a local optimum especially when the populations are small. The significance of p_c and p_m in controlling GA performance has long been acknowledged in GA research. The higher the value of p_c, the quicker are the new solutions introduced into the population. As p_c increases, however, solutions can be disrupted faster than selection can exploit them. Large value of p_m transform the GA into a purely random search, while some mutation is required to prevent the premature convergence of the GA to suboptimal solutions. The population size also affects the GA performance. Premature covergence may occur when the population size is small, while a large population size makes the algorithm slow. Usually, the choice of p_c, p_m and population size is left to the user to be determined statically prior to the execution of the GA.

To overcome the above-stated problem of difficulty in choosing the GA parameters, we suggest the following expressions which are main components of SOGA.

$$p_c = k_1(f_{max} - f')/(f_{max} - f_{min}) + k_2 \qquad (1)$$
$$k_1 + k_2 \leq 1$$
$$p_m = k_3(f_{max} - f)/(f_{max} - f_{min}) + k_4 \qquad (2)$$
$$k_3 + k_4 \leq 1$$

where f_{max} is the maximum fitness value and f_{min} is the minimum fitness value. f' is the larger of the fitness values of the solutions to be crossed. k_3 and the population size, N_{pop} are changed adaptively using the following procedure.

1. Initialize k_3.
2. $i \leftarrow i + 1$, generation i.
3. If the fittest is the same for n_{reset} generations, then $N_{pop} \leftarrow N_{pop} + n_1$ and go to step 1.
4. $N_{pop} \leftarrow N_{pop} - n_2$, $k_3 \leftarrow c * k_3$, go to step 2.

where n_1 and n_2 are positive constants of integers, and $c < 1$. As we can see from (1), p_c is a linear function of f' which varies from $k_1 + k_2$ to k_2 as f' changes from f_{min} to f_{max}. p_m varies in a similar fashion. Thus, the higher the fitness of a solution, the lower the probability of crossover or mutation of the solution. Therefore, we are able to preserve 'good' solutions of the population while the low fitness solutions prevent the GA from getting stuck at a local optimum. Note that k_3 is designed to decrease exponentially over generations. After few generations k_3 vanishes to 0 from its initial value and the mutation operator becomes to behave like a normal one. But, when the fittest is the same for n_{reset} generations, that is, the GA is getting stuck at a local optimum, p_m is enlarged to its initial value to move the search to the global optimum and, the population size N_{pop} is increased to search wider region of the search space. When the algorithm is in its normal operation state, N_{pop} is decreased at every generations to speed up the algorithm.

There have been similar works to improve the GA. [3] incorporates simulated annealing technique into the GA. [4] uses a fitness modification technique and an adaptive mutation probability. A local improvement operator is introduced in [5]. [6] proposes adaptive probabilities of crossover and mutation. However, its adaptive rules are not as general as ours and, the adaptation of the population size is not considered.

4 Simulation

The problem considered here is the same as those in [1][3]. It is presented for comparison purpose. The

object system is a discrete time system:

$$A(q^{-1})y(t) = B(q^{-1})u(t-d) \quad (3)$$

where q^{-1} is the backward shift operator and the objective is identifing $A(q^{-1})$, $B(q^{-1})$ and delay d using the given input $u(t)$ and the output $y(t)$. We define the error sequence as

$$\eta(t) = y(t) - \hat{y}(t) \quad (4)$$

with

$$\hat{A}(q^{-1})\hat{y}(t) = \hat{B}(q^{-1})u(t-\hat{d}) \quad (5)$$

The fitness function to be maximized is

$$F(t) = 1/\sum_{i=0}^{w}(\eta(t-i))^2 \quad (6)$$

where w represents window size. The system polynomials, poles and zeros in the reparameterized plane [1] are the following:

$$A(q^{-1}) = 1.0 - 1.5q^{-1} + 0.7q^{-2} \quad (7)$$

$$B(q^{-1}) = b_0(1.0 + 0.5q^{-1} + 0.0q^{-2}) \quad (8)$$

$$[p_1, p_2] = [0.75, -0.37], [z_1, z_2] = [-0.25, 0.25] \quad (9)$$

where b_0 is 1 and the delay d was set to 1. We apply a simple GA and SOGA to identify p_1, p_2, z_1, z_2, b_0 and d. b_0 is assumed to be in $[0, 2]$ and the poles and zeros in $[-1, 1]$. We use binary encoding. 7 bits were used for each parameter except for d (2 bits), so the resolution is slightly smaller than 0.02. A string consists of 37 bits. We used $p_c = 0.8$, $p_m = 0.01$, $N_{pop} = 100$ and $w = 30$ for the simple GA. We used $k_1 = k_2 = 0.5$, initial $k_3 = 0.9$, $k_4 = 0.01$, initial $N_{pop} = 50$, $n_{reset} = 5$, $n_1 = 6$, $n_2 = 2$, $c = 0.9$ and $w = 30$ for SOGA. Input for the sample data is

$$u(t) = \sin(t) - \sin(t/2.5) + random(-1 \sim 1) \quad (10)$$

We show the input and output used for the sample in Fig. 1. One simulation was done using 200 samples with 3 generations per one sample, that is, 600 generations. 10 simulations were done for each algorithm. Fig. 2 $\sim$ 4 show the average of the identification results of the poles with simple GA. The true value of p_2 is -0.371. But, the limitation on the resolution due to binary coding makes p_2 equal to -0.375. Fig. 5 $\sim$ 7 show the average of the results with SOGA. It shows the better hill-climbing and optimum finding capability than simple GA. The average of the population sizes of 10 simulations was found to be 43.4, which is much smaller than that of the simple GA though the performance of SOGA is much better than that of the simple GA.

5 Conclusion

A self organizing genetic algorithm(SOGA) was designed to prevent the premature convergence and to sustain the convergence capacity of the GA. SOGA determines p_c, p_m and N_{pop} automatically using its adaptive rule so, we do not have to determine the values for the parameter prior to the execution of GA. Simulation results indicate that SOGA has adaptive characteristics and improved hill-climbing capability compared to the simple GA. Using adaptive population size, execution time of the algorithm is significantly lowered. Furthur work includes the theoretical analysis of SOGA.

References

[1] K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 5, pp. 1033-1046, Sep., 1992.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[3] I. K. Jeong and J. J. Lee, "Genetic algorithms and neural networks for identification and control," *Proceedings of the First Asian Control Conference*, pp. 697-700, 1994.

[4] I. K. Jeong and J. J. Lee, "A modified genetic algorithm for neurocontrollers," *IEEE International Conference on Evolutionary Computing*, pp. 306-311, 1995.

[5] J. A. Miller, W. D. Potter, R. V. gandham and C. N. Lapena, "An evaluation of local improvement operators for genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 5, pp. 1340-1351, Sep., 1993.

[6] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 4, pp. 656-667, April, 1994.

[7] B. A. Julstrom, "What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm," *Proceedings of the Sixth Int. Conf. on Genetic Algorithms*, pp. 81-87, 1995.

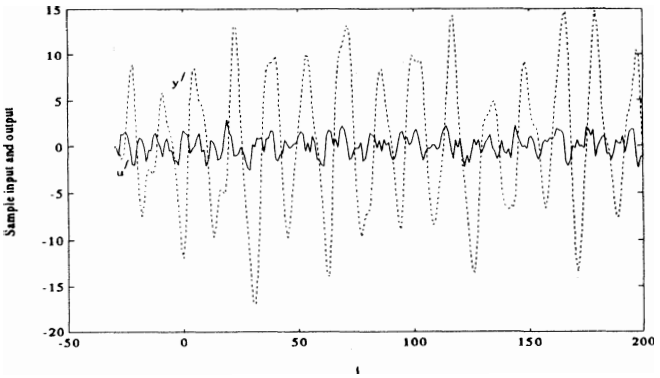


Fig. 1. The sample input and output

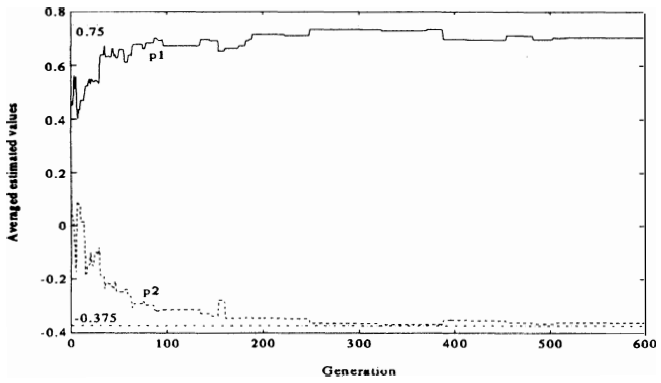


Fig. 2. Identification of the poles using a simple GA

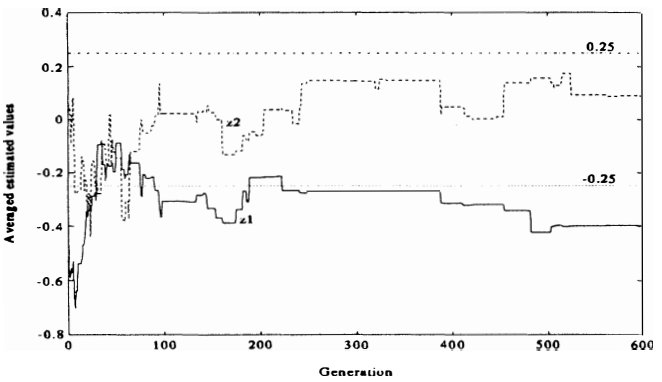


Fig. 3. Identification of the zeros using a simple GA

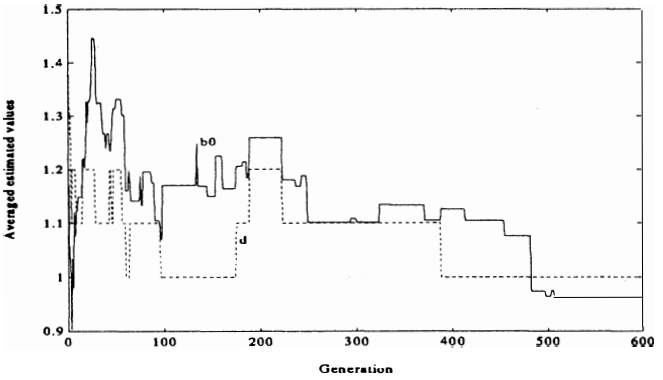


Fig. 4. Identification of b_0 and d using a simple GA

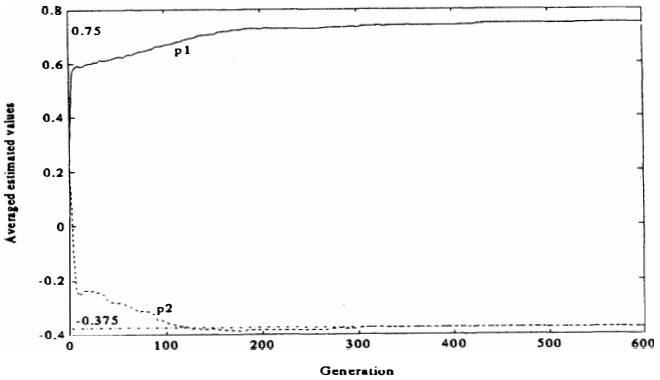


Fig. 5. Identification of the poles using SOGA

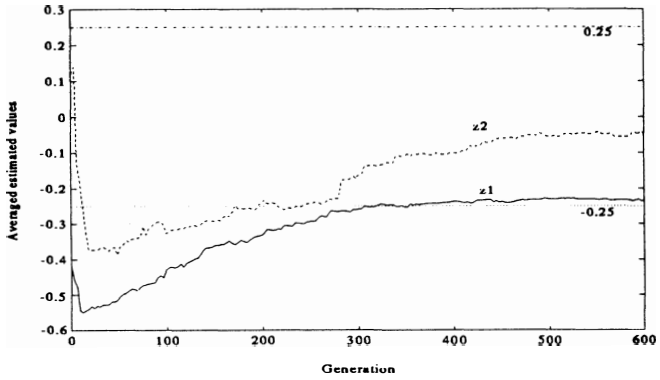


Fig. 6. Identification of the zeros using SOGA

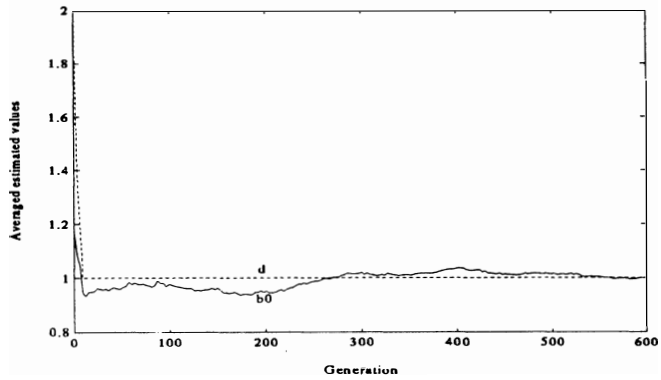


Fig. 7. Identification of b_0 and d using SOGA

Local Motion Planner for Unicycle-like Vehicle : Guaranteeing Collision Avoidance Even in Unknown Environment

Sun-Gi Hong, Changkyu Choi, Kang-Bark Park, and Ju-Jang Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373 – 1 Kusong-dong, Yusong-gu, Taejon 305 – 701, Korea
FAX: +82-42-869-3410 E-mail: sghong@odyssey.kaist.ac.kr

Abstract

This paper deals with the motion planning problem for a unicycle-like vehicle whose motion is confined by its own nonholonomic constraints. Generally, it is known that it is difficult to develop the feedback control scheme of nonholonomic systems due to the fact that nonholonomic system cannot be stabilized to a given any configuration based on a smooth time-invariant state feedback law. In this paper, we exploit a simple feedback scheme for unicycle model, based on an approximation of the desired configuration given by local holonomic planner that ignores motion constraints. To prevent the approximation error from causing the collision with obstacles, we propose an inequality constraint, based on the analysis of vehicle's motion , which is assumed to be governed by the constant control input during the sensor's sampling time. Consequently, we formulate our problem as the constrained optimization problem and the feedback scheme based on local sensor information is established by simply solving this problem. Through some simulations, we confirm the validity and effectiveness of our algorithm.

1 Introduction

This paper deals with the problem of planning constrained motions where the constraints are nonholonomic in nature. Specifically, we focus on the motion planning of the unicycle-like vehicle using the feedback control scheme.

Generally, it is difficult for the nonholonomic systems to exploit the feedback control law due to the fact that nonholonomic system cannot be stabilized to a given any configuration based on the continuous and smooth time-invariant state feedback law [1]. This motivated some complex feedback laws, for example, the time-periodic function based law [2] and so on. On the other hand, these previous methods do not take into account the presence of the unknown obstacles and require a sequence of feasible targets to complete a point-to-point motion such as a car parking problem.

In contrast to these nonholonomic planners, the local holonomic planners furnish the simple and powerful feedback scheme for the mobile robot navigating through the partially known environment. Using these merits, A. D. Luca and his collegues [3] have proposed a feasible projection strategy to modify the output of local holonomic planner. Their scheme provides a very simple and powerful feedback motion planner in a local sense. Instead, their method has a serious drawback that it cannot guarantee the collision avoidance with obstacles in spite of the collision avoidance being a basic and intrinsic demand for motion planning.

In this paper, we focus only on the goal position reaching problem. The basic idea of this paper is on the extension of A. D. Luca *et al.*'s work [3], in a sense that our proposed algorithm approximates the desired configuration given by the local holonomic planner. We propose the inequality constraints to prevent the approximation error from causing the collision with obstacles. Note that this is very different from A. D. Luca *et al.*'s work [3]. For this purpose, we analyze the local motion of vehicle, which is assumed to be governed by the constant control input during the sensor's sampling time. Finally, we formulate our problem as the constrained optimization problem under the assumption of the lower-bounded curvature due to the mechanical restrictions. Through the computer simulations, we will show the validity and effectiveness of our algorithm.

2 Motion Planner for Unicycle Model

Consider a unicycle-like vehicle positioned on the plane $\Re^2$ with respect to the base frame $< b >$, whose

motion is governed by the combined action of both the angular velocity w and the linear velocity vector v. A linear velocity vector v is assumed to be always directed as x axis of its attached frame $< a >$ where its origin is located at the center of vehicle, as depicted in Fig. 1. Then, the kinematic model of the unicycle-like vehicle, which involves the vehicle's Cartesian position x, y and its own orientation θ, is known as follows [3]:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \tag{1}$$

where v is the linear velocity and w is the angular velocity. The values of x, y, θ are all measured with respect to the base frame $< b >$. This model takes into account the nonholonomic constraint

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{2}$$

that specifies the tangent direction along any feasible path for the vehicle.

As stated earlier, A. D. Luca *et al.* [3] have presented a direct projection strategy to modify the output of local holonomic planner in the on-line manner. Their approach generates the velocity level control inputs that implement the desired motion, which is updated by a local holonomic planner, in the least-square sense. For any desired motion $\dot{\mathbf{x}}_\mathbf{d}$, their solution is given by the pseudo-inversion, as follows:

$$\begin{aligned} \mathbf{u} &= \mathbf{G}^{\#}(\mathbf{x})\dot{\mathbf{x}}_\mathbf{d} \\ &= [\mathbf{G}^T(\mathbf{x})\mathbf{G}(\mathbf{x})]^{-1}\mathbf{G}^T(\mathbf{x})\dot{\mathbf{x}}_\mathbf{d} \end{aligned} \tag{3}$$

where $\mathbf{x} \in \Re^n$ is a configuration vector of the generalized coordinate, $\mathbf{u} \in \Re^{n-m}(n > m)$ is an admissible control input vector, and $\mathbf{G}(\mathbf{x})$ is a uncycle model. Although their simple and efficient local planner furnishes a feasible motion in most situations where including even the unknown environment, no guarantee for the collision avoidance can cause some serious problems. This is because the least-square error $[\dot{\mathbf{x}}_\mathbf{d} - \mathbf{G}(\mathbf{x})\mathbf{u}]^T[\dot{\mathbf{x}}_\mathbf{d} - \mathbf{G}(\mathbf{x})\mathbf{u}]$ can drive the vehicle into the obstacle region when the desired motion $\dot{\mathbf{x}}_\mathbf{d}$ does not hold the nonholonomic constraints.

Thus, now we require to analyze the vehicle's motion, which is assumed to be governed by the constant control input vector $\mathbf{u} = (v, w)^T$ during the time interval of the command updating. This will be helpful to derive the constraints for preventing the approximation error from causing the collision. For simplicity of analysis, we fix the attached frame $< a >$ shown in Fig. 1 at the command updating instant. And then, with respect to this frame, we can obtain the vehicle's motion according to the analysis of the following equation. Note that we denote this fixed frame as $< f, a >$ through this paper.

$$\begin{pmatrix} \dot{x}_{f,a} \\ \dot{y}_{f,a} \\ \dot{\theta}_{f,a} \end{pmatrix} = \begin{pmatrix} \cos\theta_{f,a} & 0 \\ \sin\theta_{f,a} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_{f,a} \\ w_{f,a} \end{pmatrix}$$
$$(x_{f,a}(0), y_{f,a}(0), \theta_{f,a}(0))^T = (0,0,0)^T, 0 \le t \le \delta t_c \tag{4}$$

where $v_{f,a}$ and $w_{f,a}$ have the constant values respectively and δt_c is the time interval for the control input updating, that is, the sensor's sampling time. The subscript "f, a" denotes that the values of $x_{f,a}, y_{f,a}, \theta_{f,a}$ are all measured with respect to the attached frame which is fixed at the instant for the control input updating (that is, $t = 0$ in a local view). As shown in Fig. 2, a constant control input $\mathbf{u_{f,a}} = (v_{f,a}, w_{f,a})^T$ yields a circular path with a turning radius ρ during δt_c. Note that the value of $v_{f,a}$ is not negative at all because a linear velocity vector $v_{f,a}$ is assumed to be always directed as x axis of the attached frame.

Integrating the equation for the main direction of vehicle yields

$$\int_0^t \dot{\theta}_{f,a} d\tau = \int_0^t w_{f,a} d\tau \tag{5}$$

$$\theta_{f,a}(t) = w_{f,a} t \qquad (\theta_{f,a}(0) = 0) \tag{6}$$

where $0 \le t \le \delta t_c$. Then, the value of turning radius ρ can be computed as follows:

$$\rho|\theta_{f,a}(t)| = \int_0^t \sqrt{\left(\frac{dx_{f,a}}{d\tau}\right)^2 + \left(\frac{dy_{f,a}}{d\tau}\right)^2}\, d\tau \tag{7}$$

$$\rho = \left|\frac{v_{f,a}}{w_{f,a}}\right| \tag{8}$$

Thus, for the given $v_{f,a}$ and $w_{f,a}$, the vehicle tracks the following trajectory during δt_c.

$$\begin{aligned} x_{f,a}(t) &= sgn(w_{f,a})\rho\sin\theta_{f,a}(t) \\ y_{f,a}(t) &= sgn(w_{f,a})\rho(1 - \cos\theta_{f,a}(t)) \end{aligned} \tag{9}$$

where the function of $sgn(w_{f,a})$ returns the sign value.

To prevent the collision with obstacles, we should choose the control input $\mathbf{u_{f,a}} = (v_{f,a}, w_{f,a})^T$ so that the predicted locus in the equation (9) will not intersect any edges of obstacles. For this purpose, we assume that the function of $d(\psi_{f,a})$ represents the model of vehicle's surroundings and it returns a proximity measure in the direction of $\psi_{f,a}$ with respect to the fixed frame $< f, a >$. This model is updated per every δt_c. During δt_c, the following constraint should

be satisfied to guarantee the collsion avoidance for the given constant control input $\mathbf{u_{f,a}} = (v_{f,a}, w_{f,a})^T$.

$$d(\psi_{f,a}(t)) > \sqrt{x_{f,a}^2(t) + y_{f,a}^2(t)} \quad (10)$$

$$= 2\rho|\sin(\frac{\theta_{f,a}(t)}{2})| \quad (11)$$

$$= 2\rho|\sin(\psi_{f,a}(t))| \quad (12)$$

where $\psi_{f,a}(t) = \frac{w_{f,a}t}{2}$, $0 \leq t \leq \delta t_c$ and $\rho = |\frac{v_{f,a}}{w_{f,a}}|$. Fig. 2 will be helpful for the comprehensive understanding of the derivation for the above equations.

Based on the derived results so far, we can formulate our algorithm as follows:

$$\begin{array}{c} \text{Minimize } [\dot{\mathbf{x}}_\mathbf{d} - \mathbf{G(x)u}]^T[\dot{\mathbf{x}}_\mathbf{d} - \mathbf{G(x)u}] \\ \text{subject to} \\ d(\psi_{f,a}(t)) > 2\rho|\sin(\psi_{f,a}(t))| \end{array} \quad (13)$$

$$\rho = \left|\frac{v_a}{w_a}\right| = \left|\frac{v}{w}\right| \quad (14)$$

$$\psi_{f,a}(t) = \frac{w_{f,a}t}{2} = \frac{wt}{2} \quad (15)$$

where $0 \leq t \leq \delta t_c$. The solution of the equation (13) can be found based on the nonlinear programming technique. The relationship between the base frame $< b >$ and the fixed frame $< f, a >$ can be easily implemented through a simple transformation matrix.

3 Simulation Results

We demonstrated the proposed motion planner for a unicycle model in the presence with the circular obstacles, based on the potential field approach as the local holonomic motion planner.

Using the potential field approach, the desired velocity vector $\dot{\mathbf{q}}_\mathbf{d} = (\dot{x}_d, \dot{y}_d)$ can be obtained from the following equation at every sampling time.

$$\dot{\mathbf{q}}_\mathbf{d} = -\nabla_\mathbf{q}(U_a(\mathbf{q}) + U_r(\mathbf{q})) \quad (16)$$

where $U_a(\cdot)$ is an attractive potential function and $U_r(\cdot)$ is a repulsive potential function [4]. To complete finding the desired configuration, we require to assign the desired rotation of $\dot{\theta}_d$. For simplicity, we choose the desired steering input as follows:

$$\dot{\theta}_d = tan^{-1}\frac{\dot{y}_d}{\dot{x}_d} - \theta \quad (17)$$

where the function of $tan^{-1}(\cdot)$ returns the radian value from 0 to 2π. Therefore, we can compute the desired configuration $\dot{\mathbf{x}}_\mathbf{d}$ for the equation (13).

For the uncertain environment cluttered with circular osbtacles, we perform the simulation for the proposed motion planner. All the parameter values used in the simulation are given in Table 1. To solve the problem in equation (13), we use the exhaustive search technique because the serach domain is relatively small and its method is very simple for the computer analysis. We believe that another nonlinear programming techniques can also solve this problem in a short time.

In Fig. 3, we can show the successful result of the local holonomic planner. The nonholonomic motions obtained for the initial $\theta_0 = 0$ are shown in Fig. 4, with the associated desired configuration. In Fig. 4, we can see that the constructed path by the proposed algorithm is different from the holonomic one. This is because our algorithm approximates the holonomic path in local sense. Anyway, we confirm that the position error goes to zero in the terminal phase. We have applied our method to several other situations, and the satisfactory results, which guarantee the collision avoidance, were always obtained.

4 Conclusions

We have presented a efficient motion planner for a unicycle-like vehicle in the presence of unknown obstacles. Our feedback scheme can be utilized in a real time, to prevent the vehicle from colliding with obstacles. This makes our method be useful in a real application.

The proposed scheme has been applied to the unicycle-like vehicle. The potential field approach has been used as the local holonomic motion planner. The simulation results confirmed that the obtained configuration approximates the desired motion in a least-square and local sense and the proposed planner provides the satisfactory results. Since the proposed planner is based on the local strategy, even in the unknown environment the vehicle system can reach the target successfully.

Future research directions include a real application of our algorithm in a real world.

References

[1] R. W. Brockett, "Asymptotic Stability and Feedback Stabilization," in *Differential Geometric Control Theory*, R. W. Brockett, R. S. Millmann and H. J. Sussmann (Eds.), Birkhauser, Boston, pp. 181-191, 1983.

[2] J. -B. Pomet, "Explicit Design of Time Varying Stabilizing Feedback Laws for a Class of Controllable Systems without Drift," *System and Control Letters*, Vol. 18, pp. 139-145, 1992.

[3] Alessandro De Luca and Giuseppe Oriolo, "Local Incremental Planning for Nonholonomic Mobile Robots," *Proc. of IEEE Conf. on Robotics and Automation*, pp. 104-110, San Diego, California, May, 1994.

[4] O. Khatib, "Real Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90-96, 1986.

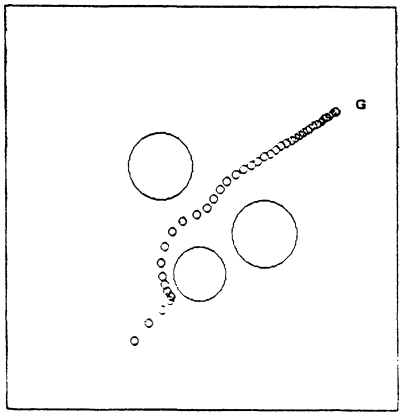


Fig. 3. A holonomic motion with potential field approach

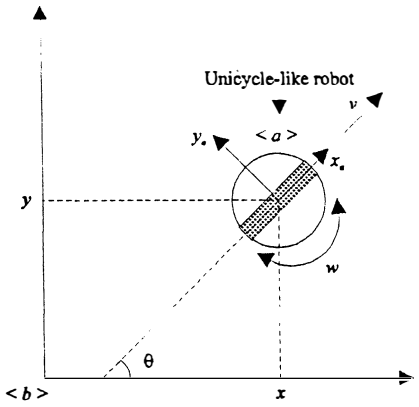


Fig. 1. A unicycle-like robot *w.r.t* the base frame

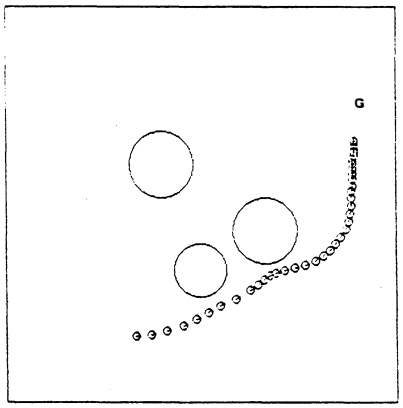


Fig. 4. A nonholonomic motion with potential field approach

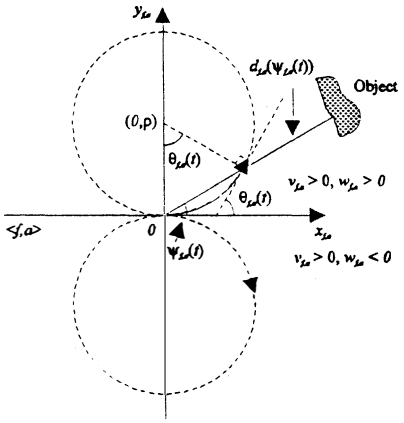


Fig. 2. A motion analysis during the time interval δt_c

Table 1. All the parameter values used in the simulations

Environment size	$300m \times 300m$
Radius of circular robot	$3m$
The number of sensors	36
The time interval for the command updating δt_c	$0.5sec$
Maximum speed v_{max}	$2.0m/sec$
Sensor range	$20.0m$

The Improved Radial Basis Function Neural Network and Its Application

Xudong Wang Huihe Shao Zhongjun Zhang
Automation Department
Shanghai Jiaotong University
Shanghai,200030,People's Republic of China

Abstract

Compared with other feed forward neural networks,the radial basis function neural network (RBFNN) has a lot of advantages which make it properer for modeling the nonlinear process,so it has recently gained considerable attention. In this paper, the RBFNN is introduced and its problems in modeling the nonlinear system are analyzed. In order to solve these problems, the structutre of the RBFNN is improved and the training method is modified. With the improved RBFNN an industrial high purity distillation column is modeled. Because of the complexities of the nonlinear system, a real time model correction method is neccessary. In this paper the Givens transformation is used to correct the model. Simulation shows that the modeling is successful.

1 Introduction

Multilayer feed forward neural network with one or more hidden layers can be used to approximate a continuous nonlinear function. So in recent years many feed forward neural networks including the radial basis function neural network have been researched. Although the radial basis function neural network is also a kind of feed forward neural networks, its structure and training method show the advantages such as the linear-in-the-parameters [1] [2] [3] and efficient training procedure. Based on the variational principle and regularization formulation, the RBFNN proves to be the best multilayer neural network to learn the input-output mapping [4].Because of the good characteristics of nonlinear mapping, the RBFNN is used in this paper to model the nonlinear industrial system such as a high purity distillation column. But the RBFNN has several problems in its structure and training algorithm. So the RBFNN should be improved. Using the improved RBFNN, the high purity distillation column is modeled. Because the working conditions of the industrial system may exceed the training field, on-line training or real time correction of RBFNN model is necessary.

2 The improved radial basis function neural network

Figure 1 is the structure of the RBFNN.

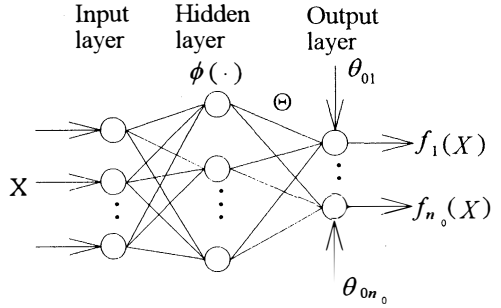


Fig. 1. The structure of RBF neural network

Such a network implements a mapping $f_j : R^n \to R$ according to :

$$f_j(X) = \theta_{0j} + \sum_{i=1}^{n_r} \theta_{ij}\phi(\|X - C_i\|) \quad 1 \le j \le n_0$$

where $X \in R^n$ is the input vector, $\phi(.)$ is a kind of nonlinear function from R^+ to R, $\|.\|$ denotes the Euclidean norm, θ_{ij}, $0 \le i \le n$, $1 \le j \le n_0$ are the weights or parameters, $c_i \in R^n$, $1 \le i \le n_r$ are the RBF centers and n_r is the numbers of centers. $\phi(.)$ is radial basis function which make the RBF networks have good approximation capabilities.

There are a lot of methods to train the RBFNN. Different method results in different training process of RBF network. The following four training methods are the most important and useful: 1. Mooky and Darken's algorithm [3]; 2. Local training method[5]; 3. Orthogonal least squares (OLS) algorithm [1] [2]; 4. Recursive Givens transformation method [6]. Usually method 3 and 4 is more efficient. But there are several problems should be solved.

First is how to calculate the centers of the radial basis function and determine the original number of neurons in the hidden layer. The rival penalized competitive learning (RPCL) can cluster a set of data [6]. This method do not need the selection of an appropriate number of clusters. So in this paper RPCL is used to calculated the RBF centers.[7]

Second is that only one kind of radial basis function is not sufficient to approximate the given input-output mapping. Improving the ability of nonlinear approximation requires the modification in the structure of the RBF neural network. In order to solve this problem, two sub-RBFNNs are combined into one RBFNN. It is important that the radial basis functions in the two sub-RBFNNs are different. Because the solution of this problem is same as that of the third problem, it is described in detail in the third problem.

The third problem is the computation problem in orthogonal least squares (OLS) algorithm which is used in the training procedure of RBFNN. When the vector orthogonalizing proceeds in the OLS algorithm [1], the module of w_k is very small. Sometimes the value of $||w_k||/||w_1||$ is too small to keep it in the group of orthogonal vectors. If such orthogonal vector is selected, the generalization of RBF network will deteriorate. This phenomena is just same as overfitting. If the problem is neglected, the performance of the network is very bad. In order to avoid overfitting, the module of w_k had better be verified. w_k will not be selected if its module is too small. Although this method can avoid overfitting problem, it usually can not satisfy the training accuracy. So when the module verifying is used, the training procedure of RBF network can not finished. In fact this problem is caused by the ill-conditioned data$\{\phi, D\}$ which are the output of the hidden layer of the RBFNN and the desired output respectively. From the hidden layer to the output layer is such a linear equation $D = \phi\Theta$. Usually the equation is an inconsistent equation. Only the approximate solution $\hat{\Theta}$ can be found so that $trace(D - \phi\hat{\Theta})$ is sufficient small. So the matrix ϕ should be chosen properly. But ϕ is the output of the hidden layer. In order to decrease the contradiction of the equation $D = \phi\Theta$, the nonlinear characteristics of the RBF neural network should be changed. So one RBFNN is decomposed into two sub-RBFNNs which have different radial basis function.

Figure 2 is the structure of the improved RBF neural network.

Structure (1) and structure (2) are two sub-RBFNNs which have the same topology. It is important that the radial basis functions in two sub-RBFNNs are different. The training procedure of this RBF network has five steps which are discribed in detail in [8].

Step 1:

Use the improved OLS algorithm with the input-output sequence $\{u, y\}$ to train structure (1).

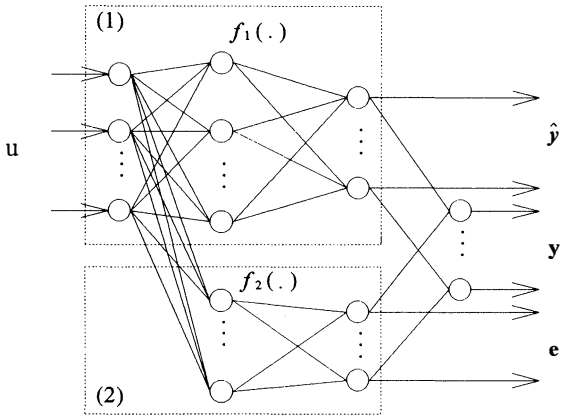


Fig. 2. The structure of the improved RBF neural network

Step 2:

If the training accuracy is sufficient, the procedure can be terminated and structure (2) is not useful.

If the training accuracy is not sufficient and the orthogonal vector w_k can not be selected then go to step 3.

Step 3:

Calculate the output sequence $\{\hat{y}\}$of structure (1) : $\hat{y} = f_1(u)$, and the error sequence $\{e\}$:$e = y - \hat{y}$.

Step 4:

Use the improved OLS algorithm with input-output sequence $\{u, e\}$ to train structure (2).

Step 5:

After structure (2) is trained the accuracy of the full network (structure (1) and (2)) is usually sufficient.The output of the RBF network is y.

3 Modeling the industrial system: high purity distillation column

In industry process the system is usually nonlinear and sometimes the knowledge that we can get from the system is only the input-output data. So structure and parameter identifying is very difficult .

It is well known that the multilayer network has the nonlinear mapping ability. So It is advantageous to use the network to identify the nonlinear system. But most of the multilayer networks need modify

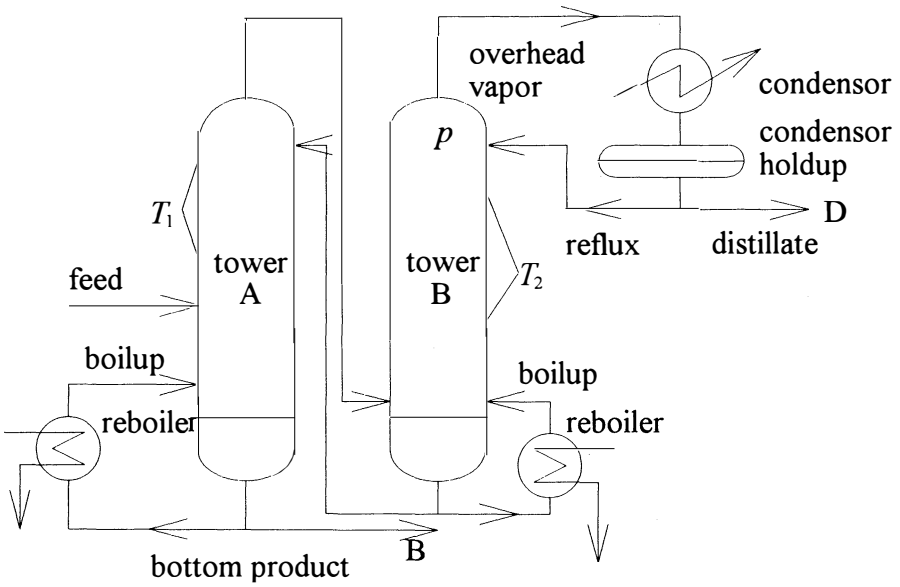


Fig 3 The distillation column

because of complexities of nonlinear system. So the RBFNN is improved to model a high purity distillation column which is an nonlinear system.

Figure 3 is the distillation column

Where overhead vapor is mainly composed of propylene and propane,bottom product is mainly composed of propane, propylene, and isobutane.

The input-output equation of the system is

$$y_B = f_1(p, T_1, T_2) \quad y_D = f_2(p, T_1, T_2) \qquad (1)$$

Where y_D is the concentraion of the propylene in the overhead vapor, y_B is the concentraion of the propane in the bottom product. T_1, T_2 are the differential temperature. p is the pressure.

The input-output equation of RBF network is as equation 2.

$$\begin{cases} y_B(t) = f_1(y_B(t-1), y_D(t-1), p(t-1), T_1(t-1), T_2(t-1)) \\ y_D(t) = f_2(y_B(t-1), y_D(t-1), p(t-1), T_1(t-1), T_2(t-1)) \end{cases} \qquad (2)$$

Figure 4 is the curve of the input T_1, Figure 5 is the output y_B of the distillation column. Figure 6 is the output y_B of the RBFNN. Comparing figure 5 with figure 6, the RBFNN model almost has the same charecteristics as the distillation system.

If the RBFNN is not improved, either the training accuracy is not enough or the performance of the RBFNN model is not good.

4 Real time correction of RBF neural network model

When the RBF network learns an industrial system it can only approximate the nonlinear system in the learning field. In fact when the system is working, its working conditions maybe exceed this field. This is usually caused by the disturbance or time varying parameter and environment. When the working conditions exceed the learning field, the RBF network usually can not work well. Even if the working conditions return into the learning field, the RBF network can not approximate the system.

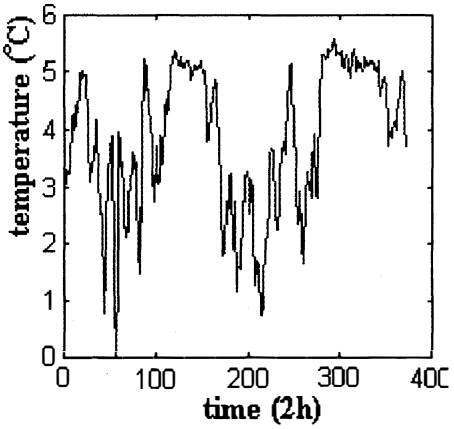


Fig 4 The curve of input T1

In order to solve this problem the recursive Givens transformation is used to correct to parameter of RBFNN in real time.

Givens transformation is very complex. it is described in detail in Gentleman's paper [9].

Using the recursive Givens transformation, the model correction strategy can been put forward . The strategy is as following.

Before the correction proceeds, the parameter $\Theta(t)$ should be stored as Θ_1. After correction, the parameter $\Theta(t)$ should be stored as Θ_2. If the conditions of the industrial system return into the training field the

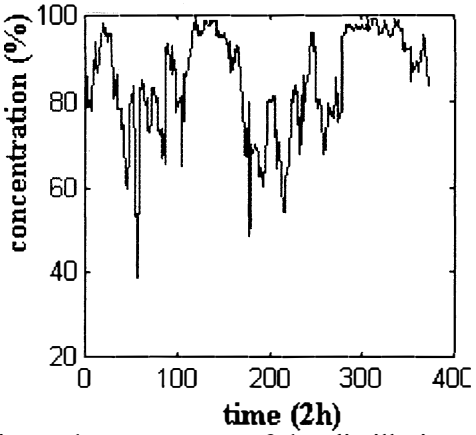


Fig 5 The output y_B of the distillation column

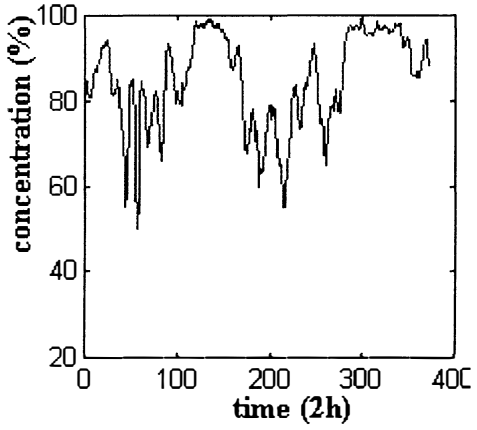


Fig 6 The output y_D of the RBFNN

correction procedure can be terminated and the parameter $\Theta(t)$ should be replaced by Θ_1.If the conditions of system is still out of the training field, the correction procedure should continue and at that time the output of RBF network should be calculated with the parameter Θ_2.

5 Conclusions

Nonlinear system modeling is an important problem because the model of the nonlinear process can be used to design the control strategy such as predictive control, adaptive control or internal model control. Also nonlinear system model is very useful in soft sensor.In practice the output of the system often can not be tested or can only be tested after a large delay. This may cause the difficulty to control the system. If an accurate model of the system can be gotten,not only the system can be easily controlled but also the system output can be tested in time so that the working condition of the system can be monitored.

Although the RBF network training method has been improved in this paper,there are some problems need to solve. For example, how to improve the generalization ability of the RBF network is an important step because the generalization ability determines the property of performance. Also how to train the RBF network so that it can predict the practical process needs more research.

References

[1] S.Chen, C.F.N.Cowan, P.M.Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," IEEE Transactions on Neural Networks, Vol.2, No.2, pp. 302-309, March, 1991.

[2] S.Chen, P.M.Grant, C.F.N.Cowan, "Orthogonal Least Squares Algorithm for Training Multioutput Radial Basis Function Networks," IEE Proceedings-F, Vol.139, No.6, pp 378-384, December, 1992.

[3] Moody and C.Darken,"Fast Learning in Networks of Locally-tuned Processing Units," Neural Computation, Vol.1, pp 281-294, 1989.

[4] S.Y.KUNG,Digital Neural Networks. PTR Prentice-Hall Inc.,1993.

[5] Tyler Holcomb and Manfred Morari, "Local Training for Radial Basis Function Networks: Towards Solving the Hidden Unit Problem," American Control Conference, pp 2331-2335, 1993.

[6] S.Chen, S.A.Billings and P.M.Grant, "Recursive Hybrid Algorithm for Nonlinear Identification Using Radial Basis Function Networks," International Journal of Control, Vol.55, No.5, pp 1051-1070, 1992.

[7] Lei Xu, Adam Krzyzak and Erkki Oja, "Rival Penalized Competitive Learing for Clustering Analysis, RBF net, and Curve Detection," IEEE Transactions on Neural Networks, Vol.4, No.4, pp 636-649, July, 1993.

[8] Xudong Wang, "Nonlinear System Modeling and Control Using the Radial Basis Function Neural Networks", Master Dissertation of Shanghai Jiao Tong University, 1995.

[9] Gentleman, W.E., "Least Squares Computations by Givens Transformations without Squares Roots," Journal of Institute of Mathematics and Applications, Vol.12, pp 329-336, 1973.

Artificial Plant Trees Based On **IFS**

Y.G.Zhang
Institute of Systems Science of Academia Sinica, Beijing, China, 100080

X.J.Li
Department of Computer Science, Beijing University of Aeronautic and Aerospace, Beijing, China, 100083

M.Sugisaka
Department of Electrical and Electronic Eng., Oita University, Oita, Japan, 870-11

ABSTRACT

In this paper we explain the notion of Iterated Function System(IFS) and created the growing units based on IFS. Then a artificial plant tree was evoluted. The transmission and control of growing information is similar to the RNA in Biological science. The comparison with L-system and future jobs for us was discussed. Three pictures showing the computational result are presented in the last part of this paper also.

Key words: Artificial Life, Genes, Fractal, Iterated Function System(IFS), L-system.

1. Introduction

In 1987, more than hundred scientists and engineers gathered in Los Alamos, New Mexico, to establish the new science of Artificial Life. In recent years the enthusiasm to study the **Artificial Life** is arising and some very interesting results were reported. Obviously, this is a very new field and it involves a deep philosophy. Some books have already been presented to introduce this new science systematically [1,2,3]. **Artificial Life, or A-life,** is devoted to the creation and study of lifelike organisms and systems built by humans. The stuff of this life is nonorganic matter, and its essence is the manner of information processing, computers are the kilns from which these new organisms emerge. By simulating a kind of life different from that with which we are familiar, A-life scientists seek to explore paths that no form of life in the universe has yet taken, the better to understand the concepts and limits of life itself. The earlier works in this field is very heuristic [4,5]. Lindenmayer's original work[4] showed a modeling plant growth using cellular automata. Dawkins' work[5] showed a vaguely lifelike creatures called Biomorph Land. He introduced the elements called "genes" into his program for biomorphic development. These genes influenced things like angle of branching, the length of a branch, the number of subbranches, and so on. [6] has showed more new works that to grow Artificial Brain by using of cellular automata in work station computer. Now a very relative research field named **Evolutionary Computation** is growing and developing. On the other hands, the **Fractal** studies have got many results in recent years[7,8,9]. Some of them were used to generate so-called plant at computer screens. Of cause, it is rather different from the Biomorphy Land and L-system. It is based on the principle of self-similarity and Iterated Function System. In spite of that some beautiful plant pictures were showed at computer screens, but they have not any meaning of Life. They are the same thing as the plants in 3-D cartoon, all of them is the computer graph skill only. Note that, some results of Fractal studies were used to explain the phenomena in the nature including the growing of alive organisms. We try to show another way to generate the plants with A-life meaning by using IFS we defined in this paper. This is what we have done in this paper.

2. IFS and Growing units

2.1 IFS in Fractal Field

In 1982, Mandelbrolt defined the word **Fractal** and used it to describe the divert world. Fractals are the sets whose Hausdorff dimension, which is generally a real number, differs from their topological dimension. An important features of fractals is that to describe a object only a little information is needed for their specification, even though the object looks like very complicated. It can be generated by a simple iterate rules. These rules are named IFS(Iterated Function System).
IFS in Fractal is defined as following:

In R^2, an affine transformation W is a linear map from R^2 to R^2:

$$(x_1, x_2) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

Here, a,b,c,d,e,f are some suitable real number. We can write W(X)=AX+B. A 2-dimensional affine transformation is thus completely specified by six real numbers. For an affine transformation and arbitrary x and y belonging to R^2, certainly at least one nonnegtive real number S can be found such that

$$\|W(X)-W(Y)\| \le S\|X-Y\|,$$

the minimum of all S that is suitable to above inequality is called as Lipschtz index S_L . If $S_L < 1$, then we call this affine transformation as a compressive affine transformation.

Let $\{W_1, W_2, \ldots, W_n\}$ be n compressive affine transformations and $\{P_1, P_2, \ldots, P_n\}$ is a probabilistic distribution on it, that is $P_i > 0$, i=1,2,... n, and $\Sigma P_i = 1$. If $(S_1)^{P1}(S_2)^{P2}\ldots(S_n)^{Pn} < 1$, then we say that $\{W_i, P_i,$ i=1,2,...,n}is a IFS codes. We can use this kind of codes to draw some Fractal graph, for example, fern plant at computer screen. It is not difficult to see that there is no any A-life meaning inside. We modified IFS to meet the requirements in A-life meaning.

2.2 Growing Units with genetic information

First of all, we try to generalized the concept of IFS. In fact, any unit which can be operated iteratively based on certain information series, whatever it is a set of function or a group of rules, could be called as a Iterated Function Unit. When we are going to simulate certain life phenomena, for example, the growing process of a tree, several kind growing units are needed. One is for the tree trunk, second for the branches, third for the subbranches and four for the leaves, above are the minimum. We designed four growing units:

T=T(high; diameter; growing direction; branch control; year),

B=B(length; diameter; growing direction; subbranch control; year),

SB=SB(length; direction; growing direction; leaves control; year),

L=L(size; shape control; direction; year).

They operate on the growing of the Trunk, Branches, SubBranches and Leaves respectively. To give them the A-life meaning, the manner of information processing must include the encoding, transcription, translation, decoding and growing etc.. The genes decide what kind of species will grow. This manner could be compared with the one for the processing in DNA. The information stored in DNA were encoded and transcribed into RNA, usually the RNA is a single-stranded molecule. In fact, the RNA sequence is the base sequence to control growing. During the second step, the ribosome makes use of genetic codes to translate a section of the RNA strand in order to form a string of amino acid(a protein). So, the growing units we have mentioned in the previous section are similar to the ribosome, the growing control sequences to be introduced in next section are similar to the RNA sequence, the growing units read the growing message from the control sequences and translate it into the growth of the tree body. The group of the growing units to proceed a growing process are still called as Iterated Function System(IFS) here. Obviously, the IFS defined here is different from that one in Fractal field.

The genes in the IFS defined here store the certain genetic information which distinguish one to others. In this paper we just considered very simple genetic information: shape(high and diameter), the position of branches starting, the direction of growing, and growing speed which is obey certain growing curves :

$$y = C/[1 + A\exp(-Bk)]$$

where A,B,C are the given constants and k is the discrete time variable(year). The constant A is the final size when the time variable tends to infinity.

2.3 Control Sequences of Growing

As what we have mentioned in last paragraph we need a control information which is similar to the RNA sequence to control the artificial tree's growing . We do not pay more attention to the transcription from the information like DNA to the RNA, shortly assume that we have already got the similar RNA sequence--Growing Control sequence. In fact, this sequence is a zero-one sequence extended with time variable :

0000101011101...

where "0" means growing only, without new branch in the time k, "1" means growing and generating a new branch in this time k. In the place where "1" appeared a new control sequence will be embedded by using the symbol "{" and "}". The sequence embedded in {} is also a zero-one sequence which is read from certain section of the original control sequence. Similarly, "1" in the embedded sequence has the same meaning as one in the original control sequence. Then a new embedded sequence in {} is generated and denote it as [ ]. Of cause, in this sequence a new sequence can be also embedded if "1" appeared in the previous embedded sequence and denote it as ( ). all of them is read from a section in the

original control sequence. The different control sequence will give different tree in shapes, but not in specie, because the specie is decided by the genes in the growing units.

2.4 The IFS in A-life Field

When we have defined the growing units and control sequences, a new Iterated Function System was established. That is IFS in A-life field : {Growing units + Control sequences in a embedded form + Probabilistic structure}. The probabilistic structure means the genetic information are randomly taken from a probabilistic space, and the control sequences is disturbanced. Based on it, we can start the works of growing of artificial plant tree.

3. Growth of the Artificial Plant Tree

After we had the definition of IFS and explained how to control the growing of artificial plant tree, we could make computer program to draw the picture according to the genetic information including in growing units. Now the genes are stored in growing units. For example, if we want to generate an artificial tree, we need to form the genes in trunk, branches, subbranches and leaves in advance. From these genes, we can generate the tree which we want.

The growth of each things which is vital in the nature depends on their own genes such as their size, structure, color, behavior etc.. For synthesized biology this is the process of generating artificial genes. For example, trunk genes control the tree's length and diameter. Different genetic information grows different tree, someone has good status, but someone is not. So suitable selection of genes information are necessary. In order to simulate the natural selection in the nature world we should have a population of certain specie of artificial tree, which has the same kind of genes but there is mutation on it. Then the Genetic Algorithm (GA) is used to get someone that has good status. GA is a search algorithm based on the mechanism of natural selection and natural genetics. A simple genetic algorithm that yields good results in many practical problems is composed of three operators, that is reproduction crossover and mutation. These operators, of course, are artificial versions of natural selection and evolution. The figures attached in the end of this paper show the some results of ours. The work has done on the SGI work station computer.

4. Discussion and Comparison with L-system

To establish more reasonable A-life theory and application, we should have more inside on the process of life information such as the encoding, transcription, translation and decoding etc.. In the "wet" life studies scientists pay more attention to the synthesized of protein subject. They simulate the real alive basic subjects in the real world through many very difficult and exploratory experiments. In the view of point of A-life scientists should pay more attention to the information processing in the real alive subjects. The more complicated the real subjects are, the more complicated the information processing are. Some simulations look like the growing of plant, but it is not artificial A-life study because the manner of information processing is rather different from the real process. Lindenmaryer[4] introduced the notion of parallel rewriting grammars for the modeling of developing biological systems, that is L-system. It needs many rules in L-system. In our work, we use IFS to grow trunk, branches, subbranches and leaves through the genes, and the growing is controlled by the control sequences. The rules in L-system are replaced by genes. We think that the growing of tree based on known L-system should not be A-life. Mainly, because the growing can not be controlled. The sequence in L-system consisted of zero, one,[and] is not the control sequence, but a result of growing only.

5. Conclusion

In this paper we redefine the Iterated Function System(IFS) which is different from that one in Fractal field. We try to follow the study on DNA to simulate the growing of plant tree. The IFS defined here has two main parts: one is the growing units with the genetic information, the another is the control sequence with embedded form. The artificial plant tree based on IFS defined here is different from L-system graphics and Fractal fern plant graph. We can get many different trees from the different genetic information. The processing of information in this paper is more close to the process in DNA. This paper is our starting work in A-life field, the computer experiments are rather simple. In the near future we will present more works in this field.

6. References

[1] Casti,J.L., "Reality Rules: Picturing the World in Mathematics", JOHN WILEY & SONS, INC. 1992

[2] Levy,S.," Artificial Life--a report from Frontier Where Computers Meet Biology", Vintage Books, Random House, Inc. 1993

[3] Langton,C., "Artificial Life" and " Artificial Life--II", Addison-Wesley, Redwood City, CA, 1989 and 1992

[4] Lindenmayer,A., "Mathematical Models for cellular Interactions in Development, Part 1 and Part 2", Journal of Theoretical Biology, 18, pp. 280-315, 1968

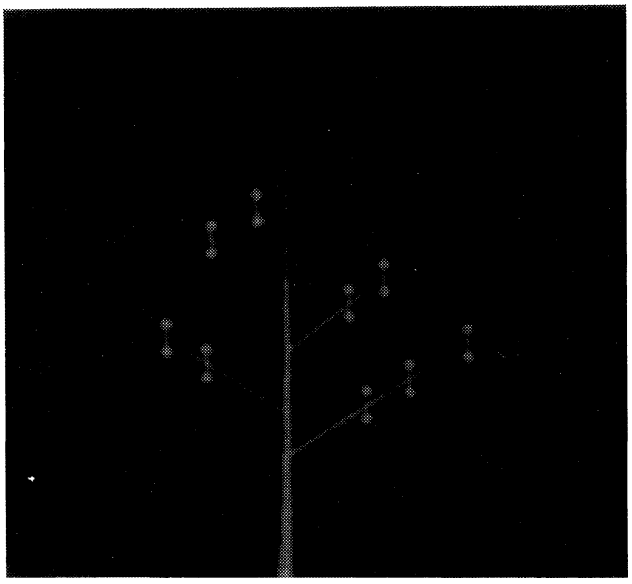
[5] Dawkins,R., "The blind Watchmaker", Longman, London, 1986

[6] Lay,T., "Artificial Brain",

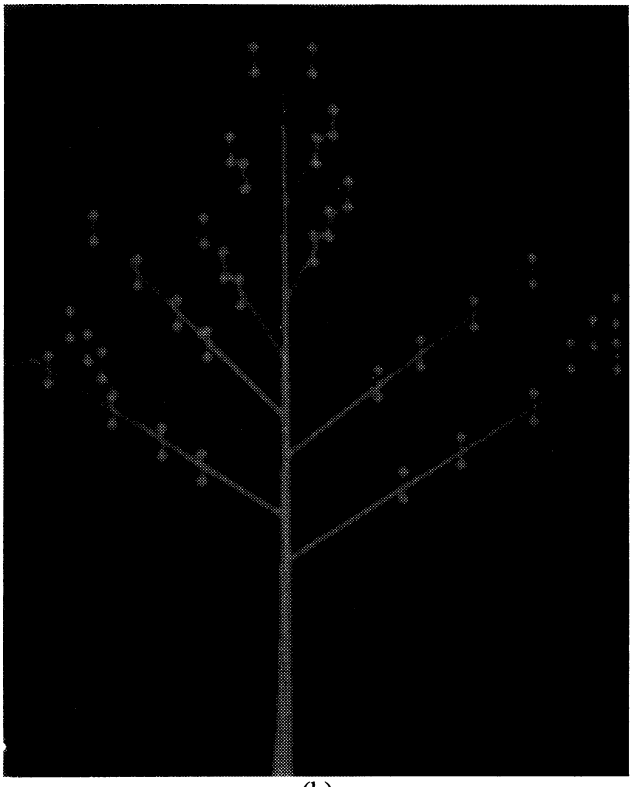
[7] Barnsley,M.F., Ervin,V., Hardin,D. and Lancaster,J. ,"Solution of an Inverse Problem for Fractals and Other Sets", *Proceedings of the National Academy of Science*, Vol.83, April, 1985

[8] Smith,A.R., "Plants, Fractals, and Formal Languages", *Computer Graphics*, 18(3), July, 1984

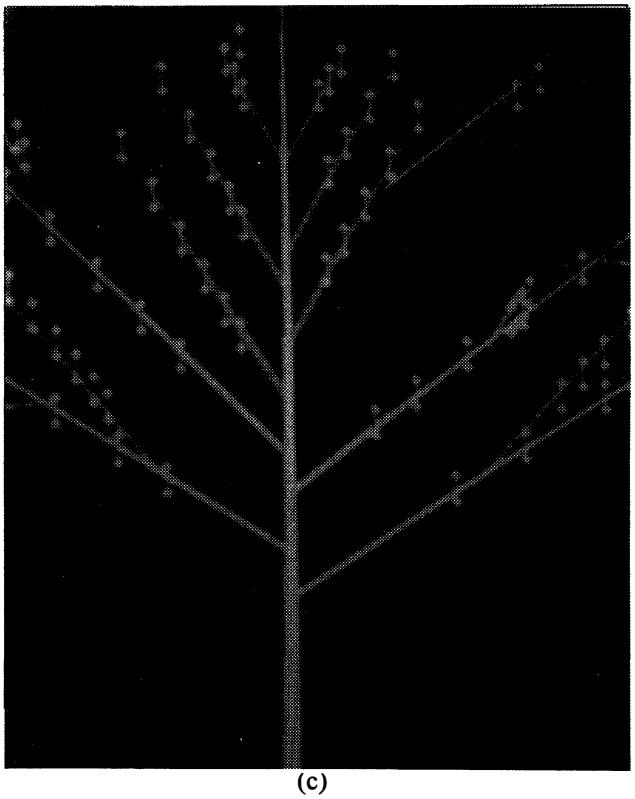
[9] Mandelbrot,B., "The Fractal Geometry of Nature", W.H.Freeman and Co., San Francisco, 1982



(a)



(b)



(c)

Fig.1 Growing of Artificial Tree based on IFS
(a) 6 years (b) 10 years (c) 12 years

Fault Diagnosis of Logical Circuit by use of Pseudorandom Signal and Neural Network

H. Kashiwagi and M.Sakata
Department of Mechanical Engineering, Faculty of Engineering
Kumamoto University, Kumamoto 860, Japan
Tel: +81-96-342-3742, Fax +81-96-342-3730
E-mail: kashiwa@gpo.kumamoto-u.ac.jp

Abstract

This paper describes a new method of pseudorandom testing of a digital circuit by use of correlation method and a neural network. The authors have recently proposed a new method of fault diagnosis of logical circuit by applying a pseudorandom M-sequence to the circuit under test, calculating the crosscorrelation function between the input and the output, and comparing the crosscorrelation functions with the references. This method, called MSEC method, is further extended by using a neural network in order to not only detect the existence of faults but also find the place or location of the faults. The author investigated the effects of using a part of fault patterns for training the neural network on the ability of fault detection of the neural network. It is shown that we can obtain more than 95 % of fault detection ration even when we use only the half of the possible training data.

1 Introduction

A functional logic testing of logical circuits is now an important technique, since the information devices including digital logical circuit are increasing so rapidly. For example, fault diagnosis of logical circuits boards is required in the process of producing any information devices.

There are mainly two methods in detecting faults of logical circuits. One is to analyse the circuits under test (CUT) logically and to design an optimal input sequence for CUT. D-algorithm, for example, belongs to this group. This method is called "deterministic method" and usually requires a large amount of computation time, and the design must be done for each kind of circuit.

The second method is to apply a random input to the circuit and the corresponding output is compared with the output of fault free circuit[3][4]. This method is called "random testing method" and has several advantages: simple hardware, low cost and applicability to any circuits.

In this connection, we have recently proposed a new method[1] of fault diagnosis of logical circuit by applying a pseudo-random M-sequence to CUT, calculating the crosscorrelation function between the input and the output, and comparing the crosscorrelation functions with the references. This method is called M-SEquence Correlation (MSEC) method, and it has a very small probability that we overlook any faults in the circuit.

In this paper, MSEC method is further extended by using a neural network in order to not only detect the existence of faults in the circuit but also find the place or location of the faults. A simple digital circuit is taken as an example, and artificial faults are inserted to the circuit. MSEC method is then applied to this circuit to obtain the crosscorrelation functions between the inputs and outputs. Since the crosscorrelation functions contain the information about faults in the circuit, they are then applied to a neural network. After training the neural network by use of the data on the faults and resulting crosscorrelation functions, the ability of the neural network to detect the location of faults is checked.

The results of the experiment show that the trained neural network has an ability not only to detect the existence of faults but also to locate the faults in the circuits.

2 Brief description of MSEC method

The basic diagram of MSEC method is shown in Fig.1. The input to the CUT is an n-th degree M-sequence a_i (a_i=0 or 1, i=1,2,...N, N=2^n-1) which is known as one of the pseudo-random sequences. The output sequence b_i (b_i=0 or 1) of the CUT is crosscor-

related with a_i. The crosscorrelation function between a_i and b_i is obtained as follows.

$$\Phi_{ab}(\tau) = N - 2(\sum_{i=1}^{N} a_i \oplus b_{i+\tau}) \quad (1)$$

where $\oplus$ denotes Exclusive OR (EOR) operation and $\sum$ is the arithmetic operation. The crosscorrelation functions $\Phi_{ab}(\tau_1), \Phi_{ab}(\tau_2), ..$ are then compared with the correct values of $\Phi_{ab}(\tau)$'s which are the crosscorrelation functions in case of fault-free circuit. When any of $\Phi_{ab}(\tau)$ differ from the correct values, we can say that there must be some fault in the CUT.

The structure of MSEC method is simple, requiring only EOR circuit and counters. In addition to this, MSEC method has a very small probability that we miss any faults in the circuit.

The undetected fault ratio of MSEC method, that is, the expected value of the probability that we overlook any faults in the CUT, is shown to be

$$D_r \simeq \frac{2^r}{(\sqrt{\pi}N)^{r+1}} \quad (2)$$

where r is the number of delay bits we use.

The undetected fault ratio of MSEC method is shown to be much smaller than the conventional one's counting, transition counting, or even CRC(cyclic redundancy check method which is rather well known as Signature Analysis method developed by Hewlett-Packerd Co. USA).

3 Data for training neural network

As is shown above, MSEC method has a very small probability that we miss any faults in the circuit, so it is very efficient to detect faults in a logical circuit. However, MSEC method can detect whether there exist any falts or not, but it cannot find the location of fault in the circuit. Since we can think that the crosscorrelation functions $\Phi_{ab}(\tau)$'s have some information on the state of the CUT, we can extract the information about the location of fault from those crosscorrelation values by applying the values of the crosscorrelation functions to a neural network. The training data for the neural network are obtained from the following experiment.

4 Experiment

As an example of CUT, a simple digital circuit (2-bit binary adder SN7482) is chosen where number of input is 5, number of output is 3(Fig.2). In this circuit, 14 switches are inserted as the simulation of possible faults. So the total number of the fault patterns is $2^{14} = 16384$. Let us denote the fault pattern as (1,2,3) in case where switches no.1, no.2, and no.3 are faulty. M-sequence of 5 degree (N=31) is applied to the input C_o, its one bit shifted veision to A_1, and likewise to B_1, B_2, A_2.

We have measured the crosscorrelation functions $\Phi_{c0-s1}(\tau), \Phi_{c0-s2}(\tau)$, and $\Phi_{c0-c2}(\tau)$, 31 values for each(total 93 values) for one fault pattern.

In order to simplify the neural network which will be used in practice, the correlation data are converted to binary ones according to whether each correlation value is positive or negative(Fig.4). Thus the corrrelation values of total 93 bits are applied to the neural network.

After measuring the correlation functions for 16384 fault patterns, it is noticed that we obtain 372 different patterns of correlation functions. That is, one correlation pattern correponds to some fault patterns. For example, the crosscorrelation pattern(Fig.3) obtained for fault (3,13,14) is the same as for (13,14), but different from those of all other fault patterns. Therefore when we obtain the correlation pattern as shown in Fig.3, we can say that switches no.13 and no.14 are faulty with 100 % probability and switch no.3 is faulty with 50 %. These probability is called here as theoretical fault probability. Thus we can calculate the theoretical probability for each correlation patterns.

These data are used for training the neural network. The neural netwok which we used has three layers: 93 nodes for input layer, 23 nodes for intermediate layer and 14 node for output layer. The algorithm used in this neural network is back propagation algorithm. The example of the output of the neural network is shown in Fig.4.

5 Effects of using partial data

One of the properties of neural network is that once all the possible inputs are used for training and the neural network converges well, the network has the ability to detect any fault with high probability. However, when only a part of possible input data are used for training, what happens about the ability of the fault detection of the neural network ?

Fig.5 shows an experimental result on the fault detection ratio vs. the number of input data for training. From this figure, we can say that we obtain more than 95 % of fault detection ratio when we use only the half of the possible input data for training.

We see from these results that the trained neural network has enough ability to find the location faults in the circuit.

6 Conclusion

A new method of fault diagnosis of logical circuit is proposed by use of pseudorandom M-sequence as its input, crosscorrelation functions between the input and output, and a neural network. This method can not only detect whether or not there exist some faults in the circuit with very small probability that we overlook any faults, but also find the location of the detected faults in the circuit.

The experiment show that we obtain more than 95 % of fault detection ratio when we use the half of the possible inpout data for training the neural network.

This is very effective in practice, since it will be very difficult to provide the complete training data for all possible faults in the circuit. Some sort of statistical data such as those concerning the frequently occurring fault points would be useful for training the neural network.

References

[1] H. Kashiwagi and I. Takahashi; A New Method of Fault Detection of a Logical Circuit by use of M-sequence Correlation Method, Trans. SICE, Vol.23, No.9, pp.993-997, 1987.

[2] H. Kashiwagi and I. Takahashi; Fault Detection of a Logic circuit by use of M-sequence Correlation Method, Proc. China-Japan Reliability Symposium held in Shanghai, pp.149-157, September 1987.

[3] T.Ohteru et.al.; Digital Circuit Test System Using Statistical Method, Proc.10th ISFTC, pp.1979-181, 1980.

[4] K.D.Wagner, C.K.Chin and E.J.McClusky; Pseudorandom Testing, IEEE Trans., Vol.C-36, No.3, pp.332-343, 1987.

[5] H.Kashiwagi and M.Sakata; Fault Diagnosis of Logical Circuit by use of Correlation and Neural Network, Proc. '92KACC held in Seoul, Korea, pp.569-572, 1992.

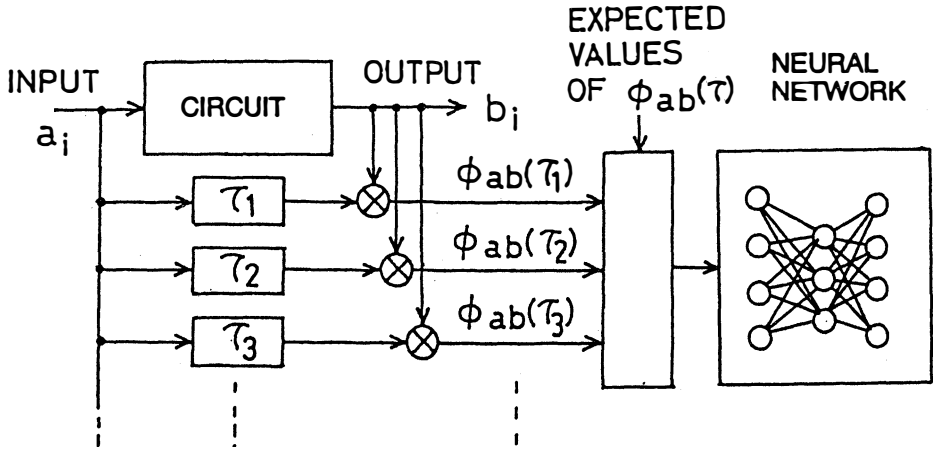


Fig.1 Basic diagram of fault diagnosis system

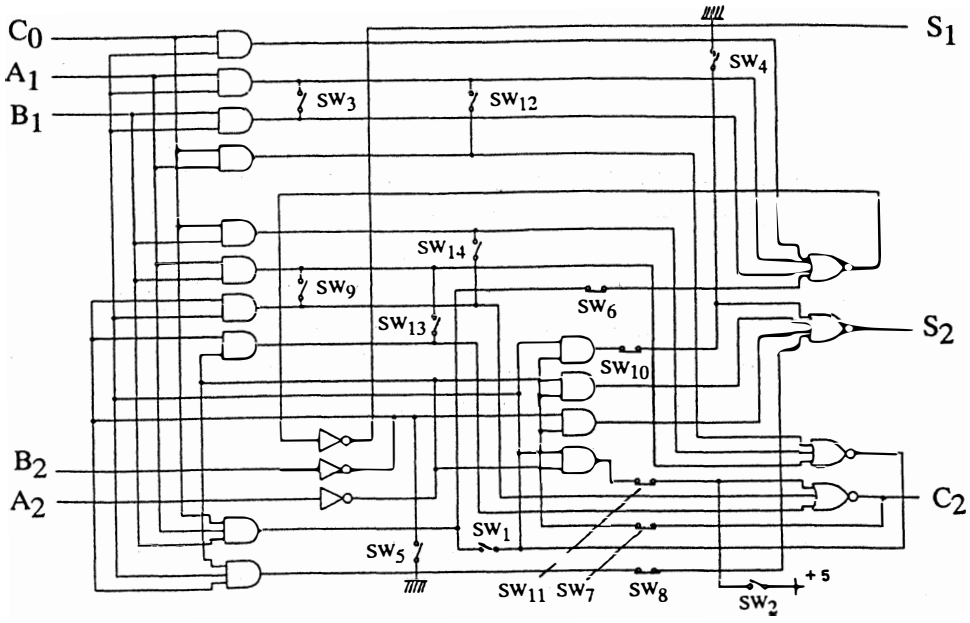


Fig.2 An example of CUT.

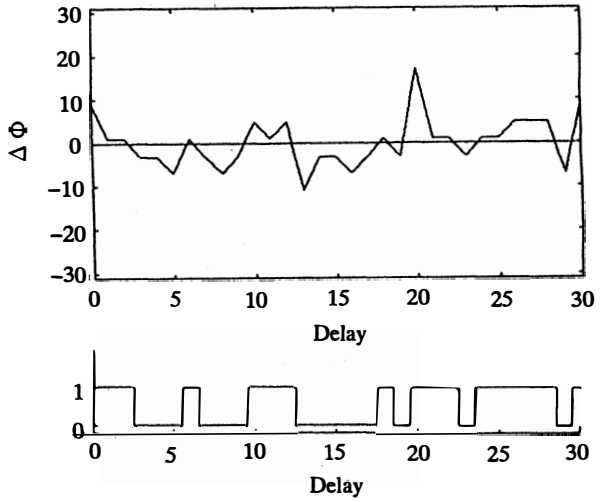


Fig.3 Difference of crosscorrelation values and its binary version

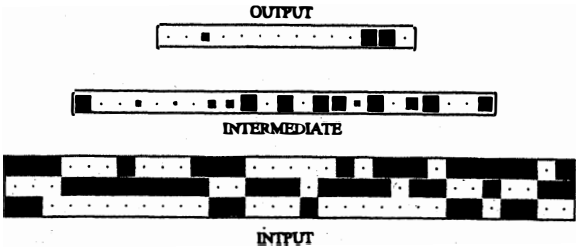


Fig.4 An example of output of the neural network

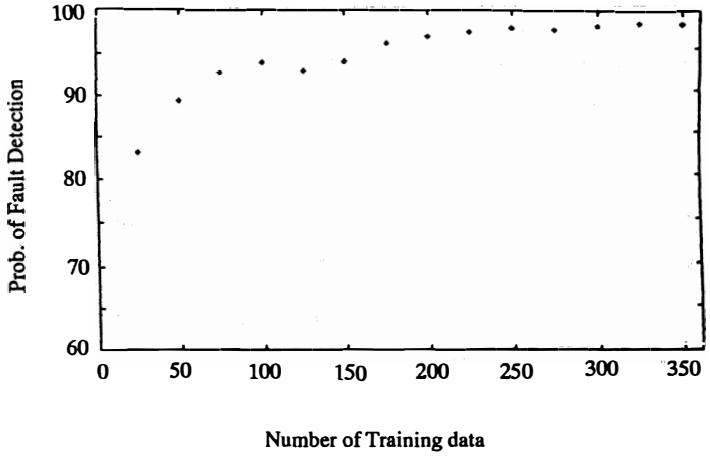


Fig.5 Probability of fault detection vs. number of training data

Computational Study on Cyclic Structure using Abstract Rewriting System on Multisets

Yasuhiro Suzuki, Shusaku Tsumoto, and Hiroshi Tanaka
Department of Information Medicine
Medical Research Institute, Tokyo Medical and Dental University
1-5-45 Yushima, Bunkyo-ku Tokyo 113 Japan
Email: {suzukcom, tsumocom}@cmn.tmd.ac.jp, tanaka@cim.tmd.ac.jp

Abstract

One of the most essential temporal structure in life systems is a cycle, which can be observed in any hierarchy of living things, such as TCA cycle in cytoplasmic level, the cell cycle in cell division, and the life cycles of living things. Although the importance of this structure has been pointed out by several researchers, they do not focus on the conditions under which such cycle structures will emerge and become stable. In this paper, an abstract rewriting system on a multiset is introduced to a model chemical reaction as a symbolic rewriting system acting on a multiset, which can be viewed as a chemical reaction system in a test tube. By use of this model, the condition of a cycle emergence, its robustness and how this system works under the probabilistic condition are examined. The results show not only that a cycle will emerge even under a simple initial condition, but also that complex behavior of a cycle, such as fusion of several cycles, is observed when input randomness is introduced.

Keywords: Abstract rewriting system, Cyclic structure, Emergent computation.

1 Introduction

One of the most essential temporal structure in life systems is a cycle, which can be observed in any hierarchy of living things. For example, in metabolic pathways inside the cell, TCA cycle is used to extract energy from organic chemicals, which plays an important role in cell metabolism[6]. Also, the cell cycle in cell division is important for the growth of life systems, and the life cycle can be observed in all living things[6]. Thus, a cycle can be viewed as a universal structure in all the hierarchies of complex systems, especially in living things, whose importance has been pointed out by several researchers of complex system, such as Eigen's hypercycle[2], Maturana's autopoiesis[5], Kauffman's NK network[4], and Fontana's Algorithmic Chemistry[3]. However, these researches do not focus on the conditions under which such a cycle will emerge and the process in which it becomes stable. In this paper, we introduce a new abstract rewriting system on multiple sets (ARMS) in order to examine the conditions of cycle emergence and the algebraic characteristics of the obtained cycle. In this model, a multiset [1] is taken as a role of a test tube, which contains "symbols", which correspond to chemical compounds. Then, rewriting rules, corresponding to chemical reaction formulae, act on this multiset, and the applied symbols are transformed into other symbols, corresponding to chemical reactions. Furthermore, we assume that the specific symbol a is input to this multiset. Interestingly, it is easily shown that even such a simple structure can emerge a cycle, although the structure of rules is strongly constrained to do that. Then, we further examine the condition of emergence of a cycle, such as the effect of inputs and that of the order of rules. The results show not only that a cycle will emerge even under a simple initial condition, but also that complex behavior of a cycle, such as fusion of several cycles and period-doubling, is observed when the initial condition is perturbed during the simulation. These results suggest that randomness of input is one of the principal factors on the complex behavior.

2 ARMS

Our computational model is based on the abstract rewriting system [1] because of the following three reasons. First, chemical reactions, which are important in our model, can be viewed as rewriting rules. For

[1] A multiset is defined as a set which is allowed to include the same elements. For example, $\{a, a, b, c\}$ is not a set, but a multiset, while $\{a, b, c\}$ is both a set and a multiset.

example, $\frac{1}{2}O_2 + H_2 \rightarrow H_2O$ is a kind of a rewriting rule. Thus, the abstract rewriting system gives a formal model of chemical reactions. Second, abstract rewriting system does not need a specific architecture like a turing machine [1] and cell automata[7]. Finally, third, we can apply the results of researches on the abstract rewriting systems, such as those on rewriting pathways, and rewriting rules. Using the concepts of abstract rewriting system, we introduce a new abstract rewriting system on multisets (ARMS) in order to examine the conditions of emerging a cycle and the algebraic characteristics of the obtained cycle. ARMS is defined as follows.

Definition 1 (ARS on Multisets(ARMS)) *Abstract rewriting system on multisets is defined as a quintuple (A, R, O_R, C, I), where A, R, O_R, C and I denote a set of symbols, a set of rewriting rules, rule order, a multiset of symbols, and a set of input strings, respectively.* □

In this model, a multiset C is used as "a test tube", which contain "symbols" in A, which correspond to chemical compounds. Then, rewriting rules R, corresponding to chemical reactions, act on this multiset, and the applied symbols are transformed into other symbols. Moreover, for simplicity, we assume that no character outputs from a multiset C. In this paper, we also focus on the effect of finiteness of a test tube, thus we assume that C is finite. Especially, for simplicity, the cardinality of C, $|C|$ is set to 10 ($|C| = 10$) in our experiments. Then, normal form can be defined as follows.

Definition 2 (Normal Form in ARMS) *If no rule in R can be applied to C and if no string in I can be input to C, then then it is said that C is in normal form.* □

Thus, normal form corresponds to the steady state, or the death of living things. Furthermore, for experiments, we specify the characteristics of ARMS in the following way.

Initial Condition As to the initial condition, A and I is set to $\{a, b, c, d, e\}$ and $\{a\}$, respectively. Then, the cardinality of $|C|$ is set to 10. Finally, rewriting is applied for 5005 steps, which is equal to the total size of search space of multiple set C.

Initial State and Final State Initial state of C is set to an empty set ($C = \{\}$). On the other hand, final state is defined as a normal form defined in the above subsection. That is, *if no rule in R can be applied to C and if no string in I can be input to C, then then C is in a final state.*

Order of Rules In ARMS, we assume that one rule is applied in each rewriting step unless no input is allowed based on a given rule order O_R. For example, let us consider the case when $R = \{r_1, r_2, r_3\}$ and $O_R = \{r_1 \rightarrow r_2 \rightarrow r_3\}$. Then, each rule is applied in the following way. In Step 1, r_1 is applied. In the next steps, Step 2 and 3 , r_2 and r_3 are applied, respectively. In Step 1, r_1 is applied again. Thus, transition of the order in each step can be viewed as a "shift" operation on a bit string.

3 Experimental Results

Throughout the experiments, R is set to $\{r_1, r_2, r_3, r_4, r_5, r_6\}$, where each rule is described as the following formulae.

$$a\,a\,a \rightarrow b : r_1, \quad b\,a \rightarrow c : r2, \quad d\,e \rightarrow a : r3,$$
$$c \rightarrow d\,d : r4, \quad d \rightarrow e : r5, \quad d \rightarrow c : r6.$$

3.1 Rule Order and Emergence of Cycles

We, first, examine the relationship between rule order and emergence of cycles. In the above rule set R, the total number of rule order is equal to $6! = 720$. For each order, we assume that a will be input after each rewriting step and check whether cycles will emerge or not.

Surprisingly , in only 9.4 per cent(68/720) of 720 kinds of order, cycles emerge, and in the rest, 90.6 per cent, rewriting calculus is terminated. In 68 kinds of order where cycles emerge, steps needed to emerge a cycle are 15 to 22 steps, and its average and its standard deviation is 19 and 3.3 steps, respectively. The average of the length of cycles is 5.5 steps, whose standard deviation is 0.49 steps.

As to terminated cases, the average of steps needed for termination is 39 steps, and its standard deviation is 6.48 steps. For further examination on the effect of inputs and rule order, we adopt the following two kinds of order, O_{R1} and O_{R2},

$$O_{R1} = \{r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_4 \rightarrow r_5 \rightarrow r_6\},$$
$$O_{R2} = \{r_3 \rightarrow r_1 \rightarrow r_2 \rightarrow r_4 \rightarrow r_5 \rightarrow r_6\},$$

because of the following two reasons:

The behavior of O_{R1} is a typical cycle structure in 68 cases. Second, the behavior of O_{R2} is completely different from the above one, although ARMS applies the same rules in these two cases.

3.2 Effect of Inputs and Rule Order

In this subsection, we examine the effect of random inputs. After that, we introduce randomness in rule order and examine the effect of rule order randomness. In order to incorporate randomness, we introduce characteristic function as follows.

Definition 3 (Characteristic Function) *Let x be a uniform random number selected from $[0,1)$. Then a characteristic function $h_\delta(x)$ is defined as:*

$$h_\delta(x) = \begin{cases} 1 & (x < \delta) \\ 0 & (x \geq \delta), \end{cases}$$

where δ denotes a threshold from the following set, $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. □

For example, when δ is equal to 0.9, then the frequency, with $h_\delta(x)$ positive, is equal to 0.9. Thus, when this function is applied to inputs, a will be input after nine of ten rewriting steps, which also means that no input will be done in one of ten steps. On the other hand, when the characteristic function is applied to rule order, the rule order will not be perturbed in nine of ten rewriting steps, which means that rule order is perturbed in one of ten steps. For example, let us consider a case of O_{R1}. When the sixth step in ten rewriting steps is perturbed, rule is applied as shown in the following sequence: r_1, r_2, r_3, r_4, r_5, $r_2(r_6)$, r_1, r_2, r_3, r_4, where r_6 is changed to r_2. Thus, both input randomness and rule randomness is equal to $1 - h_\delta(x)$. Using the above two kinds of order, we make the following three kinds of experiments: (1) Decrease the frequency of input, with rule order fixed, (2) Introduce randomness in rule order, with inputs in every step, and (3) Decrease the frequency of input, and introduce randomness in rule order. In each case, we perform 100 trials, each of which consists of 5005 rewriting steps, and measure the effect on emergence of cycles, extinction of cycles, and steps needed for termination.

Decrease Frequency of Input, with Rule Order Fixed In the above three kinds of experimental results, the most interesting one is cases when the frequency of input is decreased, with the rule order fixed. In the cases of rule order O_{R1}, rewriting steps are not terminated in all the trials, and cyclic structure also emerges. Interestingly, the length of cycle varies over 7 to 30 steps, and period-doubling of cycles is observed. On the other hand, in the cases of rule order O_{R2}, unstable cycles emerge and rewriting steps are terminated while the threshold is larger than 0.5. However, when the threshold is equal to 0.4, almost the all rewriting trials are not terminated, where cycles become stable. Moreover, at this point, the variability of the length of cycles is largest among the all trials. The peaks of the length of cycles is located around 11, 23, 34, 46, 57, and 67 steps, which shows period-doubling. We closely examine these experiments in Section 5, which shows that several kinds of cycle fusion is observed. If the threshold is less than 0.3, all the trials are stable, similar to the cases of rule order O_{R1}. These results are summarized into Table 1.

Table 1: Relation between Frequency of Inputs (δ) and Termination of Calculus

δ	$0.7 \geq$	0.6	0.5	0.4	≤ 0.3
O_{R1}	0	0	0	0	0
O_{R2}	100	100	98	13	0

Rule Order: Fixed

Table 2: Relation between Randomness of Rule Order ($1 - \delta$) and Number of Emerging Cycles

1 - δ	0.1	0.3	0.5	0.7	0.9
O_{R1}	1300(5)	325(5)	106(5)	106(5)	23(5)
O_{R2}	1269(11)	1118(5)	335(3)	223(3)	300(3)

A(B) : A and B denote the number of emerging cycles, and the median of the length of cycles, respectively.

Randomness in Rule Order, with Inputs in Every Step In these experiments, as to rule order O_{R1}, cycles become unstable and period-doubling is observed in O_{R1} when randomness of rule order is low. However, as randomness of rule order grows large, the number of emerging cycles decreases exponentially. Table 5 shows this tendency, the relation between randomness of rule order and the number of cycles emerging in 100 trials. When the randomness is larger than 0.5, almost the all trials generate only one cycle in general. On the other hand, the behavior of cycles in O_{R2} is different. As shown in Table 5, the number of emerging cycles decreases exponentially, as randomness of rule order grows large. However, the behavior of cycles when randomness of rule is higher than 0.5 is different from that when the randomness is lower than 0.3: the median of the length of cycles in the latter cases half smaller than that in O_{R1}, while that in the former cases is equal to or twice than that in O_{R1}. Interestingly, trials in the case of O_{R2} have two phase transitions: one is between 0.1 and 0.2, and the other

is between 0.3 and 0.5. It is notable that the cycles whose length is 11.0 are dominant over those whose length is 5.0.

Decrease Frequency of Input and Increase Randomness in Rule Order In the final cases when the frequency of input is decreased and randomness is introduced in rule order, the obtained results are straightforward. In the cases when randomness is introduced in rule order with an input in every step, cycles become unstable in O_{R1}, and only unstable cycles emerge in O_{R2}. However, in both cases, the number of emerging cycles decreases exponentially, as randomness of rule order grows large. Interestingly, the behavior of cycles in O_{R1} is very similar to that in O_{R2}, except when the frequency of input is very high and randomness of rule order are very low. In the exceptional cases, both cases are very similar to the cases in Subsection 4.1, except for emergence of cycles in the case of O_{R1}.

4 Discussion

As discussed in Section 4, a special case in rule order O_{R2}, with the frequency of input 0.4, generates complex behavior of cycles. Closer examination shows that a core cycle generates first and that other cycles around this cycle fuse into a core cycle in rewriting steps.The generated core cycle c_8 is given as follows:

$$c_8 : 8 \to 9 \to 10 \to \cdots \to 18 \to 8 : (11steps),$$

where each number denotes the number of rewriting steps. Around the c_8, the following cycle c_{54} is also observed:

$$c_{54} : 54 \to 55 \to 56 \to \cdots \to 66 \to 54 : (13steps),$$

It is impossible to transit from State 8 to State 54 and State 66 to State 11, with input fixed, but it will become possible when randomness of input is introduced. Then, c_8 fuses into c_{11}, and a new cycle will be generated as follows:

$$c_c : 8 \to 54 \to \cdots \to 66 \to 11 \to \cdots \to 8 : (21steps)$$

Thus, fusion of cycles depends on randomness of input: randomness of input influences the application of rules, and then change of application of rules causes the change of structure of a multiset, then it opens the hidden pathway to fusion of cycles. Although the mechanism of fusion of cycles is not fully studied, it is notable that even such a simple structure emerges a complex behavior. Further research on fusion, such as classification of cycles, will be a future work.

5 Conclusion

In this paper, we introduce a new abstract rewriting system on multisets, and model chemical reactions as a symbolic rewriting system which acts on a multiset, which can be viewed as a test tube. Using this model, we examine the condition of emerging a cycle, its robustness and how this system works under the probabilistic condition. The results show not only that a cycle will emerge even under a simple initial condition, but also that complex behavior of a cycle, such as fusion of several cycles, is observed when the initial condition is perturbed during the simulation.

Acknowledgments

This research is supported by Grants-in-Aid for Scientific Research No.07243203 from the Ministry of Education, Science and Culture in Japan.

References

[1] N. Dershowitz and J. P. Jouannaud, Rewrite Systems, in Handbook of theoretical computer science 245-309, Elsevier. 1990.

[2] M. Eigen and P. Schuster, The Hypercycle, Springer-Verlag, 1979.

[3] W. Fontana and L. W. Buss, The Arrival of the fittest: Toward a Theory of biological organization, Bulletin of Mathematical Biology, Vol.56, No.1, 1-64, 1994.

[4] S. A. Kauffman, The Origins of Order, Oxford University Press, 1993.

[5] H. R. Maturana and F. J. Varela, Autopoiesis and Cognition, D. Reidel Publishing Company, 1980.

[6] J. D. Watson, N. H. Hopkins at el, Molecular biology of the gene, The Benjamin/Cummings publishing Company, Inc, 1992.

[7] S. Wolfram, Cellular Automata and Complexity, Addison Wesley, 1994.

Identification of a Dissipative Transport System

Alan P. Wang & Cecilia Y. Wang
Department of Mathematics
Arizona State University
Tempe, AZ 85287-1804
and
Sueo Ueno
Information Science Laboratory
Kyoto School of Computer Science
Kyoto 606, Japan

Abstract

The aim of this paper is to identify the properties of an unknown medium of a transport system from measurements of outputs at the boundaries. A new method is developed here based on the inverse problem and the properties of a scattering matrix and its generator.

1 Introduction

A great deal of interest has developed in recent years in the identification of a transport system, that is, to identify the physical properties of an unknown medium based on measurements of outputs at the boundaries of a medium. Such identification problems arise in many physical problems, such as transmission lines, signal processing, acoustic propagation, radiative transfer, etc.

A transport system consists of inputs and outputs of a medium. Inputs at the boundaries are combinations of transmitted and reflected inputs. The physical properties of the medium which govern the inputs and outputs are called *medium coefficients.*

It is well known [1], [3], [6] that the reflection and transmission can be determined by solving a set of Chandrasekhar's equations with known medium coefficients. This is equivalent to saying that outputs are determined if inputs and medium coefficients are known. This is usually called the *direct problem.*

The *inverse problem* considered here is that if inputs and outputs are known at the boundaries, we wish to determine the medium coefficients inside the medium. In general, the medium is inhomogeneous with anisotropic scattering.

The method presented here is based on comparison and on the properties of a scattering matrix and its generator. We establish relations between two scattering matrices; one is known and the other is determined by measurement outputs at the boundaries. The relationship between these two scattering matrices leads to that between the two media. Hence the unknown medium is determined.

Research similar to this paper, in the sense of using scattering matrices, is by Kailath [2], Ueno and Wang [6], [8], [5]. The former developed the layer peeling and layer-adjoining methods. The latter used scattering matrices to solve the inverse problem which focused on the determination of the boundary reflection. Both have many interesting applications.

The method developed here is surprisingly simple and computational algorithms can be implemented. We have not discussed numerical convergence and computational stability. Our main purpose is to develop this unique method with the hope that it will have applications.

Since we use the scattering matrix, a section is devoted for introduction to and review of some properties which are essential for our analyses. The following section is on the development of our identification method. Mathematical analyses are presented in the appendix. The last section is for general remarks, including the treatment of an identification problem with homogeneous medium.

2 Scattering Matrix

In this section we give a review of the scattering matrix of a transport system. Certain properties are

essential to our method and our analyses. A *scattering matrix* $S = S(x,y)$ of a medium extended from x to y is given by [1], [3],

$$S = S(x,y) = \begin{pmatrix} t & \rho \\ r & \tau \end{pmatrix}. \tag{1}$$

where t and τ are left and right transmission operators and ρ and r are left and right reflection operators. The scattering matrix maps inputs to outputs at boundaries, i.e.,

$$S : \begin{pmatrix} v(x) \\ u(y) \end{pmatrix} \longrightarrow \begin{pmatrix} v(y) \\ u(x) \end{pmatrix} \tag{2}$$

inputs outputs

Physical properties of inputs and outputs are dependent on the particular transport system under consideration. In radiative transfer they are specific intensities. For example, $u(x) = f(x,\Omega)$, $\Omega = (\theta,\varphi)$, to signify that they are functions dependent on x and on direction, φ. In n-port transmission line theory, they are n-dimensional vectors of electric waves, etc. Operators are linear mappings. If there is only an input $v(x)$ at boundary x, then $tv(x) = v(y)$ which denotes the amount of input $v(x)$ transmitted to the output $v(y)$ at boundary y. Likewise, $rv(x) = u(x)$ which indicates the amount of inputs $v(x)$ is reflected to the output $u(x)$ at boundary x. Likewise, we have similar interpretations for ρ and τ.

The transport system satisfies the principle of *invariant imbedding* [1]. That is, when two media are assembled, the individual medium properties are not affected by the existence of the other. Then when two media are combined, the inputs and outputs at the boundaries of the combined media are governed by [3].

$$S(x,z) = S(x,y) * S(y,z). \tag{3}$$

The associative multiplication on the right of eq. (3) is not the usual matrix multiplication. It is called $*$ product which is

$$\begin{aligned} S * S_1 &= \begin{pmatrix} t & \rho \\ r & \tau \end{pmatrix} * \begin{pmatrix} t_1 & \rho_1 \\ r_1 & \tau_1 \end{pmatrix} \\ &= \begin{pmatrix} t_1(E-\rho r_1)^{-1}t & \\ r + \tau r_1(E-pr_1)^{-1}t & \\ & \rho_1 + t_1\rho(E-r_1\rho)^{-1}\tau_1 \\ & \tau(E-r_1\rho)^{-1}\tau_1 \end{pmatrix}, \end{aligned} \tag{4}$$

where $S = S(x,y)$ and $S_1 = S(y,z)$ for convenience and $E = \text{ident}_y$. We assume there is no "critical case," i.e., all operators are bounded.

The associative $*$ product governs the algebraic structure of the transport system at which the principle of invariant imbedding holds. The generalization of eq. (3) leads to

$$S(x,h) = S_1 * S_2 \cdots * S_n \triangleq \prod_{i=1}^{n} S_i \tag{5}$$

where $S(x,h)$ denotes the scattering matrix associated with a medium extended from x to h, $h \geq x$ and (x,h) are partitioned into $x = x_0, x_1, x_2, \ldots, x_n = h$. Each S_i denotes the scattering matrix associated with a medium extended from x_i to x_{i+1}, $x_{i+1} \geq x_i$.

In the case where a medium extended from x to y is imbedded in a *free space*, then when $x = y$ there are no reflections and all inputs are transmitted. That is,

$$S(x,x) = \begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix} \tag{6}$$

It is well known that the scattering matrix satisfies the following nonlinear system [3], [6], [7]

$$\frac{\partial}{\partial y} S(x,y) = \begin{pmatrix} (b+\rho c)t & a + b\rho + \rho d + \rho c\rho \\ \tau ct & \tau(d+cp) \end{pmatrix}. \tag{7}$$

where a, b, c and d are medium coefficients. They are related to the physical properties of a medium. Therefore medium coefficients determine the scattering matrix by solving equation (8) with initial value (7). That is called the direct problem.

3 Identification

Let us consider two transport systems, one with known medium coefficient M and the other with unknown coefficient $\tilde{M}$, and

$$M(z) = \begin{pmatrix} b & a \\ c & d \end{pmatrix}$$

and

$$\tilde{M}(z) = \begin{pmatrix} \tilde{b} & \tilde{a} \\ \tilde{c} & \tilde{d} \end{pmatrix}, \quad x \leq z \leq y', \tag{8}$$

where coefficients a, b, c and d are identical to those in eq. (8). The precise *identification problem* is that we will determined $\tilde{M}(z)$ for all $x \leq z \leq y$ from the knowledge of outputs at boundaries, x and y. We present here a step by step method to solve this problem. Some mathematical analyses are presented in the appendix.

i) *To determine* $S(x,y)$.

This is done by the direct method. For discussion of this, see [3], [6], [4]. Because eq. (8) is nonlinear the analysis is much more difficult than, say, a linear equation. However, we present an approximation method here.

The approximation method takes advantage of the algebraic structure of the $*$ product, as in eq. (6). For Δx_i small and eq. (7), we have

$$S(x_i, x_{i+1}) - S(x_i, x_i) \tag{9}$$
$$= \begin{pmatrix} [b(\bar{x}_i) + \rho(x_i, \bar{x}_i)c(\bar{x}_i)]t(x_i, \bar{x}_i) & - \\ - & - \end{pmatrix} \Delta x_i$$

for some $x_i \leq \bar{x}_i \leq x_{i+1}$ by the mean value theorem. Eq. (9) leads to

$$S_i = S(x_i, x_{i+1}) = E + M(\bar{x}_i)\Delta x_i. \tag{10}$$

And eq. (5) can be rewritten as

$$S(x, y) = \prod_{i=1}^{n} [E + M(\bar{x}_i)\Delta x_i] \tag{11}$$

The choice of $\bar{x}_i$ needs computation in the particular problem under consideration. Even with a poor choice of $\bar{x}_i$, or merely letting $\bar{x}_i = x_i$, the convergence is uniform as $n \to \infty$.

Eq. (11) involves many $*$ product multiplications. The following discussion is to facilitate the computation.

Let us use Redheffer's $\wedge$ operation [3], i.e.,

$$\wedge : S_i \longrightarrow \hat{S}_i = S_i^{\wedge} = \begin{pmatrix} t_i^{-1} & -t_i^{-1}\rho_i \\ r_i t_i^{-1} & \tau_i - r_i t_i^{-1}\rho_i \end{pmatrix}. \tag{12}$$

Eq. (12) requires t_i^{-1} to exist. This condition is always satisfied for Δx_i small, see [7]. Use the properties

$$(S_1 * S_2)^{\wedge} = \hat{S}_1 \cdot \hat{S}_2 \quad \text{and} \quad S = \hat{\hat{S}} \tag{13}$$

where $\cdot$ is the usual matrix multiplication. We obtain

$$S(x, y) = (\hat{S}_1 \cdot \hat{S}_2 \cdots \hat{S}_n)^{\wedge}. \tag{14}$$

Eq. (14) is easier to compute than eq. (5). Upon substitution of eq. (11) for S_i, we have solved for $S(x, y)$. The conclusion is

Given $M(z)$, for all $x \leq z \leq y$, we can obtain $S(x, y)$ or estimate $S(x, y)$.

ii) *To determine $\tilde{S}(x, y)$.*

To determine $\tilde{S}(x, y)$, the scattering matrix for an unknown medium, we shall consider whether inputs and outputs are vectors or functions. In the vector case, $v(x) = (v_1(x), v_2(x), \ldots, v_n(x))$, $u(y) = (u_1(y), u_2(y), \ldots, u_n(y))$, etc. Then transmission $\tilde{t} = \{t_{jk}\}$ and reflection operators $\tilde{r} = \{r_{jk}\}$ are $n \times n$ matrices and

$$t_{kj} v_j(x) = v_k(y) \quad \text{and} \quad r_{jk} v_j(x) = u_k(x) \tag{15}$$

By knowing inputs $v_j(x)$ and outputs $v_k(y)$ and $u_k(x)$, $\tilde{t}$ and $\tilde{r}$ can be obtained. Likewise, we can obtain $\tilde{\tau}$ and $\tilde{\rho}$. Therefore, $\tilde{S}(x, y)$ is determined. For the function case, $\tilde{t}, \tilde{\tau}, \tilde{\rho}$ and $\tilde{r}$ are integral operators,

$$\tilde{t} \cdot v(x) = \int \tilde{t}(\Omega - \Omega') f(x, \Omega')\, d\Omega' = g(y, \Omega) \tag{16}$$

where $\Omega = (\theta, \varphi)$ denotes the direction $f(x, \Omega')$ is the input $v(x)$ in the direction Ω', $g(y, \Omega)$ is the output in Ω, and $\tilde{t}(\Omega, -\Omega')$ is the kernel. Knowing $\tilde{f}$ and $\tilde{g}$, the transmission operator can be solved by the convolution theorem, i.e.,

$$\tilde{t} = F^{-1}(F(\tilde{t})) = F^{-1}\left\{ \frac{1}{2\pi} F_1(g) \cdot F_1(f) \right\}, \tag{17}$$

where F is the Fourier transform. In practice one may use Fast-Fourier series to speed up the computation. Likewise for other operators. Therefore, $\tilde{S}(x, y)$ is determined.

By measurements of inputs and outputs at boundaries x and y, we obtained an estimate $\tilde{S}(x, y)$.

We have used the fixed-point theorem by Wang [7] for the diagonalization of a dissipative scattering matrix. For application in radiative transfer, this condition is easily satisfied in case the volume attenuation is not zero. The other assumption we have used throughout this paper is that the transmission operator is nonsingular. This condition is always satisfied for a medium width less than the distance to the conjugate point [7]. In the case of radiative transfer, the condition that t and τ are singular restricts us to consider a finite thickness medium if the attenuation is not zero.

iii) *Solution for unknown $\tilde{M}(z)$.*

In the appendix, we establish that if $S(x, y)$ and $\tilde{S}(x, y)$ are dissipative and transmission operators are nonsingular, then there exists a constant nonsingular matrix Q such that

$$\hat{\tilde{S}}(x, y) = Q^{-1} \cdot \hat{S}(x, y) \cdot Q. \tag{18}$$

Since we have obtained S and $\tilde{S}$, $\hat{S}$ and $\hat{\tilde{S}}$, we merely employ the $\wedge$ operations on them. The matrix Q can be obtained by the standard numerical method at least for the vector input and output case. And Q can be estimated in the function case.

Once Q is obtained, equation (A-14) in the Appendix provides us a formula to compute the infinitesimal generator $\tilde{N}(z)$ for all $x \le z \le y$. The $\tilde{M}(z)$ of the unknown medium is solved,

$$\tilde{M}(z) = \begin{pmatrix} E & 0 \\ 0 & -E \end{pmatrix} \cdot \tilde{N}(z) \qquad (19)$$

where

$$\tilde{N}(z) = Q^{-1} \cdot N(z) \cdot Q. \qquad (20)$$

In case $Q = E$ = identity, then $\tilde{M}(z) = M(z)$. We can consider that the unknown medium coefficient $\tilde{M}(z)$ is a perturbation of $M(z)$ under the operation Q. Letting

$$\Delta M(z) = \tilde{M}(z) - M(z) \qquad (21)$$

and substituting $M(z)$ and $M(\tilde{z})$ into $\tilde{S}(z, z+\Delta z) = Q^{-1} \cdot S(z, z+\Delta z) \cdot Q$, and taking the limit as $\Delta z \to 0$, we obtain the following perturbation equation

$$\Delta M(z) = JQ^{-1}JMQ - M \qquad (22)$$

where

$$J = \begin{pmatrix} -E & 0 \\ 0 & E \end{pmatrix}. \qquad (23)$$

In case ΔM is small, we will expect Q to be nearly equal to the identity. This perturbation of a near identity also guarantees the existence of Q.

4 General Remarks

i) *The Method*

The result presented here on identification of a transport system is new and unique. We believe our method can be easily implemented.

The basic solution involves the following steps: a) Find a medium coefficient $M(z)$ closely related to the unknown medium coefficient $\tilde{M}(z)$. b) Compute the scattering matrix $S(x, y)$ associated with $M(z)$ by the direct method. c) Compute the scattering matrix $\tilde{S}(x, y)$ associated with $\tilde{M}(z)$ by measurements of inputs and outputs at boundaries x and y. d) Use a numerical method to obtain Q, see eq. (19). e) Compute $\tilde{N}(z)$, $\tilde{M}(z)$ by eq. (20) and (21).

ii) *The homogeneous case*

The homogeneous case is a special case considered here. The method outlined in i) applies. However, an alternate method can be used to solve this special case.

For the homogeneous case, we set the partitions on (x, y), having equal width, $\Delta x_i = \Delta x$. Then all S_i are equal, $i = 1, 2, \ldots, n$. Then, by taking the $\wedge$ operation on eq. (4), $h = y$ in this case, we have

$$\hat{S}(x, y) = \hat{S}_i \cdot \hat{S}_i \cdots \hat{S}_i = \hat{S}_i^n.$$

Since $S_i = E + M\Delta x$, we obtain

$$\hat{S}_i = \begin{pmatrix} E - a\Delta x & b\Delta x \\ c\Delta x & E + d\Delta x \end{pmatrix}$$

by eq. (13). If we choose $n = 2^k$ and assume $\hat{S}(x, y)$ is positive definite, then $\hat{S}_i$ and a, b, c, d can be solved. This is a very simple method; however, it is rather difficult to prove $\hat{S}(x, y)$ is positive definite based on some physical reason.

Appendix

If the scattering matrix is dissipative (the precise definition of dissipative used here is that it is reflective—left and right—dissipative, see [7]) in addition to the assumption that t and τ are nonsingular, then by Theorem 6 of [7], there exist two unitary matrices $U_1(z)$ and $U_2(z)$ such that

$$U_1(z) * S(x, z) * U_2(z) = D(x, z), \qquad (A\text{-}1)$$

where D is a diagonalized scattering matrix

$$D(x, z) = \begin{pmatrix} t_0(x, z) & 0 \\ 0 & \tau_0(x, z) \end{pmatrix}. \qquad (A\text{-}2)$$

This scattering matrix has zero reflections. The $\wedge$ product is well defined on $U(z)$ since $E^\wedge = E$ and $(S_1 * S_2)^\wedge = \hat{S}_1 \cdot \hat{S}_2$. Then it follows that t and τ nonsingular implies t_0 and τ_0 are nonsingular. It follows that D has an inverse under the $*$ product, and

$$U_1(z) * S(x, z) * U_2(z) * D^{-1}(x, z) = E. \qquad (A\text{-}3)$$

Likewise, for $\tilde{S}(x, y)$, we can solve for $\tilde{S}$ in terms of S, i.e.,

$$\tilde{S}(x, z) = P(x, z) * S(x, z) * Q(x, z) \qquad (A\text{-}4)$$

where

$$P(x, z) = \tilde{U}_1^{-1}(x, z) * \tilde{D}(x, z) * U_1(x, z)$$

and

$$Q(x, z) = U_2(x, z) * D^{-1}(x, z) * U_2^{-1}(x, z).$$

As $z \to x$, $\tilde{S}(x, z)$ and $S(x, z)$ both approach E, that is,

$$P(x, x) * Q(x, x) = E.$$

and

$$P(x,x) = Q^{-1}(x,x) \qquad \text{(A-5)}$$

under the usual matrix inverse.

Let N and $\tilde{N}$ be the infinitesimal generators of $S^{\wedge}(x,z)$ and $\tilde{S}^{\wedge}(x,z)$, respectively; then

$$\frac{\partial \hat{S}(x,z)}{\partial z} = \hat{S}(x,z)N(z)$$

and

$$\frac{\partial \hat{\tilde{S}}(x,z)}{\partial z} = \hat{\tilde{S}}(x,z)\tilde{N}(z). \qquad \text{(A-6)}$$

By taking the $\wedge$ operation on both sides of eq. (A-4), and differentiating with respect to z, we have

$$\begin{aligned} \frac{\partial \hat{\tilde{S}}(x,z)}{\partial z} &= \hat{P}(x,z)\cdot\frac{\partial \hat{S}(x,z)}{\partial z}\cdot\hat{Q}(x,z) \\ &\quad + \frac{\partial P(x,z)}{\partial z}\cdot\hat{S}(x,z)\cdot\hat{Q}(x,z) \qquad \text{(A-7)} \\ &\quad + \hat{P}(x,z)\cdot\hat{S}(x,z)\cdot\frac{\partial \hat{Q}(x,z)}{\partial z}. \end{aligned}$$

On the other hand, if $\tilde{N}(z)$ satisfies the equation

$$Q^{\wedge}(x,z)\cdot\tilde{N}(z) = N(z)\cdot Q^{\wedge}(x,z), \qquad \text{(A-8)}$$

this leads to

$$\begin{aligned} \hat{\tilde{S}}(x,z)\cdot\tilde{N}(z) &= \hat{P}(x,z)\cdot\hat{S}(x,z)\cdot\hat{Q}(x,z)\cdot\tilde{N}(z) \\ &= \hat{P}(x,z)\cdot\hat{S}(x,z)\cdot N(z)\cdot\hat{Q}(x,z) \\ &= \hat{P}(x,z)\cdot\frac{\partial \hat{S}(x,z)}{\partial z}\cdot\hat{Q}(x,z) \qquad \text{(A-9)} \\ &= \frac{\partial \hat{\tilde{S}}(x,z)}{\partial z} \end{aligned}$$

Comparing eq. (A-9) and eq. (A-7), it is necessary that the last two terms of eq. (A-7) be zero. This condition is easily satisfied if $\hat{P}(x,z)$ and $\hat{Q}(x,z)$ are independent of z. In view of their initial condition (A-5), we have

$$\tilde{S}(x,z) = Q^{-1} * S(x,z) * Q \qquad \text{(A-10)}$$

where $Q = Q(x)$ is independent of z. By taking the $\wedge$ operation, this leads to

$$\hat{\tilde{S}}(x,z) = Q^{-1}\cdot\hat{S}(x,z)\cdot Q \qquad \text{(A-11)}$$

for all $x \le z \le y$. For convenience we drop the $\wedge$ on Q. Eq. (A-8) now has the form

$$\tilde{N}(z) = Q^{-1}\cdot N(z)\cdot Q \qquad \text{(A-12)}$$

for all $x \le z \le y$.

References

[1] R. Bellman and S. Ueno, "Invariant Imbedding and Chandrasekhar's Planetary Problem of Radiative Transfer," Astrophys. Space Sci., Vol. 16, 1972.

[2] A. Bruckstein and T. Kailath, "An Inverse Scattering Framework for Several Problems in Signal Processing," IEEE ASSP, Jan. 1987, pp. 6–20.

[3] R. Redheffer, "On the Relation of Transmission Theory to Scattering and Transfer," J. Mathematics and Physics, Vol. XLI, No. 1, March 1962, pp. 1–41.

[4] S. Ueno, R. Kalaba, H. Kagiwada and R. Bellman, "Some Mathematical Aspects of Multiple Scattering in a Finite Inhomogeneous Slab with Anisotropic Scattering," Pub. Astr. Soc. Japan Vol. 22, 1970, pp. 75–83.

[5] S. Ueno and Y. Kawata, "Earth-atmosphere System Identification of Optical Parameters," Concise Encyclopedia of Environmental System, P.C. Young (Editor), Pergamon Press, Oxford, 1993, pp. 163–168.

[6] S. Ueno and A. P. Wang, "Scattering and Transmission Functions of Radiation by Finite Atmosphere with Reflecting Surface," J. Astrophysics and Space Science, 23 (1973), pp. 205–219.

[7] A. P. Wang, "Dissipative Properties of a Nonlinear Differential Operator Equation," J. Mathematical Analysis and Appl. Vol. 72, No. 1, Nov. 1979, pp. 75–88.

[8] A. P. Wang, "Searchlight on a Target with Diffuse Background," J. Math. Physics, Vol. 34, 1993, pp. 878–884.

Robot Path Planning through Some Places by Genetic Programming

H. Yamamoto
Dept. of Opto-Mechatronics
Wakayama University
930, Wakayama-shi, 640, Japan
tiger@center.wakayama-u.ac.jp

Abstract

This paper describes the method to determine a robot path that travels some places among machine tools of a production line FA factory. This decision is made by Genetic Algorithm with Lisp language programming. In the algorithm, the building block method to decide fitness is adopted. The method is applied to FMS system that has four machine tools and a robot.

Keywords: Genetic Algorithm, Robot Path Planning, FA Factory, Lisp Language

1 Introduction

As an example of the next generation FA factory, a decentralized autonomous production system is considered. In the next generation production system, the machine tools and robots achieve the production plan by changing their production configuration by gathering or dispersing. As the locations of the machine tools are not fixed and are changeable at a moment's notice, it is useless for us to program the robots' path while developing the production system.

In this paper, in order to decide the moving path of a robot operating by itself, the method by Genetic Programming (GP) is described. By carrying out genetic operations [1] such as crossover and mutation for the several elements of Lisp lists, the proposed method automatically finds a robot moving path through some visiting points with a Lisp program. During the genetic operations, as the original method to calculate fitness, the building block method is adopted to reduce the searching time for the solution. By applying the method to the FMS (Flexible Manufacturing System) example where a robot moves among four machine tools, the new method is ascertained to be useful.

2 Next Generation FA Factory

Many kinds of research for the next generation FA have been proposed in recent years [2]～[4]. The common element in such research is that FA systems such as a machine tool and a robot have a knowledge and make decisions or reason by themselves. As one of the prominent figures in the 21st century FA factory, the following intelligent FA system is considered in this paper. In an intelligent system, the machine tools and robots have different amounts and level of knowledge and mobilities which they use to interact with each other and determine a production system. This is called an autonomous decentralized production system. For example, in order to produce some varieties of parts, several machine tools and robots communicate and negotiate among themselves. As a result, an agreement is reached and they move to a certain place in the factory to begin production of the parts.

Under these circumstances, the location of the gathered machine tools and robots, which corresponds to the production line, cannot be forecasted beforehand. That means machine tools and robots can gather and disperse in the factory any time. Because of the changeable situation in a factory, it is useless for us to have the robot traveling path programmed beforehand in order to supply parts and tools to the machine tools.

Figure 1 shows an example of the production line which is used as the simulation example in this paper. The production line consists of four machine tools and a robot. The robot travels, as shown in Figure 1 and then returns to its starting point.

3 Program Of Robot Path Via Some Visiting Points

3.1 Expression of Path Programming with Lisp

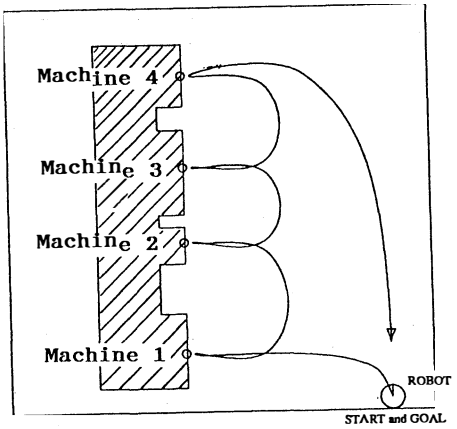


Figure 1 FMS Production Line

Consider how a robot takes its next step. The next step can be forward, backward, to the right or to the left. If the four kinds of movements are combined, the path locus of a robot can be expressed. For example, if the Lisp function for moving forward is [fw], turning right [tr] and turning left [tl], the Lisp program to express certain movements corresponds to ((fw)(tr)(fw)(tr)(fw)(tl)(fw)). The Lisp program using functions corresponding to the four kinds of movements can express the factory space even if the space is filled with machine tools and work tables.

3.2 Path Decision by GP

Because of developing the Lisp program repeatedly to carry out the four kinds of Lisp functions, [fw], [tr], [tl] and [bw] (which means moving backward), the path between two points can be expressed. Randomly developing this program, the solutions for the path become limitless.

In order to find an efficient solution, the decision method for a Lisp program by GB is adopted [5]. As "atoms" to express the Lisp program for the cycle movement of a robot, as shown in Figure 1, a set of arguments and a set of functions are used. A set of arguments consists of 17 elements, as shown in Figure 2. Figure 2 includes each meaning of the arguments. A set of functions consists of two functions, [IFLTE] and [PROGN2]. The function [IFLTE] needs four arguments. For example, it is expressed as (IFLTE a b c d) and its meaning is " Evaluate a and b. If the result is a≤b, evaluate c. If the result is a>b, evaluate d". The function [PROGN2] needs two arguments. For example, it is expressed as (PROGN2 a b) and its meaning is " Evaluate a and b and return b as the result of the evaluation". As the arguments of the two functions, the elements of an argument set are used.

mf → move 0.5m forward
mb → move 0.5m backward
tl → turn left 30°
tr → turn right 30°
s01 → minimum distance of obstacle from 0° to 30°
s02 → minimum distance of obstacle from 30° to 60°
s03 → minimum distance of obstacle from 60° to 90°
s04 → minimum distance of obstacle from 90° to 120°
s05 → minimum distance of obstacle from 120° to 150°
s06 → minimum distance of obstacle from 150° to 180°
s07 → minimum distance of obstacle from 180° to 210°
s08 → minimum distance of obstacle from 210° to 240°
s09 → minimum distance of obstacle from 240° to 270°
s10 → minimum distance of obstacle from 270° to 300°
s11 → minimum distance of obstacle from 300° to 330°
s12 → minimum distance of obstacle from 330° to 360°
d → dangerous distance

Figure 2 17 Arguments

3.3 Genetic Operations

[Initial Population]

In order to create an initial population for Lisp programs, after randomly selecting each element of an argument set, a function set and a pair of parentheses, make a list of them. The example for an individual can be described as (PROGN2 (MF) (IFLTE (PROGN2 (MB)(TR)(S01)(MF)(TL))). Every individual has the characteristic that it can be expressed with its own tree construction.

[Crossover and Mutation]

A crossover is performed by using the characteristic of an individual, as mentioned above. A crossover first randomly selects one point on the list of a Lisp program as a crossover point. With an individual by tree construction, the crossover is a little different from the one-point-crossover which is performed by a string type expression for an individual. From the view of the tree construction, the crossover finds a certain node in the tree and moves the node and the sub-tree situated below the node to another individual.

In the same manner, a mutation is carried out. After the mutation point is randomly selected, the selected node and the sub-tree whose top corresponds to the selected node are changed to new node and sub-tree. The new node and sub-tree are created just like an initial individual was created.

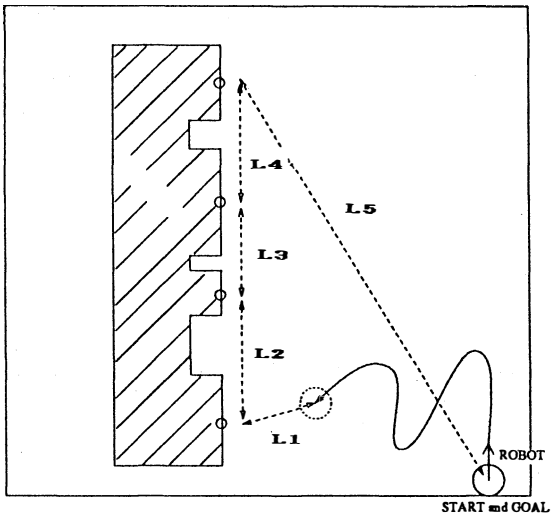


Figure 3 Ordinal method for fitness

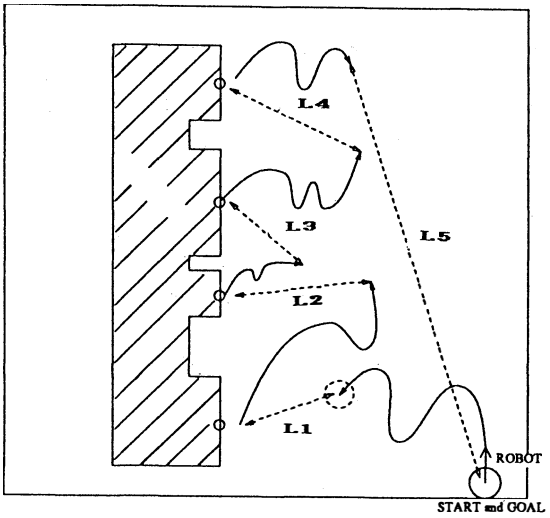


Figure 4 Building Block Method

3.4 Fitness

Fitness shows how each individual adaptable is to an environment. In this paper, the degree of adaptability to an environment is defined as how far a robot travels in one cycle movement. The ordinal method for fitness chooses fitness as the distance between the robot's starting point and the point where the robot finally arrives by a generated Lisp program. So, for example, L1 + L2 + L3 + L4 + L5 in Figure 3 corresponds to total fitness.

On the contrary, tremendous amount of paths to travel between two points in two dimensional space are existing. That means it spends tremendous amount of time for the robot to search the path from the starting point to the first traveling point in Figure 3. If the single searching time is limited, GA operations generate lethel gene that most individuals cannot arrive at the first traveling point.

In order to solve the above problem, the following building block method of stacking fitness is adopted. The building block method splits up fitness for an individual into fitness for each of the traveling points. The fitness for each traveling points means the distance from each of the robot arrival points to each of the traveling points (machine tools' ports). Each distance is considered as a "block". A single individual's fitness is decided by stacking the blocks. The split block has the time constraints for searching.

In the building block method, a robot moves a certain distance in a predetermined time. So, in its first phase, it moves from the starting point. For example, one minute and then stops. The distance from where it stops to the first traveling point is known as its fitness of a single block (which we will call L1, as shown in Figure 4). In its second phase, it moves from the first traveling point a certain distance and stops. The distance from the stopping point to the second traveling point may be called L2. And so on until it has visited all of traveling points and returns to its goal. The fitness in Figure 4 is expressed by summing up five blocks' fitness: L1, L2, L3, L4, and L5.

By calculating fitness with the building block method, the individual that has the schemata representing the distances between the arrival points and traveling points can exist. Even if an individual does not complete the cycle from start to goal, the building block method allows it to survive to the next generation. If the individual includes the block that the robot arrives at a traveling point, the block's schemata has a high possibility to be inherited by the next generation.

In this way, even though some individuals are not complete schemata as a whole, that is, they cannot complete the cycle, but they include complete schemata for a certain block, the individuals that have complete schemata for all blocks can be finally generated by repeated GA operations such as crossover and mutation.

4 Application Examples

The building block method is applied to a FMS that

has four machine tools and a robot, as shown in Figure 1. The robot travels to machine tools Machine1, Machine2, Machine3, and Machine4 in turn and then returns to its starting point. The application example adopts a population size of 400, a crossover probability of 88% and a mutation probability of 1%. Figure 5 shows the best result acquired from generation 154. Figure 6 shows the fitness curve of the application example. The bold curve is the best fitness among each generation, a big dotted line is the fitness average among each generation and a small dotted line is the worst fitness.

5 Conclusions

This paper describes the method to automatically generate a robot path programing among some machine tools called FMS. The method adopts Genetic Programing. In the programing, a building block method is used which allows individuals with partial schemata to survive to the next generation. The strategy of the building block method is " the individual that partially includes a good schemata to survive ".

After simulations of a robot path generated in FMS, the best individual is found from generation 154.

References

[1]D.E.Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

[2]P.Valeckenaers et al., IMS Test Case 5: Holonic Manufacturing Systems, Preprints of IMS'94 2nd IFAC/IFIP/IFROS Workshop, 1994.

[3]L.Kavraki and J.Latombe, Randomized Preprocessing of Configuration Space for Fast Path Planning, Proceedings of 1994 IEEE International Conference on Robotics and Automation, 1994, pp.2138-2145.

[4]H.Yamamoto, Simulator of Flexible Transfer Line Including Genetic Algorithm and its Applications to Buffer Capacity, Priprints of IMS'95 3rd IFAC/IFIP/IFROS Workshop, 1995, pp.127-132.

[5]J.R.Koza, Genetic Programming, MIT Press, 1994.

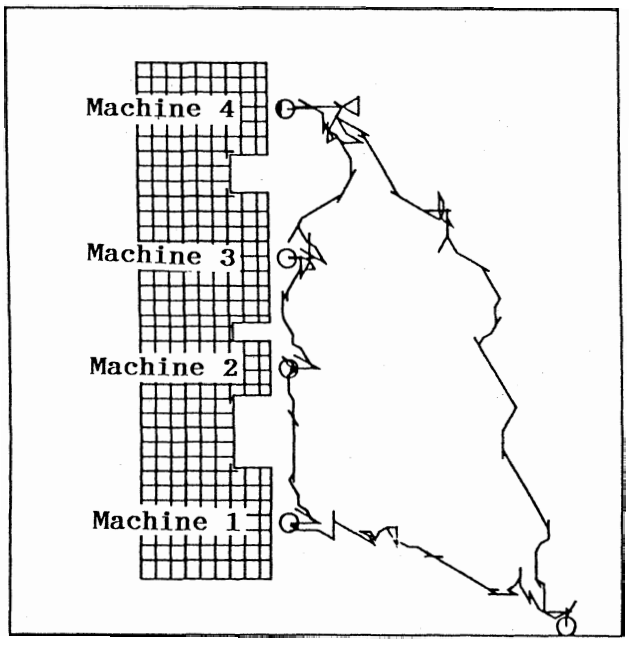


Figure 5 Final Simulation Result

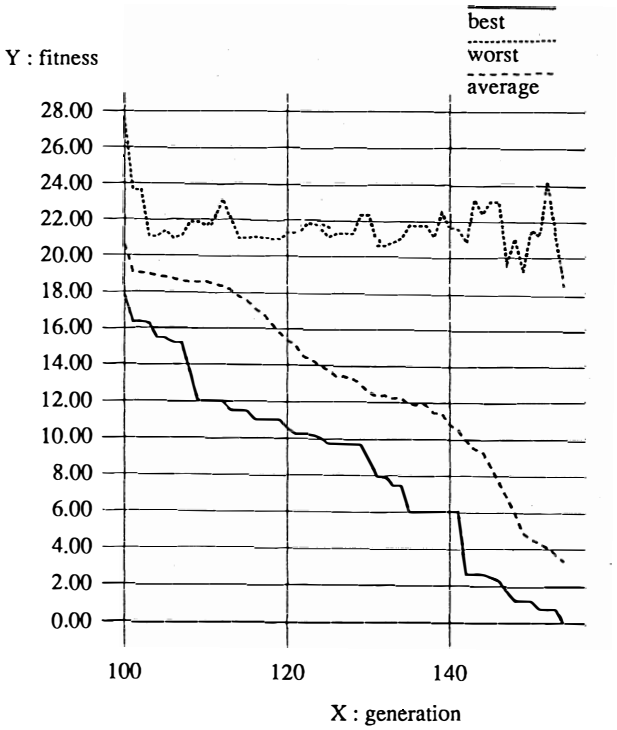


Figure 6 Fitness Curves

A Radial Basis Function Neuro-Tracker

Carl G. Looney
Computer Science Department/171
University of Nevada
Reno, NV 89557 USA
email: Looney@cs.unr.edu

Abstract

We use a radial basis function neural network to control a video camera to track a target moving at high speed. The results show considerable promise.

1 Introduction to RBF Neural Networks

We investigate the application of *radial basis function* (RBF) neural networks to video tracking of a moving target. RBFs originated in 1964 as potential functions in [1] and [2], and in [6] in 1968. Their use in neural networks was due to [3] in 1988 and to [5] in 1989. The tracking problem is taken from [7]. An RBF has the Gaussian form [4]

$$\mathbf{x} \to f(\mathbf{x};\mathbf{v}) = \exp[-\| \mathbf{x} - \mathbf{v} \|^2/(2\sigma^2)] \qquad (1)$$

where $\mathbf{x}$ is an input *feature vector*, $\mathbf{v}$ is a fixed *center vector* of the same dimension as $\mathbf{x}$, and σ^2 is the *spread* parameter. Figure 1 shows an RBF on the x_1x_2-plane. An RBF and center vector belong to each hidden neurode.

Our N-M-J RBF neural network is shown in Figure 2. Let $\{(\mathbf{x}^{(q)},\mathbf{t}^{(k(q))})\}$ be a set of Q pairs of exemplar input and output vectors for training, where $\mathbf{x}^{(q)} = (x_1^{(q)},\ldots,x_N^{(q)})$ is the qth input feature vector, $\mathbf{t}^{(k(q))} = (t_1^{(k(q))},\ldots,t_J^{(k(q))}$ is the kth *target* output for input $\mathbf{x}^{(q)}$ and there are k = 1,...,K different output classes. We select a number M of neurodes *(neuronal nodes)* in a single hidden layer. For each m = 1,...,M, there is an mth *center (weight) vector* $\mathbf{v}^{(m)}$ and a *spread* parameter σ_m^2 for the mth hidden neurode) that determine the RBF $y_m = f_m(\mathbf{x};\mathbf{v}^{(m)}) = \exp[-\| \mathbf{x} - \mathbf{v}^{(m)} \|^2/(2\sigma^2)]$ for it.

During operation, an input vector $\mathbf{x}$ is presented to the network and processed separately at each mth hidden neurode to produce the output value $y_m = f_m(\mathbf{x};\mathbf{v}^{(m)})$.

Figure 1 - The RBF for Cluster Center v

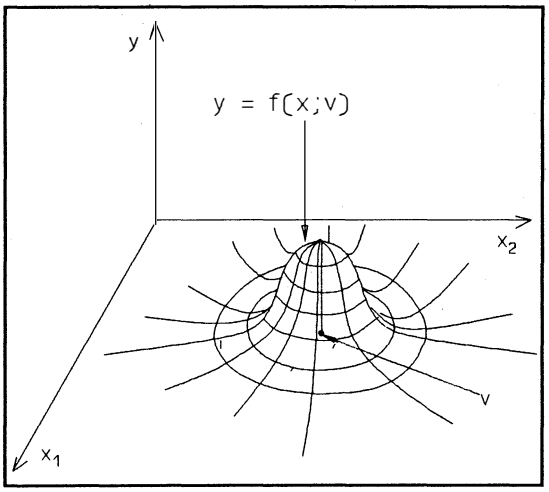
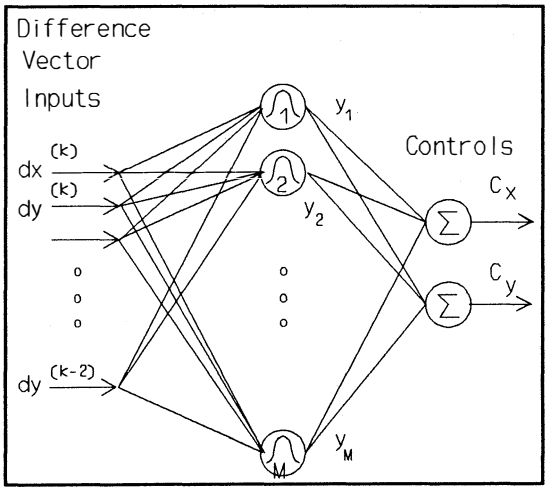


Figure 2 - Architecture of The Tracking RBF Network



When $\mathbf{x}$ is close to $\mathbf{v}^{(m)}$ then y_m is high, but when $\mathbf{x}$ is far from $\mathbf{v}^{(m)}$ then y_m is low. Each of the J perceptron type neurodes in the output layer computes a *weighted average* of the RBF components of $\mathbf{y} = (y_1,\ldots,y_M)$ via

$$z_j = (1/S)\Sigma_{(m=1,M)}u_{mj}y_m \qquad (2)$$

where either S = M or S = y_1 + ... + y_M. The RBFs partition the input space into small hyperball subclasses so the output layer can join (OR) groups of them into nonconvex classes for nonlinear separability.

The *first stage* of training selects the M centers $\{v^{(m)}\}$ and the spread parameters. The quickest method is to put M = Q and take each input exemplar feature vector $x^{(q)}$ to be a center via $v^{(m)} = x^{(m)}$, m = 1,...,Q. This method requires only the training of the output weights $\{u_{mj}\}$, which is extremely quick.

When Q is too large, we construct K different vectors $\{y^{(k)}\}$ such that each one has a single high value in the mth component of $y^{(k)} = (y_1{}^{(k)},...,y_M{}^{(k)})$ and all of the remaining components are zero and put M = K. The weight (center) vectors $\{v^{(m)}\}$ are then drawn randomly with components between 0 and 1, where $v^{(m)} = (v_1{}^{(m)},...,v_N{}^{(m)})$. Next, we present the exemplar feature vectors $\{x^{(q)}\}$ to the hidden layer, one at a time, and find the maximum RBF value y_{m*} that determines the *winning* m*-th neurode. We update the single winning neurode with center $v^{(m*)}$ via

$$v_n{}^{(m*)} \leftarrow v_m{}^{(k)} + \eta[x_n{}^{(q)} - v_n{}^{(m*)}(r)] \qquad (3)$$

(n = 1,...,N) where η is the *learning rate*. We update the other centers by subtracting the rightmost term.

Each exemplar is presented in turn about 12 times (12 epochs or 12Q iterations) to obtain good centers. The value η = 0.6 usually suffices but may be smaller or larger. The spread parameter σ^2 about 0.01 to 0.8 for a normalized feature space of inputs, depending on M (a larger number of clusters M means a smaller σ^2).

The *second stage* trains the jth output weights $\{u_{mj}\}$ over all Q inputs by steepest descent via

$$u_{mj}{}^{(r+1)} \leftarrow u_{mj}{}^{(r)} +$$

$$\eta(2/M)\Sigma_{(q=1,Q)}\Sigma_{(j=1,J)}(t_j{}^{(q)} - z_j{}^{(q)})y_m \qquad (4)$$

The output target $t^{(k(q))}$ must be known for each input exemplar vector $x^{(q)}$. The input vectors and output values do not need to be constrained as in MLPs, but we prefer to standardize (normalize) the inputs.

This training is extremely fast (up to 1000 times faster than backpropagation). The difficult training of hidden neurodes by gradient methods is not done, which circumvents the problem of local minima.

Figure 3 presents the case of XOR logic where the training uses the exemplars (0,0), (0,1), (1,0), and (1,1) as neurode centers. In contrast to this, Figure 4 shows the results of training with the exemplars (0.3,0.3), (0.3,0.7), (0.7,0.3) and (0.7,0.7), which is more robust.

Figure 3 - RBF Contours for XOR Logic

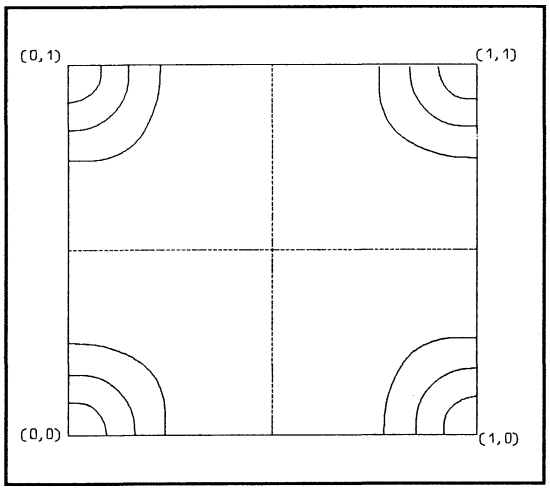
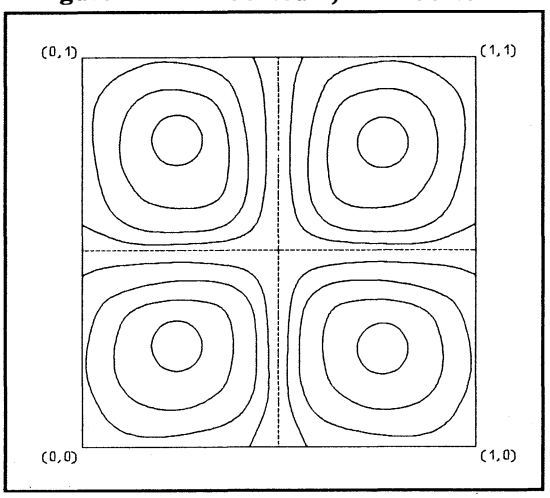


Figure 4 - RBF Contours, New Centers



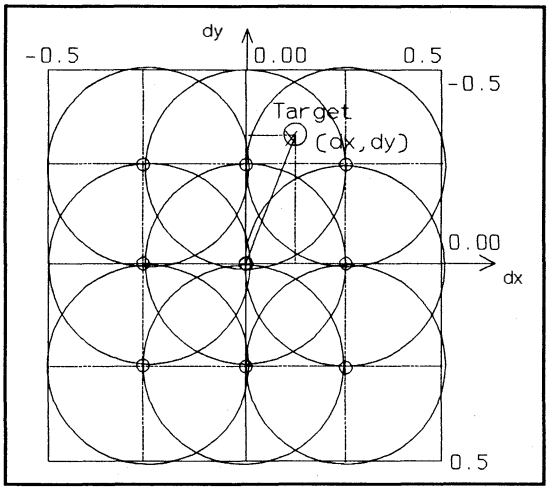
2 The Tracking Application

Tracking is implemented with a planar video display that shows the object being tracked. The *field-of-view* (FoV) is a rectangular screen with cross-hairs at the center. It is conceptually covered by M radial basis functions whose one-sigma (σ) contours are shown in Figure 5. The idea is to drive the current errors dx and dy of the target position (dx,dy) to zero using the input vector of feedback errors **e** = ($dx^{(r)}$, $dy^{(r)}$, $dx^{(r-1)}$, $dy^{(r-1)}$, $dx^{(r-2)}$,$dy^{(r-2)}$) on iteration r. The RBF network is shown in Figure 2. The outputs c_x and c_y are the respective x and y commands for the camera positioning motors.

An input positional error vector **e** activates one or more RBFs to put out higher y_m values (1 ≤ m≤ M).

The combination of outputs is fed into each of the two output neurodes, where the values are multiplied by the weights $\{u_{mj}\}$ (j = 1,2) and summed to yield the control commands, c_x and c_y. The network is trained to map the input error vector **e** into the commands c_x and c_y that are proportions of maximum turns Θ_{xmax} and Θ_{ymax} that the camera's gimbal motors can make.

Figure 5 - The Field of View with RBF σ-Contours



3 Simulation Runs and Results

A simulated high speed target was run on both the counter clockwise (CCW) path shown in Figure 6 and clockwise (CW) path that is the reflection of the path of Figure 6 about the vertical axis. The RBF network controlled the camera after training.

The centers of the nine circles shown in Figure 5 were taken as the initial center (weight) vectors of the hidden neurodes. The initial output weights were selected randomly between -1 and 1.

The network was trained by adjusting both the hidden and output weights, one at a time, to minimize the sum-squared error (SSE) over the paths (sum of dx^2 and dy^2 over the updates of an entire path). The camera tracking system was modeled simply so that the commands c_x and c_y moved the camera through the angles Θ_x and Θ_y that are proportions of Θ_{xmax} and Θ_{ymax}.

We trained for 20 iterations, where a single iteration adjusted each of the hidden and output weights in turn over a complete target path flyout to determine the sum -squared error. The spread parameter $\sigma^2 = 0.4$ yielded more stable results than smaller values. The CCW target path starts on the right, but the CW target path starts on the left. Figures 7 and 8 show the target position relative to the controlled camera on the respective CCW and

Figure 6 - The CCW Target Path (no control)

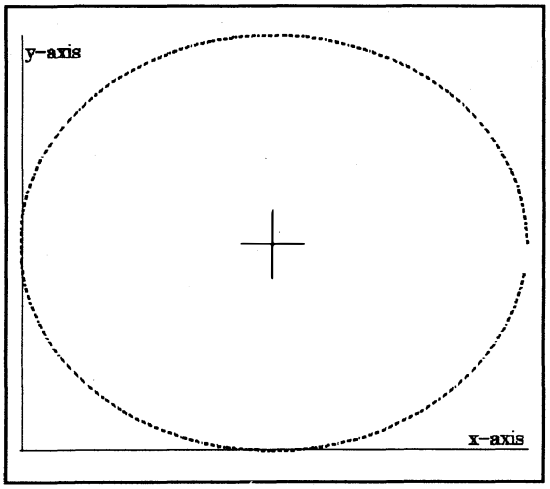


Figure 7 - The CCW Path after Training 1

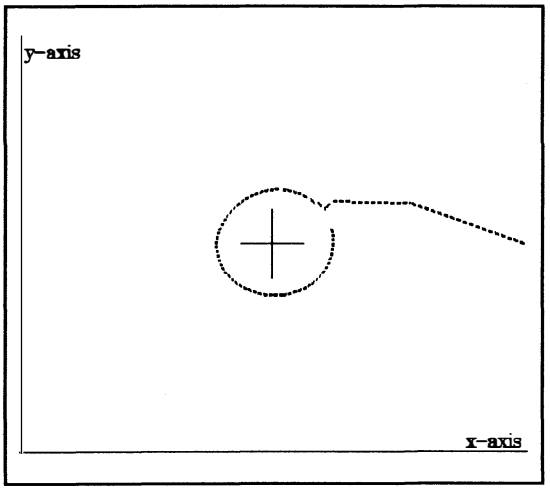
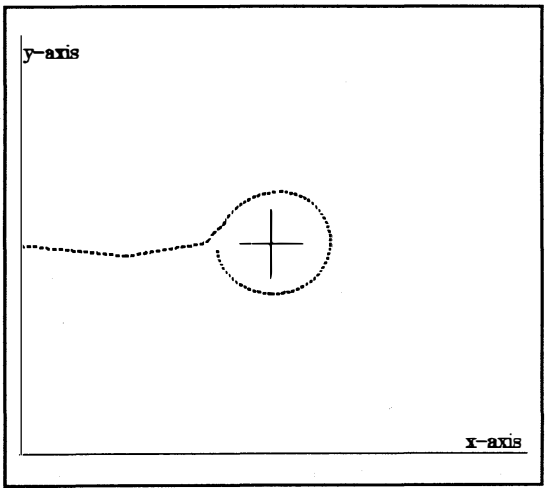


Figure 8 - CW Path after Training 1



CW paths under Training 1 runs that used the two inputs $dx^{(r)}$ and $dy^{(r)}$. Figures 9 and 10 show the results of Training 2 runs that used the four inputs $dx^{(r)}$, $dy^{(r)}$, $dx^{(r-1)}$ and $dy^{(r-1)}$. The Training 2 errors here are much smaller.

It is clear that this system is equivalent to a fuzzy rule-based control system where the RBFs are Gaussian fuzzy set membership functions and the rules connect the fuzzy set membership values to the defuzzified controls. In both neural and fuzzy systems, the fuzzy membership functions allow room for inexactitude.

Figure 9 - The CCW Path after Training 2

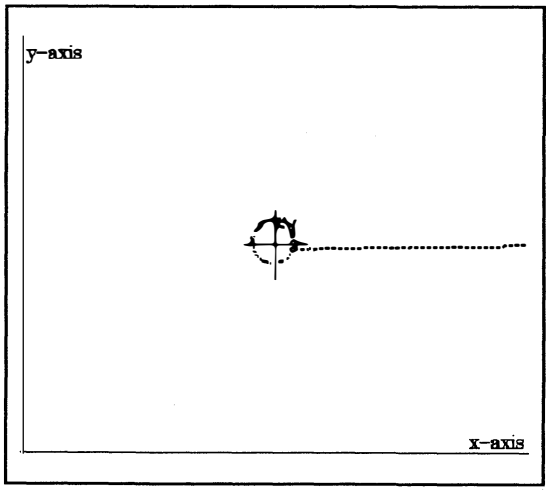
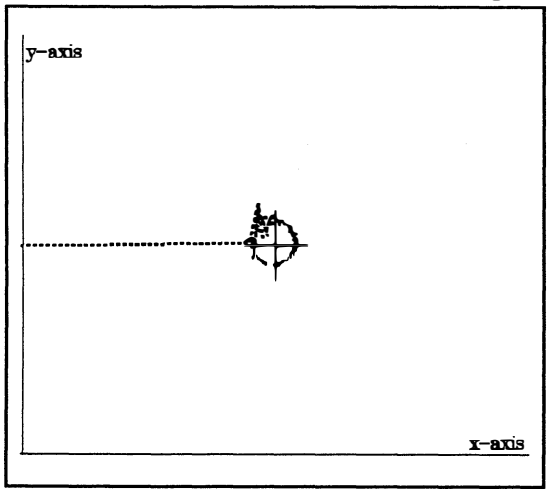


Figure 10 - The CW Path after Training 2



4 Conclusions

The use of RBF neural networks to control a video camera to track a high speed moving target demonstrates the flexibility of such networks. The use of feedback errors delayed by one time increment in addition to the current errors provided more accurate control. We may extrapolate to conclude that the use of twice delayed errors additionally would provide even more accuracy, but we expect that more delayed errors might not improve the control further or might even degrade it. The size of the network is small and the training is quick and nonproblematic in contrast to that of backpropagation multiple layered perceptrons.

It is clear that this network is equivalent to a fuzzy rule based control system and has the advantages of i) possible implementation in parallel for speed, ii) simplicity of design in that the rules become built in by the training, and iii) simplicity of execution because rules do not need to be fired by antecedents that must be fuzzified and consequents do not need to be defuzzified. The fuzzification and defuzzification are built in also. Once an RBF network is trained, the weights can be used in on-line production systems for intelligent control.

References

[1] M. A. Aizerman, E.M. Braverman, and L. I. Rozonoer,(1964), "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control* **25**, 821-837, 1964.

[2] O. A. Bashkirov, E. M. Braverman and I. B. Muchnik, "Potential function algorithms for pattern recognition learning machines," *Automation and Remote Control* **25**, 629-631, 1964.

[3] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems* **2**, 321-355, 1988.

[4] C. G. Looney, *Pattern Recognition with Neural Networks*, Oxford University Press, NY (in press, 1996).

[5] J. E. Moody, and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation* **1**, 281-294, 1989.

[6] D. F. Sprecht, *A Practical Technique for Estimating General Regression Surfaces*, LMSC-6-79- 68-6, Lockheed Missile and Space Co., Palo Alto, CA, 1968.

[7] Masanori Sugisaka, "Neural network control for recognition and tracking of moving objects," *Proc. ISCA Fourth Golden West Int'l Conf. on Intelligent Systems*, San Francisco, 189-193, 1995.

Self-Assembly Method for Mechanical Structure

Satoshi Murata, Haruhisa Kurokawa, Kohji Tomita, Shigeru Kokaji
Mechanical Engineering Laboratory, AIST, MITI
1-2 Namiki, Tsukuba, 305 JAPAN

Abstract: A novel design method for a machine system is proposed and a prototype model is constructed. Conventional machine systems are composed of various parts which are passive. In the new scheme, they are made with only one kind of active and intelligent units. The prototype of this kind of unit is called " Fractum ", which has actuators sensors and information processing functions. In the system, there is no concentrated part. These units are the same and have equivalent capacity so that any of them can replace any unit. The knowledge about the whole system is embeddedin every unit and this enables the group of these units to organize the whole system by themselves and to repair it by themselves. The algorithm for the self-organization and self-repair is designed and tested by computer simulations.

1 Introduction – Distributed Mechanical Structure

Living organism has internal maintenance mechanism. It can repair itself and reproduce itself without outside help. In contrast, most industrial products or machines rely on outside help for their maintenance. As the complexity of the machine increases, the burden of its maintenance becomes tremendous. If we can develop built-in autonomy for the machine, it will be of great use. The idea has been tried, however, with rather ad-hoc methods in conventional machines. If a supervisory system to check the machine operation is added to the original machine, then it certainly improves the machine's reliability. But this method is an automated external help and has limitations. There is no guarantee of the normal operation of the additional system. The additional system may cause trouble for the whole system. In order to avoid this, another supervisory system should be introduced to monitor the additional part and this repeats endlessly.

A solution is possible by making the whole system distributed. In this method, the machine is composed of units compatible to each other. The design information of the whole machine is stored in each unit and these units work together to establish the total system function.

There is, however, a difference between this new type of machine and a living organism. Since the machine unit cannot reproduce itself, enough units must be supplied from the beginning, also it is necessary to develop a novel way of self-assembly.

2 Hardware of Mechanical Unit

We have built actual mechanical units for a two dimensional distributed mechanical system [1,2]. The unit called "fractum" may be described as a "moving LEGO block." It is an autonomous reconfigurable unit, namely, it can change connection with other units without outside help. Each unit has 6 connection arms, an onboard micro-processor which controls these connection arms, and 6 bilateral optical communication linkages between processors of connected units. The connecting arm uses a pair of permanent magnets and an electro-magnet as an actuator. When the arms are connected mechanically, the information linkage between the units is established at the same time. Using this prototype, we confirmed its basic functions such as changing configuration and communication. We are now constructing 20 units.

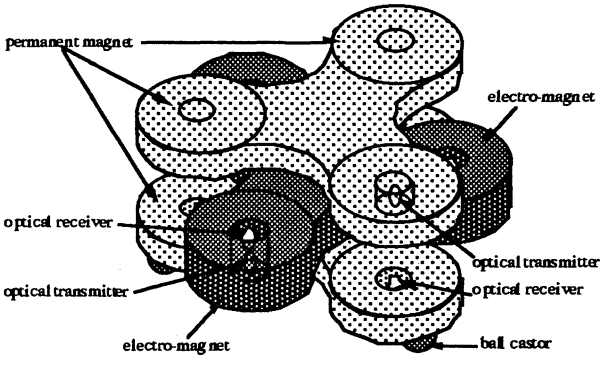


Figure 1. Schematic view of mechanical units

3 Software for Self-Assembly

A specific feature of the homogeneous structure of the system is that each unit can be used at an arbitrary position in the system. Thus, each unit must have information on the whole system, and the information must be identical in all the units.

We have developed two types of algorithms which enable "self-assembly" of the group of mechanical units. The self-assembly is a process where the group of units form a pre-determined particular destination shape from an arbitrary configuration. Only local communication between the adjacent units is assumed in these algorithms.

3.1 Algorithm for Simultaneous Assembly (Algorithm A [1,2])

On the unit, only information on local connections is available. We must describe the whole shape of the system based on the local geometry of the connection. There are 12 possible types of local connections for the unit (see Figure 2). The type transition diagram shows the 12 possible types and defines the distance between different types by the number of movement steps required to transit from one type to another.

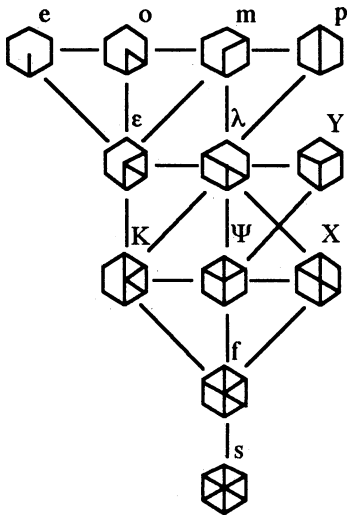


Figure 2. Type transition diagram

We developed a method to describe the whole shape of the system by these types. For example a triangular configuration made of 10 units is describedas follows.

o {K,K}
K{o,K,K,s}
s {K,K,K,K,K,K} (1)

The set of the above 3 statements (1) is a blueprint of the whole shape given to all the units. In the statement, the first letter shows one of the goal types which appears in the final configuration, and the letters in parenthesis are neighbor types of the goal type.

Each unit evaluates the difference between its current situation and the statements in (1) based on the type information obtained by local communication. If the current situation is involved in (1), the unit does not move. Otherwise, it moves randomly, but the frequency of the movement is determined corresponding to the difference between its current situation and the goal.

For small scale self-assembly, this algorithm has excellent performance (97% success in the case of 10 units). However, the larger the number of the units, the worse of the convergence and the probability of success. Also the algorithm could not deal with complex shapes. In short, the reason is combinatorial explosion. Namely, the movement of the whole system does not cease until all the unit satisfy the goal statements simultaneously, and such a situation scarcely happens when the number of units is large.

3.2 Algorithm for Developmental Assembly (Algorithm B [3])

In order to cope with the complexity of large scale self-assembly, the following developmental algorithm is devised. The algorithm is an analogy of the development processes of living organisms. By developmental assembly we can reduce the complicated large scale assembly to small sub-problems.

In the first step, which is a counterpart of fertilization, the group of units selects one particular unit by mutual voting process. The selected unit called a "kernel" becomes the starting point of the development process. The kernel gathers surrounding units to form the first stage. After the units in the first stage confirm that they have finished the first stage of assembly, they transfer to the next stage. They gather surrounding units to form the second stage and so on.

In this algorithm, the blueprint given to each unit must describe not only the final shape of the system but all transient shapes through the development process. We use the following lower

triangular matrix to describe the process of development.

```
{{ Y,---,---,---,---,---,---,---,---,---},
{ Y, e,---,---,---,---,---,---,---,---},
{ s, o, o,---,---,---,---,---,---,---},
{---, K, K, e,---,---,---,---,---,---},
{---,---, f,eps, o,---,---,---,---,---},
{---,---,---,---, K,eps, o,---,---,---},
 {---,---,---,---,---, K,eps, o,---,---},
 {---,---,---,---,---,---, K,eps, o,---},
{---,---,---,---,---,---,---, K,eps, o}}   (2)
```

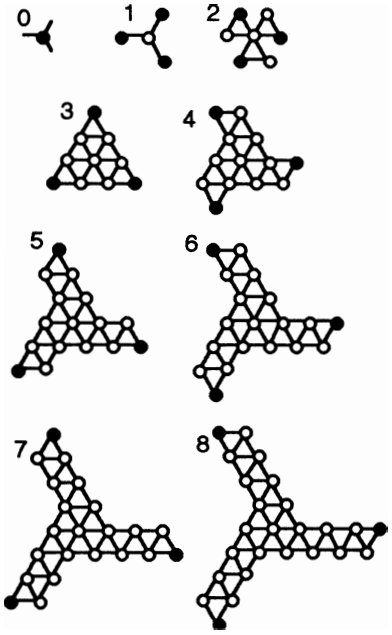


Figure 3 Stages of development described in (2)

This example shows the development process of a system made of 25 units which has 3 radial arms. Row numbers indicate the stage, and the characters in the row are the connection types that must be involved in the stage. The numbers of each types are not given in this matrix, but are determined implicitly by consistency of the configuration. Figure 3 shows the development process described in the matrix.

3.3 Hierarchical Description Method for Large Construction

By introducing multiple description matrices, we extend algorithm B to realize construction of larger structure. In this extended method, several description matrices with different order are introduced and each of them represents a part of structure. Structure of the whole system is expressed by the combination of these substructures. We can use the same description matrix for the same kind of parts, thus the size of the description can be much reduced when the system includes repeated structure and high symmetry.

To connect different description matrix, a jump command is embedded in the primal (order: 0) description matrix. If a unit is hit by the command the unit becomes a kernel of secondary (order: 1) description matrix. Of course the secondary description matrix may involve jump commands to any other description matrices including itself. In the case of jumping itself, the system reads a description matrix recursively. The following example illustrates this process. The primal matrix represents a large triangle, and "1" in the last line is the jump command to the secondary matrix, which represents small triangle. The secondary matrix called recursively, thus 3 long arms made of many triangles are formed. (see Figure 4)

```
0: {{  s, ---, ---, ---, ---, ---, ---},
    {  s,eps, ---, ---, ---, ---, ---},
    {  s,  K,  o, ---, ---, ---, ---},
    {  s,  f,  o,  o, ---, ---, ---},
    {  s,  s,  K, K,eps, ---, ---},
    {  s,  s,  s, K,  K,eps,---},
    {  s,  s,  s, 1,  K,  K,  o}}

1: {{ K, ---, ---, ---, ---},
    {  s,  o, ---, ---, ---},
    {  s,eps,  o, ---, ---},
    {  s,  K,  K,  o, ---},
    {  s,  K,  1,  o, ---}}          (3)
```

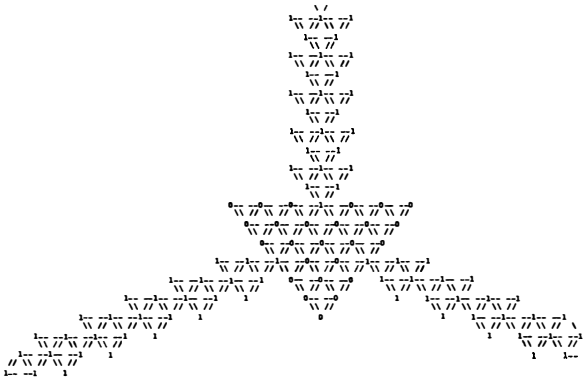


Figure 4 Example of recursive structure

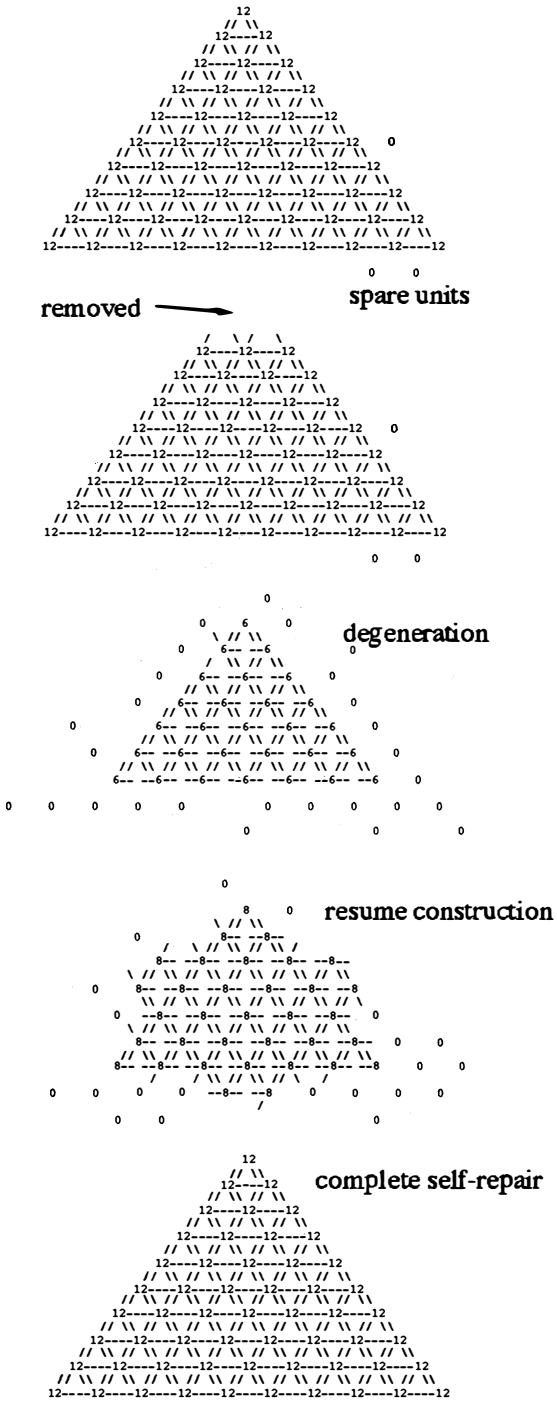


Figure 5 Self-repair by backtracking of development

4 Algorithm of Self-Repair [4]

Another extension of algorithm B is self-repair. We have developed an basic procedure of self-reconfiguration when some of the units are lost. It enables recovery from lost of arbitrary part of the system. In this procedure, we make two assumptions, 1) enough number of unit is provided at the initial time, and 2) no faulty units remains after some units are removed.

In algorithm B, construction process is divided into several stages, and the system converges to sub-goal in each stage and finally realizes the target configuration. Thus the self-repair can be naturally performed by backtracking of stages. At first, the lost of parts is detected units by checking communication channel. Then the units which detect the failure broadcast degeneration signal to other remained units. The degeneration signal includes information about a level of degeneration which is determined by the maximum stage number which is not affected by the failure. When the unit receives the degeneration signal, the connection type of the unit is reset to some previous level, and resume construction procedure again. Figure 5 illustrates a simulation results of the self-repairing process.

5 Future Study

The study of distributed intelligence will be indispensable for the refinement of self-organization and self-repair function. Together with these logical study, we envisage that the physical realization of the unit structure will be the key to the next generation technology. The organ system which is composed of vast number of organ cells are still beyond our complete understanding because of its high complexity. We hope the study of highly distributed system with active unit structure, like our unit based machine structure, might lead us to the better understanding.

References

[1] S. Murata, et al.: Self-Assembling Machine, Proc. IEEE Int. Conf. Robotics & Automation, San Diego, pp.441-448 (1994)

[2] S.Murata, et al.: Self-Repairable Machine, J. SICE 31-2 (1995) in Japanese.

[3] S.Murata, et al.: Design of Distributed Mechanical System (3rd Report), Proc. ADS Simp. SICE, Nagoya (1995) in Japanese.

[4] K.Tomita, et al.: Design of Distributed Mechanical System (4th Report), Proc. ADS Simp. SICE, Toyama (1995) in Japanese.

System Properties of Amino Acids Production.

Linear system analysis for Asparaginic acid-Isoleucine-Methionine-Lysine.

H Hirayama, Y Ishikawa, T Nozawa, Ko Li We, and Y Fukuyama.

Department of Public Health, Asahikawa Medical College
4-5 Nishi -kagura, Asahikawa 078, Japan

ABSTRACT

The system properties of amino acids production initiated from Asparaginic acid to Isoluecine, Metionine and Lysine was investigated by applying linear system. By examining the eigen values, the system was highly stable and uncontrollable. With the optimized control input that minimizes the squares of changes of the concentrations of all the components in the system and the control input, the amplitude of impulse responses and the amplitude of singular values were reduced particularly when the metabolizing enzymes do not exist. Present investigation is available to evaluate the relative magnitude of minimization of each component involved in the amino acids production system.

key words : Amino acid, feed back, stability, controllability, singular value, optimization.

[INTRODUCTION]

Amino acids are the building blocks of proteins and peptides [1]. They serve as precursors of many important molecules that have essential biological activities. Therefore a proper control to produce amino acids is indispensable to maintain the physiological operation , namely, Homeostasis in all the biological systems. The processes of amino acids production have well known as the most typical biochemical system characterized by multiple negative feed back inhibition [1][2]. Disturbances of the feed back mechanism result in serious pathological states, such as congenital enzyme deficit diseases[2].

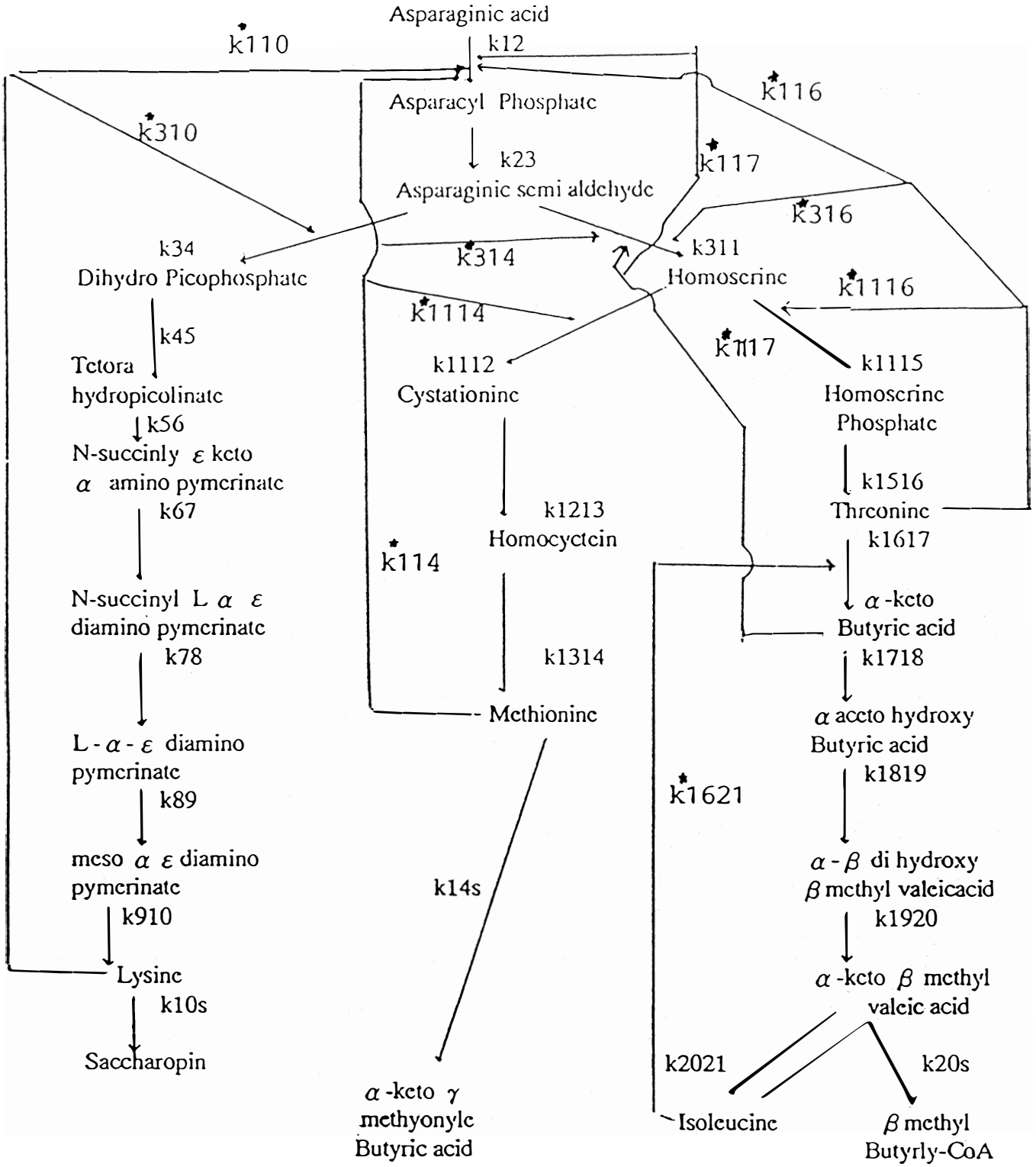
The system analysis focused on the amino acid production which is initiated from Asparaginic acid to produce Isoluecine, Mehionine and Lysine to examine, 1) the stability and controllability, 2) the effects of the optimized control input on the impulse responses, on the temporal behaviors of the system and 3) singular values of the system.

[MATHEMATICAL METHOD]

The amino acids production system is shown in Fig 1 [1][2]. Asparaginic acid is converted to produce Lysine, to Metionine and to Isoleucine. Applying the mass-action law to the reactions in Fig 1, following 21 linear rate equations are obtained

X1'= -k12 X1 + k-12 X2 + k12 (k110 X10
+ k114 X14 + k116 X16 + k117 X17)
X2' = k12 X1 - k-12 X2 - k12 (k110 X10 + k114 X14
+ k116 X16 + k117 X17) - k23 X2 + k-23 X3
X3' = k23 X2 - k-23 X3 - k34 X3 + k-34 X4 - k311 X3
+ k-311 X11 + k34 k310 X10 + k311 (k314 X14
+ k316 X16 + k317 X17)
X4' = k34 X3 - k-34 X4 - k45 X4 + k-45 X5
- k34 k310 X10
X5' = k45 X4 - k-45 X5 + k-56 X6 - k56 X5
X6' = k56 X5 - k-56 X6 + k-67 X7 - k67 X6
X7' = k67 X6 - k-67 X7 + k-78 X8 - k78 X7
X8' = k78 X7 - k-78 X8 + k-89 X9 - k89 X8
X9' = k89 X8 - k-89 X9 + k-910 X10 - k910 X9
X10' = k910 X9 - k-910 X10 -k10s X10
X11' = k311 X3 - k-311 X11 - k311 (k314 X14
+ k317 X17+ k316 X16) - k1115 X11
+ k-1115 X15 + k1115 k1116 X16
- k1112 X11 + k-1112 X12 + k1112 k1114 X14
X12'= k1112 X11 - k-1112 X12 - k1213 X12
+ k-1213 X13 - k1112 k1114 X14
X13'= k1213 X12 - k-1213 X13 - k1314 X13
+ k-1314 X14
X14'= k1314 X13 - k-1314 X14 - k14s X14
X15'= k1115 X11 - k-1115 X15 - k1516 X15
+ k-1516 X16 - k1115 k1116 X16
X16'= k1516 X15 - k-1516 X16 - k1617 X16
+ k-1617 X17 + k1617 k1621 X21
X17'= k1617 X16 - k-1617 X17 - k1718 X17
+ k-1718 X18 - k1617 k1621 X21
X18'= k1718 X17 - k-1718 X18 - k1819 X18
+ k-1819 X19

Fig 1. Schematic illustration of amino acid production initiated from Asparaginic acid.



k*110=k12 k110 k*310=k34 k310 k*1114=k1112 k114
k*114=k12 k114 k*314=k311 k314 k*1116=k1115k1116
k*116=k12 k116 k*316=k311 k316 k*1621=k1617 k1621
k*117=k12 k117 k*317=k311 k317

X19'= k1819 X18 - k-1819 X19 - k1920 X19
+ k-1920 X20
X20'= k1920 X19 - k-1920 X20 - k2021 X20
+ k-2021 X21 - k20s X20
X21'= k2021 X20 - k-2021 X21

where the variables expressing the concentrations correspond to the chemical species as shown in Table 1 and kn denotes the rate constants.

TABLE 1. List of the variables corresponding to the chemical species in the system.

X1 : Asparaginic acid ,X2 : Asparatil phosphate
X3 : Asparaginic acid semialdehyde
X4 : Dihydro di picolinate , X5 : Tetra hydropicolinate
X6 : N-succinyl ε keto α di amino pimelate
X7 : N-succinyl - ε - α di amino pimelate
X8 : Di amino pimelate
X9 : Meso ε - α di amino pimelate
X10 : Lysine X11 : Homoserine
X12 : Cystathininie X13 : Homocycteine
X14 : Methionine X15 : Homoserine phosphate
X16 : Threonine X17 : α keto butyric acid
X18 : α aceto α hydroxy butyric acid
X19 : α-β dihydroxy β-methyl valerianic acid
X20 : α keto β-methyl valerianic acid
X21 : Isolycine.

Table 2. The rate coefficients (mM)

1. The rate constants for the pathway to produce Lysine
k12=4.7 k23=0.16 k-23=2.6 k34=0.13
k45=0.101 k56=1.01
k67=5.2 k78=1.3 k89=6.7 k-89=100
k910=1.7 k110= 1.106 k310=1.101 k10s= 12
2. The rate constants for the pathway to produce Methionine
k311=2.3 k1112=4.1 k1213=1.6 k-1213=11.1
k1314=0.022 k114= 1.04 k314=0.467
k1114= 0.98 k14s=0.18
3. The rate constants for the pathway to produce Isoleucine
k1115=0.6 k1516=1.02 k1617=4.1
k1718=3.25 k1819=0.73
k1920=0.54 k2021=5.1 k316=1.051
k1116= 1.01 k1621= 1.01 k317=1.01
k116= 0.984 k117= 1.1 k20s=1.0

The physiological values of kn (mM) are shown in Table 2 [2][3]. k10s, k14s and k20s are the rate constants for catabolyzing Lysine, Methionine and Isoluecine. The state equations are expressed in vector form

$$X' = A X + B U, \quad Y = C X$$

Asparaginic acid X1, input was set to afford only to X1. The stability of the total reaction of the system was investigated by analyzing the eigen values of the eigen equation of the system. The controllability was judged by the rank of the matrix. The number of uncontrollable variables were calculated by

length (A) - rank (Co) where

$$Co = [\, B \;\; AB \;\; A^2B \;\; A^3B \;\; A^4B \;\; ---- \;\; A^{20} B \,]$$

To calculates the optimal feed back gain matrix K such that the feed back law U = -KX minimizes the cost function,

$$J = \int_0^t [\, X Q X + U R U \,] dt$$

The Q and R are the weighting coefficients vectors The system supplied by the optimized control input is obtained by solving

$$X' = (A - B K) X.$$

The initial conditions are set to be, X1 =1.0 (mM) and Xn =0.0 (2 < n <21). The impulse response of each component were compared with and without the optimized control input. The impulse responses in which the metabolizing enzymes for Lysine,Methionine and Isoleucine do not exist (k10s=k14s=k20s= 0.0) were compared in the states with and without the optimized control input. The singular values of the system were obtained by calculating the complex matrix,

$$C(j \omega I - A)^{-1} B$$

as a function of frequency (rad/sec). The singular values were analyzed for the normal and the congenital enzymes deficit (k10s=k14s=k20s=0.0) states with and without the optimized control input.

[RESULTS]

1.Stability and Controllability of the system.

Under the normal conditions (table 2), the 21 eigen values were composed of 17 negative reals and 4 sets of complexes with negative real parts. Thus the system was judged to be stable. The system was judged to be uncontrollable with 17 uncontrollable variables that were insensitive to 10 fold increases in kn.

2. The impulse responses with and without optimized control input.

Fig 2 shows the comparison of the impulse responses of X3 (Fig2-a) , X10 (Fig 2-b) between those without the optimized control input (Fig 2- left column) and those with them (Fig 2 right column). With the optimized control input, the maximum amplitude of each impulse response was depressed and the responses

Fig 2-a

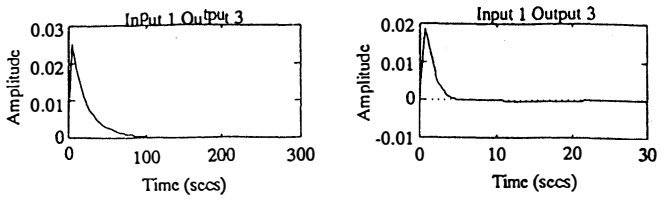


Fig 2-b

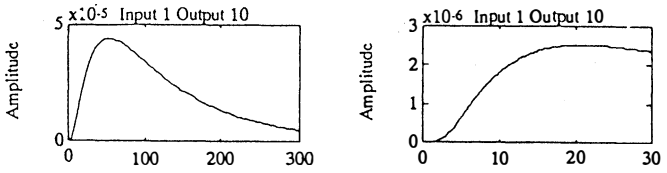


Fig 2. The impulse responses of the system.

Fig 3-a

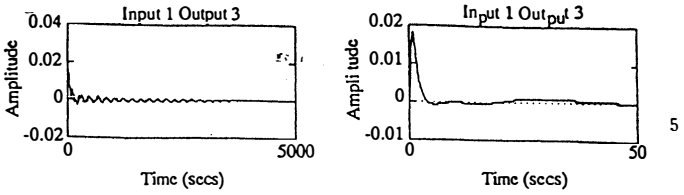


Fig 3-b

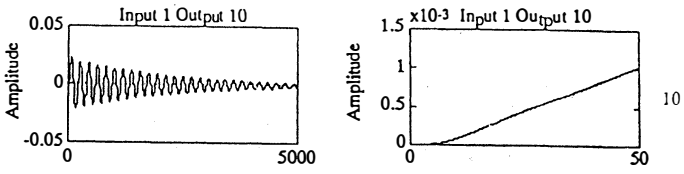


Fig 3. The impulse responses when the metabolizing enzymes do not exist (k10s=k14s=k20s=0.0)

Fig 4-a , 4-b, 4-c Fig 4-d, 4-e, 4-f

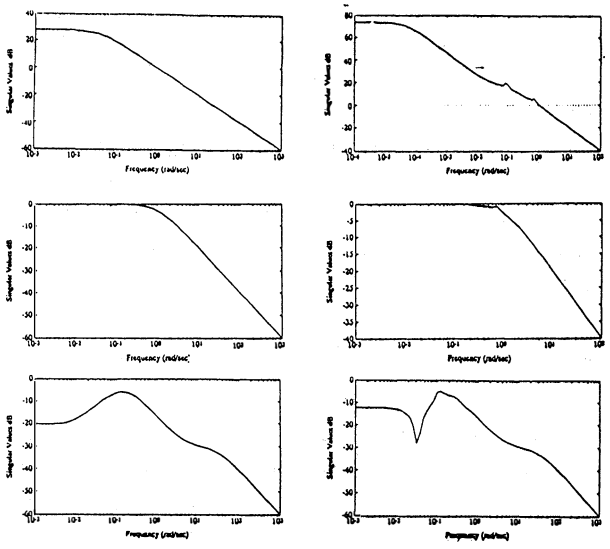


Fig 4. The singular value of the system.

terminated faster than those without the optimized control input.

3. The impulse responses when the metabolizing enzymes do not exist (k10s=k14s=k20s=0.0)but the system was supplied by optimized control input.

Without the optimized control input, the impulse responses oscillated (Fig 3-left) and lasted several 1000 secs while those with the optimized control input, these oscillations decreased and terminated earlier than those without the optimized control input. These responses were quite resemble to those when the metabolizing enzymes existed and controlled by the optimized control input (Fig 3- right column).

4. The singular value of the system.

Fig 4 shows the singular values of the system with (Fig 4-b, Fig 4-c, Fig 4-e, Fig 4-f) and without (Fig 4-a, Fig 4-d) the optimized control input when the metabolizing enzymes for Lysine, Methionine and Isoleucine exist (Fig 4-a , Fig 4-b, Fig 4-c) and do not exist (Fig 4-d and Fig 4-e, Fig 4-f). With the optimized control input, the amplitude of singular values (dB) were reduced particularly when the metabolizing enzymes do not exist (Fig 4-d to Fig 4-e). Under the optimized control input, there was little difference of the singular values between those when the metabolizing enzymes exist (Fig 4-b) and do not exist (Fig 4-e).

[DISCUSSION]

Present analysis has shown that the amino acids production started from Asparaginic acid is stable but uncontrollable. The optimized control input for Asparaginic acid shown in this study could reproduce normal physiological impulse responses even in the state of the congenital enzymes deficit. Thus for the treatment of the congenital disease, such an input as determined by U = - K X must be available.

[CONCLUSION]

The amino acids production processes started from Asrapaginic acid is stable but uncontrollable. The optimized control input diminished the oscillative impulse responses and negative values of the temporal behaviors of the system particularly in the state of enzymes deficit state. Present investigation affords important information for the medical treatment of congenital metabolic enzyme defects.

[REFERENCES]

[1]Stryler R. Biochemistry. Chapter 24. 1988.[2] Lecture on Biochemical Experiments. vol 11. . Tokyo Biochemistry Co.Limt.1974. [3]Data book of biochemistry .Chapter 13.Tokyo. Biochemical corporation. 1979.

Optimal Control in Active Transport in Biological Membrane

Number of Sites as the Control Input.

H,Hirayama., Y, Fukuyama.

Department of Public Health Asahikawa Medical College.
4-5. Nishi Kagura, Asahikawa city 078 Japan.

ABSTRACT

To analyze the transient behavior of the particles passing through biological membrane and its control principle, we have utilized optimal control theory to the 3 layers model proposed by Hill. The densities of particles in the boundary layers between the out side of the cell changed rapidly than those within membrane. Changes in reaction coefficients within the membrane influenced not only the intra membranes particle density but also those at the boundary surfaces. The 3 layers model of membrane was insensitive to changes in the weighting coefficients involved in the performance function of optimal control of membrane function.

[INTRODUCTION]

Hill , T, L and Kedem , O proposed a theoretical model for reversible thermodynamics regarding with active transport across membrane. Since this work, many investigations have been performed. However, the mechanism by which ions or molecules permeate through biological membrane had been yet not well understood. Particularly, the transient phenomenons of substrate transport through membrane is still unknown. The most critical problem has in relation to create artificial membrane is what is the control principle or control strategy of transport across the biological membrane. What do they economize or minimize to operate more efficiently.

In the present study we analyze the simplest membrane model proposed by Hill, T, L to show transient change in the state of molecules at each layer of biological membrane and optimal control principle.

[MATHEMATICAL PROCESS]

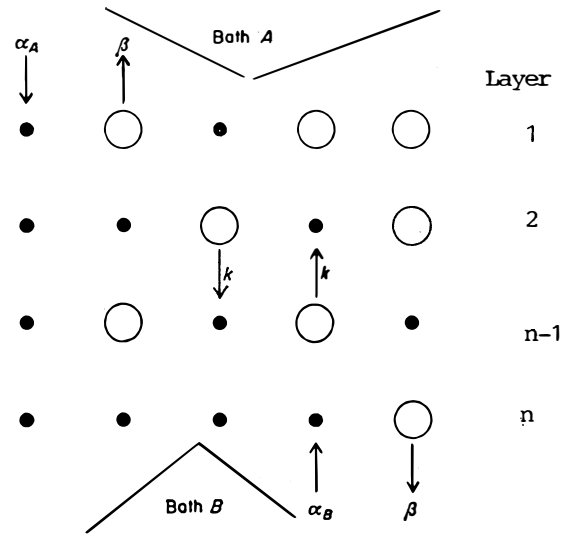
We set following coefficients. α a : the rate of active adsorption at the boundary layer facing to the bath A. α b : the rate of active adsorption at the boundary layer facing to the bath B. β a: the rate of active desorption at boundary layer A. β b: the rate of active desorption at boundary layer B. kn ; the rate of exchange of molecules between adjacent layers within the membrane. B : the number of sites in each layer. Nj : the occupation density of the j th layer. The N1 is the boundary layer facing to the bath A and N3 is another boundary layer facing to the bath B. The probable situations are shown in Fig 1. The system equations relating with Nj are

$$dN1/dt = \alpha a (B - N1) - \beta a N1 + k1 (N2 - N1) \quad \text{--(1)}$$
$$dN2/dt = k1 (N1 - N2) + k2 (N3 - N2) \quad \text{----------(2)}$$
$$dN3/dt = \alpha b (B - N3) - \beta b N3 + k2 (N2 - N3) \quad \text{-(3)}$$

Terms such as k (N2 - N1) require some explanation. Let N12 be the number of units with layer 1 and layer 2 sites both occupied . Then the rate at which layer 2 to layer 1 is k (N2 - N12) and the rate at which layer 1 to layer 2 is k (N1 - N12). Hence the difference is N2 - N1. Above equations are rearranged to

$$N1' = \alpha a B + N1 (- \alpha a - \beta a - k1) + k1 N2$$
$$N2' = N1 k1 + N2 (-k1 - k2) + N3 k2$$
$$N3' = \alpha b B + N2 k2 + N3 (- \alpha b - \beta b - k2)$$

Fig 1 : model of membrane transport of molecules. The layer 1 faces to the bath A while layer N faces to the layer B. The direction of arrows shows the way in which the reactions operate.



By setting adequate coefficients

N1' = a1 B + a2 N1 + a3 N2 ----(4)
N2' = a4 N1 + a5 N2 + a6 N3 ----(5)
N3 ' = a7 B + a8 N2 + a9 N3 ----(6)

We put B as the optimal control variable. The performance function we propose is

$$J = \int_0^{te} [\ \alpha 1\ (N1')^2 + \alpha 2\ (N2')^2 + \alpha 3\ (N3')^2\ ]\ dt$$

---------(7)

The biological significance of the performance function is to minimize the time dependent changes of concentration (density) at each layer of membrane. By expanding

d J / dt = α 1 [a1^2 B^2 + a2^2 N1^2 + a3^2 N2^2
+ 2 (a1 a2 B N1 + a1 a3 B N2 + a 2a3 N1 N2)]
+ α 2 [a4^2 N1^2 + a5^2 N2^2 + a6^2 N3^2
+ 2 (a4 a5N1 N2 + a4 a6 N1 N3 + a5 a6 N2 N3)]
+ α 3 [a7^2 B^2 + a8^2 N2^2 + a9^2 N3^2
+ 2 (a7 a8 B N2 + a7 a9 B N3 + a8 a9 N2 N3)]

Then , the integrand can be rearranged into
dJ / dt = B^2 (α 1 a1^2 + α 3 a7^2) + N1^2 (α 1 a2^2
+ α 2 a4^2) + N2 (α 1 a3^2 + α 2 a5^2 + α 3 a8^2)
+ N3 (α 2 a6^2 + α 3 a9^2) + 2 B N1 (α 1 a1 a2)
+ 2 B N2 (α 1 a1 a3 + α 3 a7 a8)
+ 2 B N3 (α 3 a7 a9) + 2 N1 N2 (α 1 a2 a3
+ α 2 a4 a5) + 2 N1 N3 (α 2 a4 a6)
+ 2 N2 N3 (α 2 a5 a6 + α 3 a8 a9)

The Hamiltonian function is
H = d1 B^2 + d2 N1^2 + d3N2^2 + d4 N3^2 + d5 B N1
+ d6 B N2 + d7 B NB3 + d8 N1 N2 + d9 N1 N3
+ d10 N2 N3
+ p1 (a1 B + a2 N1 + a3 N2)
+ p2 (a4 N1 + a5 N2 + a6 N3)
+ p3 (a7 B + a8 N2 + a9 N3)--------(8)

The optimal control can be obtained by differentiating Hamiltonian with respect to B.

d H / d B = 0. Then

2 d1 B + d5 N1 + d6 N2 + d7 N3 + a1 p1 + a7 p3 =0
----------(9)

Thus
B = d11 N1 + d12 N2 + d13 N3 + d14 p1 + d15 p3.
Putting this optimal B into state equations of N1 and N3

N1' = N1 (a1 d11 + a2) + N2 (a1 d12 + a3) + N3 (a1 d13) + p1 (a1 d14) + p3 (a1 d15) ----------(10)

N3' = N1 (a7 d11) + N2 (a7 d12 + a8) + N3 (a7 d13 + a9) + p1 (a7 d14) + p3 (a7 d15). ----------(11)

There is no change in N2. The differential equations for the co-state variables are

p1' = - d H / dN1 = - [2 d2 N1 + d5 B + d8 N2 + d9 N3 + a2 p1 + a4 p2]
= N1 (- 2 d2 - d5 d11) + N2 (-d8 - d5 d12)
+ N3 (-d9 - d5 d13) + p1 (-a2 - d5 d14) + p2 (-a4)
+ p3 (-d5 d15) ------(12)

p2' = - d H / dN2 = - [2 d3 N2 + d6 B + d8 N1 + d10 N3 + a 3p1 + a5 p2 + a8 p3]
=N1 (-d8 - d6 d11) + N2 (-2 d3 - d6 d12)
+ N3 (-d10 - d6 d13) + p1 (-a3 - d6 d14) + p2 (-a5)
+ p3 (-a8 - d6 d15) ------(13)

p3' = - d H/ dN3 = - [2 d4 N3 + d7 B + d9 N1 + d10 N2 + a6 p2 + a9 p3]
= N1 (-d9 - d7 d11) + N2 (-d10 - d7 d12) + N3 (-2 d4 - d7 d13) + p1 (- d7 d14) + p2 (-a6)
+ p3 (-a9 - d7 d15) ----------(14)

Thus N1, N2, N3, p1,p2 and p3 express the optimal state equations. For the simplicity we set two boundary conditions such that

N1(0) =0.01, N1(te) =0.8
N2(0) =0.01, N2(te)=0.9,
N3(0) =0.01, N3(te) =1.0 ------ (15)

This mathematical problem was solved numerically by multiple shooting method. For the standard conditions, we set, α 1= α 2= α 3=1.0, α a= α b=0.5, β a= β b=0.1, k1 =k2 =0.1.

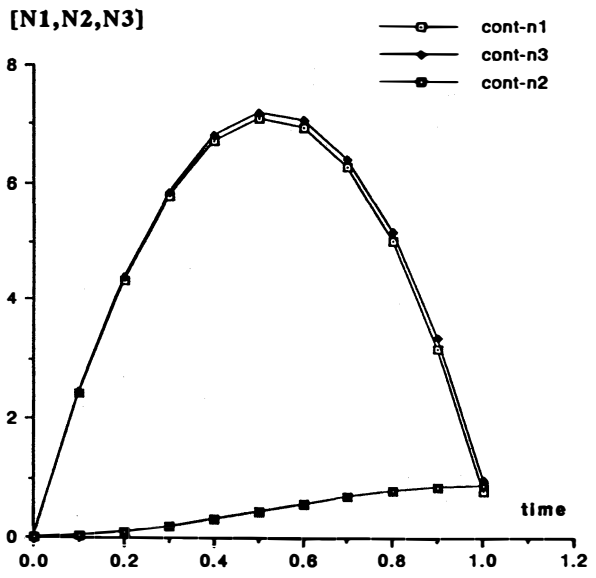


Fig 2 express the transient response of occupation density [Nns]. The reaction time was normalized to 1.0 sec.

Fig 3-a

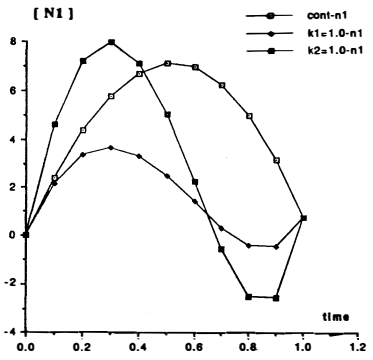


Fig 3-b

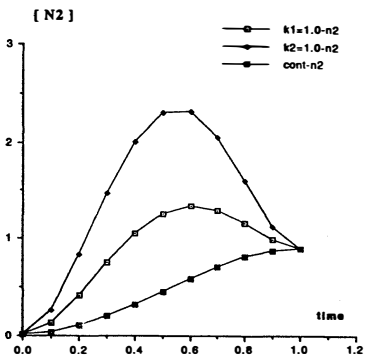


Fig 3-c

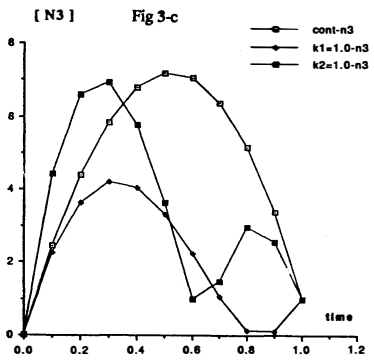


Fig 3. The effects of 10 fold increases in the k1 and k2 on the temporal changes in N1 (Fig 3-a), N2 (Fig 3-b) and on N3 (Fig 3-c). In each figure, cont-nm indicates the standard state.

[RESULTS]

1. The total behavior of N1 N2 and N3.

Fig 2 shows time dependent changes in [N1] [N2] and [N3] at control level. The [N1] and the [N3] showed parabolic pattern with definite peaks while [N2] increased monotonously.

2. Effects of reaction coefficients k1 and k2. Fig 3.

To disclose the effect of k1 and k2 , these parameters were increased ten times from kn =0.1 to 1.0. The [N1] changed into bi-phasic pattern (sinusoidal pattern)followed by negative values with increase in k1 and k2. A pronounced change was manifested with changes in k2. [N2] showed a definite peaks with increase in k1 and k2 while at control level [N2] increased monotonously. A marked increase occurred with increase in k2. [N3] also showed bi-phasic pattern but negative values. A larger hump manifested at the initial quarter of reaction time. Then a second hump developed at near the end. These changes are evident with change in k2.

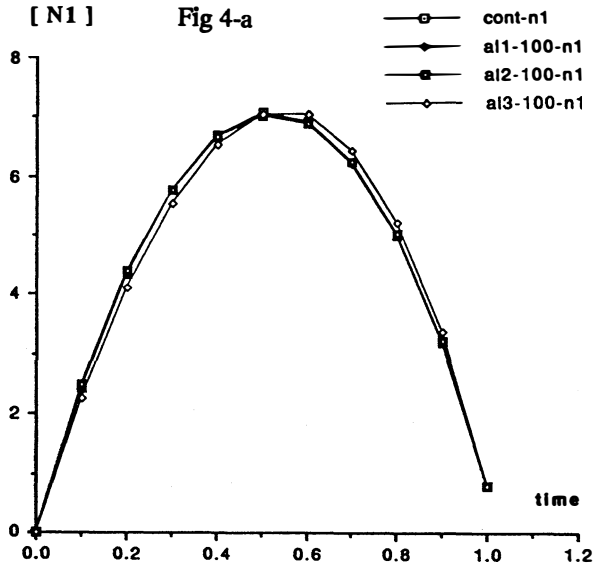
3. The effects of weighting coefficient $\alpha 1$, $\alpha 2$ and $\alpha 3$

Fig 4 shows the effect of 100 times increase in weighting coefficients of αn. With increase in αn, the time to the peak was shortened, the peak magnitudes were decreased and the curves changed total gradual. These changes, however, were small .

4. The effects of reaction coefficients α and β.

Fig 5 express effects of adsorption, desorption coefficient at both sides of boundary layer. With increase in any kinds of reaction coefficients, the [N1] showed bi-phasic pattern like sinusoidal wave. A prominent change occurred with increase in αb from 0.5 to 1.0. Since the αb is the adsorption coefficient of boundary layer facing to bath B which is opposite to A, the changes in [N1] seem to be interesting. Another influential one was the βb which is also related with bath B. Thus the [N1] (boundary layer facing to the bath A) is more affected by reaction coefficients relating

Fig 4. The effects of 100 fold increases in the weighting coefficients, $\alpha 1, \alpha 2, \alpha 3$ on the temporal changes in N1 (Fig 4-a), N3 (Fig 4-b). In each figure, cont-nm indicates the standard state.



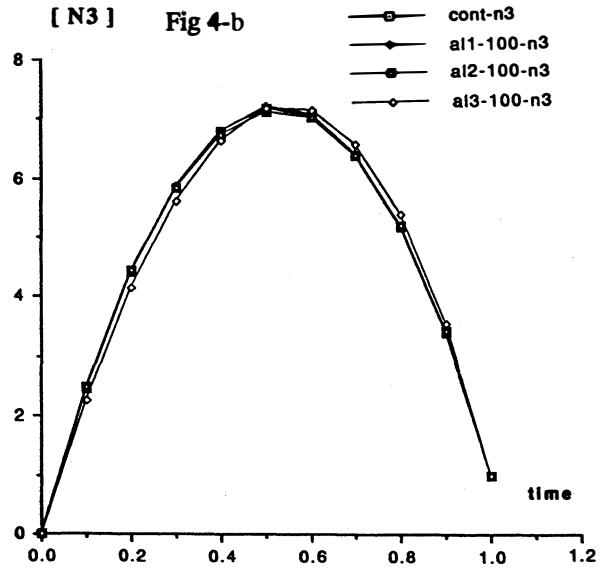


Fig 5-a

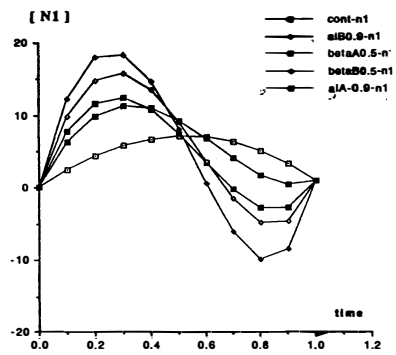


Fig 5-b

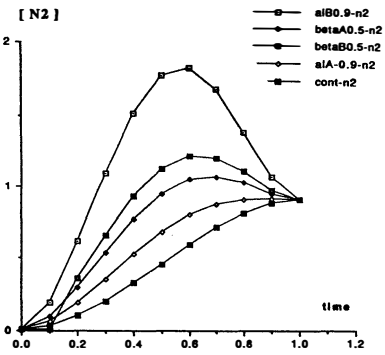


Fig 5-c

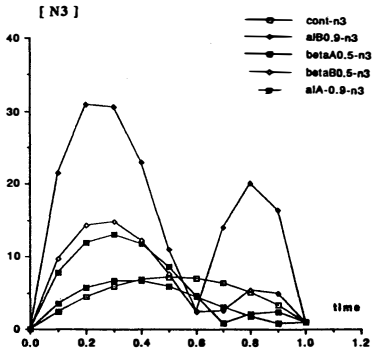


Fig 5. The optimal time courses of concnetrations in N1 (Fig 5-a), N2 (Fig 5-b) and N3 (Fig 5-c) when the coefficients were increased to $\alpha A = 0.9$, $\alpha B = 0.9$, $\beta A = 0.5$, $\beta B = 0.5$ independently while all other system parameters were at the standard level.

to bath B. [N2] increased with increase of any kinds of reaction coefficients. Here also coefficient relating to bath B had larger effect than those of bath A. What was another finding was that [N2] changed its pattern from monotonous increase to have a definite peaks. [N3] changed its pattern from parabolic pattern to bi-phasic pattern with definite two peaks with the first the larger. In this case also, the reaction coefficient relating to bath B had larger effects than those relating to bath A.

[DISCUSSION]

1. The behavior of the occupation density

There was little difference between [N1] and [N3] which are the boundary layers facing to different baths. Fig 2 showed transient pattern of optimized behavior of density of molecules. Indeed rapid changes were limited only to the boundary layers while a monotonous increase manifested at the middle layer. These changes of density at each layer must in part be dependent on the setting of the boundary values at the terminal of the reaction (t = te). The essential pattern, however, seemed to be described on this Figure.

The changes in reaction coefficient kns influenced on density [N1] and [N3] in the boundary layers. The times to the peak of two boundary layers were shortened with increase in k2 (fig 3). This indicated that the changes developed within the membrane (deeper layer) affected not only the layer within which it was involved but also, boundary layers facing to the outer world. Another importance was that the calculated density resulted in the negative values at near the terminal of reaction time. We could not offer any comprehensive explanation of this result. This result may be originated in the setting of boundary values or in the limit of the numerical calculation. (multiple shooting method). However, both k1 and k2 influenced much on the two boundary layers, it seemed likely that changes developed within the membrane surely would affect on the properties of the membrane surface.

Almost similar pattern of densities could be observed when the rate of adsorption and of desorption have been changed. A particular interest was that the density showed always larger magnitude with increase in rate coefficients relating to the bath B (Fig 5 filled diamonds and open diamonds). These asymmetricity may be originated in the mathematical structure of the biological membranes but comprehensive explanation have to wait the experimental data.

Unexpectedly there was little effect by the weighting coefficient. This must be 1), the magnitudes of weighting coefficient were too small or 2) the mathematical constitution itself was insensitive to changes in weighting coefficient in another word, the 3 layers biological membrane are created optimally from the initiation of the evolution.

[Conclusion]

1.Densities of molecules in the boundary layers changed rapidly than molecules within membrane.

2. Changes in reaction coefficients within the membrane influenced not only the intra membranes molecular density but also those at the boundary surfaces.

3. The 3 layers model of membrane was insensitive to changes in the weighting coefficients involved in the performance function of optimal control of membrane function.

[REFERENCES]

1. HILL, T,L., KEDEM, O. J. Theoretical Biology. Vol. 10. pp 399-441.

2. M.S. Seshadri,: J. Theoretical Biology. Vol. 31. pp 245-253. 1971.

3. Frehland, E and Lauger P. J. Theoretical Biology. Vol. 47. pp 189-207. 1974.

Optimization of Growth of Peptide Chain by Ribosomes on a Messenger RNA.

H,Hirayama., Y, Fukuyama.

Department of Public Health Asahikawa Medical College.
4-5. Nishi Kagura, Asahikawa city 078 Japan.

ABSTRACT

A theoretical approach was challenged to disclose the properties of temporal changes in poly peptide chain growth on a messenger RNA when more than 2 ribosomes work on. The system was stable and controllable. There was only one singular value which was depressed by the system optimization process. The impulse response terminated to 0 value earlier with smaller magnitude in the optimized system than that calculated without optimization. Present model is available to evaluate the peptide synthesis on a messenger RNA.

key words :. Peptide chain, Messenger RNA , Ribosomes, Stability, Controllability, Singular value, Optimality.

[INTRODUCTION]

The ultimate nature of biological system is determined by DNA and resulting protein synthesis on a messenger RNA. The biological information is finally converted to be a kind of protein which acts as the executor of the cell. Yet, what is the organization strategy of peptide synthesis on a messenger RNA is still unclesr. Because to know the control strategy of peptide synthesis is available to evaluate the functional properties of DNA, messenger RNA and peptide synthetic process. More over it will be possible to diagnose path physiological change and relation to hereditary disease in protein synthesis on the messenger RNA level.

In the present investigation, we analyze the sequential processes of peptide change elongation mechanism by linear system analysis. Moreover we have analyzed the effects of system optimization.

[Modeling of poly-ribosome transition]

Fig 1 shows a model for the poly-ribosome mechanism. Amino acid residues in the form of aminoacyl-S-RNA compounds (AS) are delivered to the growing peptide chain. When the first ribosome has progressed in a certain distance along the messenger RNA from its left tip, p amino acids residues will be added to the ribosome. After that a second ribosome can attach to the messenger RNA. This process continues until the full complement of ribosome per messenger RNA is achieved. When the production of each peptide chain is complete, the polypeptide and ribosome are liberated from the right-hand end of the messenger RNA while the remaining bound ribosomes continue to move on the messenger RNA. Another ribosome will attach to the left end of the messenger RNA.

Messenger RNA is linked to ribosome with time dependent rate g(t). The free messenger RNA (M) is catalyzed with the rate of kd while with the rate of km when the messenger RNA is bound to ribosome. Synthesis of peptide is started by binding a ribosome (R) to a strand of free messenger RNA. Aminoacyl-S-RNA compounds (AS) is indispensable for the this linkage process. Therefore

$$M + R + AS \xrightarrow{k0} M\text{-}R\text{-}AS \quad \text{----(1)}$$

Growth of the peptide requires adequate adhesion of amino acyl-S-RNA on the messenger RNA (M) - ribosome (R) complex (M-R) at a position adjacent

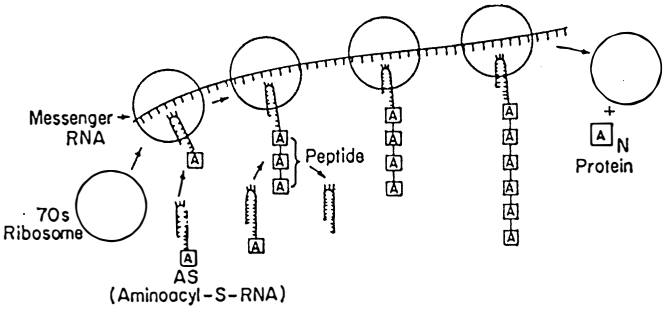


Fig 1 Present model of generalized peptide chain synthesis on a messenger RNA. Amino acid reside is expressed by A.

to the growing peptide site. Amino acid residue (A) is then added to the peptide by non specific amino acid transferase. The amino acid is linked through the carboxyly group to the soluble RNA (S) and the formation of the peptide bond is achieved by the transfer of the entire peptide chain to the amino acid group of AS. The growth step is expressed by binding at the active site followed by a transfer reaction

$$\text{M-R-AS} + \text{AS} \underset{k-1}{\overset{k1}{\rightleftharpoons}} \text{M-R-AS.AS} \quad \text{------(2)}$$

$$\text{M-R-AS.AS} \xrightarrow{k} \text{M-R-A2S+ S.} \quad \text{---(3)}$$

In (2), the AS to the right of the dot expresses the bound species. This sequential reactions are repeated during growth of the peptide. Thus generally

M-R-Ap-1 S + AS ------ M-R-Ap-1 S. AS ---(4)

M-R-Ap-1 S. AS -------- M-R-Ap S + S -----(5)

When the first ribosome has translated , p amino acids in the peptide chain is produced. Then another ribosome can be added and thereby start the second growing peptide chain. The reactions subsequent to equation (5) can be expressed by

M-R-ApS + AS + R (the second ribosome)

---------- M (R-AS) (R-ApS) ----(6)

Iteration of reactions expressed by equation (1) to (6) results in polyribosome

M (R-Ai1S) (R Ai2 S) -- (R Aiq S)

We express the adsorbed AS on the binding site of a ribosome (i1, i2, i3, ---- iq/ 1). For the simplicity of analysis, we assume that addition of the amino acid residues occurs in a sequence initiating with the ribosome which locates farthest to the right, the most left sided near to the lastly added ribosome. Then

(i1, i2 ---iq-1, iq) --> (i1 i2 iq-1, iq/1) ---> (i1 i2 iq-1, iq+1)----> (i1 i2 iq-1 + 1 , iq + 1)

----> (i1 + 1 , iq-1 + 1 , iq + 1).

The messenger RNA is divided into q regions. As the ribosome that has linked with some specific region on the messenger RNA moves along the region, the p amino acid residues are added to the growing peptide chain so that p*q peptide residues are added in all.

Generally, following 4 rules operate

1). When the most left site on the messenger RNA is empty, the newly arrived AS occupies the most left side even though other sites are empty.

(0, 3, 0) ---> (1,3,0) and not (0,4,0)

2). When all other sites are empty except the first site, and the first site is yet fulfilled, the newly arrived AS is added to the first site.

(1,0,0)--->(2,0,0)--> (3,0,0)--->(p,0,0,)

3). When a give site has been fulfilled by the amount of the product equivocal to p (amino acids residues)*q (regions), the fulfilled ribosome moves on to the next site (q + 1) and at there the amio acids attains p*q + 1. After this transition, the q th site becomes empty.

(2,4, 0/1)----> (2,0,5)

(2, 0, 0/1)----> (0,3,0)

(2, 0, 5)----->(0,3,5)

4). When all the sites are not fulfilled, the newly arrived AS ia added from the right of the total sequence

(1,3,5) ------> (1,3,6)

(1,3,6) ------>(1,4,6)

(1,3,0) ------>(1,4,0)

Thus generally for q (regions) each of which has the capacity of p (amino acids residues) ,

(0,0,0--- , 0)--> (1,0,0--,0) --> (2,0,0,--,0) ---> (p-1,0,0--, 0) -------> (p,0,0 -- , 0).

(p,0,0 -- , 0/1) ---> (0, p+1,0,0--, 0)

---> (1, p+1,0,0,--, 0) ------> (1, p+2,0,0,--, 0)

(1, p+2,0,0,--, 0/1)----> (1, p+3,0,0,--, 0)

(1, p+ p -1 ,0,0,--, 0/1)----> (1, 2p,0,0,0)

(1, 2p,0,0,0/1)----> (2, 2p,0,0 0)

----> (3, 2p, 0, 0--0)-->-->---> (p, 2p, 0, 0 0)

(p, 2p, 0, 0 0 /1) ---> (p, 0,2p+1, 0,0,0)

(p, 0,2p+1, 0,0,0)--->(0, p+1, 2p+1, 0,0,)

(0, p+1, 2p+1, 0,0,)--> (1, p+1, 2p+1, 0,0,)

(1, p+1, 2p+1, 0, 0,0 ,0/1)-->(1,p+1,2p+2,0,0)

(1,p+1,2p+2,0,0)---> (1, p+1, 2p+3, 0,0)-->-->--> (1, p+1, 3p, 0,0,0)-->(1,2p,3p,0,0,--0)

Then the differential equations of this system are

d (0,0, * *,0) / dt = g(t) - (k0 r u + km) (0,0,***, 0)

d (1,0,***,0) / dt = k0 r u (0,0,***, 0) - (k1 u + kd) (1,0,***,0) + k-1 (1,0,***, 0/1)

d (1,0,***,0/1) / dt = k1 u (1, 0,***,0) - (k-1 + k + kd) (1,0 , ***, 0/1)

d (2,0,***,0) / dt = k (1,0, *** 0/1) - (k1 u + kd) (2,0,***,0) + k-1 (2,0, ***, 0/1)

* * * * * * * * * *

d (p,0,***, 0) / dt = k (p-1,0,***, 0/1) - (k1 u + kd) (p,0,***,0)

d (0,p+1,0,***, 0) / dt = k (p, 0,***, 0/1) - (k0 r u + kd) (0, p+1, ***,0)

d (1, p+1, 0,***,0) /dt = k0 r u (0, p+1,0,***,0)
- (k1u + kd) (1, p+1, 0,***,0) + k-1 (1, p+1, 0, ***,0)
d (1,p+1,0,***,0/1)/dt = k1 u (1, p+1, 0,**,0) - (k-1 + k + kd) (1, p+1,0,***,0/1)
d(0,p+1, 2p+1, 0, ***,0) /dt = k (p,0,2p+1,0,*** , 0/1)- (k0 r u + kd) (0,p+1, 2p+1, 0, ***,0)
d(1,p+1,2p+1,0,***,0)/dt=k0ru (0, p+1, 2p+1,0, ***,0/1) - (k1 u + kd) (1,p+1, 2p+1, 0, ***,0)
+ k-1 (1, p+1, 2p+1,0,**,0/1)
d (p,2p,***, (q-1)p,0) / dt = k (p-1,2p, ****, (q-1)p,0/1) + kf (p,2p,***,qp)
- (k1 u + kd) (p,2p,***, (q-1)p,0/1)
d (p,2p,***, (q-1)p,0/1) / dt = k1 u (p,2p,***, (q-1)p,0) - (k-1 + k + kd)
(p, 2p ,*** ,(q-1)p, 0/1)
d(p,2p,***,qp) / dt = k (p-1,2p, ****, qp/1) - (kd + kf) (p,2p,***,qp)

dAn/dt = kf (p,2p, ***, qp) - kp An
d Pn / dt = kp An

In the present investigation we analyze the case when there are q (= 3) ribosomes each of which occupies a space on the messenger RNA corresponding to the designation of p (=2) amino acids. Thus full space state is expressed by vector form of (2,4, 6) and null state is by (0,0,0) respectively.

The state equations are expressed in vector form

$$X' = A X + B U, \qquad Y = C X$$

where X is the state vector of concentration of each component of the sequences reactions. U is the control input. Because, the sequential reactions are initiated from the state of (0,0,0), input g(t) was set to operate only to this state. Y is the out put vector. A is time invariant coefficient matrix of the system. B is the matrix for the control variable.

The stability of the total reaction of the system was investigated by analyzing the eigen values of the eigen equation of the system. The controllability was judged by the rank of the matrix. The number of uncontrollable variables were calculated by

$$\text{length (A) - rank (Co)} \qquad \text{where}$$

$$Co = [\, B \;\; AB \;\; A^2B \;\; A^3B \;\; A^4B \;\; ----- \;\; A^{22} B \,]$$

To calculates the optimal feed back gain matrix K such that the feed back law U = -KX minimizes the cost function,

$$J = \int_0^t [\, X Q X + U R U \,] \, dt$$

We have utilized the optimal control theory [4]. The Q and R are the weighting coefficients vectors composed of 23* 23 elements to characterize the relative magnitude of the minimization of each component X and the input U in the cost function. For the simplicity of analysis, all the elements were set to be unity at the standard level. To show the effects of relative magnitude of minimization for Xn, the elements of the weighting vector, all the elements of the matrix Q were increased to 10. The system supplied by the optimized control input is obtained by solving

$$X' = (A - B K) X$$

The initial conditions are set to be, X1 =1.0 (number for a given unit volume of a cell) and Xn =0.0 (2 < n <23). The singular values of the system were obtained by calculating the complex matrix,

$$C(j \omega I - A)^{-1} B$$

as a function of frequency (rad/sec).

[SETTING PARAMETERS]

The numerical values relating to the sequential transient movements of ribosome are still unclear because of little kinetic study. For the simplicity of analysis, we set following values of system parameters as the standard physiological state.

u=1.0 r=1.0 k0=1 k1=0.5 km1=0.01
kf=0.5 kp=0.1
kdn=0.1 (n=1 to 22)
k34=0.5 k56=0.5 k89=0.5
k1011=0.5 k1213=0.5 k1415=0.5
k1718=0.5 k1920=0.5 k2122=0.5

where u and r denote corresponding concentrations of AS and R. The kdn are those describing the catabolic process of each state and knm are the transitional rate constants.

[RESULTS]

1. Stability and Controllability

Eigen values of the sequential process are all negative without 0 value. Thus the system was stable. Surprisingly the rank of the matrix revealed that the system is controllable because of 0 uncontrollable variable.

2. The Singular values of the system.

Fig 2 shows the singular value of the system as function of frequency rad/sec. There was only one singular value. It showed a positive value below frequency 0.3 (Fig 2-a). Fig 2-b expresses the singular value when the system was optimized by minimizing the cost function and the feed back input was optimized. The singular value was depressed considerably. Fig 2-c shows the singular value when all the elements of the matrix Q were increased to 10. Further depression of the singular value was observed and it became independent from frequency.

Fig 2. The singular value of the system

Fig 2-a

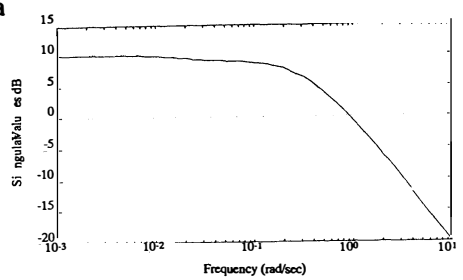


Fig 2-b

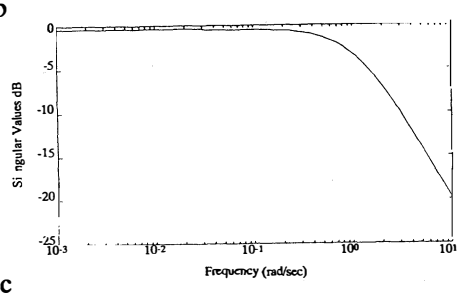


Fig 2-c

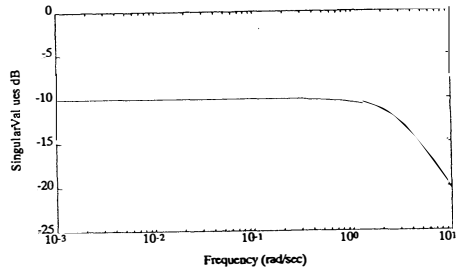


Fig 3. The impulse response of the system.

Fig 3-a

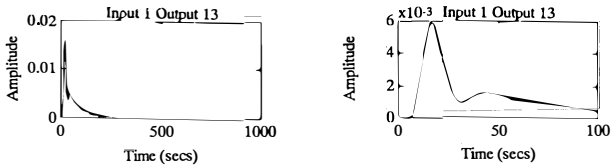


Fig 3-b

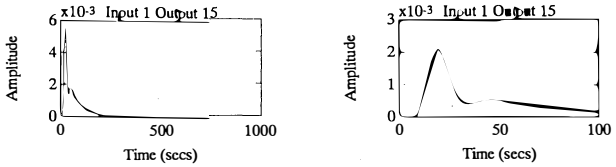


Fig 3-c

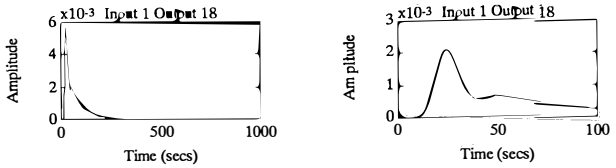


Fig 3-d

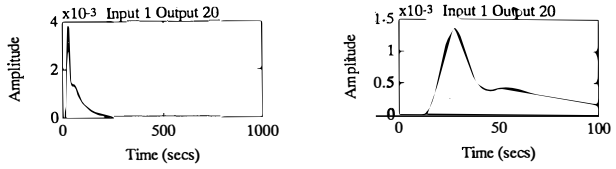
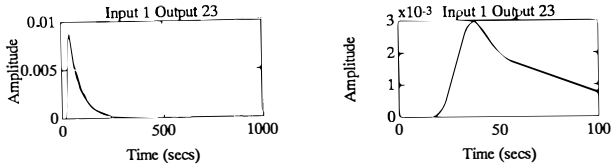


Fig 3-e



3. The impulse response of the system.

Fig3 shows the example of impulse responses of each state of the system. Fig 3-a expresses the transient change on (2,0,5) in response of impulse input. Fig 3-b, Fig 3-c, Fig 3-d and Fig 3-e show those of (0,3,5) (1,3,6), (1,4,6) and the final product respectively. The left column of the Fig 3 expresses those when the system was not optimized while the right column of the Fig 3 shows those when the system is optimized. The impulse responses with optimized system terminated earlier and their magnitude were lower than those of the non optimized impulse responses.

[DISCUSSION]

Present investigation showed that the sequential processes in growth of peptide change on a messenger RNA is stable and controllable. The impulse response showed that the processes of peptide synthesis works effectively without retention or depletion of intermediate products. Almost all the processes terminated within the same duration indicates that the synthesis process is driven to operate parallel at every step of the total processes. The biological significance to minimize the cost function is based on the concept that any biological system must operate so as to minimize the deviation from the target state simultaneously with minimize the consumption of the energy to drive the system. These features are disclosed explicitly in the Fig 4 in which the weighting matrix for the changes in each state of peptide synthesis were reinforced by 10 fold minimization (setting all the elements of the matrix Q to be 10). Thus by regulating elements in the matrix Q, simulation for optimal state in peptide synthesis on a messenger RNA must be possible.

[CONCLUSION]

1. Protein synthesis on the messenger RNA is stable and controllable.
2. The singular value of the system was sensitive to the optimization of the system in which the feed back input minimizes the deviation from the target state and consumption of energy to drive the system.
3. The impulse responses of the optimized system terminated earlier with smaller magnitude than those of non optimized system.

[REFERENCES]

1. Gerst I and Levine S,N. 1965. J. Theor. Biol. vol 9. pp 16-36. 2. Jacob, F and Mond J. 1961. J. mol. Biol. vol 3. pp 318. 3. Levinthal C., Fan, D.P. Higa., Zimmerman RA. 1963. Cold. Spr. Harb. Symp. quant Biol. XXVII. p 183.

Adaptive Evolution of Holon Networks by an Autonomous Decentralized Method

N. Honma* K. Abe*
*Dept. of Electrical Eng.
Tohoku Univ.
Sendai, 980-77, JAPAN

M. Sato
Communication Eng.
Tohoku Inst. of Tech.
Sendai, 982, JAPAN

H. Takeda
Dept. of Electrical Eng.
Tohoku Gakuin Univ.
Tagajo, 985, JAPAN

Key words: *Evolution, Autonomous decentralized system, Holon, Self-organize, Edge of chaos, Recurrent network.*

Abstract

We present a holon network that is a new model of complicated systems and their complex phenomena, such as *complex adaptive systems* or *artificial life* and their *evolution*. Holon networks have a hierarchical construction where each member of this hierarchy, called *holon*, is a network composed of lower members.

The networks are able to self-organize their structure by changing their parameters according to the environment. This is achieved by an evolution method with growth of the networks where all members, on whatever level, have their respective autonomous criteria based on properties of holons. In this method the initial holon network comprises only a few holons and the number of holons increases gradually and adaptively as occasion demands.

Some examples show that this method can lead to a network structure which has sufficient flexibility and adapts well to environments.

1 Introduction

Necessity to deal with large-scale and nonlinear complicated systems increases in many engineering fields, for instance in control and system engineering fields[6]. It is important to study what model is effectual for the problem of grasping the behavior of the large-scale nonlinear complex systems.

One of difficulties of building complex system models is caused by the variety of the system dynamics with extraordinary complexity[12, 3, 1]. The variety appears to be given rise to the various relations between the elements and the whole system. How do we describe such various relations in a "unified form" ?

The ability to explain these complex relations by reductionistic approaches may be limited. On the other hand, the notion of *holon* yields a new approach, beyond the reductionism and holism, which incorporates valid aspects of both[8]. An another concept of representing the property of complex systems using a holon character has been proposed as holonic control and holonic loop[13]. However, this holonic theory is not yet scientifically established.

In a previous paper we have proposed a model of the complex systems, called *holon network*, aiming at constructing a general system architecture based on the input-output relation with complicated interaction. Holon networks are able to self-organize their structures and adapt their parameters to the environment[4].

In this paper we present a new evolution method to achieve the evolution by adapting the number of holons comprising the network to the complexity of environments. It is shown that this method leads to network structure which adapts well to changes of environment and acquires the environmental dynamics for several estimation problems.

2 Holon network

Holon networks provide a common architecture of complex systems and a nature of components in the architecture. Fundamental properties described below, therefore, are ubiquitous in biology, psychology, ecology and wherever we find complex hierarchic systems, especially in organic systems[8].

2.1 Fundamental properties of holon network

First, the architecure of a holon network is *open-ended* multilevel hierarchical construction where each member of this hierarchy, on whatever level, is a subwhole, called *holon*, composing a network within the level(see Fig. 1).

Second, holons are Janus-faced. The face turned upward, toward the higher levels, is that of a depen-

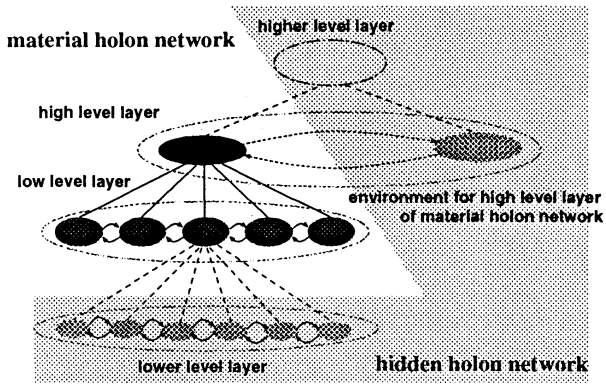


Fig. 1: open-ended multilevel hierarchical construction.

dent part; the face turned downward, towards its own constituents, is that of a whole of remarkable self-sufficiency. This implies that every holon is possessed of two opposite tendencies or potentials: an *integrative tendency* to function as part of the larger whole, and a *self-assertive tendency* to preserve its individual autonomy[8].

Finally, functions of holons are evolved by dynamical balancing the dual aspects of its holons, as wholes and as parts.

2.2 Holon networks for an estimation problem

In this paper we will consider an estimation problem whose task is estimating time sequences from an unknown binary dynamical system described in section 5. For this, two leveled holon networks are used; the lower level layer approximates arbitrary time sequences from unknown dynamics, and the mapping function of the higher level layer integrates the lower level dynamics.

We employ *binary* holons for members of the lower level. Binary holon networks constructed by one level layer are similar to *random Boolean networks*[7] or *cellular automata*[9] in respect of both of structure and function. The states of binary holons x_i are governed by

$$x_i(t+1) = B_i(t)\,(w_{i1}(t)y_1(t), \cdots w_{iN}(t)y_N(t), w_{iu}(t)u_i(t))\,, \quad (1)$$

$$y_i(t) = x_i(t), \quad (2)$$

$i = 1, \cdots, N$. N is the number of binary holons, and external inputs u_i, outputs y_i and states x_i are all *binary* variables, namely each variable takes the value either 0 or 1. B_i are Boolean functions, w_{ij} connection weights and w_{iu} weights of external inputs.

We consider the higher level composed of only one holon which has an static input-output relation given as a nonlinear function of feedforward neural networks F_{NN}. The output of the holon $\hat{y}$ is defined by

$$\hat{y}(t) = F_{NN}\,(y_1(t), y_2(t), \cdots, y_N(t))\,. \quad (3)$$

This means that the outputs of the lower level layer binary holons y_i are inputted to the higher level holon as arguments of the function F_{NN}(see Fig. 2). In this figure the function of each neuron f_{NN} is defined by

$$f_{NN}(x) = \frac{1}{1 + exp(2x/u_0)}, \quad (4)$$

where u_0 is a constant.

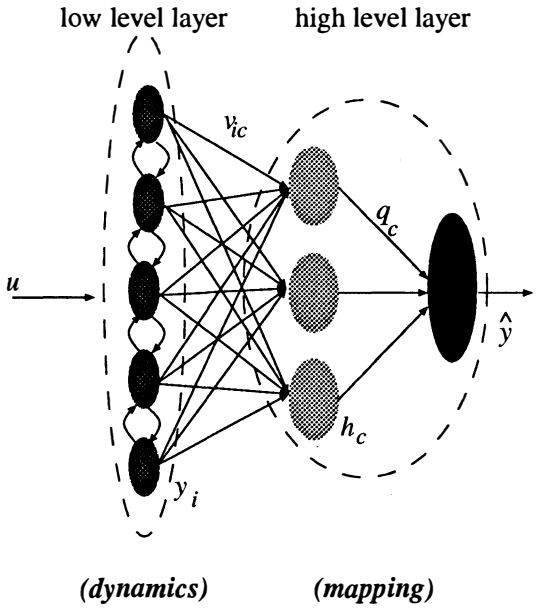


Fig. 2: Architecture of a neural holon network.

3 Dynamics of binary holon network

3.1 Parameters of the network dynamics

To parameterize dynamics of binary holon networks, we have presented two parameters, k_i and F_r[5], where k_i corresponds to parameter K in random

Boolean networks[7] and F_r corresponds to parameter λ in cellular automata[9].

The individual connection intensities k_i and the average $\bar{K}$ are defined by

$$k_i = \sum_{j=1}^{N} w_{ij}, \quad i = 1, 2, \cdots, N, \tag{5}$$

$$\bar{K} = \frac{1}{N} \sum_{i=1}^{N} k_i. \tag{6}$$

In this paper, we choose one of four Boolean functions for B_i, namely, $B_i \in \{AND, OR, XOR, \overline{XOR}\}$. Then, the parameter F_r, implying heterogeneousness of Boolean functions in the network, is defined by

$$F_r = \sum_{i=1}^{N} F_i(B_i), \tag{7}$$

where $F_i(B_i) = 1$ when B_i is either XOR or $\overline{XOR}$, and $F_i(B_i) = 0$ when B_i is either AND or OR.

3.2 Quantitative overview of binary holon network dynamics

Here, for the networks with $N = 20$, stepping through the ranges $0 \leq \bar{K} \leq N$ and $0 \leq F_r \leq N$ in discrete steps, we prepare binary holon networks for each $(\bar{K}, F_r)$ point by randomly choosing w_{ij} and B_i. Then we run these networks with $(\bar{K}, F_r)$, collecting data on the following two measures of their dynamical behavior. Finally, we investigate the relation between the behavior of these measures and each of two parameters.

We measure the complexity of the binary holon network dynamics using Shannon's entropy H and *pseudo* Lyapunov exponent λ' introduced in our previous paper[5].

Regarding the state transitions of each binary holon as a discrete process of two states, the entropy H_i and the average $\bar{H}$ are given by

$$H_i = -\sum_{s=0,1} Pr(x_i = s) \log Pr(x_i = s), \tag{8}$$

$$\bar{H} = \frac{1}{N} \sum_{i=1}^{N} H_i. \tag{9}$$

$\bar{H}$ as a function of $\bar{K}$ and F_r are shown in Fig. 3 and Fig. 4, respectively. Each point represents a distinct network structure indexed by the parameters: $\bar{K}$ and F_r.

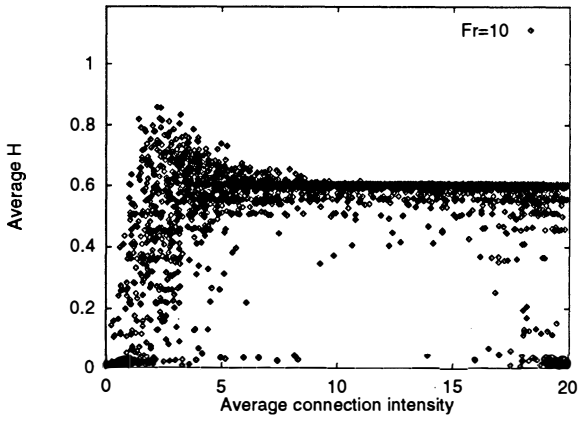


Fig. 3: Average single holon entropy $\bar{H}$ over $\bar{K}$ space for approximately 2000 binary holon network runs.

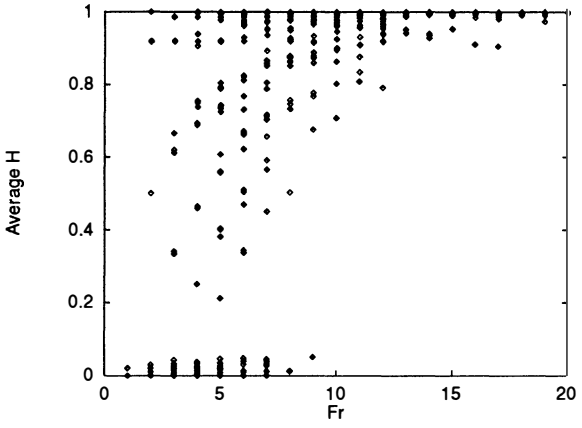


Fig. 4: Average entropy $\bar{H}$ over F_r space.

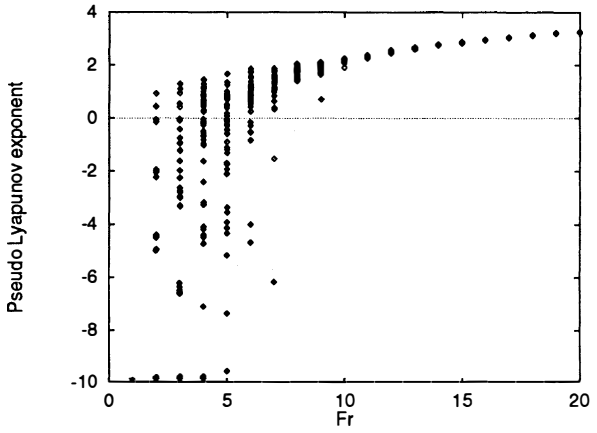


Fig. 5: *Pseudo* Lyapunov expnent λ' over F_r space.

In Fig. 3, we can see a remarkable feature in the case of around $\bar{K} = 2$. The resemble result to this feature has been reported in random Boolean networks[7]. It suggests that binary holon networks with around $\bar{K} = 2$ exhibit spontaneous collective order and that networks with large $\bar{K}$ are stochastically disordered differently from small $\bar{K}$ networks.

On the other hand, in Fig. 4, note that as F_r increases, $\bar{H}$ also increases stochastically. And the sparsely populated gap over $0 \leq F_r < 10$ and between $0.0 < \bar{H} \leq 0.4$ is observed. This distribution suggests the presence of a phase transition[9].

Pseudo Lyapunov exponent λ' measures an another complexity, which is one with respect to chaotic behavior, given by computing Hamming's distances between state transitions from an initial state and an another initial state added a minimum perturbation[5]. Fig. 5 shows *pseudo* Lyapunov exponent λ' as a function of F_r. Note that Fig. 5 bears a resemblance to Fig. 4. Also we observe a similar resemblance with respect to parameter $\bar{K}$. The relation between two measures shown in Fig. 6, however, we can conjecture an interesting fact that binary holon networks around $\lambda' = 0$, called *the edge of chaos*, where the phase transition between ordered and chaotic dynamics is observed, are able to give rise to various dynamical behavior from high entropy(*chaotic* or *disordered behavior*) to low entropy(*simple* or *ordered behavior*).

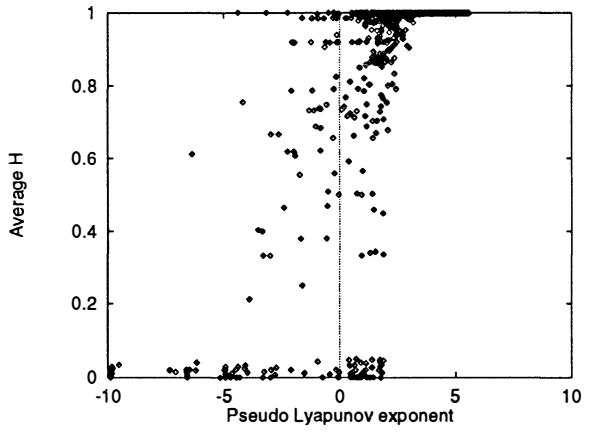


Fig. 6: Average entropy $\bar{H}$ versus *pseudo* Lyapunov expnent λ'.

Therefore, if we keep the dynamics of binary holon networks to the edge of chaos by changing parameters k_i and F_r suitably, then we can expect high efficient adaptation ability to wide range processes with complexity from high to low. In the following evolution method, as one of main strategies for adaptation to environments, we will make use of the feature of binary holon network dynamics controlled by the parameters.

4 Autonomous decentralized evolution

We propose an autonomous decentralized evolution method in order to estimate time sequences of unknown environments in desired accuracy. In this method neural networks giving the input-output relation of the higher holon are learned by the backpropagation algorithm, and the initial low level holon networks comprise only a few holons and they are grown gradually as occasion demands.

This evolution of low level networks is achieved by changing the two parameters of network structure according to two criteria based on the holon's dual character; *fitness criterion* given by estimation error, and *autonomous criterion* in order to keep the network dynamics the edge of chaos.

More details with respect to exerting of the dual characters of a holon, namely its *integrative tendency* and its *self-assertive tendency*, are discussed below. Throughout all level layer holons, the integrative tendency of a holon is affected by the self-assertive tendency of its higher level holon, while the two tendencies of a holon are independent each other[8].

In each generation $\tau = 1, 2, \cdots$, the change, denoted by δk_i, of connection intensity k_i, is calculated by

$$\delta k_i = \delta \bar{K}(\alpha_i) f_{\lambda'}(\lambda') f_k(k_i), \quad i = 1, 2, \cdots, N, \qquad (10)$$

where α_i is the backpropagation error from fitness defined by (15) on the i-th low level holon. This equation is also based on the result of section 3: As $\bar{K}$ or F_r increases, the complexity of the network dynamics also increases stochastically.

The integrative tendency of the higher holon works to improve fitness to the environment. This work has influence on $\delta \bar{K}$ in (10) as follows.

$$\delta \bar{K}(\alpha_i) = CN|\alpha_i|, \qquad (11)$$

where C is a proportional constant.

The self-assertive tendency of this higher holon has influence on the integrative tendency of its lower level layer holons as $f_{\lambda'}(\lambda') = -\lambda'$ in (10). This term as autonomous criterion acts in order to keep the network dynamics around the edge of chaos.

On the other hand, the self-assertive tendency of low level holons works to maintain the present *configuration*; values of k_i. The term of these works in (10)

$f_k(k_i)$ is given as

$$f_k(k_i) = \begin{cases} 1 + k_i - kc_i & (\lambda' < 0), \\ 1 + kc_i & (\lambda' \geq 0), \end{cases}$$

where kc_i is defined by

$$kc_i = \sum_{j=1}^{N} w_{ij} F_j(B_j). \tag{12}$$

Finally, we present an evolution method:
step 1: Set a low level layer holon network at an initial generation, by taking $N = 1$, initializing B_1 to a randomly chosen Boolean function out of the four Boolean functions, and initializing w_{11} and x_1^0 to a randomly chosen binary number.
step 2: In each generation, calculate fitness to the environment, and then give the backpropagation error α_i to the i-th low level layer holon.
step 3: If the fitness is successively greater than the desired value for generations fixed beforehand, then exit, otherwise go to *step 4*.
step 4: For all $i = 1, 2, \cdots, N$, execute *step 5* and *step 6*.
step 5: Compute $\delta k_i(\tau)$ by (10).
step 6: If δk_i is greater than 0 and if $kc_i + \delta k_i$ is greater than F_r, or if δk_i is less than 0 and if $k_i - kc_i - \delta k_i$ is greater than $N - F_r$, that is the change cannot be achieved present network configuration, then generate one low level holon with the following Boolean function decided by sign of δk_i:

$$B_{N+1} = \begin{cases} XOR \quad \text{or} \quad \overline{XOR} & (\delta k_i > 0), \\ OR \quad \text{or} \quad AND & (\delta k_i \leq 0). \end{cases}$$

Then, change N to N+1,
else change w_{ij} as follows: If δk_i is greater than 0, then let kc_i increase, else let $k_i - kc_i$ increase.
step 7: Go back to *step 2* at next generation.

5 Simulation results

We have tested the evolution method for holon networks on an output estimation problem of the following discrete time nonlinear systems with single input and single output.

$$\boldsymbol{x}(t+1) = \boldsymbol{f}\left(\boldsymbol{x}(t), u(t)\right), \tag{13}$$

$$y(t) = g\left(\boldsymbol{x}(t), u(t)\right), \tag{14}$$

where $u(t)$, $y(t)$ and $\boldsymbol{x}(t)$ denote the external input, the output and the state vector at time $t(= 1, 2, \ldots)$, respectively. For simplicity, we shall consider binary systems where each variable takes the value either 0 or 1: The input u, the output y and the components of the state vector $\boldsymbol{x}$ are binary variables. The components of transition function $\boldsymbol{f} = [f_1, f_2, \cdots, f_{nf}]^T$ and output function g are all unknown Boolean functions.

The task of solving the problem requires the network to learn to configure itself so that it may store important information obtained from the input sequence at earlier times to help determine the output at later times. In other words, the network is required to acquire approximately the dynamics of unknown system to accomplish the task.

For the above task, the evolution was made by means of a real-time estimation method that is not an iterative learning method feeding repeatedly the same pattern at every generation. We set the input signal $u(t)$ at each time $t = 1, 2, \cdots$ a randomly chosen binary value and the length $D = 100$ (see [5]). The fitness to the environment at τth generation, $fit(\tau)$ $(0 \leq fit \leq 1)$, is defined by the estimation error, $e(t) = y(t) - \hat{y(t)}$, as follows:

$$fit(\tau) = 1 - \frac{1}{D} \sum_{t=\tau-D+1}^{\tau} |e(t)|. \tag{15}$$

Here we suppose that the unit interval of the system time t is equal to the generation's unit interval.

We investigated adapting ability and the effect of the *autonomous criterion* on the simulation experiment where the environment was changed every 100 generations: the transition functions and output functions of the environmental system are changed. The learning by the backpropagation algorithm was done every generation except for the generation when the fitness at the previous generation was greater than 0.9.

The fitness as a function of evolutional generations by using the proposed evolution method is shown by the solid line in Fig. 7, and the broken line shows the fitness by using a method excepted for the term of the *autonomous criterion* effect $f_{\lambda'}(\lambda')$ in (10). This exception of the effect was made by ignoring the value of λ' and letting $f_{\lambda'}$ be 1 every generation.

Despite the changes of environment, the fitness is maintained effectively in desired values, namely not less than 0.9, for several successive generations. This fact suggests that the holon network acquires approximately not only static mapping from an input to an output sequence of the systems at a certain generation but also dynamics of the systems. Also the efficient adaptation to changes of environment suggests that the proposed self-proliferative evolution method can reduce the expensive computation time in comparison

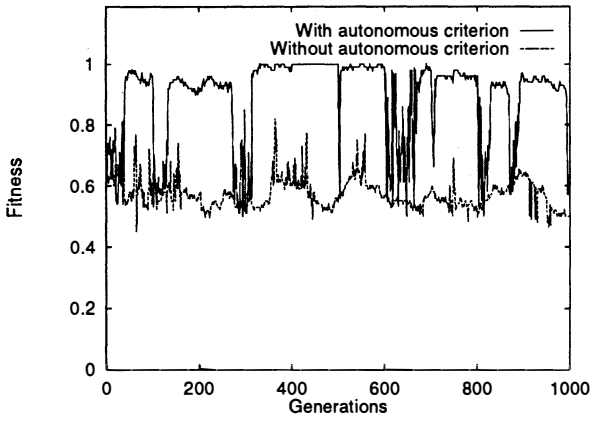


Fig. 7: Fitness as a function of generations for adaptation problem to various sequential circuits dynamics. Solid line shows evolution with autonomous criterion, and broken line shows evolution without the criterion.

with our former evolution method in which the number of low level layer holons was fixed[5].

As is clear from the results, to have the appropriate *autonomous criterion* independent of the *fitness criterion* is efficient for adaptation to changes of environment.

6 Conclusion

It is shown that holon networks are useful for modeling of complicated nonlinear dynamical systems. By using the presented evolution method, holon networks are able to have a great deal of flexibility and adaptability to changes of environment, and it is possible to reduce the expensive computation time since the number of low level layer holons is increasing gradually and adaptively as occasion demands.

The main concept of the method is to keep balance of two independent criteria; the *fitness criterion*, namely total criterion, and the *autonomous criterion*. This concept can be incorporated into other methods, such as gradient decent methods[11, 10, 14] for recurrent neural networks and genetic algorithms[2] for optimization problems. Indeed, we have proposed a method in which an genetic algorithm is combined with an evolution method where the number of low level layer holons is fixed, and illustrated the effects of the evolution method[5].

In several situations, the change of network configurations required by the *autonomous criterion* may be different from the change required by the *fitness criterion*. However, it seems that this self-inconsistency makes the evolution of holon networks be rich and flexible since the autonomous requirements are compromised with the total requirements by balancing the two faces of each holon. In fact, if the changes by the two criteria were always consistent, the *autonomous criterion* is not needed for the evolution. This notion of systems will be useful for the autonomous decentralized system theory and the emergent system theory[6].

References

[1] P.Bak and K.Chen, "Self-Organized Criticality", Scientific American, *January*, pp. 26-33, 1991.

[2] D.E.Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

[3] J.Horgan, "From Complexity to Perplexity", Scientific American, *June*, pp. 74-79, 1995.

[4] N.Honma, M.Sato, K.Abe and H.Takeda, "Identification of Nonlinear Dynamical System with Autonomous Decentralized Adapting Methods using Holon Network Model", Prep. of The 36th Japan Joint Automatic Control Conference, pp. 141-142, 1993.

[5] N.Honma, M.Sato, K.Abe and H.Takeda, "On Evolution of Holon Network by using Autonomous Decentralized Adaptation Method", Trans. of SICE, vol. 32, No. 7, pp. 908-915, 1995.

[6] S.Kitamura, At the starting point of the research project on "System Theory of Function Emergence", Proc. of the 34th SICE Annual Conference, pp. 875-876, 1995.

[7] S.A.Kauffman, "Antichaos and Adaptation", Scientific American, *August*, pp. 64-70, 1991.

[8] A.Koestler, Janus, Kousakusya, 1983.

[9] C.G.Langton, "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation", Physica D vol. 42, pp. 12-37, 1990.

[10] B.A.Pearlmutter, "Learning State Space Trajectories in Recurrent Neural Networks", Neural Computation, vol. 1, No. 2, pp. 263-269, 1989.

[11] F.J.Pineda, "Generalization of Back-Propagation to Recurrent Neural Networks", Physical Review Letters, vol. 59, No. 19, pp. 2229-2232, 1987.

[12] R.Ruthen, "Adapting to Complexity", Scientific American, *January*, pp. 110-117, 1993.

[13] H.Simizu, Life and Place, NTT Publishing Co.,Ltd., 1992.

[14] R.J.Williams and D.Zipser, "A learning Algorithm for Continually Running Fully Recurrent Neural Networks", Neural Computation, vol. 1, No. 2, pp. 270-280, 1989.

Development of A Stair Climbing Mobile Robot with Legs and Wheels

Toshihiko Mabuchi, Takeshi Nagasawa, Keizou Awa, Kazuhiro Shiraki and Tomoharu Yamada

Department of Mechanical Engineering
Kansai University
3-3-35 Yamate-cho, Suita, Osaka 564 Japan

abstract

This paper deals with the development of a stair climbing mobile robot with legs and wheels. The main technical issues in developing this type of robot are the stability and the speed of the robot while climbing stairs. The robot has two wheels in the front of the body to support its weight when it moves on plain terrain, and it also has arms between the wheels to hook onto a tread of stairs. These are two pairs of legs in the rear of body. Using not only the rotational torque of the arms and the wheels but also the force of the legs, the robot goes up and down stairs. It measures the size of stairs when going up and down the first step, therefore the measurement process does not cause this robot to lose any time. The computer which controls the motion of the robot needs no complicated calculations as other legged robots do. The mechanism of this robot and the control algorithm are described in this paper. This robot will be developed as a wheelchair with a stair climbing mechanism for disabled and elderly people in the near future.

1 Introduction

In Japan there are more than one million people who have disability with lower limb in 1993, and the number of the people is going up in this rapidly aging society. Due to the shortage of helpers in the near future, society will have a certain need of robots as a substitute for human beings that offer to support disabled and elderly people. To develop this kind of robots there are many problems which must be solved. The most important and difficult thing in designing these robots is how to make the disabled and elderly people who use robots in their daily life comfortable. On the other hand, many technical problems must also be solved.

It is not easy to move with wheel chair even at public place where all citizen lead a life because of steps. Then elevators or slopes should be equipped at these places, but under the actual situation in Japan there are few places. The ratio of stations equipped elevator is a very low: There are only 2.5 percent at JR, 5.4 percent at private railroads in 1994.

In the social background we develop a new stair climbing mobile robot. The real technical issues in developing a stair climbing robot are how simple the robot can be made and how fast it can move with stability when climbing stairs. There are many kinds of stair climbing robots: for the maintenance of nuclear reactor or other plants,[1][2] for disabled and elderly people as wheelchairs,[3]. There are three basic types of mobile mechanism: a wheel type, a crawler type and a leg type. In the wheel type an energy efficiency is higher to move the plain terrain than the others and this type has been the most popular mechanism.[4][5] In crawler types a terrain adaptivity is higher than the wheel type and this type is most useful to move on sand or mud place, but not suitable for plain terrain.[6] Legged type has the highest adaptivity to rough terrain

and can move on stair or mountain with stability, because contact points of ground where feet support the body would be safely selected. But this mechanism has problems : load weight, energy efficiency and speed of movement so far.[7][8][9] Then we will adopt the new mechanism with both legs and wheels.

2 Construction of The Robot

Figure 1 shows the structure of this robot and the robot is 1/2 scale model. Two special mechanisms to go up and down stairs are equipped with this robot.

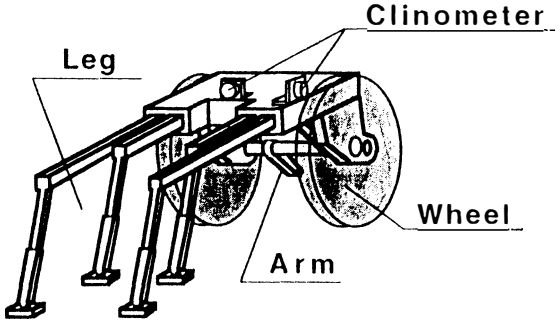


Fig. 1 Structure of SCARAB

2.1 Mechanism of Leg

This robot has two pairs of two legs in the rear of body. Each pair of rear legs has two degrees of freedom in 135㎜ as vertical and 225㎜ as horizontal direction and composed of three parts: leg(h), leg(v) and foot as shown in figure 2. The position of each leg is detected from rotary encoders of DC motor. The push and pull forces of the legs helps the robot to go up or down stairs and prevents it from slipping down. Figure 3 shows each pair of legs has four touch sensors and an angle sensor. Touch sensor detects the leg contacting with the proper place of the tread, and the angle sensor measures distances between the leg and the riser of the step.

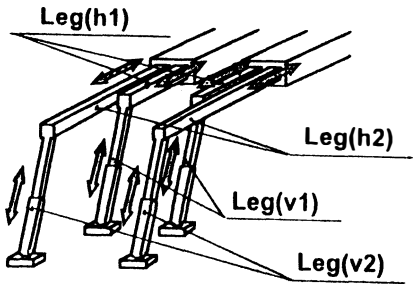


Fig.2 Legs

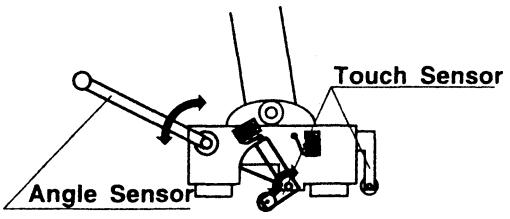


Fig.3 Foot

2.2 Mechanism Arm

Each of the two arms with a free roller at the top of itself is equipped between two wheels in front of body. And each of the arms has a touch sensor to detect the tread of the step. When the robot goes up stairs, these two arms rotate clockwise and hook onto a corner of the steps, then these two arms can help the robot to climb stairs as in figure 3. These arms also support the body of the robot in such a way as not to slip down, while the robot climbs down stairs. The position of the rotational axis and the length of the arms must be determined carefully, because the desired torque of the motor and the height which arms can lift the body are determined by them, though it is difficult to find the most suitable point and the angle for all stairs.

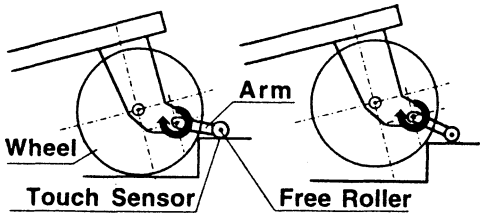


Fig.4 Arm

2.3 Movement of Stair climbing

The robot with this mechanism can climb according to the size of stairs : from 130～200㎜ in the horizontal direction and from 0～100㎜ in the vertical direction These ranges under which the robot can be controlled are decided with the measurements of stairs in about 300 places, and information from the Building Standard Act and six major railway companies in Japan. Stair climbing maneuver may divided into three categories according to robot's control algorithm: low part, middle part and high part. While the robot steps up the stair, a step is first detected with arms and the size of the steps at the low part. Then the robot goes up at middle part of the stair. When the arms can not detect the tread of the stair, the wheels are at the top of the steps. The robot goes down backwards and legs detect a steps instead of arms while goes down stairs.

Figure 5 shows how the legs can be moved in vertical and horizontal direction to step up or down at the middle part of the stair. Most of legged robots steps up or down stairs one by one. Crawler type robots move at a low speed limits because of the friction coefficient between stair and crawler. But the legs with this mechanism step up and down every two steps. Going up and down stairs, one pair of legs stays on a tread of stairs to keep hold on the body. At the same time, the other pair of long legs steps up or down to the next stair. Therefore this mechanism can go up and down faster with stability than the others.

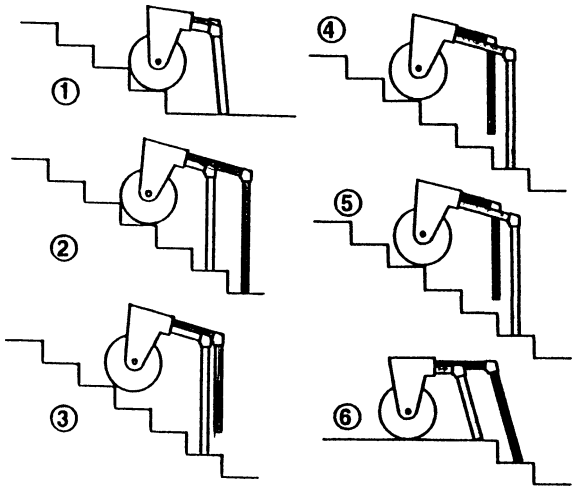


Figure.5 Movement of stair climbing

3 Control

A position control for legs and a torque control for wheels and arms are adopted respectively. The positions of legs are first estimated roughly and then decide a proper place by both external and internal sensors. Measured data of a stair include errors, and the same stair has also about 10㎜ difference of the height and the width. Touch sensors and an angle sensor of each foot work as external sensor to prevent legs slipping down, because of these error and differences. The distance between legs and the riser of the step is controlled by the angle sensor. The touch sensors beneath the foot detect the foot being on the tread in safety. Not only the rotation of wheels and arms but also movements of legs must be synchronized. While the robot goes up and down stairs, the rotational angle of wheels is determined depending on the position of the legs. Then the slip of wheel can be measured by comparing a calculated rotational angle of wheel and a measured angle of that by rotary encoder.

4. Measurement of Stairs

When the robot climb up steps, the height of a stair is measured by the arms as shown in figure 6.

$$h = t - a \cos \alpha$$

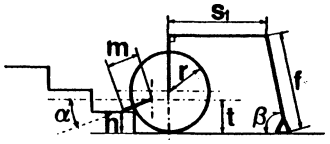


Fig.6 Measurement of stair height

A wide of the stair is calculated with following equations.

$$a^2 = r^2 + \left(f \cos \beta + s_2\right)^2$$

$$w = \left\{a^2 - (h+r)^2\right\}^{0.5} - s_1$$

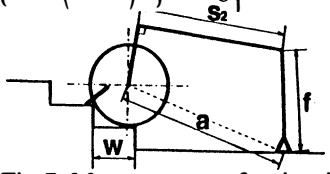


Fig.7 Measurement of stair width

When the robot climbs down stairs, the height and width of the stair are measured by shown in figure 7.

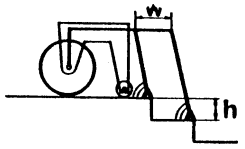


Fig.8 Measurement of stair size

5 Experimental Results

The experimental results are conducted in order to test the performance of the system. It is much difficult for mobil robots to go down stairs with stability than to go up. The both pitch and roll angle of the body and changes of Leg(h) and Leg(v) motors torque are recorded, while the robot goes down three steps at the middle part.

Experimental results with this robot are shown in if figure 10, 11 and 12. The trial robot is controlled by means of computer under these conditions: the stair height is 75㎜, the width is 150㎜ and the number of steps is 10. Figure 10 shows the both roll and pitch angle and these angles do not cause a change abruptly. This means the robot goes down the stair in safe. The relation between the torque of mortars and the horizontal

displacement of the legs are shown in figure 11 and 12. As it is clear from these figures above mentioned, the driving torque change of the legs corresponds the movement of each leg.

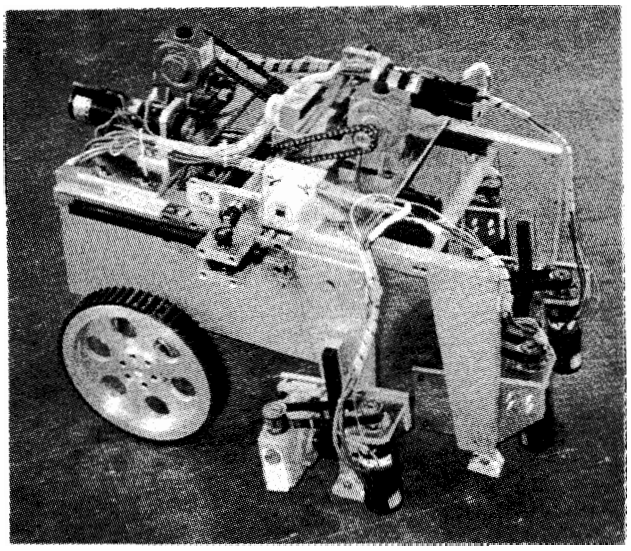


Fig.9 SCARAB

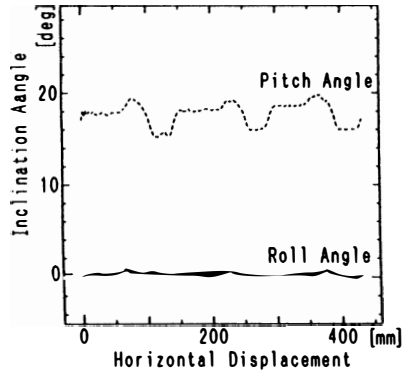


Fig.10 Inclination angle of the robot

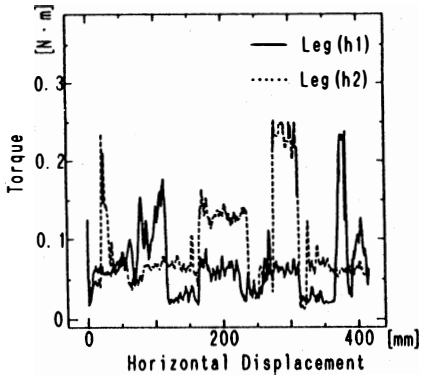
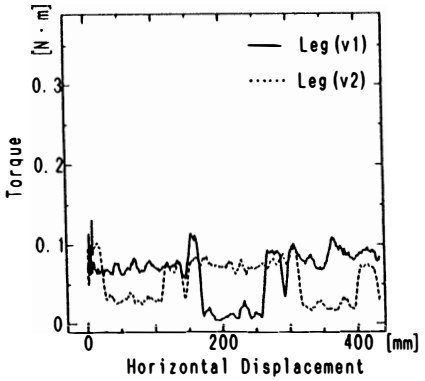


Fig.11 Torque of Leg(h)



Fig,12 Torque of Leg(v)

6 Conclusion

We proposed the prot type of this mechanism and the control algorithm in this paper. The effectiveness of this mechanism is recognized from the experimental results. We will develop the robot with this mechanism for disabled and elderly people.

References

[1] Y. Maeda,"Motion Analysis of Obstacle Striding for the Multinational Robot vehicle",*J. of Robotics Society of Japan*, Vol.7, No.2, pp.13-19, 1989.

[2] J. D. Martens,"Stabilization of a Mobile Robot Climbing Stairs",*Proc. of IEEE Int. Conf. on Robotics Research*, pp.2501-2507,1994.

[3] T. Komeda and H. Funakubo, "Wheelchair and Walking Aid Machine", *J. of Robotics Society of Japan*, Vol.11, No.5, pp.44-48, 1993.

[4] J. E. Braes,"Amber : A Walking Robot for Autonomous Planetary Exploration", *Proc. of Int. Symp. on Artificial Intelligence,Robotics and Automation in Space*, pp.127-130, 1991.

[5] S. Hirose,"A study of design and control of a quadruped walking", *Int. J. of Robotics Research*, Vol.3, No.2, pp.113-133, 1984.

[6] D. R. Pugh,"Technical description of the adaptive suspension vehicle", *Int. J. of Robotics Research*, Vol.9, No.2, pp.24-42, 1990.

[7] S. Hirose,"Design of Stair-Climbing Vehicle (Introduction of New Wheels and Modified RC Mechanism)", *Proc. of The 13th Annual Conf. of RSJ*, pp.153-154, 1995.

[8] M.Yamasina and T. Mabuchi,"Terrain Adaptive Wheel Vehicle: ALSSION",Proc. of The 71st JASME Spring Annual Meeting, pp.468-470,1994

[9] E. Gut,"Behavior Control for Robotic Exploration of Planetary Surfaces", *IEEE Transactions on Robotics and Automation*, Vol.10, No.4, pp.490-503,1994.

"CAM-BRAIN" ATR's BILLION NEURON ARTIFICIAL BRAIN PROJECT A Three Year Progress Report

Hugo de Garis

Brain Builder Group, Evolutionary Systems Department,
ATR Human Information Processing Research Laboratories,
2-2 Hikaridai, Seika-cho, Soraku-gun,
Kansai Science City, Kyoto, 619-02, Japan.
tel. + 81 7749 5 1079, fax. + 81 7749 5 1008
degaris@hip.atr.co.jp

Abstract

This paper reports on progress made in the first 3 years of ATR's "CAM-Brain" Project, which aims to use "evolutionary engineering" techniques to build/grow/evolve a RAM-and-cellular-automata based artificial brain consisting of thousands of interconnected neural network modules inside special hardware such as MIT's Cellular Automata Machine "CAM-8", or NTT's Content Addressable Memory System "CAM-System". The states of a billion (later a trillion) 3D cellular automata cells, and millions of cellular automata rules which govern their state changes, can be stored relatively cheaply in giga(tera)bytes of RAM. After 3 years work, the CA rules are almost ready. MIT's "CAM-8" (essentially a serial device) can update 200 million CA cells a second. It is likely that NTT's "CAM-System" (essentially a massively parallel device) will be able to update a *hundred billion* CA cells a second. Hence all the ingredients will soon be ready to create a revolutionary new technology which will allow thousands of evolved neural network modules to be assembled into artificial brains. This in turn will probably create not only a new research field, but hopefully a whole new industry, namely "brain building". Building artificial brains with a billion neurons is the aim of ATR's 8 year "CAM-Brain" research project, ending in 2001.

Keywords

Artificial Brains, Evolutionary Engineering, Neural Networks, Genetic Algorithms, Cellular Automata, Cellular Automata Machines (CAMs), Nano-Electronics, Darwin Machines.

1. Introduction

ATR's CAM-Brain project resulted from the experience of the author's thesis work, in which he evolved neural net modules (using concatenated bit-string weights) to control the behavior of a simulated quadruped called "LIZZY", which could walk straight, turn left, turn right, peck at food and mate [de Garis 1994]. Each of these behaviors was controlled by the time varying outputs of a single evolved neural network module, and applied to the angles of the leg components of LIZZY. (As far as he is aware, the author was the first person to evolve neural net dynamics [de Garis 1991], (in the form of walking stick-legs "Walker")). Switching between behaviors involved taking the outputs from one neural net module and feeding them into the inputs of the next module. The next step was to evolve neural net detectors, e.g. for frequency, signal strength, signal strength difference, etc. Finally, neural net "production rule" modules were evolved which could map conditional inputs from detectors to output behaviors. Thus an "intelligent" artificial creature was built, which could detect prey, mates and predators, and then approach and eat or mate, or turn away and flee. Virtually every neural net that the author tried to evolve, evolved successfully. *The evolution of these fully connected neural network modules proved to be a very powerful technique.* This success made a deep impression on the author, reinforcing his dream of being able to build much more complex artificial nervous systems, even artificial brains. However, every time the author added a neural net module to the Lizzy simulation, its speed on the screen was slowed (on a Mac II computer). Gradually, the necessity dawned on the author that some kind of evolvable hardware solution [de Garis 1993] would be needed to evolve large numbers of neural net modules and at great speed (i.e. electronic speed) in special machines the author calls "Darwin Machines" [de Garis 1993]. Evolving artificial brains directly in hardware remains the ultimate future goal of the author, but in the meantime (since the field of evolvable hardware (EHW, E-Hard) is today only in its infancy), the author compromises by using cellular automata to grow/evolve neural nets in large numbers in RAM, which is cheap and plentiful. (In a year or so, it will be quite possible to have a gigabyte of RAM in one's work-station). By using cellular automata based

neural nets which grow and evolve in gigabytes of RAM, it should be possible to evolve large numbers (thousands) of neural net modules, and then assemble them (or even evolve their interconnections) to build an artificial brain. The bottleneck is the speed of the processor which updates the CA cells. State of the art in such processors is MIT's "CAM-8" machine, which can update 200,000,000 CA cells a second. Very recently, it has been suggested by the author's ATR colleague Hemmi, that NTT's Content Addressable Memory System (CAM-System) might be able to update CA cells at a rate *thousands* of times faster than the MIT machine, i.e. at a *hundred billion* CA cells per second. NTT's machine is massively parallel. Hemmi and his programmer assistant Yoshikawa are now (December 1995) busily engaged in writing software to convert the author's CA rules (in 2D form) into Boolean expressions suitable for the NTT machine. If they succeed in applying this machine to CAM-Brain, then a new era of brain building can begin, because the ability to evolve thousands of neural net modules would become realistic and very practical (for example, to evolve a neural net module inside a cubic space of a million CA cells, i.e. 100 cells on a side, at a hundred billion cells a second, would take at most about 500 clock cycles, i.e. about five milliseconds. *So the evolution of a population of 100 chromosomes over 100 generations could all be done in about one minute.)* All the essential ingredients for brain building would be available (lots of RAM, the CA rules, and fast CA processors). Even if Hemmi does not succeed, then a new machine can be designed to be thousands of times faster than the CAM-8 machine. The author believes the CAM-Brain breakthrough is either less than a year away, or at most only a few years away (the time necessary to design and build a "Super-CAM" machine, probably with the help of NTT).

The above gives an overview of the CAM-Brain research project. What now follows is a more detailed description of CAM-Brain, showing how one grows and evolves CA based neural net modules in 2D and 3D. We begin with the essential idea. Imagine a 2D CA trail which is 3 cells wide (e.g. Fig. 2). Down the middle of the trail, send growth signals. When a growth signal hits the end of the trail, it makes the trail extend, or turn left, or right, or split etc., depending upon the nature of the signal (e.g. see Fig. 3). It was the author who hand coded the CA rules which make these extensions, turns, splits etc. happen. The CA rules themselves are *not* evolved. It is the *sequence* of these signals (fed continuously over time into an initialized short trail) that is evolved. This sequence of growth signals is the "chromosome" of a genetic algorithm, and it is this sequence that maps to a cellular automata network. When trails collide, they can form "synapses" (e.g. see Fig. 5). Once the CA network has been formed in the initial "growth phase", it is later used in a second "neural signaling phase". Neural signals move along CA-based axons and dendrites, and across synapses etc. The CA network is made to behave like a conventional artificial neural network (see Fig. 5). The outputs of some of the neurons of the complex recurrent networks which result can be used to control complex time dependent behaviors whose fitnesses can be measured. These fitness values can be used to drive the evolution. By growing/evolving thousands of neural net modules and their interconnections in an incremental evolutionary way, it will be possible to build artificial brains. According to the CAM developers at MIT, it is likely that the next generation of CAMs will achieve an increase in performance of the order of thousands, within 5 years. However, to be able to evolve a billion neuron artificial brain by 2001 (ATR's goal), it is likely that a "nano-CAM" machine (i.e. one which uses nano-scale electronic speeds and densities) will need to be developed. To this end, we are collaborating with an NTT researcher who has developed a nanoscale electronics device, who wants to combine huge numbers of them to behave as nano-scale cellular automata machines.

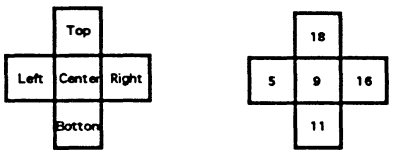
In the summer of 1994, a two dimensional CAM-Brain simulation was completed which required 11,000 hand crafted CA state transition rules. It was successfully applied to the evolution of maximizing the number of synapses, outputting an arbitrary constant neural signal value, outputting a sine wave of a desired arbitrary period and amplitude and to the evolution of a simple artificial retina which could output the vector velocity of a "white line" which "moved" across an array of "detector" neurons. Work on the 3D simulation should be completed by early 1996, and is expected to take about 150,000 hand crafted CA rules. The Brain Builder Group of ATR took possession of one of MIT's CAM8 machines in the fall of 1994. At the time of writing (December 1995) the porting of the 2D rules from a Sparc20 workstation to the CAM8 is nearing completion. If the porting of the rules of the 3D simulation to this machine is not possible, then a "SuperCAM" machine will be designed specifically for CAM-Brain, with the collaboration of the Evolutionary Technologies (ET) group of NTT, with whom our Brain Builder group of ATR's Evolutionary Systems (ES) group, collaborates closely. The complexity of CAM-Brain will make it largely undesignable, so a (directed) evolutionary approach called "evolutionary engineering" is being used. Neural networks based on cellular automata [Codd 1968], can be grown and evolved at electronic speeds inside state of the art cellular automata machines, e.g. MIT's "CAM8" machine, which can update 200 million cells per second [Toffoli & Margolus 1990]. Since RAM is cheap, gigabytes of RAM can be used to store the states of the CA cells used to grow the neural networks. CA based neural net modules are evolved in a two phase process. Three cell wide CA trails are grown by sending a sequence of growth signals (extend, turn left, turn right, fork left, fork right, T fork) down the middle of the trail. When an instruction hits the end of the trail it executes its

function. This sequence of growth instructions is treated as a chromosome in a Genetic Algorithm [Goldberg 1989] and is evolved. Once gigabytes of RAM and electronic evolutionary speeds can be used, genuine brain building, involving millions and later billions of artificial neurons, becomes realistic, and should become concrete within a year or two. The CAM-Brain Project should revolutionize the fields of neural networks and artificial life, and in time help create a new specialty called "Brain Building", with its own conferences and journals.

2. Cellular Automata Based Neural Networks

Building an artificial brain containing billions of artificial neurons is probably too complex a task to be humanly designable. The author felt that brain building would be a suitable task for the application of evolutionary engineering techniques. As mentioned briefly in the introduction, the key ideas are the following. Use evolutionary techniques to evolve neural circuits in some electronic medium, so as to take advantage of electronic speeds. The medium chosen by the author was that of cellular automata (CA) [Codd 1968], using special machines, called "Cellular Automata Machines (CAMs)", which can update hundreds of millions of CA cells a second [Toffoli & Margolus 1990]. CAMs can be used to evolve the CA based neural networks at electronic speeds. The states of the cellular automata cells can be stored in RAM, which is cheap, so one can have gigabytes of RAM to store the states of billions of CA cells. This space is large enough to contain an artificial brain. MIT's Information Mechanics Group (Toffoli and Margolus) believe that within a few years it will be technically possible to update a trillion CA cells in about 0.1 nanoseconds [p221, Toffoli & Margolus 1990]. Thus, if CA state transition rules can be found to make CA cells behave like neural networks, and if such CA based networks prove to be readily evolvable, then a potentially revolutionary new technology becomes possible. The CAM-Brain Project is based on the above ideas and aims to build artificial brains before the completion of the project in 2001. The potential is felt to be so great that it is likely that a new specialty will be formed, called "Brain Building".

For the first 18 months of the CAM-Brain Project, the author simulated a two dimensional version of CAM-Brain on a Sparc 10 workstation. This work was completed in the summer of 1994. The 2D version was used briefly (before work on the 3D version was started) to undertake some evolutionary tests, whose results will be presented in the next section. The 2D version served only as a feasibility and educational device. Since trails are obliged to collide in 2D, the 2D version was not taken very seriously. Work was begun rather quickly on the more interesting 3D version almost immediately after the 2D version was ready. Proper evolutionary tests will be undertaken once the 3D version is ready, which should be by early 1996. To begin to understand how cellular automata [Codd 1968] can be used as the basis for the growth and evolution of neural networks, consider Fig. 1 which shows an example of a 2D CA state transition rule, and Fig. 2 which shows a 2D CA trail, 3 cells wide. All cells in a CA system update the state of their cells synchronously. The new state of a given cell depends upon its present state and the states of its nearest neighbors. Down the middle of the 3 cell wide CA trail, move "signal or growth cells" as shown in Fig. 2 As an example of a state transition rule which makes a signal cell move to the right one square, consider the right hand most signal cell in Fig. 2, which has a state of 5. The cell immediately to its right has a state of 1, which we want to become a 5. Therefore the 2D state transition rule to turn the 1 into a 5 is 1.2.2.2.5-->5. These signal or growth cells are used to generate the CA trails, by causing them to extend, turn left or right, split left or right, and Tsplit. When trails collide, they can form synapses. It is the sequence of these signal cells which determines the configuration of the CA trails, thus forming a CA network.



CTRBL -> Cnext 9.18.16.11.5 -> 4

Fig.1 A 2D CA State Transition Rule

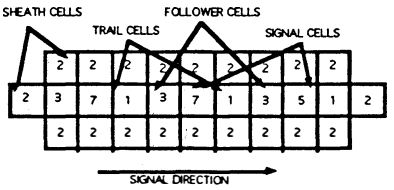


Fig. 2 Signal Cells Move Along a Cellular Automata Trail

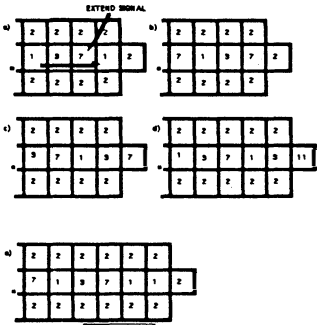


Fig. 3 Extend the Trail

It is these CA trails which later are used as neural network trails of axons and dendrites. Neural signals are sent down the middle of these CA trails. Thus there are two major phases in this process. Firstly, the CA trails are grown, using the sequence of signal cells. Secondly, the resulting CA trail network is used as a neural network, whose fitness at controlling some system can be measured and used to evolve the original growth sequence. To make this more explicit, it is the sequence of growth cells which is evolved. By modifying the sequence, one alters the CA network configuration, and hence the fitness of the configuration when it functions as a neural net in the second phase. From a genetic algorithm (GA) point of view, the format of the GA "chromosome" is the sequence of integers which code for the signaling (growth) instructions. By mutating and crossing over these integers, one obtains new CA networks, and hence new neural networks. By performing this growth at electronic speeds in CAMs, and in parallel, with one CAM per GA chromosome, and attaching a conventional programmable microprocessor to each CAM to measure the user defined fitness of the CA based neural circuit, one has a means to evolve large numbers of neural modules very quickly. Using CAMs to evolve neural circuits, is an example of a type of machine that the author labels a "Darwin Machine", i.e. one which evolves its own structure or architecture. A related idea of the author concerns the concept of "Evolvable Hardware (EHW)" [de Garis 1993] where the software instructions used to configure programmable logic devices (PLDs) are treated as chromosomes in a Genetic Algorithm [Goldberg 1989]. One then rewrites the circuit for each chromosome.

References

[Codd 1968] E.F. Codd, *Cellular Automata*, Academic Press, NY, 1968.

[de Garis 1991] Hugo de Garis, "Genetic Programming", Ch.8 in book *Neural and ntelligent Systems Integration*, ed. Branko Soucek, Wiley, NY, 1991.

[de Garis 1993] Hugo de Garis, "Evolvable Hardware : Genetic Programming of a Darwin Machine", in *Artificial Neural Nets and Genetic Algorithms*, R.F. Albrecht, C.R. Reeves, N.C. Steele (eds.), Springer Verlag, NY, 1993.

[de Garis 1994] Hugo de Garis, "Genetic Programming : Evolutionary Approaches to Multistrategy Learning", Ch.21 in book "Machine Learning : A Multistrategy Approach, Vol.4", R.S. Michalski & G. Tecuci (eds), Morgan Kauffman, 1994.

[Goldberg 1989] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, MA, 1989.

[Toffoli & Margolus 1987, 1990] T. Toffoli & N. Margolus, *Cellular Automata Machines*, MIT Press, Cambridge, MA, 1987; and *Cellular Automata Machines*, in Lattice Gas Methods for Partial Differential Equations, SFISISOC, eds. Doolen et al, Addison-Wesley, 1990.

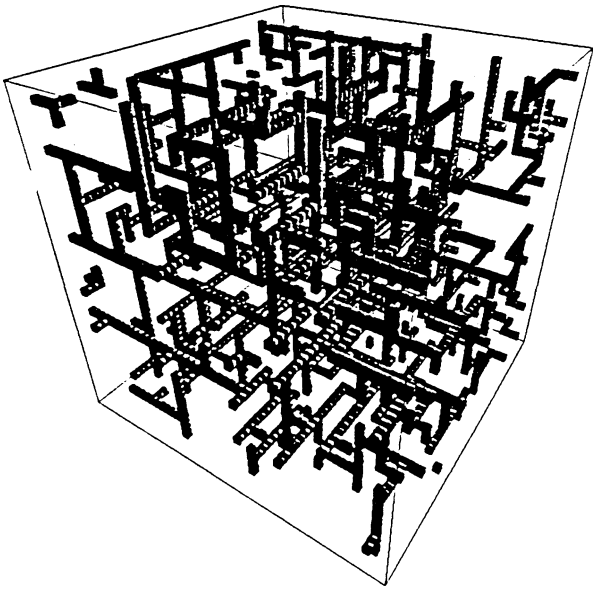


Fig. 4 3D CAM-Brain Non-Synaptic Growth

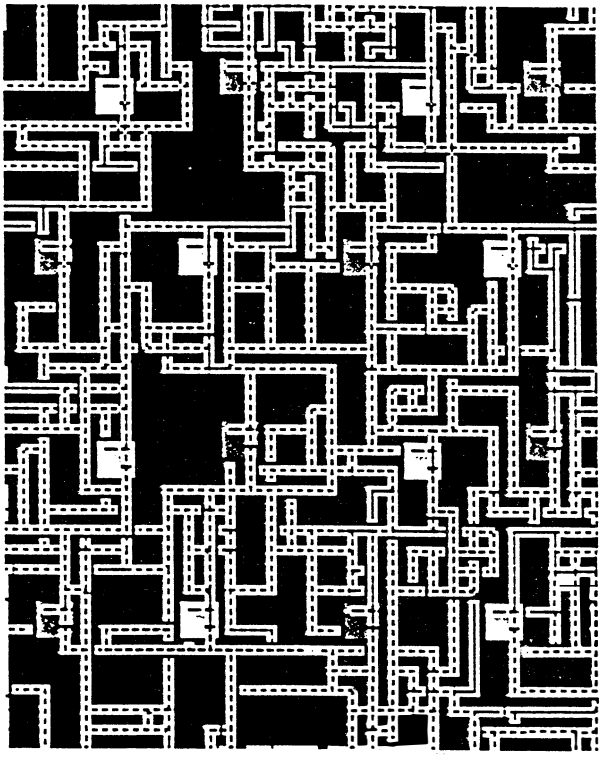


Fig. 5 2D CAM-Brain Neural Signaling

Spontaneous Behavior for Cooperation of Distributed Autonomous Robots

Takanori Shibata [*1 *2], Kazuya Ohkawa [*3], Kazuo Tanie [*2*3]

*1 Artifcial Intelligence Laboratory, MIT
545 Technology Square, NE43-814
Cambridge, MA 02139
shibata@ai.mit.edu

*2 Mechanical Engineering Laboratory, MITI
1-2 Namiki, Tsukuba 305
tanie@mel.go.jp
*3 Tsukuba University, okawa@melcy.mel.go.jp

Abstract

This paper deals with *spontaneous behavior* for cooperation among distributed autonomous robots. The robot determines its behavior depending on its experience and behaviors of other robots. Each robot feels frustration when its determination does not suit to its environment. Then, it changes its behavior to change its situation actively. Human gives the robots an evaluation function of relation of cooperation among robots. The robots acquires a model of behaviors of the other robots through learning. This model is considered as the experience.

1. Introduction

In biological societies including human beings, there are many coordinated behaviors without a supervisor for optimization. Each determines its own behavior while seeing behaviors of others and its situation. This behavior is referred to as spontaneous coordinated behavior in this paper. Coordinated behaviors are important for multiple robots to accomplish tasks effectively. The coordinated behaviors means that the robots work together or share tasks cooperatively. We assume that there is no explicit communication among robots and no supervisor.

There are many research works on coordination among robots [1-5]. In the most research, rules and strategies of behaviors are given by a human operator for effective movement. There are cases that the rules and strategies do not suit to the robots' environments. Therefore, modification of them are necessary in such cases. Strategies must be produced depending on the relations for cooperation among robots and their environments.

This paper proposes an algorithm for action selection as production of strategies for coordinated behaviors. Each robot has some patterns of behaviors correspond to some strategies. The algorithm selects the most appropriate strategy to the situation. Even though human operator gives an evaluation function for relations between robots, the robot learns a model of behaviors of other robots for strategy selection.

When the selected strategy does not fit to its environment, the robot feels frustration and recognizes that. In this case, the robot changes its strategy to change its situation actively. This change may generates a new situation and a new relation between robots. The proposed method is applied to a task for clearing up empty cans in a room by two robots. Simulations are simple but show the effectiveness of the proposed method.

2. Spontaneous Behavior

Fig. 1 shows an algorithm for generation of the spontaneous behavior. A robot has some elements of behaviors. The meaning of behaviors are vary in some research from simple action, such as move forward, to complex action, such as move to a goal avoiding obstacles. In this paper, the behavior means the later and can be seen as a strategy.

The robot is given some strategies a priori, though each strategy should be a combination of the simple behaviors. The robot has behavior rules to perform along with the strategy.

When a main goal or task is given to a robot, it selects a strategy which suits to its situation. When the selected strategy does not suit to its situation, the robot changes its strategy. While changing, the robot accomplish the main goal. The situation includes the environment and

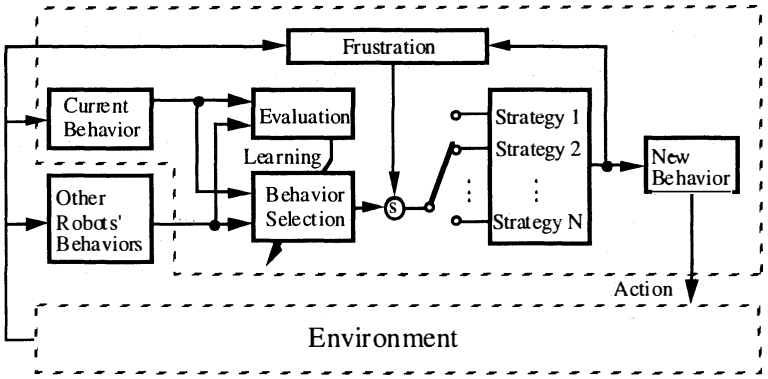


Fig. 1 An Algorithm for Generation of Spontaneous Behavior

motion of the other robots. The robot must have sensory system to know its situation. When there is only one robot in a work space, the robot selects its behavior depending on its environment. On the other hand, when there are multiple robots in a work space and they have the same main goal, the robot should take into account the other robots. The robots should work cooperatively while assisting other robots, avoiding conflicts, sharing tasks and so on. They may have interaction among themselves. As we assume there is no supervisor, they should determine their own behavior efficiently. Also, we assume there is no explicit communication among robots; each robot only indicates the current strategy and other robots can sense this.

As we assume there are multiple robots in the same work space, there are interactions among them. They must share some resources to perform selected strategies. This means they must behave cooperatively. The best strategy for one robot may influence other robot unfavorably. This depends on the aim of the robot and its situation, especially the strategies of the other robots.

We give the robot a learning capability to memorize the relation of cooperation among the robots and to use it as a criterion for the action selection. In order to generalize the criterion, this learning does not concern with the environment, just relation among the robots.

3. A Task and Its Evaluation

A main task is decomposed into some sub-tasks. In order to carry out one sub-task, the robot has one strategy which consists of behavior rules. Multiple robots work on the sub-tasks. Each robot selects a strategy and executes it to accomplish the corresponding sub-task. During the execution, the situation of the robot changes. Depending on the situation, the robot selects the next sub-task. Continuing this sequence, the robots accomplish the main task. In this case, even though one robot can complete the task, multiple robots share the task for efficiency.

We assume that each robot has an evaluation function between its behavior and other robots' one. In this paper, human operator gives the evaluation function for the relation between sub-tasks. While using this function, the robot acquires a relation model of other robots. This model is used to select the new strategy. Followings explain the evaluation method for the task which is for clearing empty cans in a room.

The task is decomposed into two sub-tasks. One is that the robot collects empty cans into the center of the room. The other is that the robot disposes of the collected cans by using a dust-pan. The sub-task of collection of the empty cans are named T1. The other is T2. We assume that there are two robots in the room and there is one dust-pan. Then, there may be conflicts between the robots if both robots select the sub-task T2.

We assume that the T2 is preferable for the robot to T1 and the relation of evaluation value of the sub-tasks is T2 > T1. This assumption determines order of the evaluation values which expresses cooperation and conflict between robots. While considering the order, we assigned the evaluation values as follows:

(strategy (sub-task) of itself, strategy of the other robot) ---> Evaluation value
(T1, T1) ---> p (equal cooperation)
(T1, T2)---> -q (unfavorable cooperation)

(T2, T1)---> q (favorable cooperation)
(T2, T2)---> -p (conflict) (1)
where $q > p > 0$.

Each robot is given this evaluation function. The robot can select its strategy (sub-task) arbitrarily though the evaluation value is determined by the relation between the other and itself. It is desirable to gain higher evaluation value.

4. Learning for Strategy Selection

In order to get high evaluation value, the robot had better select a strategy while predicting the other robot's behavior. For this purpose, Q-learning is applied for modeling relation to the other robot's starategy [6]. This chapter explains the method.

A is a set of strategies for each robot. In this paper, **A** is (T1, T2). **X** is a set of finite discrete state of the robots. In this paper, **X** is the selected sub-tasks by the robots. The strategies of the robots at time t are expressed as a state x_t. Depending on x_t, the robot selects its behavior whose Q value is the maximum. The robot selects a strategy a_t and recognizes the state of the other robot. As an assumption, each robot has a sign which indicates the state of its behavior, so that other robots can recognize it. Q value is modified depending on the evaluation function (1) as learning as follows:

if $c_t = a_t$,

$$Q_{new}(x_t, c_t) \leftarrow Q_{old}(x_t, c_t) + \Delta Q$$

else,

$$Q_{new}(x_t, c_t) \leftarrow Q_{old}(x_t, c_t) - \frac{1}{n-1}\Delta Q \qquad (2)$$

where c_t is the possible strategy to be selected at time t and n is the number of the possible strategies. ΔQ is calculated as follows:

$$\Delta Q = \alpha\left\{\max_{b_t \in A} E(a_t, b_t) + \gamma \max_{a_{t+1} \in A} Q_{old}(x_{t+1}, a_{t+1}) - Q_{old}(x_t, a_t)\right\} \qquad (3)$$

where α is learning rate and γ is discounting factor. The evaluation value $E(a_t, b_t)$ is given by eq. (1).

5. Frustration of Robot and New Action to Its Environment

The previous chapters described the evaluation method for relation between robots concerning with cooperation and conflict, and its usage for learning the model of the other robots. The model is used for the behavior selection. However, this selection does not consider robots' environments. In human beings, he may be frustrated when his decision does not suit to his situation. We give a similar function to the robots to take into account their environment in the strategy selection.

As a model of the frustration, we defined a monotonous increasing function. If the selected strategy is not accomplished for a certain time, the selection is considered that it does not fit to the environment. In this case, the robot changes to other behavior which has the second high Q value. This function means that the robot acts to its environment to change its situation spontaneously.

6. Simulations

Simulations were carried out to show the effectiveness of the proposed method. It was applied to a task for clearing up a room by two robots.

Fig. 2 shows the initial state of robots and their environments. There are two robots at left side. Small cylinders are the empty cans. At upper right side, there is a dust-pan.

Each robot has two sub-tasks. One is to collect the empty cans in the center of the room (T1). The other is to dispose of the collected cans to the lower right of the room by using the dust-pan (T2). The robot does not know the locations of cans but knows that of the dust-pan and the disposing place.

At first, each robot learned the behavior of the other robot while changing its behavior. They obtained Q values in the learning process which is on-line process. Then, they learned to collect the empty cans cooperatively as seen in Fig. 3. For a while, each robot engaged in the collection task, T1, as equal cooperation. In Fig. 4, the right robot (Robot 2) could not find empty cans and it was

frustrated. Then, it changed its strategy to take the dust-pan for disposition. Fig. 5 shows that the robot got the collected cans by the dust-pan and moved them to the appropriate place. After all, the task was carried out.

7. Conclusions

This paper proposed an algorithm for the spontaneous behavior of the autonomous robot. The robot learns relations of cooperation among robots by using Q-leaming to obtain models of other robots' behaviors. Also, the robot has a function of frustration which works as a fitness function to the environment. In the case that the robot is frustrated, the robot changes its behavior to change its situation actively. The algorithm is applied to distributed autonomous robot system. The simulation was simple but the result showed that robots worked cooperatively as coordinated system even though there was no supervisor.

As the future work, the proposed method will be applied to more complicated tasks where many robots works together in the same work space. A generation method of the evaluation function will be also investigated.

Reference

[1] H. Asama, et al., Collision Avoidance among Multiple Mobile Robots Based on Rules and Communication, IEEE IROS'91, pp. 1215-1220 (1991)

[2] S. Yuta, et al., Coordinating Autonomous and Centralized Decision Making to Active Cooperative Behaviors Between Multiple Mobile Robots, IEEE IROS'92, pp. 1566-1574 (1992)

[3] Y. Kuniyoshi, et al., Cooperation be Observation, -The Framework and Basic Task Patterns, IEEE ICRA'94, pp. 767-774, (1994)

[4] D. McFarland, Towards Robot Cooperation, From Animal to Animat 3, Int'l Conf. on SAB, pp. 440-444 (1994)

[5] K. Sekiyama and T. Fukuda, Adaptability Enhancement of Multiple Robots in Dynamical Environment, Proc. of the IEEE ICRA'95, pp. 1656-1661 (1995)

[6] C. Watkins, Leaming from Delayed Rewards, PhD Thesis, Cambridge University (1989)

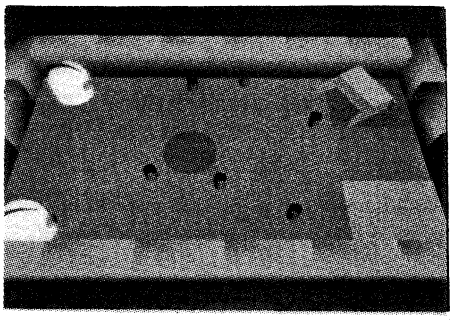


Fig. 2 Initial State for Simulation

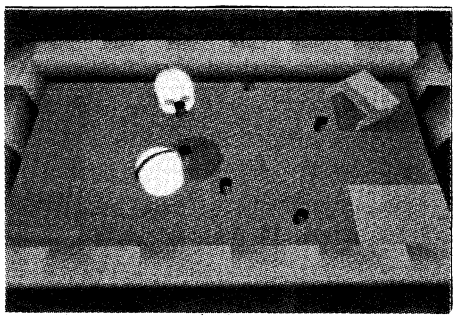


Fig. 3 Cooperative Behavior to Collect Empty Cans by Two Robots

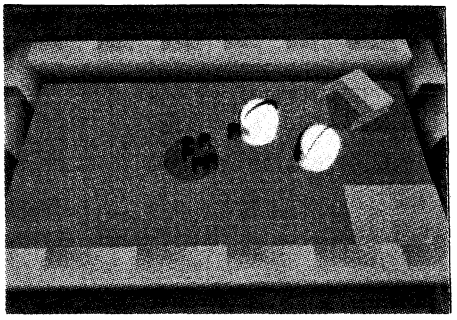


Fig. 4 Change of Behavior because of Frustration (Robot 2)

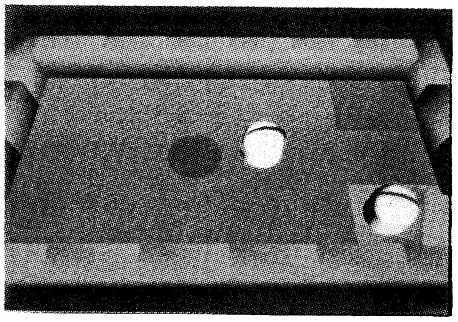


Fig. 5 Accomplishment of Task

Nonlinear Numerical Optimization with Use of a Hybrid Genetic Algorithm Incorporated the Modified Powell method

Masahiro Okamoto*, Taisuke Nonaka, Shuichiro Ochiai, Daisuke Tominaga
Department of Biochemical Engineering & Science, Kyushu Institute of Technology,
Iizuka-city 820, Japan
*E-mail : okahon@bse.kyutech.ac.jp

Abstract

We proposed the hybrid numerical optimization technique which is incorporated the Genetic Algorithm (GA) into the Modified Powell method. Escaped from the trapping in local minimum, this procedure can seek out the global minimum with considerably fast convergence. The effectiveness was shown especially for the minimum search problem of a variable-separable multi-peak(bottom) function having over twenty thousands local minima within the parameter searching region.

1 Introduction

Among numerical optimization techniques, the direct-search method which need not to evaluate the gradient is much suitable for the analysis of dynamics of complex nonlinear control systems. The Powell method[1] is well known to have an ultimate fast convergence among direct-search methods, especially where the cost function is approximated well by a quadratic form of parameters to be estimated. It is proved for the Powell method that at most m iterations, where m is the number of parameters to be estimated, yield the optimal solution to the problems with cost function J of quadratic form, if the directions of m-dimensional vectors are linearly independent at every iteration step. Some problems, however, are not always assured of optimal solutions because the direction vectors are not always linearly independent. To overcome this, Powell[2] has revised the algorithm by introducing new criteria for formation of linearly independent direction vectors; this revised method is called "The Modified Powell method".

Keyword : hybrid numerical optimization, function optimization, global minimum, genetic algorithm, modified Powell method

However even if this method is applied, the minimum points obtained may not be the global minimum; most of the conventional numerical optimization can easily be trapped in local optima or by constraints in a region of the parameter space far from the optimal solution. The Genetic Algorithm (GA)[3] is known as one of the algorithms which can seek out the global minimum. The GA is based on the heuristic assumptions that the "best" solutions will be found in regions of the parameter space containing a relatively high proportion of "good" solutions and that these reginos can be explored by the genetic operators of selection, crossover, and mutation. The GA offers a number of advantages; GA searches from a set of designs (not from a single design), and it is not derivative-based, and it can explore and exploit the parameter space without trapping in local optima[4]. However this method has major disadvantage that the computational cost (computational time) of the large numbers of runs of the design code is considerably large.

In the present study, we shall propose the hybrid numerical optimization technique which is incorporated the GA into the Modified Powell method. It is highly expected that a unique combination of the GA and the Modified Powell offers all the advantages of both optimization techniques while offsetting their disadvantages; the proposed procedure can not only help the algorithm to escape from the trapping in locla minima but also seek out the global minimum with considerably fast convergence.

2 Procedures

The procedures of proposed hybrid numerical optimization can be summarized as follows:

(step 1-1) : Set an initial or a re-initial points of parameter space.

(step 1-2) : Search optimal, near optimal, or better solutions with use of the Modified Powell method.

(step 1-3) : Evaluate the estimated values (solutions) of parameters and calculate the cost function f.

(step 1-4) : If the value of f is large and does not remarkably decrease during searching procedures in the Modified Powell method, switch to the GA and go to step 2-1.

(step 1-5) : If the absolute value, $|f-f_{opt}|$, is smaller than the threshold value ϵ; $|f-f_{opt}| < \epsilon$, where f_{opt} represents the desired value (optimal value) of f, output the results. Otherwise, continue to search (back to step 1-2)

(step 2-1) : Transform the real value, obtained value by the Modified Powell method (= obtained value trapping in local minimum) into 32-bit standard floating point format; the sign is represented by a single bit, the biased exponent and the fraction will have 8 bits and 23bits, respectively;

1	8	23
(sign)		

(step 2-2) : Generate 32-bit m initial genes involving the genetic code obtained in step 2-1

(step 2-3) : Generate crossover gene; randomly select two genes and crossover each other at one point (ont-point crossover).

(step 2-4) : Generate mutation gene; randomly select genes and randomly select the mutation point and mutate at that point.

(step 2-5) : Evaluate the genes and proceed to the next generation (introduce the elitist strategy; where the best evaluated member (gene) of each generation is copied into the succeeding generation).

(step 2-6) : During steps 2-3 to 2-5, if the value better than one in step 2-1 is obtained, switch back to the Modified Powell method and back to step 1-1 by translating the genetic code (32-bit strings) into real number.

3 Function optimization

In order to examine the effectiveness of our proposed procedures, they were applied to the minimum search problem of the following function :

$$f = \sqrt{0.01 \times ((u-25)^2 + (v-40)^2)} + 10 \\ -10 \times \sin(0.5\pi u)\cos(0.75\pi v) \quad (1)$$

This function has 20480 local minimum points in the region where u and v are within [-256, 256]. The graph is plotted in Figure 1 in a region with u and v in [0, 50]. The global minimum value is one when $u = 25$ and $v = 40$.

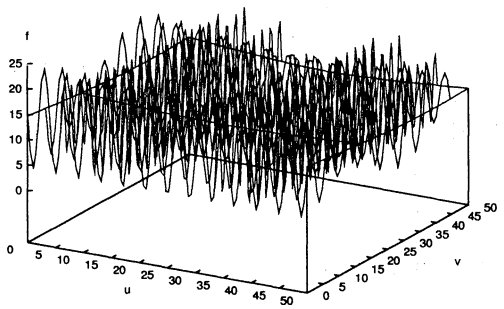


Figure 1: Illustration of the shape of function f(Eq.(1)).

At this optimal point, the value of function f takes 0. In computer simutation, we searched in a region with u and v in [-256, 256]. In the Modified Powell method, the Rosenbrock method (Success and Faults method)[5] was used as one-directioal search, and the judgement of that the obtained value is trapped into local minimum was as follows:

$$|f - f_{opt}| > \epsilon \quad and \\ |f_i - f_{i-1}| < \theta \quad (2)$$

, where f_i and f_{i-1} show the values of cost function f at i-th and (i-1)th iteration, respectively and f_{opt} represents the desired value of f. In the present study,

Table 1: Summary of convergence frequency in the function optimization (Eq.(1)).

Optimization method	Mutation rate	CPU-time (sec)	Convergence frequency (%)
Simple GA	0.05	4.31	50
	0.10	3.21	69
	0.15	3.50	57
Hybrid-method	0.05	3.10	74
	0.10	2.88	79
	0.15	2.85	81

Table 2: Summary of convergence frequency by using the hybrid method with additional procedures.

Optimization method	Additional procedures	CPU-time (sec)	Convergence frequency (%)
Hybrid-method	none	2.85	81
	+(1)	2.57	82
	+(2)	2.74	86
	+(1)+(2)	2.59	89
GA	none	3.50	57
	+(2)	5.76	66

the threshold values of ϵ and θ are fixed at 10^{-3} and 10^{-5} respectively.

In the GA procedures, the total numbers of genes at each generation is fixed at 60 and the crossover rate is 0.5 (crossover occurs among half of the genes at each generation). The rate of bit-mutation in each gene is set to $0.05 \sim 0.15$. All the computations have been performed in UNIX workstation Titan2-300LX/HX (Kubota Computer Inc., Japan, 75 SPECfp92). We terminated the simulation when the total number of generations in the GA is over 500. Table 1 shows the convengence frequency (percentage of getting opimal solutions) in the cases of simple GA method and the hybrid method (our proposed method). These values represent the convergence frequency among 100 runs by simple GA or hybrid method.

The results indicate that the hybrid optimization technique outperforms the simple GA technique. In order to improve the convergence frequency, the following two additional procedures were added to the procedures at section 2 :

(additional procedures)

(1) At step 2-6, the best three values obtained are back to step 1-1, that is, best three candidates obtained in the GA procedures are taken back to the Modified Powell method.

(2) At step 2-4, if the value of cost function f becomes considerably lower than the initial value, change the mutation rate at each bit; the mutation rate at the biased exponent changes to small, while the mutation rate at the fraction changes to large. This procedure will appreciably narrow the searching region.

Table 2 shows the effects of the introduction of the additional procedures above on the convergence frequency. The initial mutation rate was fixed at 0.15. In this simulation, if the value of cost function f becomes to be lower than 1.0, the following function, which calculate the mutation rate at each bit, was applied as the additional procedure (2):

$$Y(i) = M\_rate \times [\frac{2}{\pi}\{tan^{-1}\frac{i-8}{2}\} + 1] \quad (i = 5, 6, ..., 31) \qquad (3)$$

, where $Y(i)$ represents the mutation rate at the i-th bit from most significant bit (MSB), and M_rate shows the initial mutation rate. The other conditions were the same as in Table 1. As shows in Table 2, by combining the additional procedures (1) and (2), the hybrid optimization method can seek out the global minimum at considerably high frequency (89%); in the case of simple GA method, the convergence frequency is 57%.

4 Conclusions

(1) We proposed the hybrid numerical optimization

technique which is incorporated the GA into the Modified Powell method.

(2) The effectiveness was shown especially for the minimum search problem of a variable-separable multi-peak function; without trapping and staying in local minimum, the proposed method can explore and exploit the global minimum (optimal solution) with considerably fast convergence.

(3) This technique will be highly applicable to nonlinear complex numerical optimization in the field of process control and robotics.

References

[1] M.J.D.Powell "An efficient method for finding the minimum of a function of several variables without calculating derivatives", *Computer J.*, 7, pp155-162, 1964.

[2] M.J.D.Powell "On the calculations of orthogonal vectors", *Computer J.*, 11, pp302-304, 1968.

[3] J.H.Holland "Genetic algorithms", *Scientfic American*, pp44-50, 1992.

[4] L.Davis "Handbook of genetic algorithms", Van Nostrand Reinhold, New York, 1991.

[5] H.H.Rosenbrock "An automatic method for finding the greatest or least value of a function", *Computer J.*, pp175-184, 1960

A Language for Describing Motions of Mechatronic Systems: A Nonlinear Position-dependent Circuit Theory

Suguru Arimoto
Faculty of Engineering, The University of Tokyo
Bunkyo-ku, Tokyo, 113 Japan

Abstract

Dynamics and control of nonlinear mechanical and advanced mechatronics systems can be investigated more efficiently by using corresponding nonlinear position-dependent circuits that describe Lagrange's equations of motions and interactions with objects or/and task environments. Such expressions of Lagrange's equations via nonlinear circuits are indebted to lumped-parameter discretization of mechanical systems as a set of rigid bodies through equations of motion due to Newton's second law. This observation is quite analogous to validity of electric circuits that can be derived as lumped-parameter versions of Maxwell's equations of electromagnetic waves. Couplings of mechanical circuits with electric circuits through actuator dynamics are also discussed. In such electromechanical circuits the passivity should be a generalization of impedance concept in order to cope with nonlinear properties of circuits and play a crucial role in motion control problems. In particular, it is shown that the passivity as an input-output property gives rise to a necessary and sufficient characterization of H_∞-tuning for disturbance attenuation, which can give another systems-theoretic interpretation of the energy conservation law.

Key words: Nonlinear Circuit, Mechanical Systems, Robot Motion, Passivity.

1 Introduction

It is widely known as explained in many textbooks that there is an analogy between dynamics of a simple linear mechanical system of mass-damper-spring (MCK) and that of a simple lumped-parameter linear electric circuit of inductor-resistor-capacitor (LRC) (see Fig.1). Nevertheless, dynamics of actual mechanical and mechatronics systems such as robot arms and mechanical hands have never been analyzed intensively based on their corresponding circuit-theoretic expressions. Indeed, there is a dearth of papers that have discussed directly any possibility of expressing Lagrange's equation of motion for nonlinear mechanical systems via lumped-parameter circuits like electric circuits. Conversely, there is a well-known approach called "Euler-Lagrange formalism" for analyzing electric circuits (see Meisel [1]) and "Lagrange-Formalism" for discrete electromechanical systems (see Maisser and Steigenberger [2]). More classically, there was a proposal for description of mechanical systems via a network called "bond graph" (see Paynter [3], Karnopp and Rosenberg [4]). However, all these were proposed before opening a new "robot era" and therefore have not yet been extended to cope with dynamics of nonlinear mechanical and mechatronics systems. Indeed, dynamics of actual mechatronics systems such as anthropoid robot arms and hands are nonlinear and have strong couplings between joint variables. As a matter of course in electric circuit theory, generalization of electric circuits towards nonlinear circuits has been attempted in 1960's and 1970's (see Brayton and Moser [5] and Chua and Green [6]), but only a limited class of circuits was investigated by introducing individually a few of single nonlinear elements like a nonlinear resistor, an inductor with nonlinear flux, and a nonlinear capacitor.

This paper shows first that a mechanical system governed by Lagrange's equation of motion implies the existence of a nonlinear position-dependent circuit and its dynamics corresponds to that of the circuit. In such a nonlinear circuit the velocity vector can be regarded as the current vector that passes along a bundle of branches and flows into and out of such blocks (see Fig.2) with n input and n output terminals as kinetic inductor, resistor, virtual resistor, and gravity capacitor. These linear or nonlinear blocks are considered to be operators that operate on a domain of velocity vectors and generate voltage drops. A nonlinear kinetic inductor can be regarded as a block with n inputs and n outputs being dependent on the position and retains a kinetic energy. A gravity capacitor is also dependent on the position and retains a poten-

tial energy. Both Kirchhoff's current and voltage laws are valid to this class of nonlinear position-dependent circuits, correspondingly to d'Alambert's principle. In addition, a new concept called "displacement capacitor" is introduced in order to cope with set-point control problems of nonlinear mechanical systems. The displacement capacitor can be interpreted as an extension of a mechanical spring. Thus, it is shown that a similar method to the direct-current analysis for electric circuits can be applied for the stability analysis of set-point control without compensation for the gravity force. However, the proposed approach is not a simple extension of this analogy because mechanical circuits as a language for describing dynamics should have more flexible syntax and general context to cope with various nonlinearities in electromechanical systems and three underlying physical variables "velocity", "torque", and M'position" differently from electric circuits that have only two kinds of underlying physical variables "current" (corresponding to "velocity") and "voltage" (corresponding to 'Mtorque"). A physical variable "electric charge" appears as an integral of "current" in time in electric circuits like 'Mposition" as an integral of "velocity" in time, but a vector of position variables signifies a real configuration of the system in space. Hence, a kinetic inductor is a genuinely new concept of inductor that is dependent on not only "velocity" but also 'Mposition", that is, the inductance operator that induces "voltage drop" depends on the present configuration of the systems.

Secondly we show an example of nonlinear circuit that expresses coupling of the motion of a robot arm driven by a set of voltage-controlled DC motors as joint actuators with a set of linear armature electric circuits. Thirdly we present a necessary and sufficient characterization of H_∞ −tuning of PD control with offline gravity compensation in the sense of disturbance attenuation. This in turn means that the H_∞ −tuning for mechatronics systems can be interpreted as another characterization of "energy conservation law".

2 Coupling of Mechanical Circuits with Electric Circuits

The Lagrange equation of motion of a robot arm (see Fig.3) is described by

$$\frac{d}{dt}\{\frac{\partial}{\partial \dot{q}}L(q,\dot{q})\} - \frac{\partial}{\partial q}L(q,\dot{q}) = F \qquad (1)$$

where $L(q,\dot{q}) = K(q,\dot{q}) - P(q), K(q,\dot{q})$ denotes the kinetic energy, $P(q)$ the potential energy, and F the

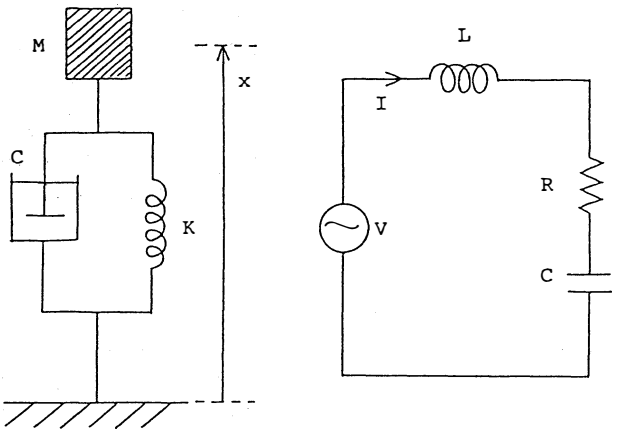


Fig.1 Analogy between a Mass-Damper-Spring mechanical system and a linear Induetor-Resistor-Capacitor electric circuit.

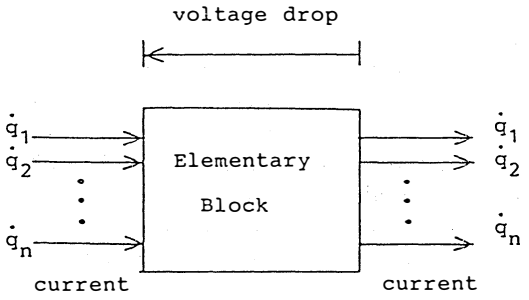


Fig.2 Elementary block of nonlinear position-dependent circuits.

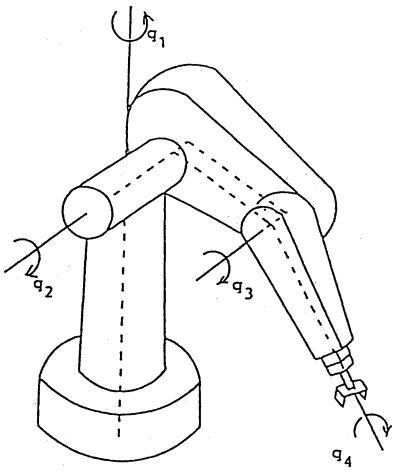


Fig.3 Anthropoid manipulator with all revolute-type joints

generalized force vector whose i-th component stands for a torque generated by an actuator at joint i. In detail, (1) can be written as

$$H(q)\ddot{q} + \dot{H}(q)\dot{q} - \frac{\partial}{\partial q}K(q,\dot{q}) + \frac{\partial}{\partial q}P(q) = F. \quad (2)$$

Further, note that (1) can be expressed as

$$\{H(q)\frac{d}{dt} + \frac{1}{2}\dot{H}(q)\}\dot{q} + S(q,\dot{q})\dot{q} + g(q) = F \quad (3)$$

where

$$S(q,\dot{q})\dot{q} = \frac{1}{2}\{\dot{H}(q)\dot{q} - \frac{\partial}{\partial q}\dot{q}^T H(q)\dot{q}\}. \quad (4)$$

It is well known that $S(q,\dot{q})$ becomes skew-symmetric (see Arimoto and Miyazaki [7]).

Now, the Lagrange equation of motion for the robot of Fig.3 is expressed by the nonlinear circuit depicted in Fig.4. Conversely, applying Kirchhoff's voltage law for the circuit of Fig.4 leads to equation (3).

Next consider coupling of a mechanical motion of a robot arm with a set of electric circuits of joint actuators. The dynamic equation of the armature circuit corresponding to the i-th joint actuator can be expressed as

$$L_i\dot{I}_{ai} + R_i I_{ai} + K_{bi}\dot{\theta}_i = v_i, \qquad i = 1,\cdots,n \quad (5)$$

where $L_i, R_i, I_{ai}.K_{bi}$, and v_i denote the armature inductance, armature resistance, armature current, constant of motor back electro-motive force, and armature voltage of the i-th motor armature circuit. We denote the transmission gear ratio $n_i : 1$ from the motor shaft angle to the joint angle by a number $r_i(= 1/n_i)$ and assume that the arm link is sufficiently stiff to be considered as rigid and hence $r_i\dot{\theta}_i = \dot{q}_i$ where q_i denotes the i-th component of position coordinates q of the arm. On the other hand, the torque generated from the i-th joint motor is equivalent to $K_{\tau i}I_{ai}$, where $K_{\tau i}$ denotes the torque constant. According to the d'Alambert principle, the torque $K_{\tau i}I_{ai}$ is equal to the load due to the inertia moment and damping from the motor side itself and the i-th contribution $r_i F_i$ of the motion of the arm, where F_i denotes the i-th component of F in equation (3). This can be expressed as

$$\begin{aligned} NK_\tau I_a &= F + H_0\ddot{q} + B_0\dot{q} \\ &= \{H_0 + H(q)\}\ddot{q} + \{\frac{1}{2}\dot{H}(q) + B_0 + S(q,\dot{q})\}\dot{q} \\ &\quad + g(q) \end{aligned} \quad (6)$$

where $N = diag(r_1^{-1},\cdots,r_n^{-1}), K_\tau = diag(K_{\tau 1},\cdots,K_{\tau n}), H_0 = diag(J_{m1}r_1^{-2},\cdots,J_{mn}r_n^{-2})$,

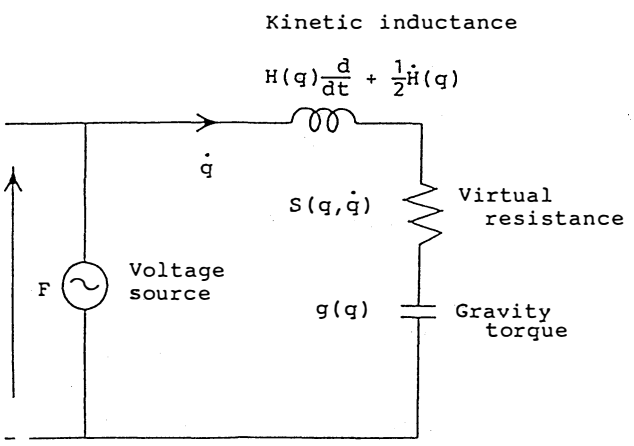


Fig.4 A nonlinear circuit that expresses Lagrange's equation of motion.

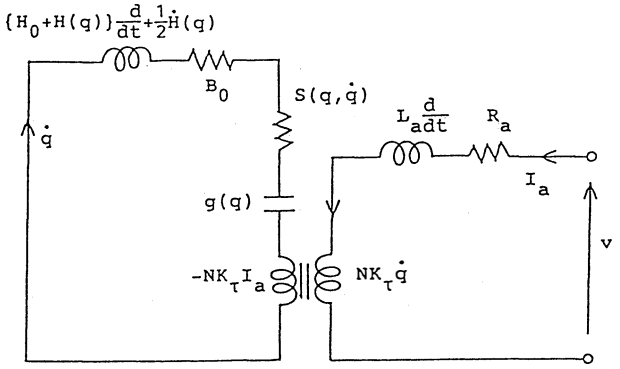


Fig.5 The transformer connects the motion of a nonlinear mechanical system with a set of linear electrical circuits. The input-output pair $\{v, I_a\}$ satisfies the passivity.

$B_0 = diag(b_{m1}r_1^{-2},\cdots,b_{mn}r_n^{-2})$ provided that all joints are driven by such voltage-controlled DC servomotors,

Now, the nonlinear position-dependent circuit governed by equation (6) can be coupled with the linear electrical circuit governed by (5). Since $r_i\dot{\theta}_i = \dot{q}_i$ for $i = 1,\cdots,n$ and in principle the motor torque constant $K_{\tau i}$ is equivalent to the constant K_{bi} of back electro-motive force, the circuit of the total dynamics can be depicted in Fig.5, where coupling between the mechanical nonlinear circuit and electrical armature circuit is realized by a "transformer" at the velocity level. The voltage drop (torque) induced by the transformer at the side of mechanical motion is equivalent to $-NK_\tau I_a$ (proportional to the current vector I_a) and the actual voltage drop induced by the transformer at the armature circuit is considered to be

equivalent to the voltage $NK_r\dot{q}$ induced by the back electro-motive force. According to equation (5) and $K_{ri} = K_{bi}$, the set of armature circuits can be described in the following form:

$$L_a\dot{I}_a + R_aI_a + NK_r\dot{q} = v \tag{7}$$

where $I_a = (I_{a1}, \cdots, I_{an})^T$, $v = (v_1, \cdots, v_n)^T$, $L_a = diag(L_1, \cdots, L_n)$, and $R_0 = diag(R_1, \cdots, R_n)$. At this stage it is natural to expect that the input and output pair $\{v, I_a\}$ for the total circuit of Fig.5 may satisfy the passivity. In fact, it is possible to show the validity of this property by taking inner products between equation (7) and I_a and between equation (6) and $\dot{q}$ and summing up these two resulting quantities, which results in the form

$$\begin{aligned} I_a^Tv &= \frac{d}{dt}[\frac{1}{2}I_a^TL_aI_a + \frac{1}{2}\dot{q}^T\{H_0 + H(q)\}\dot{q} \\ &+ P(q)] + I_a^TR_aI_a + \dot{q}^TB_0\dot{q}. \end{aligned} \tag{8}$$

Since $min_qP(q) = 0$, it follows from (8) that

$$\int_0^t I_0^T(\tau)v(\tau)d\tau \geq -E(0) = -\gamma_0^2 \tag{9}$$

where $E(0)$ denotes a nonnegative quantity at $t = 0$ of the total energy defined by the content inside the square bracket [ ] in (8). A setpoint control problem for the total system can be solved by taking advantage of this passivity property though the total system seems more complicated with $3n$ state variables $(q, \dot{q}, I_a)$. This has been treated in Arimoto [8].

3 Disturbance Attenuation in a Sense of H_∞ Control

In this section we treat a special case when the inductance matrix L_a is small enough in comparison with R_a so that the first term of (7) can be ignored in the armature circuit dynamics. Then, (7) can be rewritten in the form

$$NK_rI_a = F - \bar{B}_0\dot{q} \tag{10}$$

where $F = NK_rR_a^{-1}v$ and $\bar{B}_0 = NK_rR_0^{-1}NK_r$. Substituting (10) into (6) yields

$$\{H_0 + H(q)\}\ddot{q} + \{\frac{1}{2}\dot{H}(q) + B_1 + S(q,\dot{q})\}\dot{q} + g(q) = F \tag{11}$$

where $B_1 = B_0 + \bar{B}_0$. We implicitly assume the exact knowledge on N, K_r and R_a and thereby the input F can be composed of the input voltage v through $F = NK_rR_a^{-1}v$. Further, we must bear in mind that there are other nonlinear frictional forces that can not be described and hence can not be included in (11). Therefore, if we consider a PD feedback with gravity compensation defined by

$$F = g(q_d) - A\Delta q - B\dot{q} + u \tag{12}$$

where q_d is a given desired target position and $\Delta q = q - q_d$ and the closed-loop system (obtained by substituting (12) into (11))

$$\begin{aligned} &\{H_0 + H(q)\}\ddot{q} + \{\frac{1}{2}\dot{H}(q) + S(q,\dot{q}) + \bar{B}\}\dot{q} \\ &+g(q) - g(q_d) + A\Delta q = u = h(\Delta q, \dot{q}) + \Delta u \end{aligned} \tag{13}$$

where $\dot{B} = B_1 + B$, then extra input u must be a sum of unknown but structured frictional forces $h(\Delta q, \dot{q})$ and another genuine disturbance Δu as described in (13). It should be also remarked that a position feedback of time delay $A\Delta q(t - \Delta t)$ induces a positive velocity feedback as shown by

$$-A\Delta q(t - \Delta t) \cong -A\Delta q(t) + \Delta tA\Delta\dot{q}(t). \tag{14}$$

Since such a positive velocity feedback must be taken into account in the structured disturbance $h(\Delta q, \dot{q})$, we suppose a class of structured disturbances satisfying

$$\gamma^2\int_0^t h(\Delta q(\tau), \dot{q}(\tau))^T\dot{q}(\tau)d\tau \leq \int_0^t \| \dot{q}(\tau) \|^2 d\tau \tag{15}$$

for a fixed γ^2 and any $t > 0$.

Definition 1 If the input-output pair $\{\Delta u, \dot{q}\}$ concerning the system of (13) satisfies the passivity for any structured disturbance $h(\Delta q, \dot{q})$ satisfying (15), then it is said that it satisfies the passivity with margin γ^2.

At the same time, we are concerned with the performance of disturbance attenuation in general in the sense L^2 norm of the output $\dot{q}$ versus that of the total disturbance u in (13) over any time interval $[0, t]$.

Definition 2 Assume that $q(0) = q_d$ and $\dot{q}(0) = 0$. If for a fixed $\gamma^2 > 0$ and any time interval $[0, t]$ it holds that

$$b_0\int_0^t \| \dot{q}(\tau) \|^2 d\tau \leq \gamma^2\int_0^t \| u(\tau) \|^2 d\tau, \tag{16}$$

then we say that the system described by (13) establishes the H_∞-tuning with level γ^{-2}. Here in (16) b_0 is defined as b_0 =minimum eigenvalue of $\bar{B}$, $(b_0 = \gamma_m(\bar{B}))$.

The most important result concerning these two concepts can be stated in the following:

Theorem 1 A necessary and sufficient condition for the system of (13) to satisfy the passivity with margin γ^{-2} and to establish the H_∞-tuning with level γ^{-2} simultaneously is that

$$\gamma^{-2} \leq b_0 \quad (= \gamma_m(\bar{B})), \tag{17}$$

provided that $A > 0$ is chosen to satisfy the following two inequalities with a given small $a > 0$ simultaneously:

$$\frac{1}{2}\Delta q^T A \Delta q + P(q) - P(q_d) - \Delta q^T g(q_d) \geq a \parallel \Delta q \parallel^2,$$
$$\Delta q^T \{g(q) - g(q_d)\} + \Delta q^T A \Delta q \geq a \parallel \Delta q \parallel^2 . \tag{18}$$

We omit the proof (see Arimoto[9]).

Theorem 1 presents a physical interpretation that if the energy dissipation rate $\dot{q}^T \bar{B} \dot{q}$ has a margin beyond $\gamma^{-2} \parallel \dot{q} \parallel^2$ for any $\dot{q}$ then the H_∞-tuning is established. This is valid for the nonlinear dynamics of (13) in a global sense. Another noteworthy feature of Theorem 1 is that the sufficient condition for establishment of the H_∞-tuning is expressed in a very simple form, which need neither solve any of Riccati's matrix equations (see Doyle et al. [10]) nor use solutions to a Hamilton-Jacobi-Issacs equation (van der Shaft [11][12], Isidori and Astolfi [13], and Astolfi and Lanari [14]). Readily from the proof of the necessity part of Theorem 1 the following result follows (the proof will be given in Arimoto[9]):

Theorem 2 As for the circuit depicted in Fig.6, the input-output pair $\{\Delta u, \dot{q}\}$ satisfies the passivity if and only if $\gamma^{-2} \leq b_0 (= \gamma_m(\bar{B}))$, provided that $A > 0$ is large enough satisfying (18) and (19) simultaneously with a specified $a > 0$.

4 Conclusion

In this paper we have introduced a framework of nonlinear position dependent circuits as another language for describing nonlinear dynamics of mechanical and advanced mechatronic systems. Instead of Fourier and Laplace transforms that are used effectively in analyzing linear lumped-parameter electric circuits, the concept of passivity becomes fundamental and can be used in effect as a basic tool of characterizing input-output properties of such nonlinear position-dependent circuits.

Reference

[1] J. Meisel, *Principle of Electromechanical Energy Conversion*, McGram-Hill, 1966.

[2] P. Maisser and J. Steinberger, "Lagrange-Formalismus fur diskrete electromechanishe System," *Z. Angem. Math. Mech.*, Vol.59, pp.717-730, 1979.

[3] H. M. Paynter, *Analysis and Design of Engineering Systems*, The MIT Press, 1960.

[4] D. Karnopp and R.C. Rosenberg, *Analysis and Simulation of Multiport Systems*, The MIT Press, 1968.

[5] R.K. Brayton and J. K. Moser, "A theory of nonlinear networks Prt I and Part II", *Quart. Appl. Math.*, Vol22, pp.1-33 and pp.81-104, 1964.

[6] L.O. Chua and D.N. Green, "A qualitative analysis of the behavior of dynamic nonlinear networks : Stability of autonomous networks", *IEEE Trans. of on Circuits and Systems*, Vol.23, pp.255-279, 1976.

[7] S. Arimoto and F. Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capabilities", in *Robotics Research: First Int. Symp..* (M.Brady and P.P. Paul eds.), MIT Press, pp.783-799, 1984.

[8] S. Arimoto, "Nonlinear position-dependent circuits for mechanical motion control", *Proc. of IFAC Workshop Motion Control*, Munich, Germany, Oct.9-11, 1995, pp.1-20.

[9] S. Arimoto, "Nonlinear position-dependent circuits: A language for describing motions of nonlinear mechanical systems", to be presented at the *IUTAM Symp. on Advanced Mechanical Systems*, April 21-26, 1996. Eindhovenm Netherlands.

[10] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. on Automatic Control*, Vol.34, pp.831-847, 1989.

[11] A.J. van der Schaft, "On a state-space approach to nonlinear H_∞ control, *Systems and Control Letters*, Vol.16, pp.1-8, 1991.

[12] A.J. van der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control." *IEEE Trans. on Automatic Control*, Vol.37, pp.770-784, 1992.

[13] A. Ishidori and A. Astolifi, "Disturbance attenuation and H_∞ control via measurement feedback in nonlinear system, *IEEE Trans. on Automatic Control*, Vol.37, pp.1283-1293, 1992.

[14] A. Astolifi and L. Lanari, "H_∞ control of rifid robot", *Proc. of IFAC Symp. on Robot Control*, Sept. 19-21, 1994, Capri, Italy, pp.199-204.

Reinforcement Learning of Dynamic Behavior by Using Recurrent Neural Networks

Ahmet ONAT Hajime KITA and Yoshikazu NISHIKAWA
Department of Electrical Engineering
Graduate School of Kyoto University
Yoshida-Honmachi, Sakyo,
Kyoto 606-01 Japan

Abstract

Reinforcement learning is a learning scheme for finding an optimal policy to control a system, based on a scalar signal such as a reward or a punishment. If the observation of the system by the controller is sufficiently rich to represent the internal state of the system, the controller can achieve the optimal policy only by learning reactive behavior. However, if the state of the controlled system cannot be assessed completely using current sensory system outputs, the controller must learn a dynamic behavior pattern to achieve the optimal policy.

We propose a dynamic controller scheme, which utilizes memory to uncover the hidden states by using information about the past output, and make control decisions using the memory. This scheme integrates Q-learning proposed by Watkins, and recurrent neural networks of several types. It performs favorably in simulations which involve a task with hidden states.

1 Introduction

Reinforcement learning is an attractive method for autonomous learning agents because a teacher that shows the correct policy is not necessary. It uses only the reinforcement signal, a scalar performance indicator. It works as follows: The controller observes the output of the system, and decides on an action. The system changes its state according to the taken action, and yields a new output, and also a scalar reinforcement signal to the controller. The aim is to find an *optimal policy*, i.e., a mapping of the output of the system to the action, which maximizes a value function, usually the *cumulative discounted sum* of the reinforcement signals. There have been several learning techniques proposed, to realize reinforcement learning, e.g., TD-learning by Sutton[8], and Q-learning by Watkins. [9].

Studies on reinforcement learning usually focus on cases where the state of the controlled system is fully available to the controller as the output of the system, and try to obtain an optimal mapping of system outputs to controller actions. However, in realistic situations, it is usually impossible to access the entire state variables of the system. Some of the states may be hidden from the controller, because of sensory limitations. In other words, multiple system states may be perceived as the same by the controller. This is called *perceptual aliasing*[10]. While perceptual aliasing may be advantageous when the aliased states are irrelevant to the task at hand, states requiring different actions might also be aliased. This impairs the performance of the controller, and is clearly disadvantageous.

Various methods have been proposed for learning in presence of perceptual aliasing. Whitehead et. al.[10] have proposed a reconfigurable sensory system to utilize the available sensors more efficiently. Their "lion" algorithm discovers and avoids passing through states with perceptual aliasing. Chrisman[2] and McCallum [3] propose methods which split the modeled state space of the controlled system, based on the gathered statistics. Our approach to cope with perceptual aliasing, is to incorporate memory into reinforcement learning by using recurrent neural networks. It is trained by Q-learning [9] modified for neural networks using the error backpropagation technique.

2 Q-Learning for Recurrent Neural Networks

In this paper, we propose a dynamic controller which reconstructs the missing state information based on the past outputs of the controlled system. The controlled system is assumed to be finite-state, discrete-time, and it is characterized by the transition proba-

bilities;

$$\text{Prob}\,[X_{t+1} = x_j | X_t = x_i\ ,\ A_t = a_k],\ t = 1, 2, \ldots \quad (1)$$

where, $x_i,\ x_j \in \mathcal{X}$ are the states; X_t, denotes the states of the system at time t; $A_t = a_k \in \mathcal{A}$ is the action performed at time t. The system returns the reinforcement signal $r_{t+1}(X_t, A_t)$ to the controller for action A_t from state X_t. The observed output of the system $Y = CX$ is calculated from the state vector X. We assume that at some points in time, X_t can not be determined by considering only Y_t, but the system is observable, i.e., the state X_t can be estimated from the past output $\{Y_{t-1}, Y_{t-2}, ...\}$. The aim of the controller is to maximize the *cumulative discounted reinforcement* (or, the *utility*) given by;

$$V_t = \sum_{k=0}^{\infty} \gamma^k r_{(t+k)} \quad (2)$$

where, $0 < \gamma \leq 1$ is a constant called the *discount factor.*

The proposed architecture of the controller is shown in Fig. 1. It integrates a recurrent network of the Elman type[4] and Q-learning. The main idea is to use Q-learning to tune the connection weights of the recurrent neural network for a dynamic behavior pattern.

Q-learning is a reinforcement learning scheme which estimates the optimal utility ($Q(x_i, a_k)$) to be obtained by taking action a_k at state x_i. Lin[5] has modified it for usage with feedforward neural networks and error backpropagation technique. In this modification, the network has one output unit for each action a_k, which represent the predicted utility Q of doing action a_k for the current input. The actions are selected randomly, with a bias toward stronger outputs. This helps avoid local optimal policies. Initially, actions are chosen with little respect to their Q values, because these do not yet describe the controlled system. This allows for *experimentation* as the controller assesses the outcome of each action in all contexts. As learning proceeds, and the internal model of the controlled system is improved, actions with high Q values are chosen most of the time.

The Elman neural network is similar to multilayer feedforward networks, except that there is feedback from the hidden layer to the input, through a one step time delay. The layer of time delay units is called the context register.

We use an Elman network to learn Q values. It has input units corresponding 1) to the output of the system, 2) the last action, and 3) to the context register. Each output unit corresponds to an action, and represents its Q value. Its output is then fed to the stochastic action selector (SAS), which selects one action according to the Boltzmann distribution[1];

$$\text{Prob}(a_k) = \frac{exp(\frac{Q(X_t, a_k)}{T_t})}{\sum_{l=0}^{n} exp(\frac{Q(X_t, a_k)}{T_t})} \quad (3)$$

where, T_t is a parameter called the temperature which controls the randomness of the action selection routine.

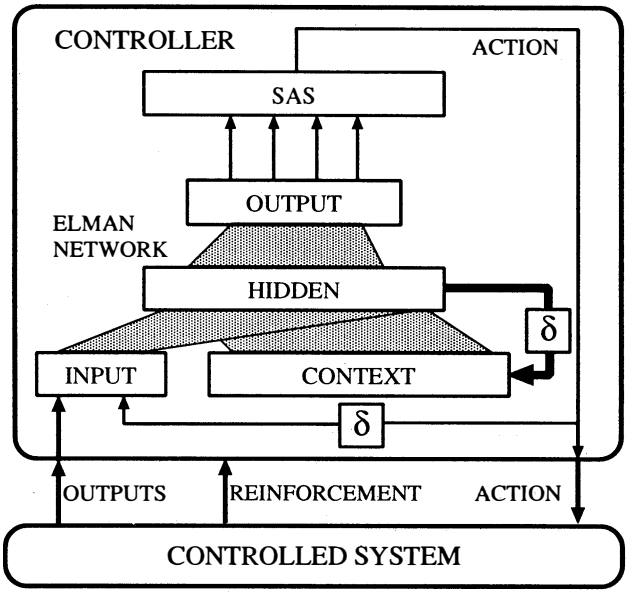


Figure 1: The proposed Q-learning recurrent network model. δ represents a one step delay.

The training algorithm is as follows:

1. Get the current output of the system Y_t. Concatenate Y_t, the past action A_{t-1}, and the outputs of the context units as vector Z_t . Feed Z_t into the network, and obtain $Q(Z_t, a_k)$ values for each action $a_k \in \mathcal{A}$.

2. Using the stochastic action selector, choose an action $A_t = a_l$ and apply it to the system.

3. Get the new outputs Y_{t+1}, and the reinforcement r_{t+1} from the system.

4. Obtain the network output; $Q(Z_{t+1}, a_k)$ for each action $a_k \in \mathcal{A}$.

5. Calculate the error term $\Delta Q = \{r_{t+1} + \gamma\ \max_{a_k} Q(Z_{t+1}, a_k)\} - Q(Z_t, A_t)$.

6. Construct the error vector $\bar{e}(t)$ by using ΔQ for the output unit corresponding to the taken action, and 0 for the rest.

7. Update the weights using the error backpropagation technique, and go to 1.

The above technique can be used with different types of recurrent neural networks. We have also used the *information loss minimization method*, proposed by Noda[6] as an improvement over the Elman network. This method improves network performance by retaining relevant information longer. It uses the X-model architecture shown in Fig. 2, which consists of an Elman network with two additional output unit groups called the context reconstruction layer, and the input reconstruction layer. These groups are trained to minimize the error function

$$E = \langle (y_O - d_O)^2 + (y_{RI} - x_I)^2 + (y_{RC} - x_C)^2 \rangle \quad (4)$$

where $y_O, d_O, x_I, x_C, y_{RI}$ and y_{RC} are activation pattern vectors of the output, desired output, input, context, reconst-input, and reconst-context layers, respectively. Since the output of the network is a transformation of the output of the hidden layer, the two additional groups make sure that enough information about the input and the previous state(context layer) is retained in the context layer. This results in related information being retained for a long time.

In our model,the sigmoid function is used as the activation function of the neurons in the hidden layer, and a linear function for the output units..

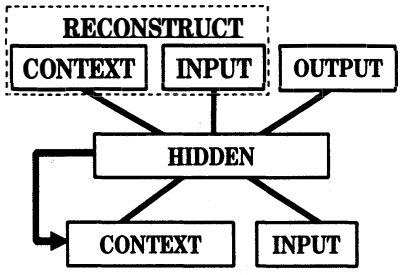


Figure 2: X-model network architecture

3 Simulation

3.1 The Learning Task

The environment consists of a building with a number of rooms connected by doors in a chain as shown in Fig. 3. Each room is accessible only from the previous one, and is connected to the next one. The door to the next room is always closed when the controller(an agent, shown as the man in Fig. 3) enters into the room. Also, the doors have locks, and certain doors are locked. The controller receives two types of inputs; the state of the door (open or closed), and the number of the room the controller is in. In some rooms, the controller cannot detect the room number. This causes perceptual aliasing. The task is to go through the rooms, and exit the building as quickly as possible.

The controller has three actions; "*open*", "*toggle lock*", and "*go ahead*". Reinforcement signals are given when the controller gets into the next room, or exits the building (reward +1), or if it bumps into a closed door (punishment −1). No reinforcement is given otherwise. The state of the lock is not observable. Each trial starts from a room selected randomly out of those having a detectable room number. To achieve the optimal performance, the controller must identify the rooms and remember the states of the locks.

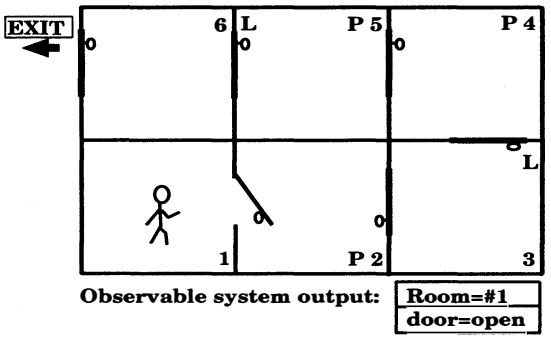


Figure 3: A graphical sketch of the simulation environment. "L" represents a locked door, "P" indicates rooms which cannot be distinguished from one another.

3.2 Simulation Results

We have carried out computer simulations using the environment of Fig. 3. The parameters in the simulation are; $\gamma = 0.8$, $\eta = 0.1$(the learning rate for the network), and 17 hidden units. Figure 4 is an example of the variation of the redundant moves taken by the controller. The number of *redundant moves* taken to reach the goal state, averaged over 100 trials, is plotted against the number of moves.

The performance is improved as learning proceeds. As training proceeds, the controller settles on a

non dynamic policy, which uses an extra action to probe the controlled system state for a while. Later it finds the optimal policy for this task. The behavior pattern finally acquired by the controller is almost optimal.

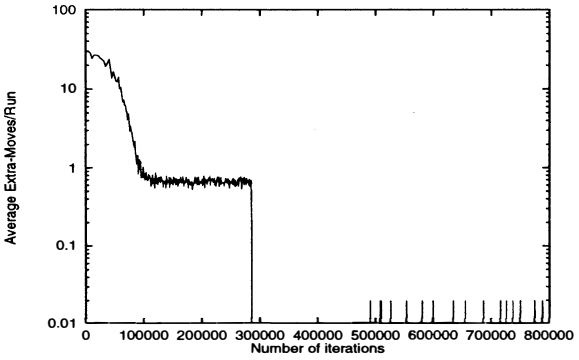


Figure 4: Variation of the redundant moves; Elman network.

We have repeated the simulations using the X-model network instead of the Elman network. Parameters used are the same as above. The result is shown in Fig. 5. Convergence takes place in a shorter time. Also, it can be seen that convergence is smoother, as compared to the simple Elman network.

We have also examined learning with sparse reinforcement signals, as a more difficult learning task; the controller receives a reinforcement only after it gets to the goal or if it bumps into doors. In this test, the convergence has not been achieved correctly. The policy discovered by the controller is suboptimal, i.e., it includes one extra move to check the state of the lock.

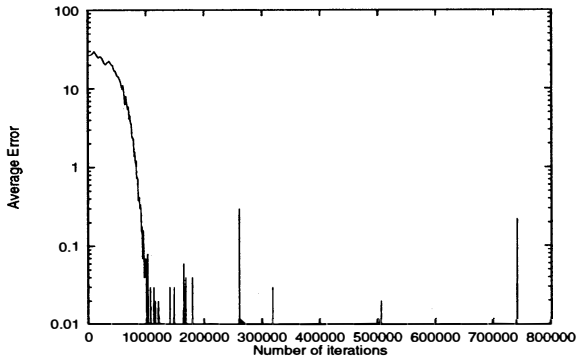


Figure 5: Variation of the redundant moves; the X-model (information loss minimization method).

4 Conclusion

In this study, we construct a *dynamic controller* that acquires an optimal dynamic control policy under the reinforcement learning framework. For the controller and its learning method, we use recurrent neural networks and Q-learning. Computer simulations show that the controller succeeds in learning the optimal policy for a task that causes perceptual aliasing.

In the simulation, information loss minimization technique is shown to improve learning speed compared to the simple Elman network. However, we have also found that, learning gets difficult in a task with sparse reinforcement signals. Further study is needed to cope with this difficul[illegible].

References

[1] A.G.Barto, S.J.Bradtke, S.P.Singh: "Learning to Act using Real-Time Dynamic Programming", *Technical Report*, Dept. of Computer Science Univ. of Massachusetts, 1993.

[2] L. Chrisman, "Reinforcement Learning with Perceptual Aliasing: The Perceptual Distinctions Approach", *Proc. AAAI'92*, pp. 183-188, 1992

[3] A. McCallum, "Instance Based Utile Distinctions for Reinforcement Learning", *Proc. of 12th International Machine Learning Conference*, 1995

[4] J.L. Elman, "Finding Structure in Time", *Cognitive Science*, Vol. 14, pp. 179-211, 1990

[5] L.Lin: "Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching", *Machine Learning*, Vol. 8, pp. 293-321, 1992

[6] I Noda "Neural Networks that Learn Symbolic and Structured Representation of Information", *Researches of the Electrotechnical Laboratory*, No 968, Agency of Industrial Science and Technology, Japan, 1994

[7] A. Onat, H.Kita, and Y. Nishikawa, "Reinforcement Learning of Dynamic Behavior by Using Recurrent Neural Networks", Proc. World Conference on Neural Networks, 1995

[8] R.S.Sutton, "Learning to Predict by the Methods of Temporal Differences", *Machine Learning*, Vol. 3, 1988

[9] C. Watkins and P. Dayan, "Q-learning", *Machine Learning*, Vol. 8, pp. 279-292, 1992

[10] S.D. Whitehead, and D.H. Ballard, "Learning to Perceive and Act by Trial and Error", *Machine Learning*, Vol. 7, pp. 45-83, 1991

Building Ears for Robot: Sound Localization and Separation

Jie Huang †, Noboru Ohnishi † ‡ and Noboru Sugie ‡‡
†Bio-Mimetic Control Research Center, The Institute of Physical and Chemical Research (RIKEN)
3-8-31, Rokuban, Atsuta-ku, Nagoya, 456 Japan, E-mail: huang@nagoya.riken.go.jp
http://www.bmc.riken.go.jp/BMC/SENS/Huang/huang.html
‡Department of Information Engineering, Faculty of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01 Japan, E-mail: ohnishi@nuie.nagoya-u.ac.jp
‡‡Faculty of Science and Technology, Meijo University
1-501, Shiogamaguchi, Tenpaku-ku, Nagoya, 468 Japan, E-mail: sugie@nuie.nagoya-u.ac.jp

Abstract

This paper describes our research on robot audition, and introduces sound localization and separation methods which mimic some effects in the human auditory system such as the precedence effect and the cocktail party effect. We implemented a DSP-based real system and conducted experiments in guiding a camera, approaching a minimum path, localizing and separating multiple sound sources in a real reverberant environment. We demonstrated the system's ability and potential applications, as well as the need of improvement and further studies.

1 Introduction

Audition is one of the most important senses used by humans and animals to recognize their environment. Sound localization and separation ability is particularly important. Biological researches have revealed that the evolution of mammalian hearing is related to the need to localize sound, and the evolution of the localization acuity appears to be related to vision [3]. Thus sound localization acuity enables a mammal to direct its field of best vision to a sound source. This ability is important for humans, as well. A robot, designed to move around our living space and communicate with human beings, also must be equipped with an auditory system capable of sound localization and separation.

Compared to the vision system, audition has several unique properties. It is all-directional. When a sound source emits energe the sound fills the air, and a sound receiver (microphone) receives the sound energy from all directions. Some specialized cameras can also receive an image from all directions, but still have to scan the total area to locate a specific object [8]. Audition mixes the signals into a one-dimensional time series, making it easier to locate any urgent or emergency situation. Audition requires no illumination, enabling a robot to work in darkness or low light. Audition also is not effected by obstacles, so a robot can perceive audition information from sources behind obstacles. One example of this is localization of a sound source outside of a room or around a corner. The robot first localizes the sound source in the area of the door/corner, and then travels to that point and listens again, finally locating the sound source.

Few researches have been done untill very recently in the field of robot audition. Brooks stressed the importance of a sound processing system for a "thinking" robot [1]. The authors of this paper have done base researches on sound localization and separation [5] [6] [7], and found that a sound source can be localized in a reverberant environment by mimicking the precedence effect of the human auditory system. Concurrent multiple sound sources can also be localized correctly in that environment by using a histogram method. We also found that the perceptual sound segment grouping method, which mimicking the cocktail party effect, has potential for sound separation in a reverberant environment.

In this paper, we briefly introduce the methods we

have developed and discribe a robot system equipped with audition, as well as vision and sonar system; we also discribe some related experiments to demonstrate potential applications of the auditory system.

2 Modeling and Mimicking the Human Auditory System

Modeling and mimicking a biological system is an useful way to implement a flexible and environment adaptable system. The signal processing method used by the auditory systems and brains of animals or humans is usually heuristic in the result of evolution. A model or a mimicking method should not be constructed using complex mathematical methods which would not normally exist in a neural system, therefore the methods we propose are simple, practical, and easily achieved.

2.1 Modeling the Precedence Effect

In a reverberant environment, sound mixed with echoes is perceived by our ears (microphones). The spatial information in continuous portions of a sound is corrupted by echoes. The "precedence effect" suggests that the human auditory system can detect the beginning of a sound and mask the subsequent portion of that sound [9], reflecting the adaptation of the human auditory system to reverberant environments.

A model of the precedence effect has been proposed by Zurek [11] . According to the model, the sound portion after an onset will be inhibited for a period of about 10ms. There is, however, no quantitive criterion for onset detection. A model which can detect an 'echo-free' onset has been proposed by the authors [7]. This model is based on the amplitude pattern of a typical impulse response in a normal room. We assume that reflections of an impulse sound will delay more than a certain time, which depends on the distance from the sound source to walls, comparing to the direct sound. It is because a sound source is usually distanced from walls; a nearby wall in almost cases does not change the direction of the first reflection sound very much. We also assume the strength of reflections decreases exponentially over time.

The amplitude pattern of the reflections of an impulse sound is described as:

$$i_{ech}(t) = \begin{cases} 0 & ,0 \le t < t_{fe} \\ \alpha_{fe} e^{-\frac{t-t_{fe}}{\tau}} & ,t \ge t_{fe} \end{cases} . \quad (1)$$

where α_{fe} and t_{fe} denote the amplitude and delay time of the first echo, and τ is time constant. If the sound (including echoes) is given as $s(t)$, the maximum possible echo at time t_0 is estimated as:

$$e_{ech}(t_0) = Max\{s_p(t) i_{ech}(t)\}, \quad 0 < t < \infty \quad (2)$$

where $s_p(t) = s(t_0 - t)$. If the ratio $s(t_0)/e_{ech}(t_0)$ is small enough, signal $s(t_0)$ will be masked by echoes of the preceding signal $s_p(t)$. Thus, the masking time of the precedence effect can be modeled as:

$$\frac{s(t)}{e_{ech}(t)} < r_{thr} \quad (3)$$

where r_{thr} is the threshold ratio of the precedence effect. A masking pattern for echo-free onset detection is then obtained by $\frac{s(t)}{e_{ech}(t)} \ge r_{thr}$.

Note that $s(t)$ itself includes echoes and that $e_{ech}(t)$ is therefore the estimation of echoes and their higher order reflections. In onsets, however, there is no echo and the ratio of $s(t)$ and $e_{ech}(t)$ will be the exact ratio of sound to echo.

2.2 Mimicking the Cocktail Party Effect

A human's ability to separate a specific sound from a noisy environment is known as the "cocktail party effect" [2]. It is also known as a binaural effect.

To mimic the cocktail party effect, sound is represented by segments of enhanced continuous local peaks in the time-frequency space. The onset of a segment is its starting point and the offset is its endpoint. Candidates of arrival temporal disparity (ATD) are calculated at echo-free onsets by a zero-crossing method ($\Delta tc_n = \Delta t + n\lambda/C$, where Δtc_n are ATD candidates. Δt is the real ATD, n is an integer, λ is wavelength, and C is sound velocity.) The number of candidates N is calculated from the wavelength of each band and the distance between the microphones D, i.e. $N = D/\lambda$. Each candidate contributes to the ATD histogram by a value of one Nth, and is also weighted by its amplitude level and sound to echo ratio. The ATD histograms of different microphone pairs are then integrated into an azimuth histogram by a histogram mapping method to resolve the anterior-posterior ambiguity.

The azimuth histogram is divided into blocks, each having a peak and a dominant region. Because the directions of segments belonging to each block are close, as the psychological proximity principle states, all of them are integrated into a single group. Sound segments, which have no sufficient echo-free onset, are

regrouped by the harmonic histograms which are calculated from grouped segments. Sound segments of each group are then synthesized together by the waveform overlapping method to form a sound from each source.

3 System and Experiments

3.1 System Implementation

The sound processing portion of the robot is implemented with three microphones as shown in Figure 1. Each microphone is set at a vertex of an equilateral triangle with a side length of 13.5cm, about as far apart as a human's ears. Since the microphones of current system are not equipped with artificial pinna, amplitude disparities are not sufficient as localization cue. A second version of the robot is now being developed and will have four microphones, each of which will be attached with an artificial pinna to enhance amplitude disparities.

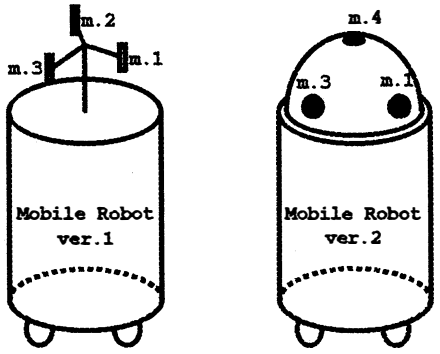


Figure 1: A robot equipped with ears (microphones).

The system of sound processing is shown in Figure 2. The DSP is used for real time processing and the remote computers for non-real time processing. The robot is also equipped with sonar system for obstacle avoidance, and camera system for visual processing.

3.2 Some Experiments and Results

3.2.1 Guiding a Camera to a Sounding Object

Real time processing is required to guide a camera to a sounding object. Figure 3 gives a fast algorithm for echo-free onset detection. The sound signal is first filtered and the envelope $s(t)$ is calculated. The symbol '$*$' means operation of multiplying by a value, t_{fe} and

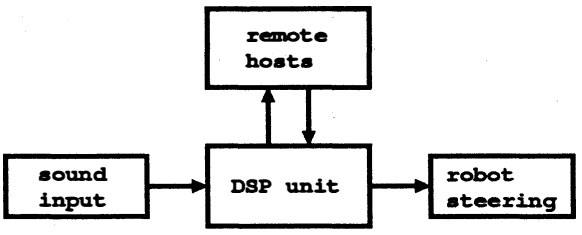


Figure 2: Sound Processing Units.

t_s (the sampling time) mean operation of time delay, and $d = i_{ech}(t_{n+1})/i_{ech}(t_n) = e^{-t_s/\tau}$.

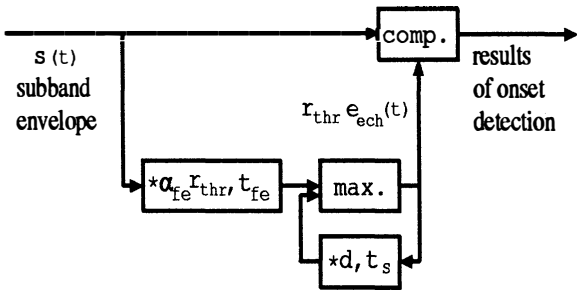


Figure 3: A real time DSP implementation for echo-free onset detection.

We used a single subband (1kHz) system and performed this test in a normal room. It was difficult to detect speech correctly by only a single subband, however, sounds such as hands clapping, knocking on the door, or an transient voice could be detected well. The localization was accurate within a few degrees and robust against background noise and non-transient sound. This system is very simple and can be used to alert humans to unexpected events or intrusions.

3.2.2 Minimum Path Approaching

We tested the previous system to localize a source which was invisible to the robot. The robot was positioned outside of a room, with the sound source (a speaker emitting transient sound) positioned inside (Figure 4).

In this situation, the direction of the sound onset depends on its minimum path in theoretically. In fact, the system did tend to localize sound at the turning point of the minimum path, the edge of the door. However, due to the narrow corridor, there were many echoes with directions differing slightly from the minimum path sound. The localization accuracy in the corridor was poorer than in the tests conducted in-

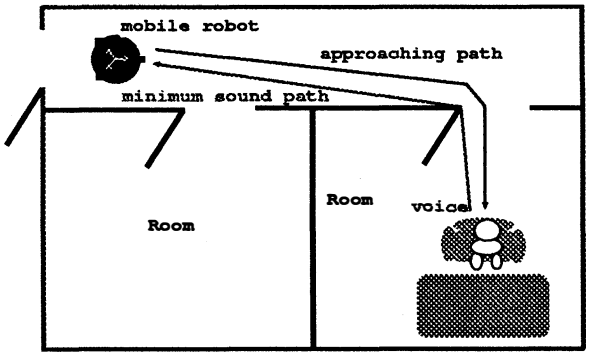


Figure 4: Robot navigation by sound localization.

side a room. After some approach and correction, the robot was able to find the turning point and then finally located the sound source inside the room [10].

3.2.3 Localizing and Separating Multiple Sound Sources

Localication experiments with multiple sound source were conducted in a normal room and in an anechoic chamber. The sound sources were set in two different positions. Sound source 1 was a recording of a radio weather forecast presented by a male announcer (speaker 1), and sound source 2 was a recording of a radio talk show with one male and one female host (speaker 2). Each sound lasted about 20 seconds. The azimuth between sound source 1 and sound source 2 was about 38 degrees. The distance from the sound sources to the center of the microphone set was about 2.9m. For simplifing, sound sources are assumed to be located in front side of the microphone set and two microphones were used. The processing of this test was non-real time and off-line, in contrast to the two provious tests.

Figure 5 shows the results as ATD histograms of pronunciations of "anoseikatsuo" (male) and "tyusin" (male). The left one was in an anechoic chamber, the right one in a normal room. Both figures show two large peaks, each corresponding to the direction of a sound source. While the localization errors in the anechoic chamber were within a few degrees, the errors in the normal room decayed into about double.

Figure 6 shows the waveforms before and after separation, with the left one being in an anechoic chamber and right one in a normal room. The top plots in the figures show the mixed sound before separation, and the middle and bottom plots are of sound 1 and sound 2 after separation.

The figures show that the sound sources are clearly separated in the anechoic chamber. The results obtained in the normal room are similar to the results obtained in the anechoic chamber, despite the existence of strong reverberant sound. However, we can find some distortions in the separated waveforms. The distortions were partially caused by the reverberant sound, which colors tembre and does not effect the perception very much. The distortions were also caused by the misgrouping of sound segments, which should to be removed. Due to the misgrouping of sound segments, the separation quality remained low. We need to develop a method for decreasing the misgrouping of sound segments to improve the quality.

4 Summary

We have developed a robot auditory system and demonstrated its ability to localize and separate sound in a reverberant environment. Our experiments involved guiding a canera to a sounding object, minimum path approaching, and multiple sound source localization and separation. We also have indicated that the quality of sound separation in a reverberant environment remains low and requires improvement.

The potential applications are not limited to the above-mentioned situations. Further studies, including sound understanding or recognition are also required for building an auditory system in a robot. We believe that the development of robot audition will contribute to the human-robot communication and the environment recognition capability of a robot.

References

[1] Brooks R. A. and Stein L. A. (1993), *Building Brains for Bodies*, MIT A.I. Memo No.1439.

[2] Cherry E. C. (1953), *Some experiments on the recognition of speech with one and with two ears*, J. Acoust. Soc. Am., vol.25, pp.975-979.

[3] Heffner R. S. and Heffner H. E. (1992), *Evolution of Sound Localization in Mammals*, in Webster and Popper (Eds.), The Evolutionary Biology of Hearing, New York: Springer-Verlag, pp.691-715.

[4] Huang J., Ohnishi N. and Sugie N. (1988), *Research on a Sound Source Localization System*

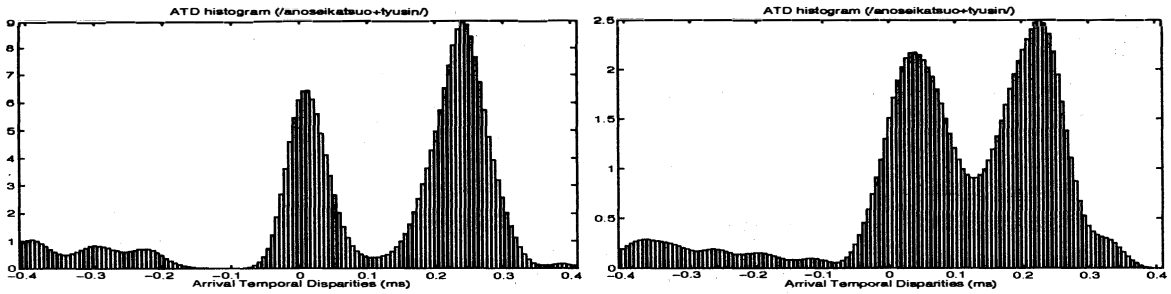


Figure 5: ATD histograms (left in an anechoic chamber and right in a normal room). The two large peaks are contributed by sound source 1 sound source 2 respectively.

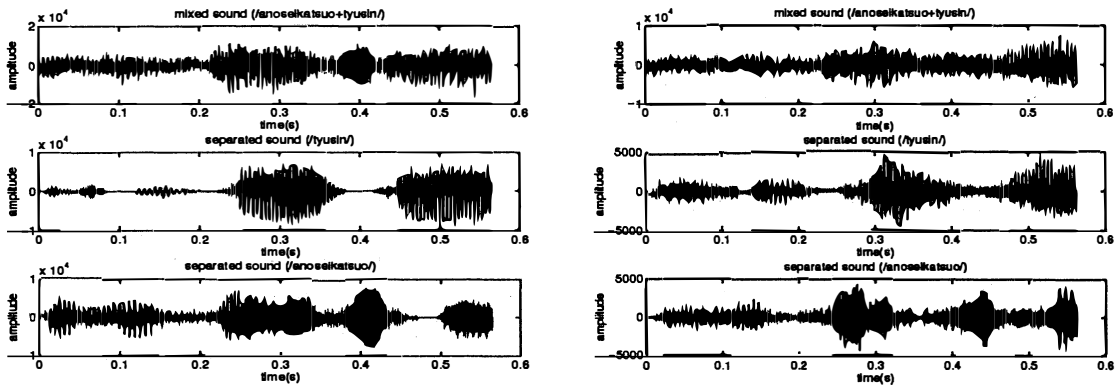


Figure 6: Waveforms before (first plot) and after (second and third plots) the separation (left one in an anechoic chamber and right one in a normal room).

Suggested by Living Orgamisms — Localization of a Single Sound Source in an Echoic Environment —(Japanese), IEICE Trans. vol.J71-A, no.10, pp.1780-1789.

[5] Huang J., Ohnishi N. and Sugie N. (1991), *A Multiple Sound Source Localization System Using Temporal Disparity Histograms*(Japanese), J. Robotics Soc. Japan, vol.9, no.4, pp.409-414.

[6] Huang J., Ohnishi N. and Sugie N. (1995), *A Biomimetic System for Localization and Separation of Multiple Sound Sources*, IEEE Trans. Instrumentation and Measurement, vol.44, no.3, pp.733-738.

[7] Huang J., Ohnishi N. and Sugie N. (1995), *Sound Separation Based on Perceptual Grouping of Sound Segments*, Submitted for NOLTA'95.

[8] Nishizawa Y., Yagi Y., Yachida M. (1993), *Generation of Environmental Map and Estimation of Free Space for a Mobile Robot Using Omnidirectional Image Sensor COPIS* J. Robotics Soc. Japan, vol.11, no.6, pp.868-874.

[9] Wallach H., Newman E. B. and Rosenzweig M. R. (1949), *The Precedence Effect in Sound Localization*, J. Psychol. Am., vol.62-3, pp.315-336.

[10] Wang F., Ohnishi N. and Sugie N. (1994), *Autonomous Mobile Robot Localizing and Approaching Sound Sources*, Proc. 12th Annual Conf. Robotics Soc. Japan, pp.1095-1096.

[11] Zurek P. M. (1993), *The Precedence Effect*, in Yost and Gourevitch (Eds.), Directional Hearing, New York: Springer-Verlag, pp.85-105.

Evaluation of Robust Control by Universal Learning Network

Masanao OHBAYASHI*,Kotaro HIRASAWA*, and Junichi MURATA*

* Dep. of Electrical Eng., Kyushu University, Higashiku, Fukuoka 812, JAPAN
Tel: +81-92-641-1101(ext.5330); Fax: +81-92-631-2790; E-mail:oba@scel.ee.kyushu-u.ac.jp

Abstract: Characteristics of control system design using Universal Learning Network(U.L.N.) are that a system to be controlled and a controller are both constructed by U.L.N. and that the controller is best tuned through learning. U.L.N. has the same generalization ability as N.N.. So the controller constructed by U.L.N. is able to control the system in a favorable way under the condition different from the condition of the control system at learning stage. But stability can not be realized sufficiently.

In this paper, we propose a robust control method using U.L.N. and second order derivatives of U.L.N.. The proposed method can realize better performance and robustness than the commonly used Neural Network. Robust control considered here is defined as follows. Even though initial values of node outputs change from those at learning,the control system is able to reduce its influence to other node outputs and can control the system in a preferable way as in the case of no variation. In order to realize such robust control, a new term concerning the variation is added to a usual criterion function. And parameter variables are adjusted so as to minimize the above mentioned criterion function using the second order derivatives of criterion function with respect to the parameters. Finally it is shown that the controller constructed by the proposed method works in an effective way through a simulation study of a nonlinear crane system.

Keywords Robust control , Neural networks, Second order derivative

1. Introduction

Universal Learning Network(U.L.N.) and a computing method for its higher order derivatives have been proposed,[1]which can be used as a fundametal tool in modelling and control of large-scale complicated systems such as economic, social and living systems as well as industrial plants.

Each nodes in U.L.N. is allowed to have any nonlinear functions. In case of designing a control system using U.L.N., the system to be controlled and the controller are both constructed by U.L.N., and the controller is best tuned through learning to minimize a criterion function which is assumed to be function of the target value of system node output, actual value of system node output and output of the controller. U.L.N. has the same generalization ability as Neural Network(N.N.). So the controller constructed by U.L.N. is able to control the system in a favorable way under the conditions different from those of the control system at learning stage. But stability can not be realized sufficiently under the conditions much different from those at learning stage.

Robust control considered here belongs to the control which is able to control the system stably under the conditions much different from those of the control system at learning stage. The difference in conditions considered here is the difference between initial values of node outputs at control stage and those of node outputs at learning stage. A robust control design method is proposed where the parameters of the control system are tuned to minimize a criterion function, using second order derivatives of the criterion function with respect to parameters, which consists of two terms,one is a usual criterion function and the other is a new term which evaluates the influence of the above differences for all node outouts of the system. Finally it is shown that the controller constructed by the proposed method works in an effective way through a simulaton study of a nonlinear crane system.

2. Structure of Universal Learning Network

Basic structure of U.L.N. which consists of nonlinearly operated nodes and branches that may have arbitrary time delays is shown in **Fig.1**.

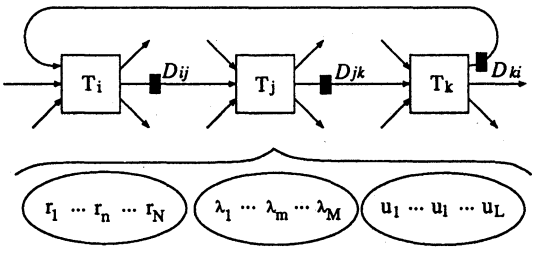


Fig.1 Structure of Universal Learning Network

Basic equation of U.L.N. is represented by Eq.(1):

$$h(T_j,t) = O_j(\{h'(T_j,t)\},\{r_n(t)\},\{u_l(t)\},\{\lambda_m(t)\}), \quad (1)$$

where
- $h(T_j,t)$: Output of T_j node at time t,
- $h'(T_j,t)$: Input of T_j node at time t,
- $r_n(t)$: External input variable at time t,
- $u_l(t)$: Control variable at time t,
- $\lambda_m(t)$: Parameter variable at time t,
- O_j : Nonlinear function of T_j node,

If T_i node is one of the nodes connected with input side of T_j node,then,$h'(T_j,t)$ via a branch with time delay D_{ij} is represented by Eq.(2):

$$h'(T_j,t) = h(T_i, t - D_{ij}) \tag{2}$$

Let a criterion function be written in Eq.(3):

$$E = E(\{h(T_k,s)\},\{u_l(s)\},\{\lambda_m(s)\}) \tag{3}$$

- $k \in K_o$ K_o : Set of nodes related with evaluation,
- $l \in L_o$ L_o : Set of control variables related with evaluation,
- $s \in S_o$ S_o : Set of sampling times related with evaluation.

In the following chapters,a computing method for derivatives of criterion function E with respect to parameter variable $\lambda_m(t_0)$ is presented,which is essential to design a robust control system using U.L.N..[1]

3. Computation of First Order Derivative

First order derivative of E with respect to parameter $\lambda_1(t_0)$ can be written in the form of Eq.(4),assuming t_0 to be designated sampling time,

$$\frac{dE}{d\lambda_1(t_0)} = \sum_{k\in K_o}\sum_{s\in S_o}\left(\frac{\partial E}{\partial h(T_k,s)}\frac{dh(T_k,s)}{d\lambda_1(t_0)}\right) + \frac{\partial E}{\partial \lambda_1(t_0)} \tag{4}$$

As $\frac{\partial E}{\partial h(T_k,s)}$ and $\frac{\partial E}{\partial \lambda_1(t_0)}$ can be calculated easily from Eq.(3),it is matter of importance to calculate $\frac{dh(T_k,s)}{d\lambda_1(t_0)}$.

$\frac{dh(T_k,t)}{d\lambda_1(t_0)}$ can be transformed into Eq.(5).

$$\frac{dh(T_k,t)}{d\lambda_1(t_0)} = \sum_{j\in J}\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\frac{dh(T_j,t-D_{jk})}{d\lambda_1(t_0)}\right) + \frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}, \tag{5}$$

where

J : Set of nodes connected with input side of T_k node.

Putting $P_1(T_k,t,\lambda_1(t_0)) = \frac{dh(T_k,t)}{\lambda_1(t_0)}$, iterative equation of P_1 by forward propagation can be obtained from Eq(5).

$$\begin{aligned} &P_1(T_k,t,\lambda_1(t_0)) \\ &= \sum_{j\in J}\left[\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}P_1(T_j,t-D_{jk},\lambda_1(t_0))\right] \\ &+ \frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)} \end{aligned} \tag{6}$$

$k = 1,2,.....,R,$
$t = 1,2,.....,T,$

where

- R : Number of nodes of the system,
- T : Number of sampling times.

Since $h(T_j,t_0-1)$ does not depend on $\lambda_1(t_0)$, initial value of Eq.(6),

$$P_1(T_j,t_0-1,\lambda_1(t_0)) = 0, \quad j = 1,2,.....,R. \tag{7}$$

4. Computation of Second Order Derivative

Second order derivative of E with respect to parameter variables $\lambda_1(t_0)$, $\lambda_2(t_0)$ can be obtained by differentiating Eq.(4) with respect to $\lambda_2(t_0)$,

$$\begin{aligned} \frac{d^2E}{d\lambda_1(t_0)d\lambda_2(t_0)} &= \sum_{k\in K_o}\sum_{s\in S_o}\left[\frac{d\left(\frac{\partial E}{\partial h(T_k,s)}\right)}{d\lambda_2(t_0)}\frac{dh(T_k,s)}{d\lambda_1(t_0)}\right. \\ &+ \left.\frac{\partial E}{\partial h(T_k,s)}\frac{d^2h(T_k,s)}{d\lambda_1(t_0)d\lambda_2(t_0)}\right] \\ &+ \frac{d(\frac{\partial E}{\partial \lambda_1(t_0)})}{d\lambda_2(t_0)} \end{aligned} \tag{8}$$

$\frac{d^2h(T_k,s)}{d\lambda_1(t_0)d\lambda_2(t_0)}$ in Eq.(8) can be transformed into Eq.(9) by differentiating Eq.(5) with respect to $\lambda_2(t_0)$,

$$\begin{aligned} \frac{d^2h(T_k,t)}{d\lambda_1(t_0)d\lambda_2(t_0)} &= \sum_{j\in J}\left[\frac{d\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{d\lambda_2(t_0)}\frac{dh(T_j,t-D_{jk})}{d\lambda_1(t_0)}\right. \\ &+ \left.\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\frac{d^2h(T_j,t-D_{jk})}{d\lambda_1(t_0)d\lambda_2(t_0)}\right] \\ &+ \frac{d\left(\frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}\right)}{d\lambda_2(t_0)} \end{aligned} \tag{9}$$

Putting $P_1(T_k,t,\lambda_1(t_0)) = \frac{dh(T_k,t)}{d\lambda_1(t_0)}$,and $P_2(T_k,t,\lambda_1(t_0),\lambda_2(t_0)) = \frac{d^2h(T_k,t)}{d\lambda_1(t_0)d\lambda_2(t_0)}$, as in the case of first order derivatives, iterative equation of P_2 by forward propagation can be obtained from Eq.(9),

$$\begin{aligned} &P_2(T_k,t,\lambda_1(t_0),\lambda_2(t_0)) \\ &= \sum_{j\in J}\left[\frac{d\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{d\lambda_2(t_0)}P_1(T_j,t-D_{jk},\lambda_1(t_0))\right. \\ &+ \left.\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}P_2(T_j,t-D_{jk},\lambda_1(t_0),\lambda_2(t_0))\right] \\ &+ \frac{d\left(\frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}\right)}{d\lambda_2(t_0)}, \end{aligned} \tag{10}$$

$$k = 1, 2, ..., R,$$
$$t = 1, 2, ..., T.$$

$$P_2(T_j, t_0 - 1, \lambda_1(t_0), \lambda_2(t_0)) \doteq 0, \quad (11)$$
$$j = 1, 2, ..., R.$$

$\frac{d\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{d\lambda_2(t_0)}, \frac{d\left(\frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}\right)}{d\lambda_2(t_0)}$ in Eq.(10) can be calculated by the computation of first order derivatives putting $E = \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}, E = \frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}$ respectively.

Substituting $\frac{dh(T_k,t)}{d\lambda_1(t_0)}$, $\frac{d^2h(T_k,t)}{\lambda_1(t_0)\lambda_2(t_0)}$ obtained from Eq.(6), (7) and Eq.(10),(11) respectively into Eq.(8), $\frac{d^2E}{d\lambda_1(t_0)\lambda_2(t_0)}$ can be calculated.

5. Robust Control Method

5.1 criterion for suppressing changes of nodes outputs

E is a usual criterion function, E_H is a new term which takes charge of suppressing changes of node outputs of the system caused by the changes of particular node outputs at time t_1.Then a new criterion function L is defined as follows:

$$L = E + E_H, \quad (12)$$

$$E_H = C_H \sum_{s=S_1}^{S_N} \sum_{r \in R_s} \Big(\sum_{i \in R_p} \frac{dh(T_r,s)}{dh(T_i,t_1)} \triangle h(T_i,t_1) \Big)^2, \quad (13)$$

R_p : Set of nodes related with perturbation,
R_s : Set of nodes related with suppression,
$S_i (i = 1, \cdots, N)$: Set of sampling times related with suppression,
$C_H > 0$: coefficient.

$\frac{dh(T_r,S_i)}{dh(T_i,t_1)} \triangle h(T_i,t_1)$ means the change of T_r node output in the case of a change of T_i node output at t_1, namely $\triangle h(T_i,t_1)$. Eq.(13) is the sum of those squared.

5.2.Learning Algorithm

The aim of the optimization learning in U.L.N. used here is to search for the parameters which make the above criterion function L minimal. (From now on the parameter variables are considered to be time invariant.)

The parameter variables in order to minimize Eq.(12) should be calculated by a gradient method.

$$\lambda_m \leftarrow \lambda_m - \gamma \frac{dL}{d\lambda_m}, \quad (14)$$
$$where \quad \frac{dL}{d\lambda_m} = \frac{dE}{d\lambda_m} + \frac{dE_H}{d\lambda_m},$$
$\gamma > 0$: coefficient.

Now,computation of $\frac{dE}{d\lambda_m}$ and $\frac{dE_H}{d\lambda_m}$ can be carried out by making use of the first and the second order derivatives in chapter.3,.4.

< Computation of $\frac{dE}{d\lambda_m}$ >

Putting $\lambda_1(t_0) = \lambda_m$, $\frac{dE}{d\lambda_m}$ is able to be computed using Eq.(4),(6)

< Computation of $\frac{dE_H}{d\lambda_m}$ >

First order derivative of E_H with respect to λ_m can be obtained by differentiating Eq.(13) with respect to λ_m,

$$\frac{dE_H}{d\lambda_m} = 2\, C_H \sum_{s=S_1}^{S_N} \sum_{r \in R_s} \Big[\Big(\sum_{i \in R_p} \frac{dh(T_r,s)}{dh(T_i,t_1)} \triangle h(T_i,t_1) \Big) \times \Big(\sum_{i \in R_p} \frac{d^2h(T_r,s)}{dh(T_i,t_1)d\lambda_m} \triangle h(T_i,t_1) \Big) \Big] \quad (15)$$

Now, $\frac{dh(T_r,s)}{dh(T_i,t_1)}$, $\frac{d^2h(T_r,s)}{dh(T_i,t_1)d\lambda_m}$ are needed in order to compute the $\frac{dE_H}{d\lambda_m}$

[1] Computation of $\frac{dh(T_r,s)}{dh(T_i,t_1)}$

Putting $E=h(T_r,s)$, $\lambda_1(t_0)=h(T_i,t_1)$ and making use of the first order derivative in chapter.3, Eq.(16),(17) can be obtained,

$$\frac{dh(T_r,s)}{dh(T_i,t_1)} = P_1(T_r, s, h(T_i,t_1)). \quad (16)$$

$$P_1(T_k,t,h(T_i,t_1)) = \sum_{j \in J} \Big[\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})} P_1(T_j, t-D_{jk}, h(T_i,t_1)) \Big] + \frac{\partial h(T_k,t)}{\partial h(T_i,t_1)}. \quad (17)$$

[2] Computation of $\frac{d^2h(T_r,s)}{dh(T_i,t_1)d\lambda_m}$

Putting $E=h(T_r,s)$, $\lambda_1(t_0)=h(T_i,t_1)$, $\lambda_2(t_0) = \lambda_m$ and making use of the second order derivative in chapter.4, Eq.(18),(19) can be obtained,

$$\frac{d^2h(T_r,s)}{dh(T_i,t_1)d\lambda_m} = P_2(T_r, s, h(T_i,t_1), \lambda_m). \quad (18)$$

$$P_2(T_k,t,h(T_i,t_1),\lambda_m) = \sum_{j \in J} \Big[\frac{d\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{d\lambda_m} P_1(T_j, t-D_{jk}, h(T_i,t_1)) + \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})} P_2(T_j, t-D_{jk}, h(T_i,t_1), \lambda_m) \Big] + \frac{d\left(\frac{\partial h(T_k,t)}{\partial h(T_i,t_1)}\right)}{d\lambda_m} \quad (19)$$

The coefficient of P_1 in Eq.(19), $\frac{d\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{d\lambda_m}$, can be calculated by computing the first order derivative of $E = \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}$ using Eq.(4),(6).

6. Numerical Example

6.1 Controled System

The controled system is a nonlinear crane system. A position of the crane stand, an angle between the rope and vertical line and a position of the load are represented as x, θ, ℓ respectively. Then the nonlinear crane system is described as follows:

$$\begin{aligned}
\frac{d^2x}{dt^2} &= -\frac{mg}{M}\theta - \frac{D+G}{M}\frac{dx}{dt} + \frac{G}{M}u_d, \\
\frac{d^2\theta}{dt^2} &= -\frac{M+m}{\ell M}g\theta - \frac{D+G}{\ell M}\frac{dx}{dt} + \frac{G}{\ell M}u_d, \\
\frac{d^2\ell}{dt^2} &= -\frac{C+G_m}{m}\frac{d\ell}{dt} + \frac{G_m}{m}u_m,
\end{aligned} \tag{20}$$

where, u_d, u_m are input voltage to a motor for moving the crane stand and to a motor for rolling up the load respectively,and C, G, G_m, D, M, m are appropriate system parameters.

Putting as follows,

$$\begin{aligned}
&h(T_1,t) = x(t), \quad h(T_2,t) = \dot{x}(t), \quad h(T_3,t) = \theta(t), \\
&h(T_4,t) = \dot{\theta}(t), \quad h(T_5,t) = \ell(t), \quad h(T_6,t) = \dot{\ell}(t),
\end{aligned}$$

Eq.(20) can be transformed into discrete type equations as follows:

$$\begin{aligned}
h(T_1,t) &= a_{11}h(T_1,\hat{t}) + a_{21}h(T_2,\hat{t}), \\
h(T_2,t) &= a_{22}h(T_2,\hat{t}) + a_{32}h(T_3,\hat{t}) + b_1u_d(\hat{t}), \\
h(T_3,t) &= a_{33}h(T_3,\hat{t}) + a_{43}h(T_4,\hat{t}), \\
h(T_4,t) &= a_{24}\frac{h(T_2,\hat{t})}{h(T_5,\hat{t})} + a_{34}\frac{h(T_3,\hat{t})}{h(T_5,\hat{t})} \\
&\quad + a_{44}h(T_4,\hat{t})\frac{b_1}{h(T_5,\hat{t})}u_d(\hat{t}), \\
h(T_5,t) &= a_{55}h(T_5,\hat{t}) + a_{65}h(T_6,\hat{t}), \\
h(T_6,t) &= a_{66}h(T_6,\hat{t}) + b_2u_m(\hat{t}). \\
&\quad Where\ \hat{t} = t-1.
\end{aligned} \tag{21}$$

A recurrent type control model of the nonlinear crane system using U.L.N. is shown in **Fig.2**. Each control input u_d, u_m is constructed by two control nodes respectively ,one is the node with linear function,the other is the node with tanh function. (All branches have one sampling time delay.)

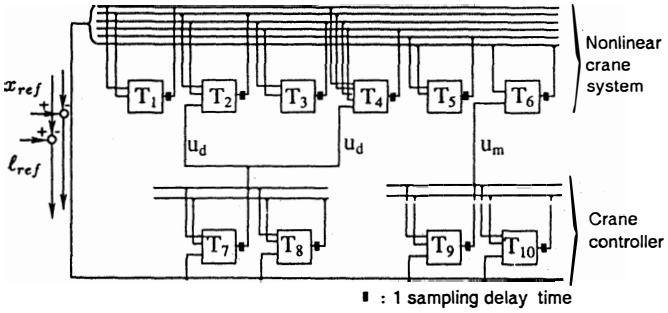


Fig.2 Recurrent type control model of a nonlinear crane system using Universal Learning Network

6.2 Criterion Function

M=40[kg], D=300[kg/sec], G=700[N/V], m=2[kg], g=9.8[m/sec^2],G_m=0.98[N/V], C=0.42[kg/sec] are used, and reference of moving the crane stand(x_{ref}) is 1[m], reference of rolling up the load(ℓ_{ref}) is 0.5[m].

When the parameter variables are tuned through learning, initial values of node outputs of the system are set up as follows:

$$h(T_5,0) = 1.0, \ h(T_i,0) = 0.0 \ \ (i = 1,2,3,4,6). \tag{22}$$

In numerical example, a change of the initial position of the load ,namely $\Delta h(T_5,0)$ is assumed.

The criterion E to achieve the desired dynamics of the system and E_H to achieve the suppressing the perturbation of the system caused by the change of the initial position of the load are defined respectively as follows,

$$\begin{aligned}
E &= \frac{1}{2}\Big[\sum_{s\in S_o}\{Q_{11}(x_{ref} - h(T_1,s))^2\} + Q_{12}(h(T_2,t_f))^2 \\
&\quad + \sum_{s\in S_o}\{Q_{13}(h(T_3,s))^2 + Q_{14}(h(T_4,s))^2\} \\
&\quad + \sum_{s\in S_o}\{Q_{15}(\ell_{ref} - h(T_5,s))^2\} + Q_{16}(h(T_6,t_f))^2 \\
&\quad + \sum_{s\in S_o}\{R_1(h(T_7,s))^2 + R_2(h(T_9,s))^2\}\Big]
\end{aligned} \tag{23}$$

$$E_H = C_H\sum_{r\in R_o}\left(\frac{dh(T_r,s)}{dh(T_5,0)}\,\Delta h(T_5,0)\right)^2 \tag{24}$$

where, S_o : Set of all sampling times, t_f : final time, $Q_{11} \sim Q_{16} = 1.0$, $R_1 \sim R_2 = 0.001$, r : $1,2,3,4,5,6$. $C_H = 100$, $\Delta h(T_5,0) = 1$ $s = 1.0$ [sec] in Eq.(24). A total number of learning is five thousand.

Using these values, two cases have been learned, one, named $case_E$, is the case of using the criterion function E, the other, named $case_L$, is the case of using the criterion function L.

6.3 Simulation Results

Relation between number of learning and values of the criterion functions E and L at learning is shown in **Fig.3**. Using the control parameter variables obtained at learning number equals to five thousand, control simulations have been carried out for $\ell(0) = 1, 2, 3, 4, \cdots$ and $\ell_{ref} = 0.05, 1, 2, \cdots$. In case of $\ell(0) = 1, \ell_{ref} = 0.5$, that is ,the conditions at control stage are the same as those at learning stage, control results are shown in **Fig.4**. ($\ell(0)$ means the initial position of the load at control stage.) Both result of $case_E$ and $case_L$ are almost the same. In **Fig.5**, system stable operation area is shown as the triangle , for example, in case of **Fig.5(a)**, such an area is shown as shadow one. That area means that the system operates stably, when initial value of ℓ changes between 0.1 and 7.0 ,and reference value of ℓ is fixed to 0.5, and other conditions at control stage are the same as those at learning stage. **Fig.5** shows that in case of perturbation of $\ell(0)$, the area in **Fig.5(b)** is larger than one in **Fig.5(a)**. Conversely in case of perturbation of ℓ_{ref} , the area in **Fig.5(b)** is narrower than one in **Fig.5(a)**. It is thought this result has been caused by the suppression of perturbation of $\ell(0)$. In **Fig.6**, control results of **Fig.5(b)**($\ell(0) = 20.0, \ell_{ref} = 0.5$) show that the system operates stably and gets its goal.

Thus in the event of a large change of node outputs,the effect of E_H to stability of the system appears remarkably.

7. Conclusion

A robust control method using forward propagation U.L.N. is proposed,and through simulation study,it has been shown that the proposed method is very useful. In numerical example, the controller obtained makes the system operate stably in the event of $\ell(0) = 20$. ($\ell(0) = 20$ is twenty times as large as $h(T_5, 0) = 1$ in Eq.(22)). This means that the proposed method is the method for practical use.

In this paper we proposed the robust control method for a change of nodes outputs. The robust control method for perturbation of reference, system parameter variables and external inputs can be derived in the same way as in the case of a change of node outputs.

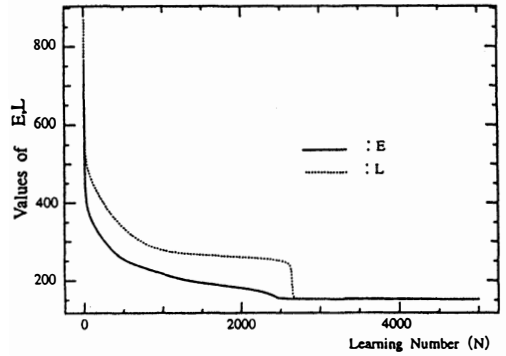


Fig.3 Values of criterion function E, L

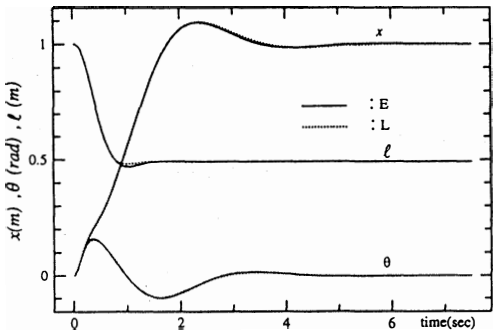


Fig.4 Results of control in case of $\ell(0) = 1, \ell_{ref} = 0.5$

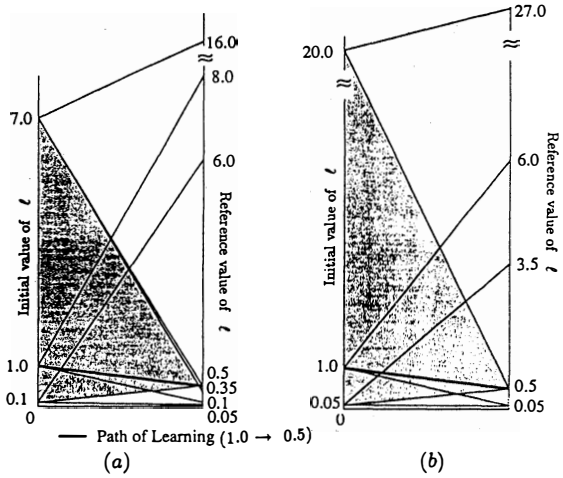


Fig.5 System stable areas (a): $case_E$, (b): $case_L$, when $\ell(0)$ and ℓ_{ref} change.

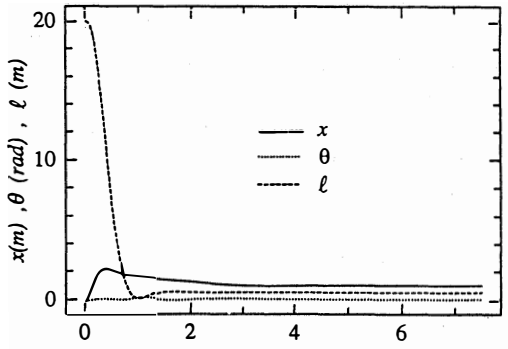


Fig.6 Control results in case of $case_L$($\ell(0) = 20, \ell_{ref} = 0.5$)

References

[1] K.Hiraswa,M,Ohbayashi,M,Koga:Forward Propagation Universal Learning Network ,Memoirs of the Faculty of Engineering,Kyushu University,Vol.55,No3,pp225-234,1995

[2] M.Ohbayashi,K.Hirasawa,J.Murata:Robust Control by Universal Learning Network,Proc. of 1995 Korea Automatic Control Conference, pp123-126,1995

Chaos Control on Multi-Branch Universal Learning Network

Masaru KOGA, Kotaro HIRASAWA, Junichi MURATA and Masanao OHBAYASHI
Department of Electrical Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, Japan
Tel : +81-92-641-1101 (Ext 5329), Fax : +81-92-631-2790
E-mail : masa@scel.ee.kyushu-u.ac.jp

ABSTRACT

In this paper, the chaos control already proposed is extended to multi-branch Universal Learning Network (U.L.N.) for the purpose of investigating the influence of increasing the number of branches and changing delay times. Controlling chaotic phenomena on U.L.N. can be realized by changing Lyapunov Number of U.L.N., which is accomplished by adjusting U.L.N. parameters so as to minimize a criterion function that is the difference between the desired Lyapunov Number and its actual value. Simulations are done for the single and multi-branch U.L.N.. Based on the simulation results of the single, so-called the ordinary recurrent neural network, and multi-branch U.L.N., suitable U.L.N. parameters and structures have been made clear that can realize generation and die-out of chaotic phenomena more efficiently.

1. Introduction

The authors already proposed a Chaos Control Method on Universal Learning Network(U.L.N.)[1]. The method can control chaotic phenomena on such U.L.N. that have bounded functional nodes.

The method can control generation and die-out of chaotic phenomena on U.L.N. by changing the Lyapunov Number of U.L.N. to positive and negative values respectively.

From the simulation, it has been certified that the controlling chaotic phenomena can be easily realized on the single-branch U.L.N., that is, the ordinary recurrent neural network; no special terms are needed, whereas chaotic neural network model proposed by Aihara has a few special terms[3]. And the mechanism of controlling chaotic phenomena has been investigated; some insight has been obtained into the condition that is necessary for controlling chaotic phenomena on U.L.N..

However, a recurrent neural network is only a particular type of U.L.N.. U.L.N. can represent more general and more realistic large-scale complicated systems which may have multiple interconnections between their nodes and various delay times.

In this paper, the chaos control is extended to multi-branch U.L.N. for the purpose of investigating the influence of increasing the number of branches and changing delay times.

Simulations are done for the single and multi-branch U.L.N.. The results show that, in the case of generating chaotic phenomena, the Lyapunov Number can be more quickly changed to the positive desired value on the multi-branch U.L.N. than on the single-branch U.L.N., so-called the ordinary recurrent neural network, that is, generating chaotic phenomena is easier on the multi-branch U.L.N.. The results also show that the larger the time delay is, the more difficult changing the Lyapunov Number to desired positive Lyapunov Number is, in other words, generating chaotic phenomena becomes more difficult.

Based on the simulation results of the single and the multi-branch U.L.N., suitable U.L.N. parameters and structures e.g., the number of branches and delay times, have been made clear that can realize generation and die-out of chaotic phenomena more efficiently.

2. Multi-Branch Universal Learning Network

U.L.N.[2] is briefly described in this section.

U.L.N. is a network which consists of arbitrary nonlinear functional nodes and arbitrary delay elements between the nodes, and is aimed to model and control various large-scale complicated systems like industrial plants as well as economical, social and

life phenomena. Structure of a multi-branch U.L.N. is shown in **Fig.1**, and basic equation of multi-branch U.L.N. is represented as follows,

$$h(T_j,t) = O_j(\{h'(T_j,t)\},\{\lambda_m(t)\},\{r_{m'}(t)\},\{u_{m''}(t)\} \quad (j \in R)\ (t \in T), \tag{1}$$

$h(T_j,t)$: T_j node's output at sampling time t $(j \in R)$,
$h'(T_j,t)$: T_j node's input at sampling time t,
$\lambda_m(t)$: m parameter at sampling time t $(m \in M)$,
$r_{m'}(t)$: m' external input at sampling time t $(m' \in M')$,
$u_{m''}(t)$: m'' control variable at sampling time t $(m'' \in M'')$,
O_j : T_j node's nonlinear function,
R : set of nodes,
T : set of sampling times,
M : set of parameters,
M' : set of external inputs,
M'' : set of control variables.

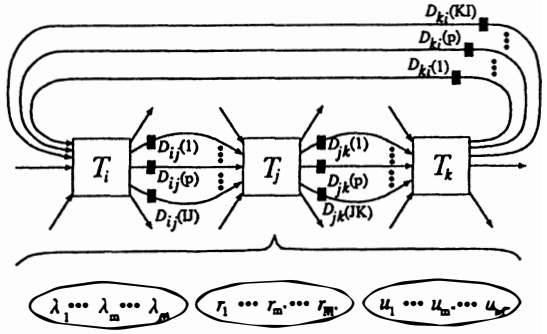


Fig.1 Structure of Multi-Branch Universal Learning Network

$D_{ij}(p)$ $(p \in IJ)$ means the delay time from T_i to T_j nodes. The input to T_j node $h'(T_j,t)$ is expressed as follows,

$$h'(T_j,t) = h(T_i, t - D_{ij}(p)), \tag{2}$$

IJ : set of branches from T_i to T_j.

Criterion function of a system is represented as follows,

$$E = E(\{h(T_r,s)\},\{u_{r'}(s')\}), \tag{3}$$

$r \in R_o,\ r' \in R'_o,\ s,\ s' \in S_o$,

$R_o,\ R'_o,\ S_o$: set of nodes, set of control variables, and set of sampling instances related to the criterion function.

Learning of U.L.N. is to search for the parameters, $\lambda_m(t)$ and $u_l(t)$, which minimize the criterion function by the gradient method.

3. Chaos Control Method

The method already proposed can control chaotic phenomena on such U.L.N. that have bounded function nodes[1]. The method is that generation of chaotic phenomena is realized by changing Lyapunov Number of U.L.N. to positive value, and die-out of the chaotic phenomena is realized by changing the Lyapunov Number to negative one. The gradient method utilizes second order derivatives of U.L.N. because Lyapunov Number of U.L.N. is calculated by first order derivatives.

3.1. Lyapunov Number and its Derivatives

Lyapunov Number of U.L.N. is defined as follows (see **Fig.2**),

$$L_i = \lim_{N\to\infty} \frac{1}{NS} \times \sum_{n=0}^{N-1} \left[\sum_{t=s_{(n)}}^{s_{(n+1)}-1} \ln \sqrt{\sum_{r\in R} \left\{ \frac{dh(T_r, s_{(n+1)})}{dh(T_i,t)} \right\}^2} \right], \tag{4}$$

S : $s_{(n+1)} - s_{(n)}$,
N : number of S,
R : number of whole network nodes.

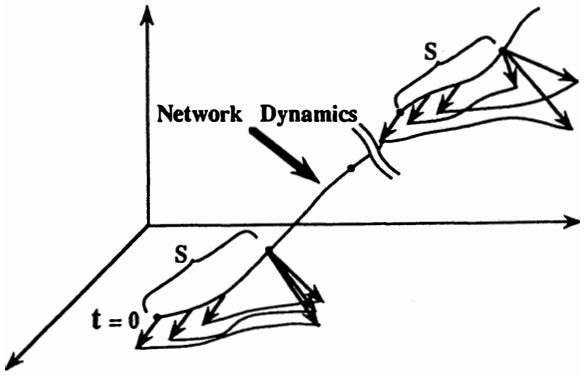


Fig.2 Explanation of Lyapunov Number

The criterion function E_L is defined as follows,

$$E_L = \sum_{r\in R} (L^o_r - L_r)^2, \tag{5}$$

L^o_r : desired values of L_r.

Parameter λ_m is adjusted to minimize criterion function by the gradient method,

$$\lambda_m \leftarrow \lambda_m - \gamma \frac{dE_L}{d\lambda_m}, \tag{6}$$

$$\frac{dE_L}{d\lambda_m} = 2 \sum_{r\in R} (L_r - L^o_r) \frac{dL_r}{d\lambda_m}. \tag{7}$$

If $dL_i/d\lambda_m$ can be calculated, parameter λ_m can be learned. From equation (4), $dL_i/d\lambda_m$ is obtained as follows,

$$\frac{dL_i}{d\lambda_m} = \lim_{N\to\infty}\frac{1}{NS} \times \sum_{n=0}^{N-1}\left[\sum_{t=s(n)}^{s(n+1)-1}\frac{\sum_{r\in R}\frac{dh(T_r,s(n+1))}{dh(T_i,t)}\frac{d^2h(T_r,s(n+1))}{dh(T_i,t)d\lambda_m}}{\sum_{r\in R}\left\{\frac{dh(T_r,s(n+1))}{dh(T_i,t)}\right\}^2}\right].$$

If $dh(T_r,s)/dh(T_i,t)$, $d^2h(T_r,s)/dh(T_i,t)d\lambda_m$ are calculated, λ_m is able to be learned.

$dh(T_r,s)/dh(T_i,t)$ is calculated as follows,

$$\frac{dh(T_r,s)}{dh(T_i,t)} = \delta_1(T_i,t), \tag{8}$$

$$\delta_1(T_j,t) = \sum_{k\in K}\sum_{p\in JK} \left\{\frac{\partial h(T_k,t+D_{jk}(p))}{\partial h(T_j,t)}\delta_1(T_k,t+D_{jk}(p))\right\}, \tag{9}$$

$$\delta_1(T_r,s) = 1.$$

$d^2h(T_r,s)/dh(T_i,t)d\lambda_m$ is calculated as follows,

$$\frac{d^2h(T_r,s)}{dh(T_i,t)\lambda_m} = \delta_1(T_i,t), \tag{10}$$

$$\delta_1(T_j,t) = \sum_{k\in K}\sum_{p\in JK} \left\{\frac{\partial h(T_k,t+D_{jk}(p))}{\partial h(T_j,t)}\delta_1(T_k,t+D_{jk}(p))\right\} + \sum_{k\in K}\sum_{p\in JK} \left[\frac{d\left\{\frac{\partial h(T_k,t+D_{jk}(p))}{\partial h(T_j,t)}\right\}}{d\lambda_m}\delta_{12}(T_k,t+D_{jk}(p))\right], \tag{11}$$

$$\delta_{12}(T_j,t) = \sum_{k\in K}\sum_{p\in JK} \left\{\frac{\partial h(T_k,t+D_{jk}(p))}{\partial h(T_j,t)}\delta_{12}(T_k,t+D_{jk}(p))\right\},$$

$$\delta_{12}(T_r,s) = 1. \tag{12}$$

In equation (11), by putting
$E' = \partial h(T_k,t+D_{jk})/\partial h(T_j,t)$,
$d\{\partial h(T_k,t+D_{jk})/\partial h(T_j,t)\}/d\lambda_m$ is calculated as follows,

$$\frac{d\left\{\frac{\partial h(T_k,t+D_{jk}(p))}{\partial h(T_j,t)}\right\}}{d\lambda_m} = \sum_{t'\in T}\left[\sum_{d\in D(\lambda_m)}\left\{\frac{dh(T_d,t')}{d\lambda_m}\delta'(T_d,t')\right\}\right] + \frac{\partial E'}{\partial\lambda_m},$$

$D(\lambda_m)$: Set of nodes including λ_m, (13)

$$\delta'(T_j,t) = \sum_{k\in K}\sum_{p\in JK} \left\{\frac{\partial h(T_k,t+D_{jk}(p))}{\partial h(T_j,t)}\delta'(T_k,t+D_{jk}(p))\right\} + \frac{\partial E'}{\partial h(T_j,t)}. \tag{14}$$

3.2. Random Search Method

It is difficult to optimize the parameters successfully by using only the gradient method because Lyapunov Number is very sensitive to changes in the parameters and the criterion function is not convex. Therefore, a random search method is combined with the gradient method.

During iteration of the gradient based learning, if Lyapunov Number is not improved over its previous value, the previous parameters are randomly and uniformly changed within a fixed range. And new Lyapunov Number is calculated with new parameters. The Lyapunov Number has to be calculated repeatedly by changing parameters randomly until it gets better than the previous Lyapunov Number.

If the Lyapunov Number is better than the previous one, learning is resumed by the gradient method again. If not, the parameters should be chosen which give the best Lyapunov Number among those tried in the random search.

4. Simulations

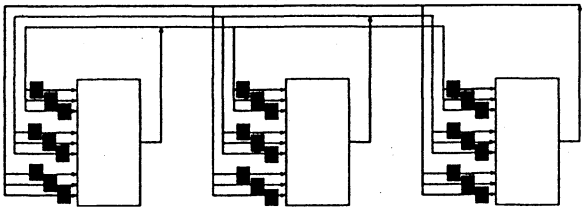
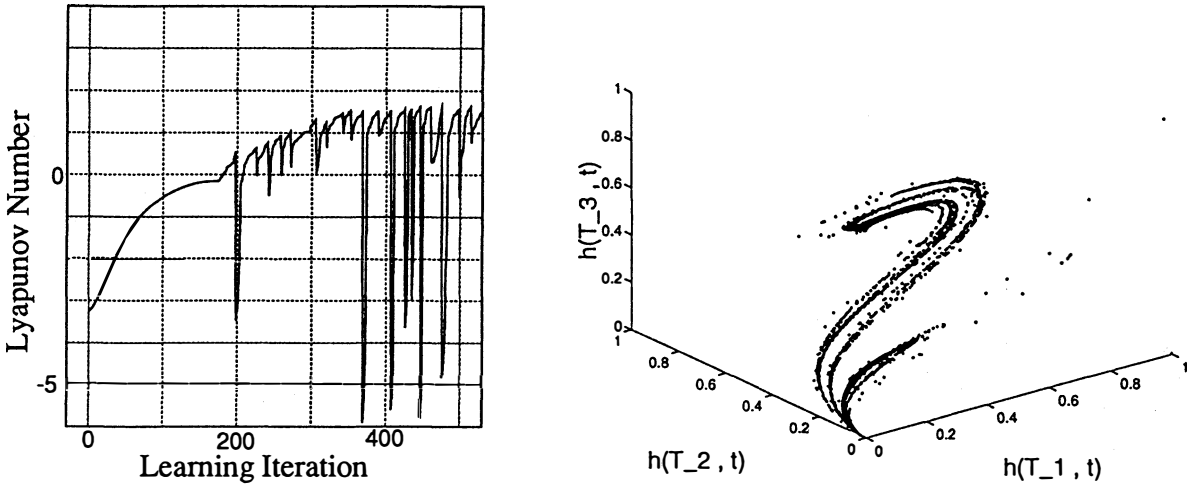


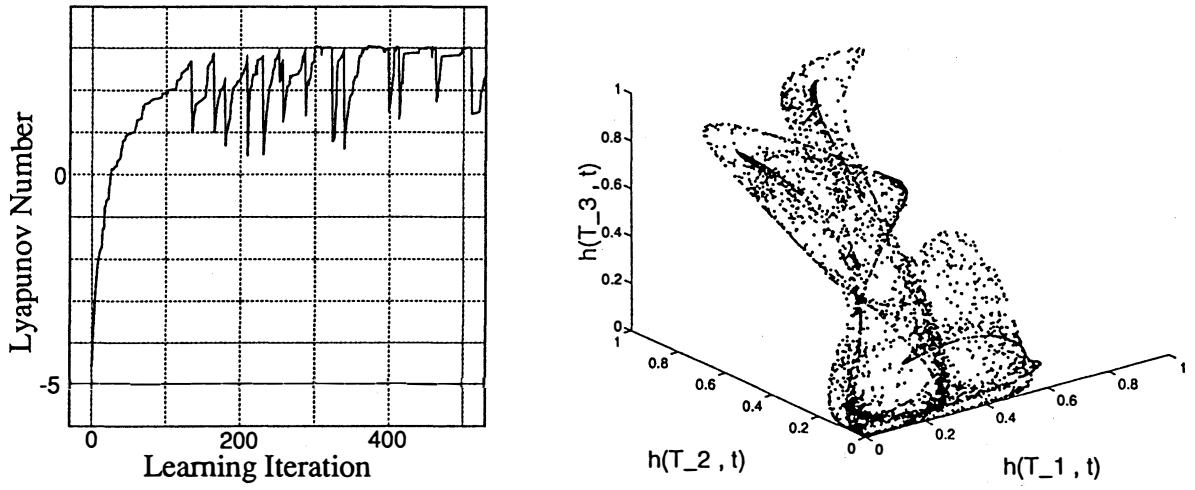
Fig.3 Multi-Branch U.L.N.

Simulations are carried out to generate chaotic phenomena on multi-branch U.L.N. shown in **Fig.3.** Sigmoid functions are adopted as U.L.N. node functions. The equation describing the network is as follows,



(a) (b)

Fig.4 Generation of chaos on Single-Branch U.L.N.



(a) (b)

Fig.5 Generation of chaos on Multi-Branch U.L.N.

$$h(T_j,t) = f(\alpha_j) = \frac{1.0}{1.0 + exp(-\phi_j \alpha_j)}, \quad (15)$$

$$\alpha_j = \sum_{i \in I} \sum_{p \in IJ} h_{ij}(p) h'(T_j,t) + \theta_j, \quad (16)$$

f : sigmoid functions,
I : set of nodes connected to input side of T_j node,
IJ : set of branches,
h_{ij}, θ_j, ϕ_j : weight, threshold and slope of sigmoid functions.

There are three branches between the nodes (see **Fig.3**), and weights, thresholds and slopes of the sigmoid functions are optimized.

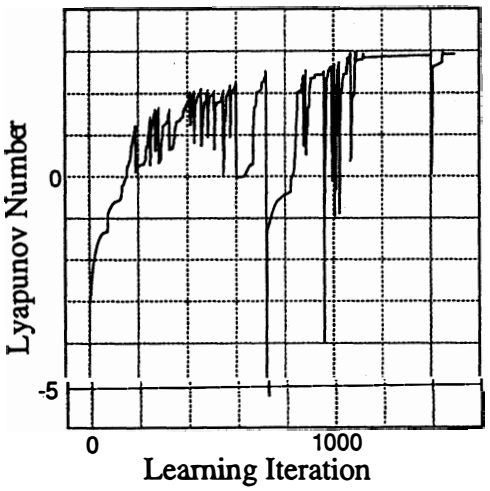
Desired Lyapunov Number is set to positive number three.

4.1. One-sampling delay time

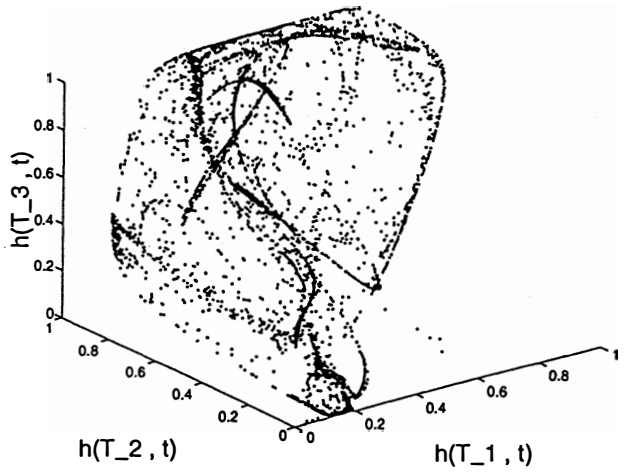
At first, every delay time is set to one-sampling time.

Learning curve of Lyapunov Number and network dynamics after 500 epochs of both single-branch U.L.N. and multi-branch U.L.N. are respectively shown in **Fig.4(a)**, **(b)** and **Fig.5(a)**, **(b)**. **Fig.4(a)** shows that it takes 200 epochs to change the Lyapunov Number to positive value in the single-branch U.L.N.. However, from **Fig.5(a)**, in the multi-branch U.L.N., it takes less than 50 epochs to exceed zero value of Lyapunov Number.

Furthermore, in the single-branch U.L.N., the Lya-

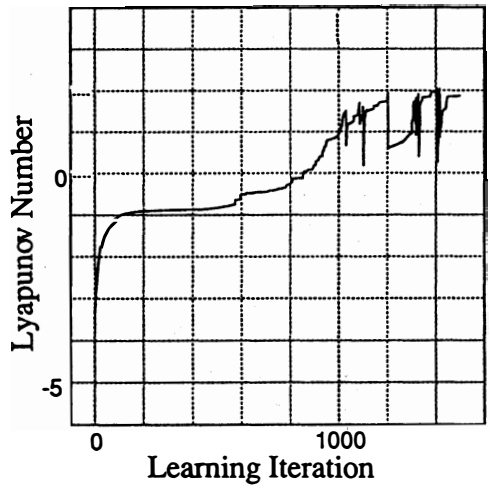


(a)

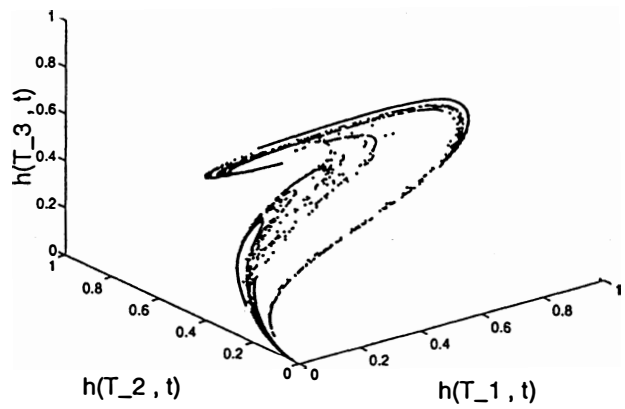


(b)

Fig.6 Results of D_1 network (one sampling delay time is dominant)

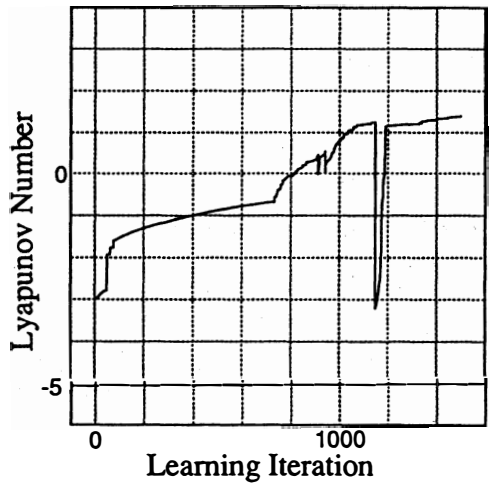


(a)

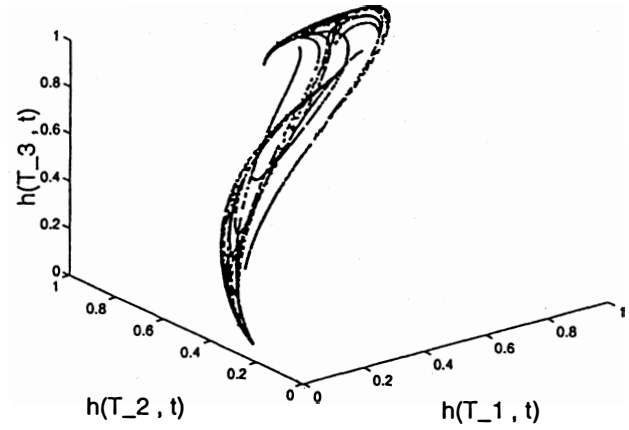


(b)

Fig.7 Results of D_2 network (two sampling delay time is dominant)



(a)



(b)

Fig.8 Results of D_3 network (three sampling delay time is dominant)

punov Number cannot be reached to the desired value within 500 epochs. On the other hand, the Lyapunov Number can be rearched near the desired value at about over 150 epochs.

Accordingly, in case that every time delay is set to one-sampling time, the Lyapunov Number can be more quickly changed to the positive desired value on the multi-branch U.L.N. than on the single-branch U.L.N., that means generating chaotic phenomena is easier.

The reason can be explained as follows; changing the Lyapunov Number to positive value is caused by making δ_1 in equation (9) a large value. From the equation (9), it is clear that the larger the number of branches connecting to output side of T_j node is, the easier changing δ_1 to a large value is. Thus generating chaotic phenomena becomes easier.

4.2. Various delay times

Consider the case where the value of delay time is varied in the multi-branch U.L.N..

The multi-branch U.L.N. has 27 branches. Three cases are selected for assigning delay times to these branches. In the first case, delay time $D_{ij} = 1$ is dominant, where 21 out of the 27 delay times are equally set to one and each of the other 6 is randomly set to either one, two or three. In other cases, delay time $D_{ij} = 2$ and 3 are dominant, respectively. The multi-branch U.L.N. obtained in these three cases are referred to as D_1, D_2 and D_3 networks, respectively.

Learning curves of Lyapunov Number and the network dynamics after 1500 learning epochs of the D_1, D_2 and D_3 networks are respectively shown in **Fig.6(a), (b), Fig.7(a), (b) and Fig.8(a), (b).**

The largest Lyapunov Number of the D_1, D_2 and D_3 networks are respectively 3.0, 2.0 and 1.5. The results show that the larger the time delay is, the more difficult changing the Lyapunov Number to desired positive Lyapunov Number is, in other words, generating chaotic phenomena becomes more difficult. The reason is that changing Lyapunov Number to large value is more difficult because the influence of magnification of the dynamics by the perturbation cannot appear in the range S in the case of the large value of delay time.

5. Conclusions

In this paper, the chaos control is extended to multi-branch U.L.N. for the purpose of investigating the influence of increasing the number of branches and changing delay times.

Simulations are carried out to generate chaotic phenomena on the single and multi-branch U.L.N..

The following insights are obtained:

1. in the case of generating chaotic phenomena, the Lyapunov Number can be more quickly changed to the positive desired value on the multi-branch U.L.N. than on the single-branch U.L.N., so-called ordinary recurrent neural network, that is, generating chaotic phenomena is easier,
2. the larger the time delay is, the more difficult changing the Lyapunov Number to desired positive Lyapunov Number is, in other words, generating chaotic phenomena becomes more difficult.

Based on the simulation results of the single and multi-branch U.L.N., suitable U.L.N. parameters and structures, e.g., the number of branches and delay time, have been made clear that can realize generation and die-out of chaotic phenomena more efficiently.

References

[1] Masaru.KOGA, Kotaro HIRASAWA, Junichi MURATA and Masanao OHBAYASHI, "Chaos Control using Second Order Derivatives of Universal Learning Network", *ICNN'95*, Vol.3, 1287–1292. 1995.

[2] Kotaro HIRASAWA, Junichi MURATA and Masanao OHBAYASHI, "Universal Learning Network and Computation of its Higher Order Derivatives", *ICNN'95*, Vol.3, 1273–1277. 1995.

[3] K. Aihara, T. takabe and M. Toyoda, "Chaotic Neural Networks", *Phys.Lett.A*, 144, 6, 7, pp. 333–340. 1990.

A Step Toward Human - Robot Cooperative System

Masaru Ishii,
Fukuoka Inst. of Tech.,
Fukuoka, Japan

Hironori Saita
Fukuoka Inst. of Tech.,
Fukuoka, Japan

Abstract

In near future, we will have to live with many machines such as home robots, hospital robots and so on. User friendly communication between man and those machines is very important for cooperative system utilizing their features and abilities. As an example of human - robot cooperative system, we propose an intuitive approach to robot teaching method with multi-media tools.

Keywords: Robot Teaching, Multimedia, Interface, Human - Robot Cooperation

1. Introduction

In the conventional research on an intelligent robot system, many autonomous robots were made for carrying out all tasks automatically by themselves. These robot systems do not consider a role of a man as an operator except for master-slave type robot systems.

In near future, we will have to live with many working machines such as home robots and hospital robots and so on. User friendly communication between man and those machines is very important for human - robot cooperative system that utilizes their features and abilities in order to accomplish their common operational tasks.

Realization of such cooperative system is eagerly required for future robot systems. In this system, various communicating functions between man and robot system are necessary for exchanging their commands, advises and comments about their common operational goals with user friendly interfaces.

As a first step for constructing above system, robot teaching system with multi-media tools is described in this paper. Clicking buttons on the display by a mouse device mean controlling a robot system. After we make many primitives of robot's operations using this method, we can create any necessary tasks easily by a macro operation that combined those primitives. Also, we can select a wanted task from robot macro operation's database. One of our aims is to make such a simple command for executing complicated robot tasks.

This robot teaching method using multimedia tools provides a foundation for realizing user friendly human - robot cooperative system.

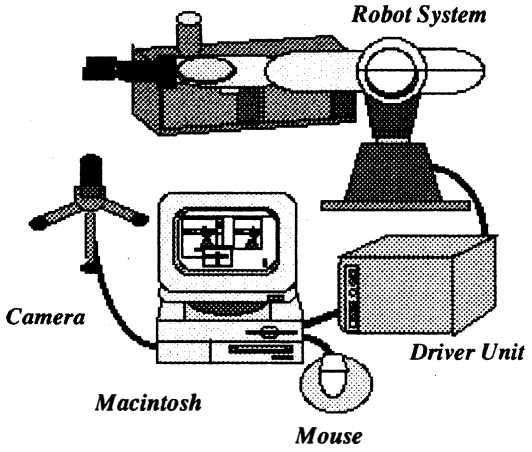


Fig.1. A robot teaching system

2. A new robot teaching metod

Teaching is essential in order for robots to carry out various tasks with several kinds of moving sequences. Robot teaching by showing or guiding is called playback control. Playback control

method is widely used in many industries and a great deal of operator time and labor is required. Teaching box, however, is not so flexible to expand for various teaching commands easily and must be improved in aspects of user interfaces. We have developed a structual robot teaching method using multi-media tools in a Macintosh computer as showed in ***Fig.1***.

In our system, we use a robot system (Mitsubishi - movemaser EX) with five degrees of freedom. The robot has six movable parts: waist, shoulder, elbow, list-pitch, list-roll and hand (open or close). The robot has twenty-four position's control commands, four hand's control commands and six reading commands using a RS232C cable.

Hyper card system of a Macintosh computer is a multi-media information exchangeable base software. It has fields for texts, pictures, button for managing cards and sounds. Hyper-talk is an object oriented type programming language of hyper card system for stack control. Its scripts are defined as an attribute of a button. It is designed conveniently to transform a command from users. For example, an objective of scripts written in hyper talk is carried out by selecting a button with a mouse device.

It is necessary for robot control to send a serial signal of control commands of a robot system from hyper card system. Hyper-talk has not such a sending signal command and thus XCMD: extended command of hyper talk, is used. XCMD is a machine language image compiled by C program and added to hyper card for high speed processing such as signal sending. We use a send-signal XCMD for sending serial data.

Ideally, operator gives an outline of task to a robot system and then robot system must carry out the delicate and complicated works. In practical situation, man and a robot system have to cooperate each other with their features and abilities.

At first, we must teach a primitive operation of a robot system. We select a registration card stack shown in ***Fig.2***. For moving five movable joints such as waist, shoulder, elbow, list's pitch and list's roll of a robot system, we slide each lever of small windows on the display of a Macintosh computer. These slide operations are carried out by a mouse device. In this registration card, we can teach robot's primitive operations such as point to point positioning.

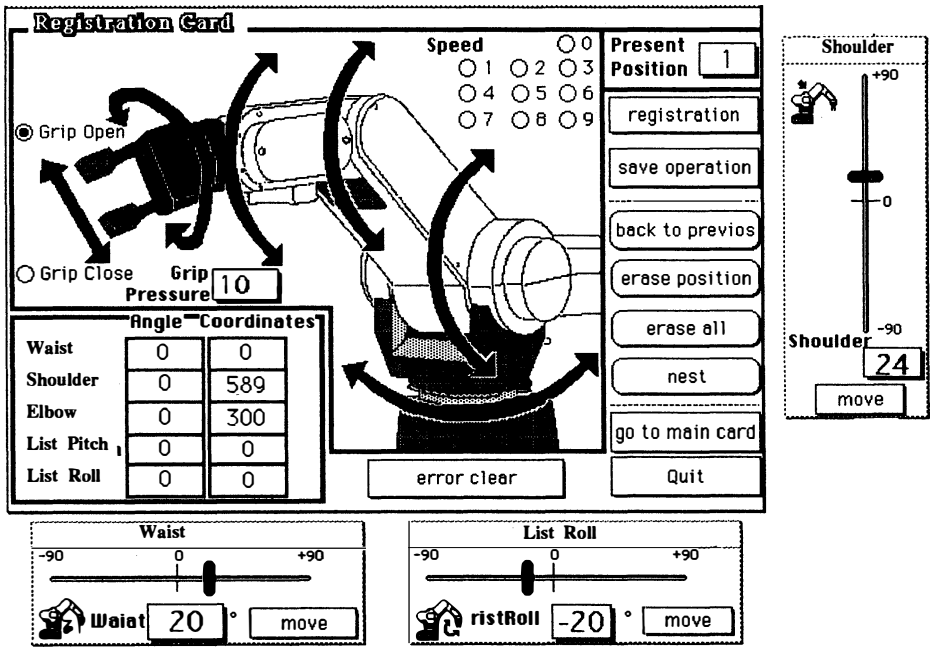


Fig.2. Robot control by sliding levers on the display of a Macintosh computer

3. Robot macro operation

Actually, a complicated task of a robot system consists of several primitive robot operations. So we can replace a robot complex work with a sequence of these primitive operations. In robot macro card in ***Fig.3***, we can give a name to a macro operation. Left lower part of this stack shows macro operation's names such as **put-into-video-deck**, **put-back-video-tape** for example. Operation **put-into-video-deck** means that an operator teaches 3D positions and orientation of a robot system for putting a tape into video-deck by our teaching method. As soon as we push a **put-into-video-deck** button, robot system plays back the same operation quickly.

As this system has a hierarchy structure, we can make a higher macro operation such as handling a **Video Tape** operation by selecting and collecting optimal operations from lower macro ones. This operation has manipulating sequences as follows: grasping a video tape, inserting it and so on. So we can make a simple command for such a complicated task.

Also, robot moving speed can be set and standard pose button means all zero state of all movable parts. In right upper side of this stack, current positions and angles of each joint are displayed. We can get 3D positions and orientation of a robot system to receive these data from a drive unit by a signal handling external command.

In this way, a robot operation's sequence is created as one card stack by this teaching method. A registered task is performed automatically by clicking a button of above operation on the main menu card.

4. Sensor and voice control of a robot

In the human - robot cooperative system, man's commands are expected more simple and a robot have to carry out complicated tasks utilizing their features and abilities in order to accomplish their common operational tasks. Our system make use of optical sensors attached at a robot hand to find out and grasp an object automatically (***Fig.4***). After we move a robot 's hand near an obeject , robot system can execute fine operations using sensor feedback.

In an intelligent cooperative system, various functions communicating between a man and robot systems are necessary for exchanging their commands, advises and comments about their operational goals with user friendly interfaces.

In this system, we use voice control utilities for realizing robot operations by voice commands instead of selecting buttons of the hyper-card system. Commands' name must be registered by

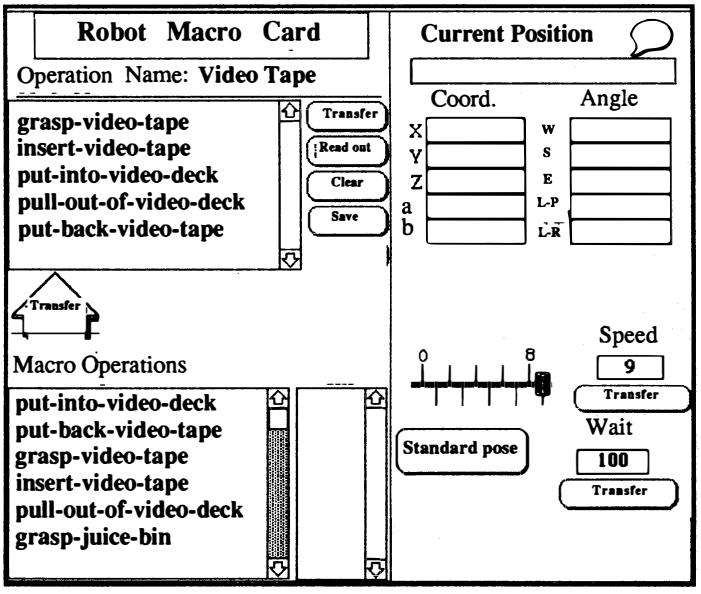


Fig.3. Layout of a robot macro operation card

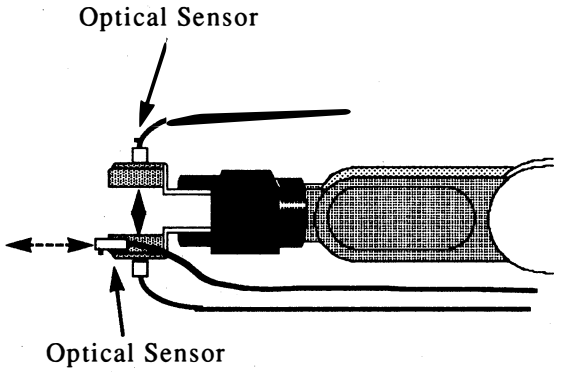


Fig.4. Optical sensors attached at a hand

operator's voice in advance.

Besides this operator's voice command, several voice responses from a robot system and a teaching system are necessary as follows:

(1) Guidance about teaching sequence,
(2) Report of moving action's results,
(3) Reports of current robot's states
(4) Help requirement
(5) Questions and so on.

5. Experimental results

In ***Fig.5***, we show a complete view of an experimental robot teaching system composing of a robot system, a Macintosh computer, a voice navigator and a monitoring camera set.

Multimedia display of our system is shown in ***Fig.6***. In left side of this display, hyper card stack is seen for controlling and teaching a robot system. Monitoring scene of this operation is in right up side. As an experiment example, we make macro operations about handling a video tape from teaching stacks mentioned above. An experimental task is to grasp a video tape, insert and push it into a video deck. Next, these systems pull a video tape from a deck and put it back to the shelf. As shown in ***Fig.7***, this system can insert a video tape into a deck successfully.

6. Conclusion

This paper describes an intuitive robot teaching method with multimedia tools for user friendly interfaces. Simple example of robot teaching is carried out 1/3 teaching time in comparison with the conventional method because of using only a mouse device for all robot teaching.

The goal of our system is making human - robot cooperative system for overcoming common operational tasks with mutual conversation and collaboration. Sharing and learning ability of robot system by operator's advice are expected to solve common problems in our system.

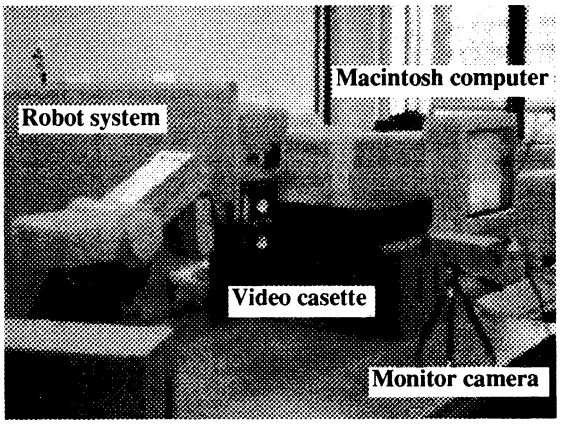


Fig.5. Overview of an experimental system

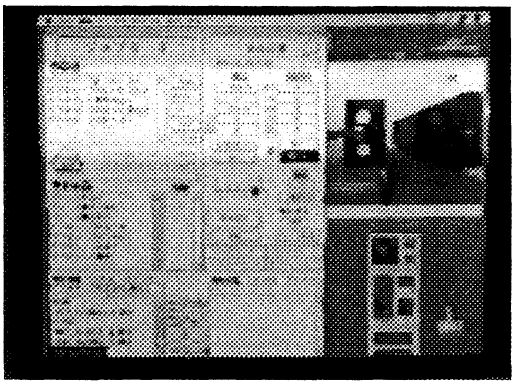


Fig.6. Display layout of our system

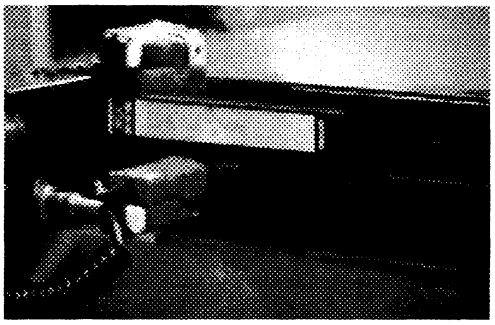


Fig.7. Inserting operation

Acknowledgment

The authors would like to thank to the Language and Information Laboratory of Fukuoka Inst. of Tech. for supporting the research.

References

[1] M.,Ishii,"A robot teaching method using Hyper Card system," Proc. of IEEE Workshop on Robot and Human Communication, pp.410-412, 1992.

[2] S.,Hirai and T.,Sato,"Motion Understanding for World Model Management of Telerobot," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS), pp.124-131, 1989.

Proposal of Decentralized Consensus-making Mechanisms Based on Immune System
~Application to a Behavior Arbitration of an Autonomous Mobile Robot~

Akio ISHIGURO, Yuji WATANABE, Toshiyuki KONDO and Yoshiki UCHIKAWA

Dept. of Information Electronics, School of Eng., Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01, Japan
Phone: +81-52-789-3167, Fax: +81-52-789-3166
E-mail: {ishiguro/ yuji/ kon/ uchikawa}@bioele.nuee.nagoya-u.ac.jp

Abstract

In the field of recent artificial intelligent (AI) technology, much attention has been focused on reactive planning systems (e.g. behavior-based AI, emergent computation, and so on). However, the arbitration between competence modules (simple behavior/action) in these approaches encounters its difficulty. On the other hand, according to the recent studies in the biological fields, it has been clarified that the immune system works as self-preservation system against dynamically changing environments through the interaction among lymphocytes/antibodies. Therefore, the immune system would be expected to provide a new methodology suitable for dynamic problems, for example, robotics.

In this paper, we propose a new, decentralized consensus-making system inspired from the biological immune system. And we apply our proposed method to action arbitration for an autonomous mobile robot as a practical example. We furthermore try to evolve the proposed artificial immune system by adjusting affinities among antibodies through genetic process. To confirm the validity of our method, we carry out some simulations.

***Key words**: immune network, decentralized consensus-making network, autonomous mobile robot, action arbitration, bottom-up manner*

1. Introduction

In the field of artificial intelligent (AI) technology, conventional approaches based on a functional decomposition, leading to a so-called "*sense-model-plan-act*" cycle have been criticized on many drawbacks over the last decade. Typical criticisms are that these systems show brittleness under environmental changes, and required much computation time for mapping obtained sensory inputs onto complex internal models before action can be taken. Therefore, in recent year much attention has been focused on reactive planning systems (e.g. behavior-based AI, new AI, emergent computation, animat approach, and so on), which have already been demonstrated robustness and flexibility against dynamically changing world. In these approaches, intelligence is regarded to emerge through a dynamic and/or instantaneous process resulting from mutual interactions among competence modules (simple behavior/action) [4][12][13]. However, the arbitration between competence modules in these approaches encounters its difficulty.

On the other hand, immune system has various interesting features such as immunological memory, immunological tolerance, pattern recognition, and so on viewed from the engineering standpoint. In addition to the above, recent studies on immunology have clarified that the immune system does not just detect and eliminate the not-self materials called *antigen* such as virus, cancer cells and so on, rather plays important roles to maintain its own system against dynamically changing environments through the interaction among *lymphocytes/antibodies*. Therefore, immune system would be expected to provide a new methodology suitable for dynamic problem dealing with unknown/hostile environments rather than static problem.

Based on the above facts, we have been trying to engineer methods of the immune system, and apply to robotics and so on[1]-[3]. We expect that there would be an interesting AI technique suitable for dynamically changing environment by imitating the immune system in living organisms. In this paper, we propose a new, decentralized inference/consensus-making system inspired from the biological immune system. To confirm the feasibility our proposed method, we apply to action arbitration for an autonomous mobile robots as a practical example. Moreover, we try to evolve the proposed artificial immune system equipped with the mobile robot by adjusting affinities among antibodies through genetic process. Finally, the validity of our proposed methods are confirmed by carrying out some simulations.

2. Overview of biological immune system

The basic components of the immune system are lymphocytes that are mainly classified into two types, namely *B-lymphocytes* and *T-lymphocytes*. B-lymphocytes are the cells produced from *bone marrow*. Roughly 10^7 distinct types of B-lymphocytes are contained in a human body, each of which has distinct chemical structure and produces "Y" shaped *antibodies* from its surfaces. The antibody recognizes specific *antigen*, which are the foreign material that invade living creature. In other words, the response can be regarded just as a *key and lock relationship* (see Fig. 1). To cope with continuously changing environment, living systems possess enormous repertoire of antibodies in advance. On the other hand, T-lymphocytes are the cells produced from *thymus*, and they generally perform to regulate the production of antibodies from B-lymphocytes as outside circuits of B-lymphocyte network (idiotypic network) discussed later.

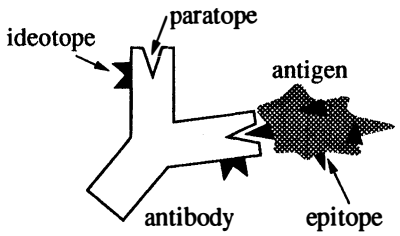


Fig.1: Structure of antigen and antibody.

For the sake of convenience in the following explanation, we furthermore introduce several terminology in immunology. The portion on the antigen recognized by the antibody is called *epitope* (antigen determinant), and the one on the antibody that recognizes the corresponding antigen determinant is called *paratope*. Recent studies on immunology have clarified that each type of antibody has also its specific antigen determinant called *idiotope* (see Fig.1).

Based on this fact, *N.K Jerne*, a Danish immunologist, proposed a remarkable hypothesis which he has called the "*idiotypic network hypothesis*", sometimes called "*immune network hypothesis*"[5]-[9]. This network hypothesis is the concept that antibodies/lymphocytes are not just isolated, namely they are communicating to each other among different species of antibodies/lymphocytes. This idea of *Jerne's* is schematically shown in Fig.2. The idiotope Id1 of antibody 1 stimulates the B-lymphocyte 2, which attaches the antibody 2 to its surface, through the paratope P2. Viewed from the standpoint of antibody 2, the idiotope Id1 of antibody 1 works simultaneously as an antigen. As a result, the B-lymphocytes 1 with antibody 1 are suppressed by antibody 2. On the other hand, antibody 3 stimulates antibody 1 since the idiotope Id3 of antibody 3 works as an antigen viewed from antibody 1. In this way, these stimulation and suppression chains among antibodies form a large-scaled chain loop and works as a self and not-self recognizer. Again, the heart of *Jerne's* idea is that the self-nonself recognition in the immune system is carried out at system level. Therefore, the immune system would provide a new parallel distributed processing.

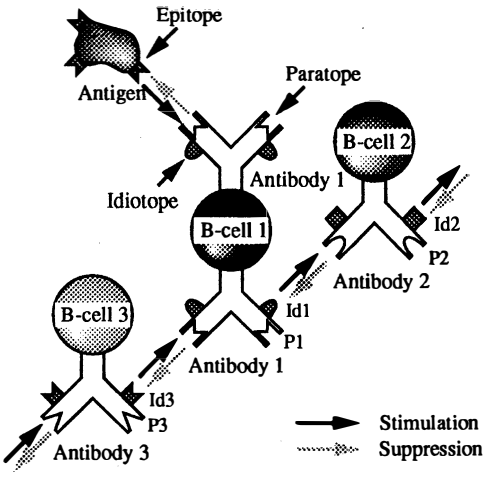


Fig.2: *Jerne's* idiotypic network hypothesis.

3. Action selection and Immune system

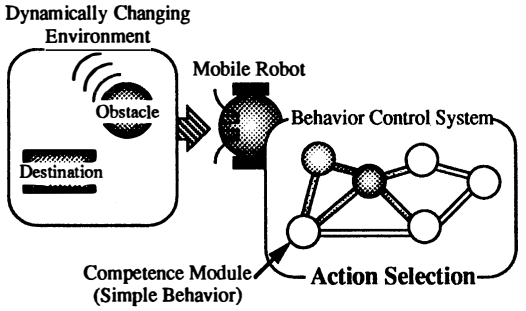
Recently, fatal limits of conventional artificial intelligence approaches based on the functional decomposition have been pointed out. Therefore, much attention has focused on the behavioral decomposition approaches. These behavior-based AI techniques demonstrate their robustness and flexibility against hostile environments, however, arbitration among competence modules (simple behavior/action) arises difficult problems. Namely, it is difficult to select a competence module suitable for the current situation and goals. To overcome this problem, several methods have been proposed. Among them, *Maes* proposed an interesting method; *behavior network system*, under which an action suitable for the current situation and the given goals emerges as the result of the interaction among competence modules[10][11].

Against the above-mentioned stream of works, we

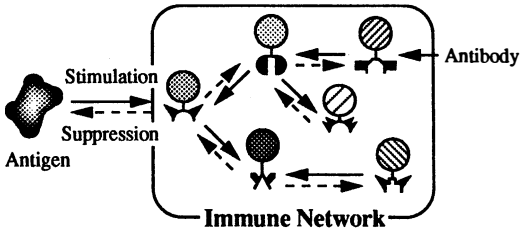
approach to this problem from the immunological standpoint, namely using immune network. Fig.3 schematically shows the action selection system for autonomous mobile robot and the immune network system.

As shown in this figure, current situations, for example, distance and direction to the obstacle detected by installed sensors work as antigens, and a prepared competence module can be regarded as an antibody (or B-lymphocyte), while interaction between modules is represented by stimulation and suppression between antibodies.

The basic concept of our method is that the immune system equipped with the autonomous mobile robot selects the competence module (antibody) which is suitable for the current situation (antigen).



(a) An autonomous mobile robot with an action selection mechanism.



(b) Immune network.

Fig.3: Basic concept of our proposed method.

4. Proposed consensus-making network based on biological immune system

4.1 Problems

For convenience, we dub the autonomous mobile robot with the artificial immune system "*immunobot*". To verify the ability of our proposed immune system against hostile environments quantitatively, in this study we use simulated environment for the immunobot as shown in Fig.4.

In this simulated environment, there are following three kinds of objects: 1) *predators*, 2) *obstacles* and 3) *foods*. And we assume that prespecified quantity of initial energy is given to the immunobot at the beginning of each simulation, and energy level zero means the death of immunobot. For quantitative evaluation we use the following assumption:

(1) if the immunobot moves, it consumes energy E_m.
(2) if the immunobot is captured by a predator, it consumes energy E_p.
(3) if the immunobot collides with an obstacle, it loses energy E_o.
(4) if the immunobot picks up a food, it obtains energy E_f.
(5) Obtaining food behavior does not emerge if the energy level of the immunobot is high.(this is for preventing over-charging)

The predators attack the immunobot if they detect the immunobot within the prespecified detectable range. Fig.4 also indicates the structure of the immunobot used in the simulated environment. We equipped this immunobot with external and internal detectors. External detectors are installed in eight direction, and each can detect the existence of predators, obstacles and foods. And we assume that each detector can also detect the distance to the objects in three degrees, so to speak: *near*, *mid* and *far*. On the other hand, internal sensor detects the energy level. For simplicity, we assume that the immunobot can move toward the above eight directions. The aim of immunobot is to survive as long as possible. To realize this aim, the immunobot must select an action (antibody) suitable for the current detected situation (antigen).

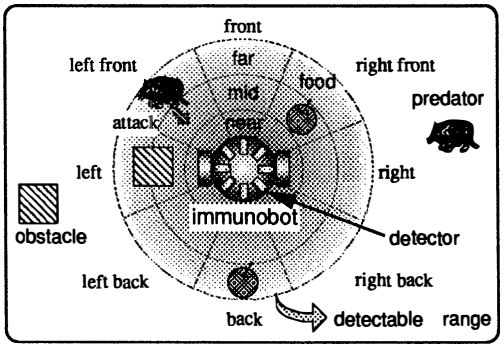


Fig.4: Simulated environment.

4.2 Definition of antibodies

As described earlier, the detected current situation

and prepared competence module work as antigen and antibody, respectively. To make the immunobot select a suitable antibody against the current antigen, we must look carefully into the definition of the antibodies. Moreover, we should notice that our immunological arbitration mechanism selects an antibody in a bottom-up manner by communicating among antibodies. To realize the above requirements, we defined the description of antibodies as follows. As mentioned in section 2, the identity of a specific antibody is generally determined by the structure (e.g. shapes) of its paratope and idiotope. Fig.5 depicts our proposed definition of antibodies.

As in the figure, we assign desirable condition to the paratope, and disallowed conditions to the idiotope, respectively. In addition, we divide the structure of paratope and idiotope into three potions: *objects*, *direction* and *distance*.

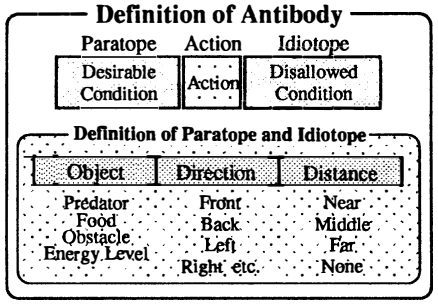


Fig.5: Definition of antibodies.

For the ease of understanding, an example of the prepared antibody is listed in Fig.6. This antibody is activated if the immunobot detects the food in the front direction and mid range, and makes the immunobot move forward to pick it up. However, if a predator exists in front and near/mid range, or if a food exists in near range, this antibody might hesitate to be activated. Another antibodies are designed in the same way.

Note that typical inference systems (e.g. fuzzy inference) adopt a *condition-action* description framework. Against these conventional systems, our proposed system uses *condition-action-condition* fashion. This description method provides decentralized dynamic inference in a bottom-up manner.

Paratope			Action	Idiotope		
Food	Front	Mid.	Move Forward	Predator	Front	Near, Mid.
				Food	#	Near

Fig.6: An example of the prepared antibody.

4.3 Dynamics

For adequate selection of antibodies, we assign one state variable called *concentration* to each antibody. In this study, selection of antibodies is simply carried out in a *winner-take-all* fashion. Namely, only one antibody is allowed to activate and act its corresponding behavior to the world if its concentration surpasses the prespecified threshold. The concentration of i-th antibody, which is denoted by a_i, is calculated as follows:

$$\frac{da_i}{dt} = \left(\sum_j^N m_{j,i} \cdot a_j - \sum_k^N m_{i,k} \cdot a_k + m_i - k_i \right) \cdot a_i \qquad (1)$$

Where N denotes the number of antibodies, and $m_{j,i}$ and m_i denote matching ratios between antibody j and antibody i and between antibody i and antigen, respectively. The first and second terms of the righthand side denote the stimulation and suppression from other antibodies, respectively. The third term represents the stimulation from the antigen, and the forth term the natural death (see Fig.7).

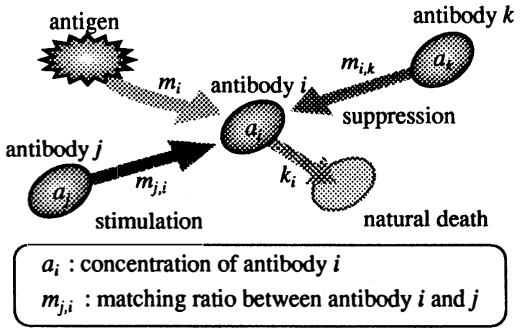


Fig.7: Dynamics.

4.4 Basic mechanism of the proposed consensus-making networks

Next, we explain the fundamental mechanism of our immunological consensus-making networks in detail.

For the ease of understanding, we assume that the immunobot is placed in the situation shown in Fig.8 as an example. In this situation, four antigens listed in the same figure invade into the immunobot's interior, and the listed five antibodies mainly participate in the consensus-making. For example, antibody 1 means that if the immunobot detects food within far range in front direction, it is permitted to move forward; However, if it recognizes other food within nearer range/ predator in front direction/ current energy level is high, this antibody would give way to other antibodies, whose paratopes

represent such situations.

Suppose that the current energy level is high. In this situation, antibodies 1, 2, 3 and 5 are stimulated by the antigen. Note that each of the above antibodies increases its concentration just according to its related antigen. However, to select an appropriate antibody, communication among the antibodies is indispensable. In the figure, the interactions between the antibodies are indicated by arrows. And the solid arrow represents stronger interaction than the shade one. Through these interactions, concentration of each antibody varies. Consequently, antibody 5 will have the highest concentration, and then permitted to be activated.

In the case the robot's current energy level is low, antibody 3 tends to be selected in the same way.

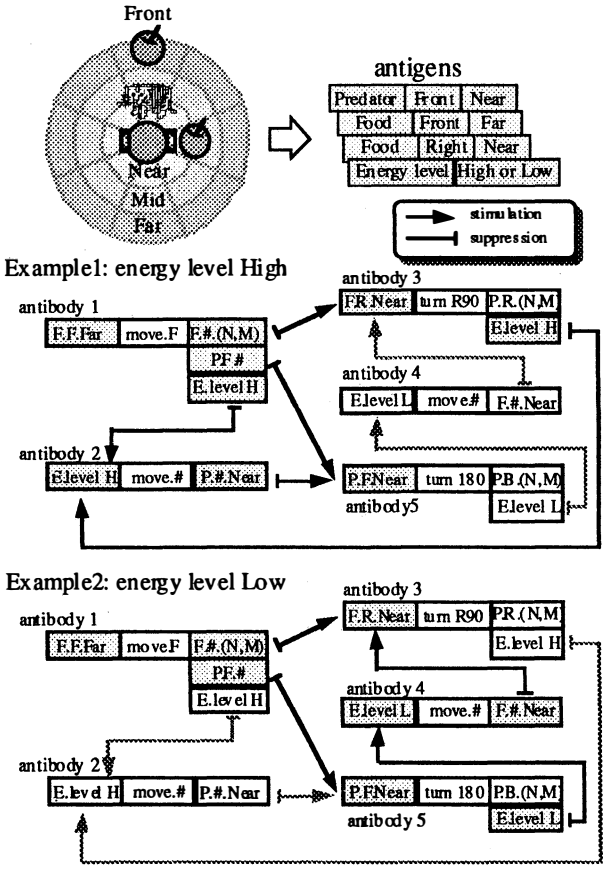


Fig.8: Examples of consensus-making networks by communicating among antibodies.

5. Evolution of affinities among antibodies

As described before, idiotopes denote disallowed conditions. However, we should notice that there should be various disallowance among idiotopes. For the appropriate selection of antibodies, degree of disallowance is a significant parameter, since this parameter determines the *affinity* (i.e. strength) between antibodies (see Fig.9). In spite of this, it is very difficult to determine the magnitude of this parameter *a priori*. Therefore, in this study, we tried to adjust the affinities among antibodies using genetic algorithms. In the evolution process, we used the sum of resultant life time and remaining energy level of the immunobot as fitness function:

$$fitness = life\ time + remaining\ energy\ level \quad (2)$$

Simulation conditions are listed in Table 1. Fig.10 represents the simulation results when we applied the immune networks before and after evolution to 10 different randomly-arranged unknown environments and recorded survival ratio. From the figure, it is understood that the immunobot obtained more appropriate consensus-making system through the genetic process. Fig.11 shows the another comparison of before and after evolution. Obviously, the result obtained after evolution is superior to that obtained before evolution in the area of the number of times captured by predators, the number of times obtaining food, and the number of collisions.

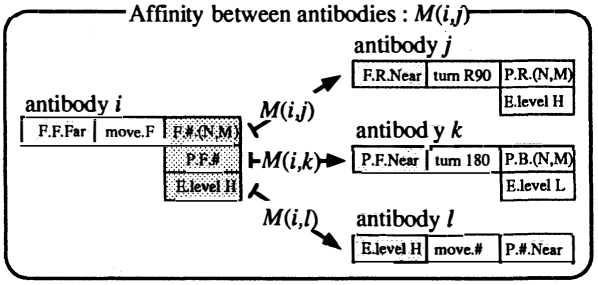


Fig.9: Affinities among antibodies.

Table 1: Simulation conditions.

population	20
crossover	uniform crossover
crossover rate	5-50 %
mutation rate	20 %
maximum life time	1000 steps
magnitude of affinity	0~1.0
# of antibodies	91
# of predators	5
# of obstacles	10
# of foods	10

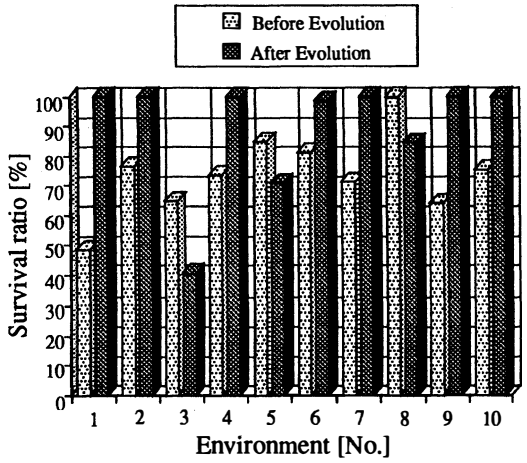


Fig.10: Simulation results (survival ratio).

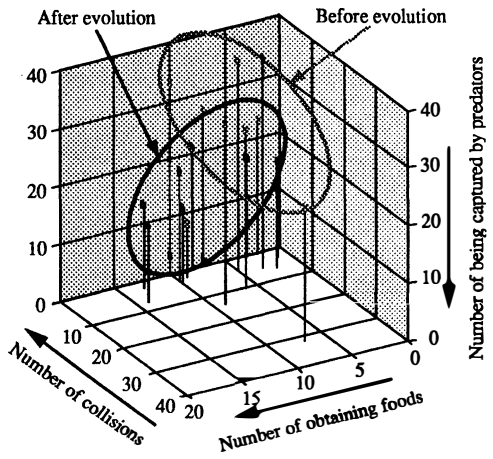


Fig.11: Quantitative comparison between before and after evolutionary processes.

6. Conclusions and Further Work

In this paper, we proposed a new decentralized inference/consensus-making network based on the biological immune system and confirmed the usefulness of our proposed system by applying to a action arbitration for autonomous mobile robots under hostile environments. And we determined pseudo-optimal values of the affinities among prepared antibodies by using genetic algorithms.

Since this study is still in a rudimentary stage of investigation, we designed antibodies *a priori* in a top-down manner. We are now investigating to construct the immune system of the immunobot in a bottom-up manner by introducing *gene recombination* and *meta-dynamics function*. Moreover, we are also trying to endow the immunobot with an on-line learning ability by introducing *T-lymphocytes* and *cytokines* from the engineering point of view.

Acknowledgments

This research was supported in part by a grant from the Japanese Ministry of Education (No. 07243208).

References

[1] A. Ishiguro, Y. Watanabe and Y. Uchikawa, "Fault Diagnosis of Plant Systems Using Immune Networks", *in Proc. of 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 34-42 (1994)

[2] A. Ishiguro, S. Ichikawa and Y. Uchikawa, "A Gait Acquisition of 6-Legged Walking Robot Using Immune Networks", *Journal of Robotics Society of Japan*, Vol. 13, No. 3, pp. 125-128 (1995) (in Japanese), also *in Proc. of IROS '94*, Vol. 2, pp. 1034-1041 (1994)

[3] A. Ishiguro, Y. Watanabe and Y. Uchikawa, "An Immunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots", *in Proc. of IROS '95*, Vol. 1, pp. 495-500 (1995)

[4] D.Cliff, et al.(eds), *From Animal to Animat 3*, MIT Press (1994)

[5] H.Fujita and K.Aihara, "A distributed surveillance and protection system in living organisms", *Trans. on IEE Japan*, Vol. 107-C, No. 11, pp. 1042-1048 (1987) (in Japanese)

[6] J.D.Farmer, N.H.Packard and A.S.Perelson, "The immune system, adaptation, and machine learning", *Physica* 22D, pp. 187-204 (1986)

[7] N.K.Jerne, "The immune system", *Scientific American*, Vol. 229, No. 1, pp. 52-60 (1973)

[8] N.K.Jerne, "The generative grammar of the immune system", *EMBO Journal*, Vol. 4, No. 4 (1985)

[9] N.K.Jerne, "Idiotypic networks and other preconceived ideas", *Immunological Rev.*, Vol. 79, pp. 5-24 (1984)

[10] P.Maes, "The dynamic action selection", *Proc. of IJCAI-89*, pp. 991-997 (1989)

[11] P.Maes, "Situated agent can have goals", *Designing Autonomous Agents*, pp. 49-70, MIT Press (1991)

[12] R.Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of R&A*, Vol. 2, No. 1, pp. 14-23 (1986)

[13] R.Brooks, "Intelligence without reason", *Proc. of IJCAI-91*, pp. 569-595 (1991)

A Method of Gait Coordination of Hexapod Robots Using Immune Networks

A.Ishiguro, S.Kuboshiki, S.Ichikawa and Y.Uchikawa.
Dept. of Information electronics, School of Eng., Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-01

Abstract

Biological information processing systems can be regarded as one of the ultimate decentralized systems, and have been expected to provide various fruitful ideas to engineering fields. Among these systems, the immune system plays an important role to cope with dynamically changing environments by constructing self-nonself recognition networks among different species of antibodies, and has a lot of interesting features such as learning, self-organizing abilities and so on, viewed from engineering standpoints. However, it has not yet been applied to engineering fields so far.

Therefore in this study, we pay close attention to the immune system. We propose a new interpretation of roles of antibodies in terms of self-assertion and subordination and apply this idea to a gait coordination problem of a hexapod robot as a practical example.

Key words: idiotypic network, self-assertion, subordination, genetic algorithms, hexapod robot

1 Introduction

Recently, systems have become increasingly complex and large scale. As a result, conventional, centralized control methodology has reached its limit. To overcome this problem, it has become necessary to turn to developing decentralized processing systems.

In a sense, biological systems in living organisms can be regarded as the ultimate in distributed information processing systems, and are expected to provide various feasible ideas to engineering fields. These systems can be mainly classified into the following four subsystems: 1) brain-nervous system, 2) genetic system, 3) endocrine system, and 4) immune system. Among these systems, brain-nervous and genetic systems have already been applied to engineering fields by modeling as neural networks and genetic algorithms, respectively, and they have been widely used in various fields. The other two systems, however, have not been applied to engineering fields notwithstanding their important roles, viewed from engineering standpoints.

Among these two systems, the immune system works to protect living organisms and cope with dynamically changing environments against countless unknown materials such as virus and so on. This ability is just what we need for autonomous mobile robots. Therefore, we expect that the immune system would provide a novel information processing methodology to engineering fields, especially robotics. Based on the above facts, we have been trying to construct engineering models of the immune system, and apply them to robotics and so forth [1]-[3].

In this paper, we propose a description method of antibodies based on the interpretation in terms of self-assertion and subordination and apply this idea to a gait coordination of hexapod walking robots as a practical example. We take a quite different way from conventional methods by paying close attention to reactions between antibodies to realize a successful walking patterns. To confirm the validity of our proposed method, some simulations are carried out.

2 Immune networks

2.1 Structure of the immune system

The main task of the immune system is to detect and eliminate the non-self materials called the "antigens" such as virus and cancer cells and so on which come from inside and outside of the living organisms. The basic components of the immune system are lymphocytes and these are mainly classified into two types, called B-lymphocytes and T-lymphocytes (Fig.1).

B-lymphocytes are the cells produced from the bone marrow. Each type of lymphocyte produces "Y" shaped protein called the "antibodies" from its surfaces if specific antigens invade.

On the other hand, T-lymphocytes are the cells produced from the thymus, and are classified into three different types; suppresser T-lymphocytes, helper T-lymphocytes and killer T-lymphocytes. Among these T-lymphocytes, suppresser and helper T-lymphocytes perform to regulate the production of antibodies from B-lymphocytes.

The antibody produced from the B-lymphocyte has a portion called the "paratope" which recognizes the corresponding antigen determinant called the "epitope". The paratope reacts to the epitope in a way similar to a lock and key relationship, as a result of the interaction, the antigen is neutralized.

2.2 Idiotypic network hypothesis

Recent studies on immunology have clarified that each type of antibody also has its own specific antigen determinant called the "idiotope" (Fig.1). Based on this fact, N.K.Jerne, who is an immunologist, proposed a remarkable hypothesis called the "idiotypic network hypothesis"[4]-[5]. This network hypothesis, which is fundamentally based on clonal selection theory[6],

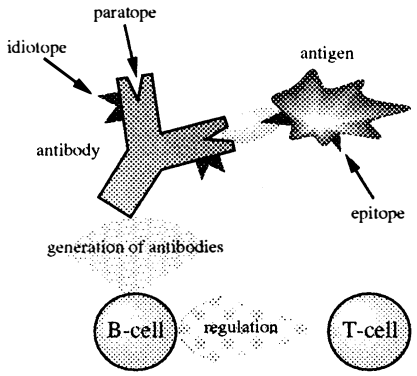


Fig.1: Structure of immune system.

employs the concept that antibodies are not simply isolated, rather, they communicate with each other through their paratopes and idiotopes. This idea of Jerne's is schematically shown in Fig.2. The idiotope Idi of antibody i (Abi) stimulates the B-lymphocyte i-1, which attaches the antibody i-1 (Abi-1) to its surface, through the paratope Pi-1. Viewed from the standpoint of Abi-1, the idiotope Idi of Abi works simultaneously as an antigen. As a result, the B-lymphocytes i with Abi are suppressed by Abi-1. On the other hand, antibody i+1 (Abi+1) stimulates Abi since the idiotope Idi+1 of Abi+1 works as an antigen for Abi. In this way, these stimulation and suppression chains among antibodies form the large closed chain loop which works as a self and non-self recognizer. The heart of Jerne's idea is that the self-nonself recognition in the immune system is carried out at system level.

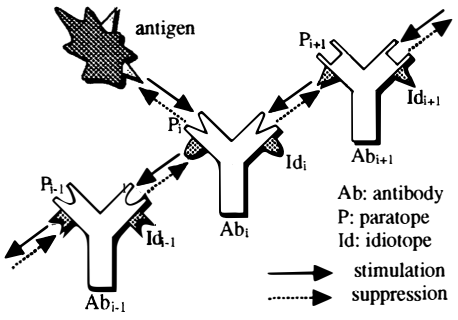


Fig.2: Idiotypic network hypothesis.

2.3 Interpretation of roles of paratope and idiotope

As described above, antibodies interact with each other through their paratopes and idiotopes. Now, suppose that the idiotope of antibody i and the paratope of antibody j are interacting. In this case, the concentration of antibody i will be decreased due to suppression from the paratope of antibody j, whereas the concentration of antibody j will be increased as a consequence of stimulation from the idiotope of antibody i. Looking at the interaction in a different way, we can say that the concentration of an antibody is increased by own paratope and decreased by own idiotope. Regarding the concentration as the degree of self-assertion, an antibody asserts itself by own paratope and subordinates to others by own idiotope. That is, an antibody has two complementary, in other words Janus-faced, parts.

In this method, we attempt to apply this idea to a gait control problem.

3. Gait control for a multi-legged walking robot

Recently, multi-legged walking robots have been expected to expand working fields of robots since they can walk on rough terrain such as another planet and so on. Therefore, much attention has been paid to development of their control methods. In unstructured environments, we must consider the situation that legs of the walking robot are sometimes disturbed. As a result, the robot often can not walk forward and/or fall down. To avoid such situations, techniques that automatically maintain adequate leg coordination are highly required. Note that this problem also provides one of good examples of self-organizing problems, since each leg must coordinate its movement adequately to maintain reliable locomotion using information about the states of other legs.

So far various control methods for gait control of multi-legged walking robots have been proposed [7]-[13]. In contrast to the aforementioned stream of works, we utilize the dynamics of immune networks to obtain reliable gait patterns. As a rudimentary stage of investigation, we take a hexapod walking robot as a practical example (see Fig.3), and assume that this robot walks maintaining static stability (i.e. duty ratio > 0.5).

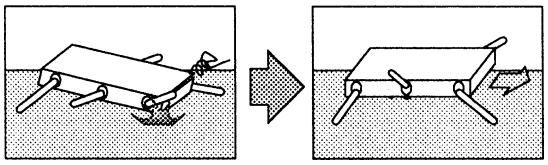


Fig.3: Hexapod walking robot.

4 Immunological approach to a gait coordination problem

4.1 Description of antibodies

Based on the interpretation described in section 2.3, we suppose that self-assertion corresponds to the requests of phase change to other legs, while subordination corresponds to the obedience to requests from other legs. Based on these ideas, we consider the structure of antibodies as follows. We assign the state of "own leg" to idiotope and the states of "other legs", which the own leg wants to eliminate, to paratope

(Fig.4). In this study, we assume that the state of a leg can be categorized as either 'support' or 'transfer'. Support means that a leg is on the ground, while transfer signifies off the ground. Explaining our proposed description of antibodies more concretely, the antibody shown in Fig.4 means that R1 support requests L2 transfer and R2 transfer to chang their phases. Each leg produces antibodies, and they form networks through paratopes and idiotopes. We expect that this immune system works to maintain suitable gaits by eliminating the states of legs which disturb successful walking.

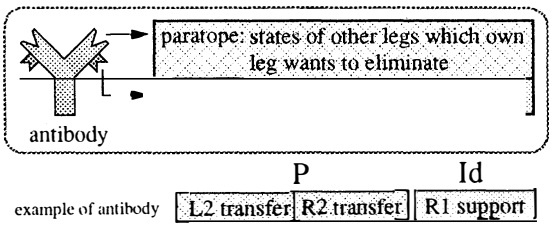


Fig.4: Description of antibody.

4.2 Algorithms

In this study, we assume that B-cells, which exist in each leg, generate antibodies when the robot falls down. The concentration of the generated antibodies decreases as the concerned leg approaches its limit position because we suppose that the leg in such a state cannot make a request to others so strongly. The concentration changes through the stimulation and suppression. The leg state changes on the spot if the concentration of the antibody concerned with the leg becomes lower than the pre-specified threshold level. In addition, other legs move a certain distance in the walking direction.

The concentration of antibodies is determined by a differential equation as follow:

$$\frac{dB_i}{dt} = \left(\sum_j a_{ij} f(B_j) - \sum_j a_{ji} f(B_j) - k \right) f(B_i) + supply_i \quad (1)$$

where B_i denotes the concentration of antibody i. a_{ij} represents the affinity between the paratope of antibody i and the idiotope of antibody j. k and $supply_i$ determine the natural death and supply from the bone marrow, respectively. f means a squashing function such as sigmoid function. In this equation, the first term denotes requests to others and second term is requests from others. The third and fourth terms denote the natural death and the supply from B-cells, respectively.

Summarizing the main point of our idea, each leg has a B-cell and each B-cell generates antibodies when the robot falls down. Antibodies are interacting with each other through networks and the phase change is carried out using the concentration of antibodies. From this, we can see that each B-cell must generate adequate antibodies for successful gait patterns. So, in this study, we determine the description of the paratopes of each antibody using Genetic Algorithms.

4.3 Decision of antibodies using GA

For simplicity, we use the following assumptions for GA-search:

(1)All the affinities between antibodies are the same.

(2)Requests from legs are symmetric about the longitudinal body axis.

(3)No leg makes complementary requests to a same leg.

From these assumptions, we represent the requests by a one-dimensional chromosome of length 30, and assign one of the following values into each locus: 1(request to a stance leg), -1(request to a swing leg), 0(no request). For the evaluation of chromosomes, we use the falling ratio as fitness.

$$fitness = \frac{duration\ of\ falling\ down}{pre\text{-}specified\ test\ duration} \quad (2)$$

5 Simulation results

Fig.5 shows the transition of average and best fitness under the condition of duty ratio 0.6. From this graph, we can see that the chromosome at the 20th generation realizes an adequate gait pattern. Fig.6 shows a set of antibodies determined by decoding the obtained best chromosome. This set of antibodies is an example, and we have confirmed that there are several sets of antibodies to produce a successful gait.

Next, in order to verify the robustness of the obtained immune network, we forcibly fixed the movement of each leg for a certain duration when the robot was walking successfully. Fig.7 illustrates the transition of falling ratio in the case of each leg. In this figure, the disturbance was inputted at the 20th time step. From these results, we can see that the robot acquired a suitable gait again after the input of disturbance.

Fig.8 illustrates the convergence rate to suitable gaits when varying the duty ratio. Note that this immune system was obtained under the condition of duty ratio 0.6. From these results, we can see that the robot can walk successfully even if the duty ratio drops to about 0.55.

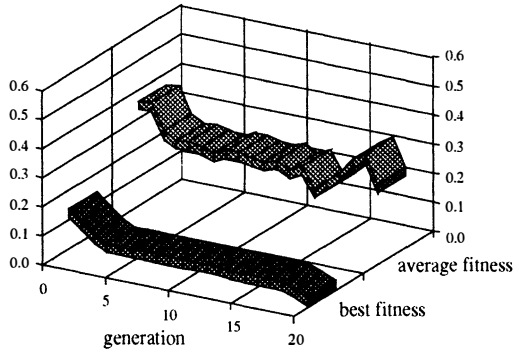


Fig.5: Transition of fitness.

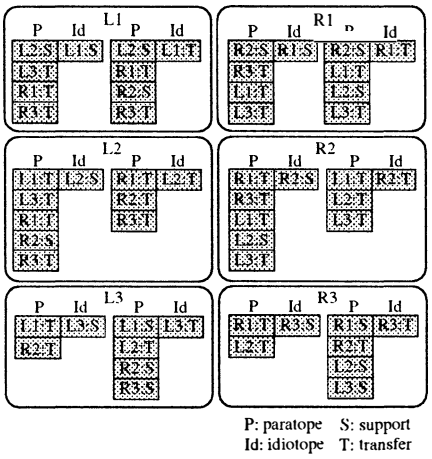


Fig.6: An example of obtained antibodies.

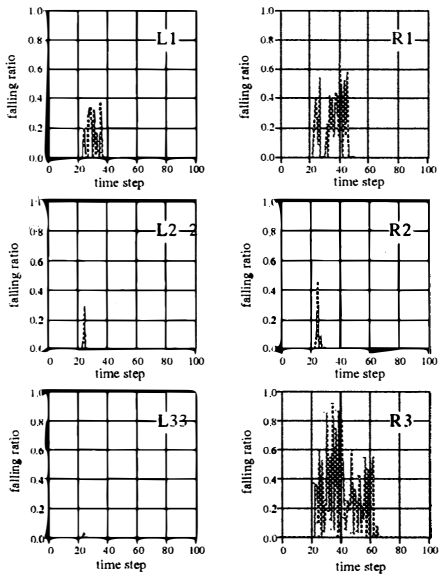


Fig.7: Robustness against disturbance.

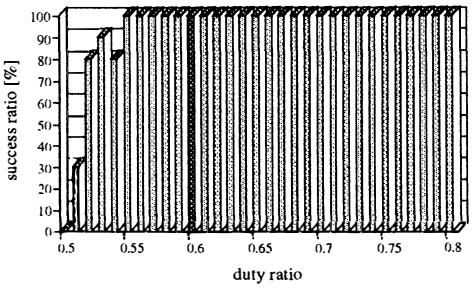


Fig.8: Robustness against the duty ratio.

6 Conclusions and future work

In this study, we have proposed a new description method of antibodies in terms of self-assertion and subordination, based on the fact in immunology. We have applied the proposed method to a gait control problem and confirmed that the robot can acquire a suitable gait and maintain it in spite of the input of disturbance and change of the duty ratio.

In the future, we will construct methods to achieve suitable gaits based on on-line learning, without using genetic algorithms, and attempt when one leg breaks down. In addition, we will experimentally verify the feasibility of our method using a real walking robot.

References

[1]S. Ichikawa, A. Ishiguro and Y. Uchikawa, "A Gait Acquisition of 6-Legged Walking Robot Using Immune Networks", Journal of Robotics Society of Japan, Vol. 13, No. 3, pp. 125-128 (1995) (in Japanese), also in Proc. of IROS '94, Vol. 2, pp. 1034-1041 (1994)

[2]A. Ishiguro, Y. Watanabe and Y. Uchikawa, "Fault Diagnosis of Plant Systems Using Immune Networks", in Proc. of 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 34-42 (1994)

[3]A. Ishiguro, Y. Watanabe and Y. Uchikawa, "An Immunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots", to appear in Proc. IROS '95 (1995)

[4]N. K. Jerne, "The immune system", Scientific American, Vol. 229, No. 1, pp. 52-60 (1973)

[5]N. K. Jerne, "Idiotypic networks and other preconceived ideas", Immunological Rev., Vol. 79, pp. 5-24 (1984)

[6]F. M. Burnet, "The clonal selection theory of acquired immunity", Cambridge University Press (1959)

[7]R. D. Beer, "Intelligence as adaptive behavior: An experiment in computational neuropathology", Academic press (1990)

[8]R. A. Brooks, "A robot that walks: emergent behavior from a carefully evolved network", Neural Computation, Vol. 1, pp. 253-262 (1989)

[9]H. Yuasa and M. Ito, "Coordination of many oscillators and generation of locomotory patterns", Biological Cybernetics, Vol. 63, pp. 177-184(1990)

[10]P. Maes and R. Brooks, "Learning to Coordinate Behaviors", in Proc. of AAAI, pp. 796-802 (1990)

[11]S. Hirose, O. Kunieda, The International Journal of Robotics Research 1991, pp. 3-12 (1991)

[12]M. A. Lewis, A. H. Fagg and G. A. Bekey, "Genetic Algorithms for Gait Synthesis in a Hexapod Robot", in Advanced Mobile Robots - Theory and Applications: International Prospectives, Chapter 11, pp. 317-331, World Publishing (1994)

[13]K. Berns, St. Cordes, W. Ilg, "Adaptive, Neural Control Architecture for the Walking Machine LAURON", in Proc. of IROS '94, Vol. 2, pp. 1172-1177 (1994)

An Immunological Approach to Dynamic Behavior Arbitration for Autonomous Mobile Robots
~ Construction of Immune Networks through Learning ~

A. ISHIGURO, T. KONDO, Y. WATANABE and Y. UCHIKAWA

Dept. of Information Electronics, School of Eng., Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01, Japan
Phone: +81-52-789-3167, Fax: +81-52-789-3166
E-mail: {ishiguro/kon/yuji/uchikawa}@bioele.nuee.nagoya-u.ac.jp

Abstract

Conventional AI system have been criticized for its brittleness in dynamic changing environments. Therefore, in recent years much attention has been focused on the reactive planning systems. In this paper, we propose a new decentralized consensus-making system for the behavior arbitration of an autonomous mobile robots inspired from biological immune system. In addition, we propose a new learning method for reactive planing systems, and try to construct suitable immune networks using reinforcement signals.

Key Words: *Autonomous mobile robots, Idiotipic networks hypothesis, Decentralized consensus-making, Reinforcement learning*

1. Introduction

Conventional AI (artificial intelligence) systems based on a functional decomposition, leading to a so-called "*sense-model-plan-act*" cycle have been criticized on many drawbacks over the last decade. Typical criticisms are that these systems show brittleness under environmental changes, and required much computation time for mapping obtained sensory inputs onto complex internal models before action can be taken. Therefore, in recent years much attention has been focused on *reactive planning systems* (e.g. *behavior-based AI*, *new AI*, *emergent computation* ,and so on), which have already been demonstrated robustness and flexibility against dynamically changing world [1][2].

On the other hand, biological information processing systems such as human beings have many interesting functions and are expected to provide various feasible ideas to engineering fields, especially robotics. Biological information system in living organisms can be mainly classified into the following four systems: (1) *brain-nervous system*, (2) *genetic system*, (3) *endocrine system*, and (4) *immune system*. Among these systems, brain-nervous and genetic systems have already been applied to engineering fields by modeling as *neural networks*[3] and *genetic algorithms*[4], and they have been widely used in various fields. Little attention, however, has been paid to application of the other systems (*i.e.* endocrine and immune systems) notwithstanding their important characteristics.

Immune system, in particular, have various interesting features such as immunological memory, immunological tolerance, pattern recognition, non-hierarchical distributed structure, and so on viewed from the engineering standpoint. In addition to the above, recent studies on immunology have clarified that the immune system does not just detect and eliminate the not-self materials called *antigen* such as virus, cancer cells and so on, which come from inside/outside of the living system, rather plays important roles tomaintainits own system against dynamically changing environments. Therefore, immune system would be expected to provide a new methodology suitable for dynamic problem dealing with unknown/hostile environments rather than static problem.

Based on the above consideration, we have been trying to engineer methods of the immune system, and apply to robotics and so forth[5]-[9]. We expect that there would be an interesting AI technique suitable for dynamically changing environment by imitating the immune system in living organisms. In this paper, we propose a new decentralized consensus-making system inspired from immune system in living organisms. To confirm the feasibility of our proposed method, we apply

to behavior control of an autonomous mobile robots as a practical example. Moreover, we try to evolve the proposed artificial immune system equipped with the mobile robot by adjusting affinities among antibodies using reinforcement signals. Finally, the validity of our proposed methods are confirmed by carrying out simulations.

2. Overview of Immune System

2.1 Structure of Antigen and Antibody

The basic components of the immune system are *lymphocytes* that are mainly classified into two types, namely *B-lymphocytes* and *T-lymphocytes*. B-lymphocytes are the cells produced from *bone marrows*. Roughly 10^7 distinct types of B-lymphocytes are contained in a human body, each of which has distinct chemical structure and produces "Y" shaped *antibodies* from its surfaces (see Fig.1). The antibody specifically recognizes an *antigen* like a *lock and key relationship*. To cope with continuously changing environment, living systems possess enormous repertoire of antibodies in advance. On the other hand, T-lymphocytes are the cells produced from *thymus*, and they generally perform to regulate the production of antibodies from B-lymphocytes as outside circuits of B-lymphocyte network (idiotypic network) discussed later.

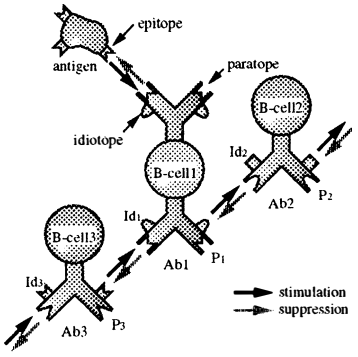


Fig.1 *Jerne's* idiotype network hypothesis.

For the sake of convenience in the following explanation, we furthermore introduce several terminology in immunology. The portion on the antigen recognized by the antibody is called *epitope* (antigen determinant), and the one on the antibody that recognizes the corresponding antigen determinant is called *paratope*. Recent studies on immunology have clarified that each type of antibody has also its specific antigen determinant called *idiotope*.

2.2 Jerne's idiotipic networks hypothesis

Based on this fact, *N.K.Jerne*, who is an immunologist, proposed a remarkable hypothesis: *idiotypic network hypothesis* [12]-[16]. This network hypothesis is the concept that antibodies are not just isolated, namely they are communicating to each other among different species of antibodies. In other word, the immune system is constructed as a large-scaled system of lymphocytes through mutual interaction between different species of lymphocytes. This idea of *Jerne's* is schematically shown in Fig.1. The idiotope Id1 of antibody 1 (Ab1) stimulates the B-lymphocyte 2, which attaches the antibody 2 (Ab2) to its surface, through the paratope P2. Viewed from the standpoint of Ab2, the idiotope Id1 of Ab1 works simultaneously as an antigen. As a result, the B-lymphocytes 1 with Ab1 are suppressed by Ab2. On the other hand, antibody 3 (Ab3) stimulates Ab1 since the idiotope Id3 of Ab3 works as an antigen viewed from Ab1. In this way, these stimulation and suppression chains among antibodies form a large-scaled chain loop and works as a self and not-self recognizer. Again, the heart of *Jerne's* idea is that the self, and not-self recognition in the immune system is carried out at system level.

3. Action selection and Immune system

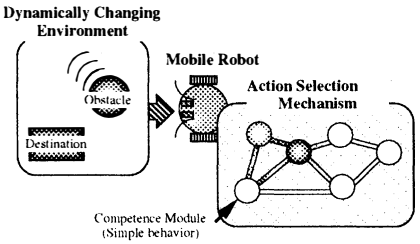
Recently, behavior-based AI techniques demonstrate their robustness and flexibility against hostile environments, however, arbitration among competence modules (simple behavior) arises difficult problems. Namely, it is difficult to select a competence module suitable for the current situation. To overcome this problem, several methods have been proposed.[10][11]

Against the above-mentioned stream of works, we approach to this problem from the immunological standpoint, namely using immune networks. Fig.2 schematically shows the autonomous mobile robots and the immune network system.

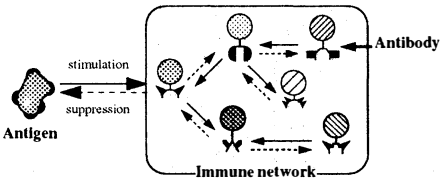
As shown in this figure, current situation, for example,

distance and direction to the detected obstacle work as antigens, and competence module (simple behavior) and interaction between modules can be recognized as antibody and stimulation/suppression among antibodies, respectively.

The basic concept of our method is that the immune system equipped with the autonomous mobile robot selects the competence module (antibody) which is most suitable for the current situation (antigen).



(a) An autonomous mobile robot with action selection mechanism.



(b) Immune networks.

Fig.2: Basic concept of our proposed method.

4. Proposed consensus-making networks based on biological immune system

4.1 Problem

For convenience, we dub the autonomous mobile robot with the artificial immune system "*immunobot*". To verify the ability of our proposed immune system against hostile environments quantitatively, in this study we use simulated environment for the immunobot as shown in Fig.3.

In this simulated environment, there are following three kinds of objects: 1) *predators*, 2) *obstacles* and 3) *food*. And we assume that prespecified quantity of initial energy is given to the immunobot at the beginning of each simulation. For quantitative evaluation we use the following assumption:

(1) if the immunobot moves, it consumes energy E_m.
(2) if the immunobot is captured by a predator, it consumes energy E_p.
(3) if the immunobot collides with an obstacle, it loses energy E_o.
(4) if the immunobot picks up a food, it obtains energy E_f.

The predators attack the immunobot if they detect the immunobot within the prespecified detectable range. Therefore, to survive as long as possible, the immunobot must select a competence module (antibody) suitable for the current detected situation (antigen).

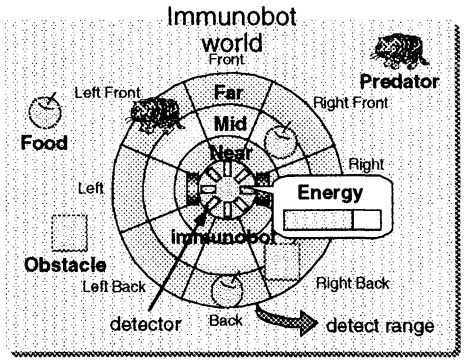


Fig.3: Simulated environment.

Fig.3 also indicates the structure of the immunobot used in the following simulations. We equipped this immunobot with external and internal detectors. External detectors are installed in eight direction shown in Fig.3, and each can detect the distance to the objects in three degrees: *near*, *mid* and *far*. On the other hand, internal sensor detects the current energy level. For simplicity, we assume that the immunobot can move toward the above eight directions in the following simulations.

4.2 Description of antibodies

As described earlier, in this study, the detected current situation and prepared competence modules work as antigens and antibodies, respectively. To make the immunobot select a suitable antibody against the current antigen, it is highly important how we describe the antibodies. Moreover, we should notice that our immunological arbitration mechanism selects an antibody in a bottom-up manner by communicating

among the antibodies. To realize the above requirements, we defined the description of antibodies as follows. As mentioned in section 2, the identity of a specific antibody is generally determined by the structure of its paratope and idiotope. Fig.4 depicts our proposed description of antibodies

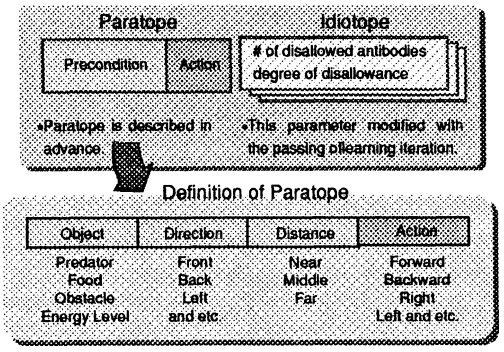


Fig.4: Description of antibodies.

As shown in this figure, we assign a pair of desirable condition and action to paratope, the number of disallowed antibodies and the degree of disallowance to idiotope, respectively. In addition, we divide the structure of paratope into four potions: *objects*, *direction*, *distance*, *and action*.

4.3 Dynamics

For adequate selection of antibodies, we assign one state variable called *concentration* to each antibody. In this study, selection of antibodies is simply carried outin a *winner-take-all* fashion. Namely, only one antibody is allowed to activate and act its corresponding behavior to the world if its concentration surpasses the prespecified threshold. As shown in Fig.5, concentration of the antibody is influenced by the stimulation and suppression from other antibodies, the stimulation from antigen, and the dissipation factor (i.e. *natural death*). The concentration of i-th antibody, which is denoted by a_i, is calculated as follows:

$$\frac{da_i}{dt} = \left(\sum_j^N m_{j,i} \cdot a_j - \sum_k^N m_{i,k} \cdot a_k + m_i - k \right) \cdot a_i . \quad (1)$$

The first and second terms of right hand side denote the stimulation and suppression from other antibodies, respectively. The third term represents the stimulation from antigen, and the forth term the natural death.

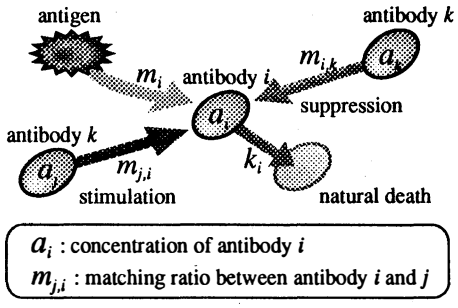


Fig.5: Dynamics.

5. Proposed Learning method

5.1 Learning Procedure

In this section, we attempt to determine the priority among antibodies on an on-linelearningbasis.To realize this aim, we propose a new learning method using the advantages of the prementioneddescription of antibodies as follows.

For the ease of the followingexplanation, we assume that the immunobot is placed as in Fig.6. In this situation, as antigen 1 and 2 invade into the immunobot's interior, antibody 1 (Ab1) and 2 (Ab2) simultaneously are stimulated by each antigen, concentration of each antibody become higher. Note that the notation [P, F, M] represents that a predator exists in the front direction and mid range. However, as the priority between Ab 1 and 2 is unknown (because idiotopes are initially *tabula rasa*, there are no stimulation/suppression chain), in this case, either of them is selected randomly.

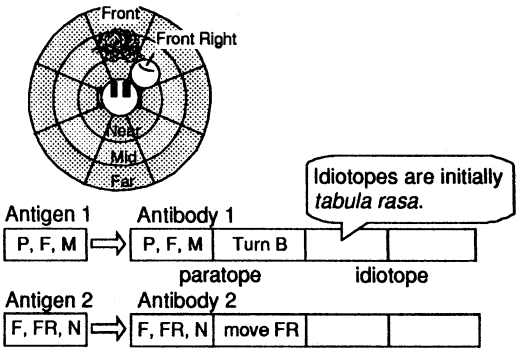


Fig.6: An Example of antibody selection.

Now, assuming that Ab 2 is randomly selected and the immunobot successfully avoids the predator and obtains food, In this case, the immunobot can receive the positive reinforcement signals as a reward. To make the immunobot tend to select Ab 2 in the same/similar situation, we record the number of Ab 2 (i.e. 2) in the idiotope of Ab 1. This procedure works to raise the priority of Ab 2 against Ab 1.
In the case where the immunobot receives the negative reinforcement signal, we record the number of Ab 1 (i.e. 1) in the idiotope of Ab 2 in the same way. This works to decrease the priority of Ab 2 against Ab 1.

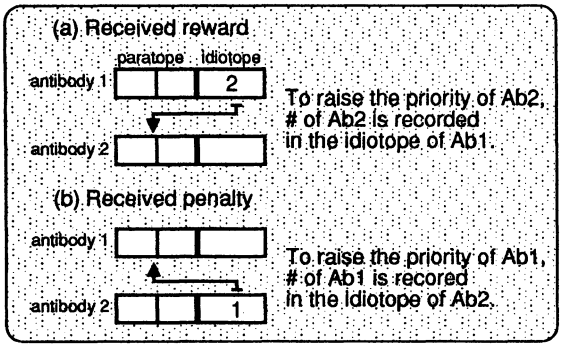


Fig.7: learning process.

For simplicity, in this study, we give the reward/penalty signal to the immunobot in the situation listed in Table 1.

Table 1: Reward/Penalty Criterion.

Reward	Obtain food
	Approach food
	Away from predator/obstacle
Penalty	Captured by predator
	Collide with obstacle
	Approach predator/obstacle

5.2 Simulation Result

To verify the feasibility of our proposed learning method, we carried out same simulations. Fig.8 denotes the transition of life time of the immunobot. From the figure, it is understood that the life time gradually increases.

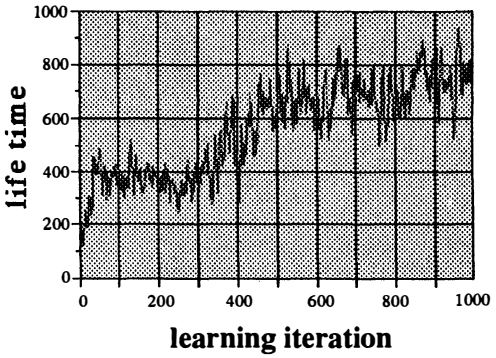


Fig.8: Transition of life time.

Fig.9 represents an example of obtained networks. Detail investigation of the obtained immune network is currently undergoing.

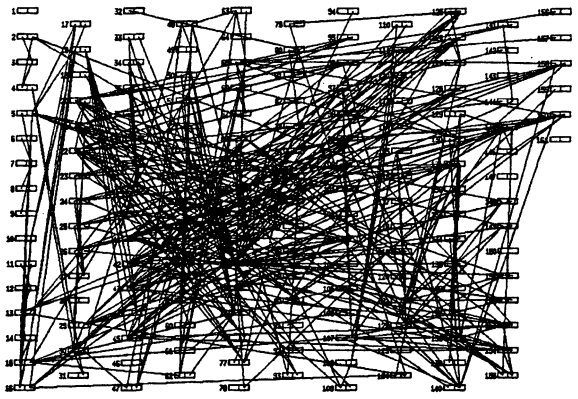


Fig.9: example of obtained networks.

6. Conclusions and Further Work

In this paper, we proposed a new learning method for the decentralized consensus-making network based on the biological immune system. And we applied the proposed method to the behavior arbitration of an autonomous mobile robot in the simulated environment and confirmed the feasibility by carrying out simulations. Through the simulations, we showed that the immune networks can be automatically constructed using reinforcement signals.

To reveal the validity of our proposed method quantitatively, detail analysis of the resultant immune networks is highly important. This is currently under investigation. And we are new planning to apply this ides to a real experimental mobile robot. This project is also

currently undergoing.

Acknowledgments

This research was supported in part by a grant from the Japanese Ministry of Education (No. 07243208).

References

[1] R.Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE Journal of R&A, Vol.2, No.1, pp.14-23, 1986

[2] R.Brooks, "Intelligence without reason", Proc. of IJCAI-91, pp.569-595, 1991

[3] D.E.Rumelhart, et al., "Parallel distributed processing", MITPress, 1986

[4] D.E.Goldberg, "Genetic algorithms in search, optimization and machine learning", Addison-Wesley, 1989

[5] A. Ishiguro, Y. Watanabe and Y. Uchikawa, "Fault Diagnosis of Plant Systems Using Immune Networks", in Proc. of 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 34-42, 1994

[6] S. Ichikawa, A. Ishiguro and Y. Uchikawa, "A Gait Acquisition of 6-Legged Walking Robot Using Immune Networks", Journal of Robotics Society of Japan, Vol. 13, No. 3, pp. 125-128, 1995 (in Japanese), also in Proc. of IROS '94, Vol. 2, pp. 1034-1041, 1994

[7] A. Ishiguro, S. Kuboshiki, S. Ichikawa and Y. Uchikawa, "Gait Coordination of Hexapod Walking Robots Using Mutual-Coupled Immune Networks", Proc. of ICEC'95, Vol. 2, pp. 672-677, 1995

[8] A. Ishiguro, Y. Watanabe and Y. Uchikawa, "An Immunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots", Proc. of IROS '95, Vol.1, pp.495-500, 1995

[9] A. Ishiguro, T. Kondo, Y. Watanabe and Y. Uchikawa, "Dynamic Behavior Arbitration of Autonomous Mobile Robots Using Immune Networks", Proc. of ICEC'95, Vol.2, pp. 722-727, 1995

[10] P. Maes, "The dynamic action selection", Proc. of IJCAI-89, pp.991-997, 1989

[11] P. Maes, "Situated agent can have goals", Designing Autonomous Agents, pp.49-70, MIT Press, 1991

[12] N.K.Jerne, "The immune system", Scientific American, Vol.229, No.1, pp.52-60, 1973

[13] N.K.Jerne, "The generative grammar of the immune system, EMBO Journal, Vol.4, No.4, 1985

[14] N.K.Jerne, "Idiotypic networks and other preconceived ideas", Immunological Rev., Vol.79, pp.5-24, 1984

[15] J.D.Farmer, N.HPackard and A.SPerelson: The immune system, adaptation, and machine learning, *Physica* 22D, pp.187-204, 1986

[16] H.Fujita and K.Aihara, "A distributed surveillance and protection system in living organisms", *Trans. on IEE Japan*, Vol.107-C, No.11, pp.1042-1048, 1987 (in Japanese)

Artificial Evolution and Real Robots

Inman Harvey
School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH, UK

Abstract

Artificial evolution as a design methodology allows the relaxation of many of the constraints that have held back conventional methods. It does not require a complete prior analysis and decomposition of the task to be tackled, as human designers require. However this freedom comes at some cost; there are a whole new set of issues relating to evolution that must be considered. Standard Genetic Algorithms may not be appropriate for incremental evolution of robot controllers. SAGA, Species Adaptation Genetic Algorithms, has been developed to meet these special needs.

The main cost of an evolutionary approach is the large number of trials that are required. Simulations — especially those involving vision in complex environments, or modelling detailed semiconductor physics — may not be adequate or practical.

Examples of evolved robots will be discussed, including a specialised piece of equipment allowed for the testing of a robot using simple vision in real time, and what is believed to be the first successful example of an evolved hardware controller for a robot.

1 Why Evolutionary Robotics?

Humans are naturally evolved creatures, and the selection criteria under which our ancestors were judged did not include the ability to design complex systems —- in fact, we are not very good at it. A common and useful trick to overcome our shortcomings is that of *divide and conquer*: a complex problem is decomposed into separate easier sub-problems.

However, the interactions between such sub-problems must be few in number, so that the human designers can temporarily ignore them while solving one sub-problem at a time. When it comes to designing such complex systems as a cognitive control system for a robot, there are at least three major problems.

- It is not clear *how* a robot control system should be decomposed.
- Interactions between separate sub-systems are not limited to direct connecting links, but also include interactions mediated *via the environment.*
- As system complexity grows, the number of potential interactions between sub-parts grows *exponentially.*

Classical approaches to robotics have often assumed a primary decomposition into Perception, Planning and Action modules. Many people now see this as a basic error. Brooks' subsumption approach [1] is explicitly claimed to be inspired by natural evolution. Initially simple behaviours are 'wired into' a robot, and thoroughly debugged, before adding the next behaviour. This incremental approach echoes the phylogenetic history of complex cognitive creatures, some of whose behaviours we are trying to emulate in robots. Nevertheless, each new layer of behaviour is wired in by hand design; despite the heuristics used to minimise interactions between layers, it seems that unpredictable interactions may become insuperable when the number of layers gets much bigger than 10.

So an obvious alternative approach is to explicitly use evolutionary techniques to incrementally evolve increasingly complex robot control systems, rather than attempt to figure out each evolutionary step by hand design. Unanticipated and elusive interactions between sub-systems, though tricky or perhaps impenetrable for human designers, need not directly bother an evolutionary process where the only benchmark is the behaviour of the whole system.

2 Artificial Evolution for Robots

Genetic Algorithms (GAs) are the most common form of algorithm which uses evolutionary ideas for search, optimisation and machine learning — the fields covered in [2]. However, recently concerns have been voiced to the effect that GAs, when originally proposed by Holland [3], were intended as algorithms for complex *adaptive* systems, and their use for function optimisation is perhaps not best suited to their

strengths. Evolutionary robotics typically needs adaptive improvement techniques, rather than optimisation techniques, and this critical but little-understood distinction needs to be made clear.

Most published GA work, both applications and theoretical analysis, refers to optimisation problems which can be seen as search problems in some high-dimensional search space, of known (usually enormous) size. Each dimension typically corresponds to some parameter which needs to be set, which is coded for on a small section of the genotype, a 'gene'. What such optimisation problems share is the well-defined finite nature of the search space. This allows the choice of a genotype coding, such that a genotype, often binary, of fixed length can encode any potential solution within the space of possibilities. In robotics, a genotype specifies the characteristics of a control system.

The GA works with a population of such genotypes, each of which is evaluated in terms of how good is the potential solution that it encodes. Genotypes which happen to be fitter in the current population (which initially may be generated at random) are preferentially selected to be parents of the next generation. Offspring inherit genetic material from their parents; usually inheriting part of this from each of two parents. A small number of random mutations are applied to the genotypes of the offspring. This cycle of selection, reproduction with inheritance of genetic material, and variation, is repeated over many generations, with the population remaining the same size as old members are replaced by new ones, the offspring of those members demonstrated to be fitter.

A GA optimisation problem has typically been seen as starting with a population of random points effectively spanning and crudely sampling the whole search space. Successive rounds of selection, reproduction and mutation focus the population of sample points towards fitter regions of the space, homing in on an optimum or near-optimal region. One consequence of this approach has been the primary reliance on recombination as the genetic operator, which mixes and matches information from different samples in order to move towards regions of expected higher fitness; mutation is typically relegated to the rôle of a background genetic operator.

However, some domains — including much of evolutionary robotics — do not always fall into this convenient picture of a fixed-dimensional search space. Standard GA theory does not necessarily then apply.

In evolutionary robotics a genotype will specify the control system (possibly more, see below) of a robot which is expected to produce appropriate behaviours when tested in its environment. However the evaluation of fitness is in terms of the robot's *behaviour*; for all except toy problems there is unlikely to be any obvious way to predict in advance the necessary complexity of control system for a given behaviour. Hence it is often appropriate to choose a genetic encoding which allows for, and encodes the characteristics of, a variable number of components. This has the added benefit of making incremental evolution possible: initially simple robots are evolved under a selection criterion based on simple tasks, and then the same robot population is allowed to increase in complexity in response to a gradual and continuing increase in task complexity. Such incremental evolution calls for *GAs as adaptive improvers* rather than *GAs as optimisers.*

3 SAGA

The conceptual framework of SAGA was introduced to deal with the dynamics of a GA when genotype lengths are allowed to increase [4]. It was shown, using concepts of epistasis and fitness landscapes drawn from theoretical biology [5], that progress through such a genotype space will only be feasible through relatively gradual increases in genotype length. A general trend towards increase in length is associated with the evolution of a *species* rather than global search — the population will be largely genetically converged.

Evolutionary search can be thought of as searching around the current focus of a species for neighbouring regions which are fitter (or in the case of neutral drift, not less fit) while being careful not to lose gains that were made in achieving the current *status quo.* The population can be visualised as moving around on a mountainous fitness landscape, where altitude represents fitness, and movements measured in horizontal directions loosely represent movements in genotype space. Selection is a force which tends to move a population up hills, and keep them centred around a local optimum; mutation produces offspring exploring outwards from the current population.

To increase exploration mutation rates should be increased; but if they are too high then the population disperses completely, losing the current local optimum, and the search becomes random. For any given selection pressure, there is a maximum rate of mutation which simultaneously allows the population to retain a hold on its current hill-top, while maximising search along relatively high ridges in the landscape, potentially towards higher peaks [6]. In SAGA, this means that rank-based or tournament selection should be used to maintain a constant selective pressure (the

expected number of offspring of any individual should depend on its current ranking within the population, rather than the ratio of its fitness to the average fitness); and mutation rates should be maintained at a rate of about 1 mutation per genotype [7].

4 What building blocks?

We are relying on evolution for the design of a control system, but we must choose appropriate building blocks for it to work with. There is good reason to believe that the primitives manipulated by the evolutionary process should be at the lowest level possible. Any high level semantic groupings inevitably incorporate the human designer's prejudices. Primitives that are equivalent to a programming language give rise to a rugged fitness landscape with steep precipices. A program taken as a linear string of characters *can* be treated as a genotype, but typically a single mutation in a working program is fatal — a 'precipice'. Genetic programming [8] relies on recombination rather than mutation, but typically relies both on clever, domain-specific, choice of primitives, and on enormous population sizes which are difficult when evaluating robots.

With Brooks [1], we dismiss the classical Perception, Planning, Action decomposition of robot control systems. Instead we see the robot — body, sensors, motors and control system or 'nervous system' — as a dynamical system coupled (via the sensors and motors) with a dynamic environment [9]. This coupled interaction generates the robot behaviour which is to be evaluated. The control system is itself a dynamical system, and hence its genetic specification should be at the level of the primitives of a dynamical system.

One convenient form of dynamical system is an (artificial) neural net (NN). If this takes the form of a feedforward net, from sensors, perhaps via hidden nodes, to motors, then such a control system would have no internal state, and be capable only of generating reactive behaviour. However if a recurrent net is used, with temporal specifications to determine the timescales on which internal feedback is propagated, then non-reactive behaviour is also possible. Dynamic recurrent NNs (DRNNs), with temporal delays on links between nodes, are a class of dynamical systems capable in principle of replicating to an arbitrary degree of accuracy the dynamical behaviour of any other dynamical system with a finite number of components. Such DRNNs are equivalent (only trivial transformations are needed) to Brook's subsumption architectures using Augmented Finite State Machine (AFSMs). The temporal properties can derive either from within each AFSM (Brooks), or through the time-delays on links between them (DRNNs).

One significant difference from subsumption architecture is the deliberate introduction of internal noise at the nodes of DRNNs, with two effects. First, it makes possible new types of feedback dynamics, such as self-bootstrapping feedback loops and oscillator loops, which would not initiate themselves without the noise. Second, it helps to make more smooth the fitness landscape on which the GA is operating; a mutation which deletes a link or a node is comparable to a lot of noise, and hence the change in behaviour due to such a mutation is more closely correlated in the presence of noise than it would otherwise have been.

Thus genotypes need to specify a finite number of nodes, together with their thresholds or other details of a non-linear activation function transforming summed node inputs into node outputs; and links and connections between nodes, specifying weights and time-delays on the links. This can be generalised to include weight-changing rules. A specified subset of the nodes are designated as input nodes, receiving sensory inputs; similarly there is a set of output or motor nodes. Other nodes ('hidden') can be arbitrary in number, and genetically specified links are not necessarily restricted to feedforward ones.

5 Experiments

Using these ideas, a series of experiments were performed at Sussex with the gantry-robot using low bandwidth vision in a noisy real-world domain. A sequence of simple navigational tasks of increasing complexity was presented to the robot, and artifical evolution used to develop the control system and the visual morphology appropriate for success at these tasks. This work is reported elsewhere [10], and is believed to be the first example of artificial evolution for a robot using real vision.

The work used a genetically specified dynamical system as the control system, which is conceptualised as a DRNN, but in practice been implemented on a computer. There is a related approach of evolving control systems directly onto hardware, which has been taken within our group by Thompson [11].

This work is *intrinsic* hardware evolution, in that for each genetically specified piece of hardware, the actual hardware is tested *in situ*; as contrasted with extrinsic hardware evolution, where simulations of the hardware are evaluated during evolution. The actual low-level physics of the hardware can be utilised,

and the realtime dynamics operate at their proper timescales. A human designer usually constrains and controls through clocking such features as switching transients; but by using artificial evolution, such design constraints can be relaxed in an unclocked system.

Thompson used artificial evolution to design a real EHW circuit as an on-board controller for a two-wheeled autonomous mobile robot required to display simple wall-avoidance behaviour in a wide corridor. The robot's only sensors were two sonars mounted on the left and right sides which fire simultaneously five times a second; the output from each sonar to the control system changes when the echo returns. In a conventional system the time of flight would be processed to estimate the range of obstacles, but in Thompson's implementation the pulses are fed directly into the hardware control system, termed a Dynamic State Machine (DSM). This is related to a Finite State Machine, except that each internal signal may (under genetic control) be clocked or unclocked. The global clock frequency, where used, is also under genetic control.

The DSM accepts pulses directly from the sonars, and outputs signals directly (no post-processing) to the motors for the left and right wheels which guide the robot. During evolution a population of DSM specifications (instantiated one at a time on the real hardware) is evaluated at the task of navigating to the centre of the corridor. Success was achieved within 35 generations; for full details see [11]. One successful evolved DSMs was found to be using just 32 bits of RAM and 3 flip-flops, excluding the clock generation. This minimal hardware produced the appropriate sensorimotor coupling between sonar echo signals and motor pulses, to guide the robot in its task. This is believed to be the first ever artificial evolved hardware controller for a robot.

6 Conclusions

Evolutionary robotics allows the relaxation of conventional design constraints, but new theoretical issues need to be studied. The SAGA framework allows for incremental evolution, and robot control systems should be treated as a class of dynamical system. Examples have been given of simple evolved robot behaviours in noisy real world conditions, including the use of evolvable hardware.

Acknowledgements

I acknowledge funding from the E.P.S.R.C., and discussions with Phil Husbands and Adrian Thompson.

References

[1] Brooks, R. A. "A robust layered control system for a mobile robot," *IEEE J. Rob. Autom.*, Vol. 2 pp. 14–23, 1986.

[2] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading MA, 1989.

[3] Holland, J. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, USA, 1975.

[4] Harvey, I. "Species adaptation genetic algorithms: The basis for a continuing SAGA," In Varela, F., Bourgine, P. (eds.), *Toward a Practice of Autonomous Systems*, pp. 346–354. MIT Press/Bradford Books, Cambridge, MA, 1992.

[5] Kauffman, S. "Adaptation on rugged fitness landscapes," In Stein, D. L. (ed.), *Lectures in the Sciences of Complexity*, pp. 527–618. Addison Wesley, 1989.

[6] Eigen, M., McCaskill, J., and Schuster, P., "Molecular quasi-species," *Journal of Physical Chemistry, 92*, pp. 6881–6891, 1988.

[7] Harvey, I. "Evolutionary robotics and SAGA: the case for hill crawling and tournament selection," In Langton, C. (ed.), *Artificial Life III*, pp. 299–326. Addison Wesley, 1993.

[8] Koza, J. R. "Genetic programming," Tech. Report STAN-CS-90-1314, Dept. of Comp. Sc., Stanford University, 1990.

[9] Beer, R. D., and Gallagher, J. C. "Evolving dynamic neural networks for adaptive behavior," *Adaptive Behavior, 1*(1), pp. 91–122, 1992.

[10] I. Harvey, P. Husbands, and D. Cliff. "Seeing the light: Artificial evolution, real vision." In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, (eds.), *From Animals to Animats 3*, pp. 392–401. MIT Press/Bradford Books, Cambridge MA, 1994.

[11] Thompson, A., Harvey, I., and Husbands, P. "Unconstrained evolution and hard consequences," In Sanchez, E., and Tomassini, M. (eds.), *Towards Evolvable Hardware.* Springer-Verlag, 1996.

Design principles of autonomous agents: a case study of classification

Rolf Pfeifer
AI Lab, Computer Science Department
University of Zurich, Winterthurerstrasse 190
CH-8057 Zurich, Switzerland
pfeifer@ifi.unizh.ch

Abstract

Building artificial life-like autonomous agents is still considered an art, rather than a science. A generally accepted precise methodology is missing, and—given the properties of the real world—it is doubtful, whether there will ever be one. Nevertheless, it is possible to define criteria and provide heuristics for good designs. We have developed a number of design principles which, when applied, lead to what we would consider good designs from a cognitive science or ALife perspective. The paper illustrates some of these principles using a case study of classification.

Keywords: autonomous agents, sensory-motor coordination, value-based learning, ecologically balanced designs.

1 Introduction

One of the main research interests in ALife is building artificial creatures. We have been interested in particular types of artificial creatures, namely real-world automous agents. By building artificial autonomous agents, we are trying to understand naturally intelligent agents. We hope that this endeavor will lead to insights about principles of intelligence that include, but are not restricted to, natural systems.

The field of automous agents is still in its infancy. There is no concensus whatsoever on how to build intelligent agents, and a theory is entirely lacking. While some believe that the theory is to be found in the area of non-linear dynamics, others bet on principles from the theory of evolution. The information processing paradigm that held great promise in the early days of artificial intelligence and cognitive science, has been shown to be too limited to explain intelligence. One of its major problems was to deal with behavior in the real world. So, it seems, that we are currently lacking good candidates for a theory of intelligence. But the fact that there is no generally accepted unified framework or theory around does not imply that no relevant knowledge exists. It is only unsystematic and widely distributed in various research communities and individual laboratories. The communities where pertinent knowledge can be found include robotics and engineering, neurobiology, biology, artificial intelligence, psychology, and, of course, ALife (even though the latter does not yet constitute a coherent research community).

As a first step in our understanding of intelligent artificial real-world agents, we have started establishing a set of design principles, i.e., a set of principles stating what we consider to be the important ideas underlying good designs. They represent an initial approximation and are not comprehensive. Moreover, we realize, that some researchers will not agree with them.

2 Design principles: overview

There are three sets of principles. The first is about the types of agents that are of interest. In the field of ALife, researchers study virtual as well as real-world agents. We are interested in the latter kind. The most prominent real-world autonomous agents are robots. Out of the many types of robots, we are particularly interested in autonomous mobile robots, rather than stationary industrial ones.

The first set of design principles states that those agents are of relevance to the study of intelligence that are embodied, autonomous, self-sufficient, and situated. They are of high interest, because they display behaviors that we normally associate with intelligence (for detailed arguments, see, for example, [2], [7]).

The second set of principles concerns the quality of the designs themselves, i.e. the morphology, the architecture and the mechanisms. The main principles in this set that we will discuss are:

(1) Intelligence is to be viewed as emergent from a large set of loosely coupled processes.

(2) The agent should be designed as a learning system whose learning processes are based on its own value system.

(3) The sensory system, the motor system, and the neural substrate should be "ecologically balanced."

Finally, the third set of principles contains design heuristics, strategies, and metaphors. When employed, they should lead to good designs. We will discuss a particularly important principle; it concerns the frame-of-reference problem. Rather than explaining the principles in abstract terms we illustrate them with a case study on categorization, which is inspired by a garbage-collecting robot that has to learn a number of distinctions.

3 Case study: a garbage collecting robot

One of the fundamental abilities of natural living systems, is classification. Food needs to be distinguished from non-food, conspecifics from enemies, the nest from the rest of the environment, and so on. A garbage-collecting robot should learn the distinctions between garbage and non-garbage, various locations, like the incinerator, the sites where garbage is deposited, the charging station, etc. We designed an agent capable of learning some basic distinctions in a particular environment. Let us now discuss the agent design and subsequently summarize the essential points in terms of design principles. For more detail, the reader is referred to [8].

3.1 Basic set-up

The basic set-up is shown in figure 1. There is a wheeled robot, equipped with eight IR sensors, two individually driven motors, and a gripper with two degrees of freedom. There is also a sensor in the "hand" of the gripper for detecting whether there is an object in the "hand". In the environment, there are small, large, and very large pegs. Our agent should pick up the small ones and cary them to a light source. The large ones are too heavy to carry, so it should push them towards the periphery of the arena. The very large ones are too heavy for anything and should be left alone. We want the robot to *learn* these distinctions, because we are specifically interested in learning. Moreover, it should learn the distinctions from its own perspective, without instruction from the experimenter. In this simple case, it might be possible to actually program the agent with these distinctions, but first we want to study learning, and second, as the situations and the types of objects get more complex, pre-programming distinctions will get more and more difficult.

3.2 Perceptual aliasing

If the agent is to learn the distinctions between the various types of pegs, it somehow has to solve the problem of perceptual aliasing. Perceptual aliasing has two forms. On the one hand, the proximal stimulus, i.e., the particular sensory stimulation is ambiguous and may correspond to several situations of the environment. On the other hand, the sensory stimulation from one object varies greatly. It strongly depends on distance and orientation. A large number of states in sensory space correspond to one type of object.

The standard approach is to take an array of sensor readings, e.g., a pixel array, and to perform some processing on it, in order to map it onto an internal model. As we know from computer vision, this has not worked very well. The deeper reason why it does not work well is that the problem is treated as one of information processing which is done "on the input side". A close look at the recent literature (and some not so recent one), suggests that perception is not so much an information processing phenomenon, but rather a matter of *sensory-motor coordination* ([3], [4], [6], [11]).

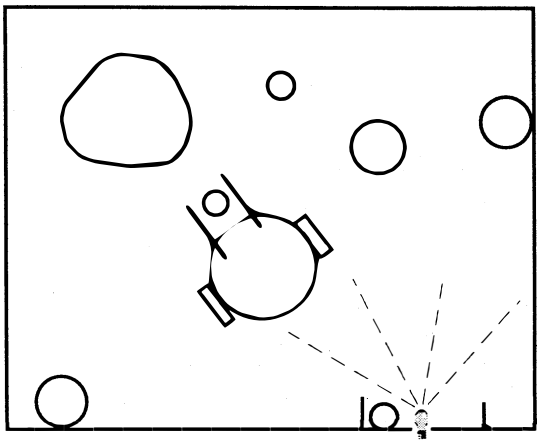


Figure 1: Simplified set-up for the garbage collecting robot. Details, see text.

3.3 Architecture

The architecture that we have used is shown in figure 2. The underlying idea is that behavior—and ultimately intelligence— is, in essence, an emergent property from a large number of parallel processes. There is no set of individual actions from which the agent has to select one for execution, i.e. there is no process of action selection. Rather, there are potentially large numbers of loosely coupled processes which all influence the motor variables. What the agent does is determined by the values of these variables. Because these ideas can be traced back to Braitenberg's seminal book [1], it is called an Extended Braitenberg Architecture (EBA) (for more detail, see [8], [10]).

3.4 Value-based learning and sensory-motor coordination

A situated agent should learn from its own perspective. For this purpose, it needs a value system (e.g. [5], [9],). Figure 3 depicts the process through which the agent learns to form the categories. Value is associated with three types of behaviors, grasping and carrying, pushing, and ignoring. Value is used here to designate what matters to the robot. Take grasping for example. Grasping is intrinsically rewarding which means that whenever the agent manages to pick up a peg, a value signal is generated by the value system. Likewise for pushing. Ignoring is slightly different, but the idea is that avoiding uninteresting objects also has value to the agent.

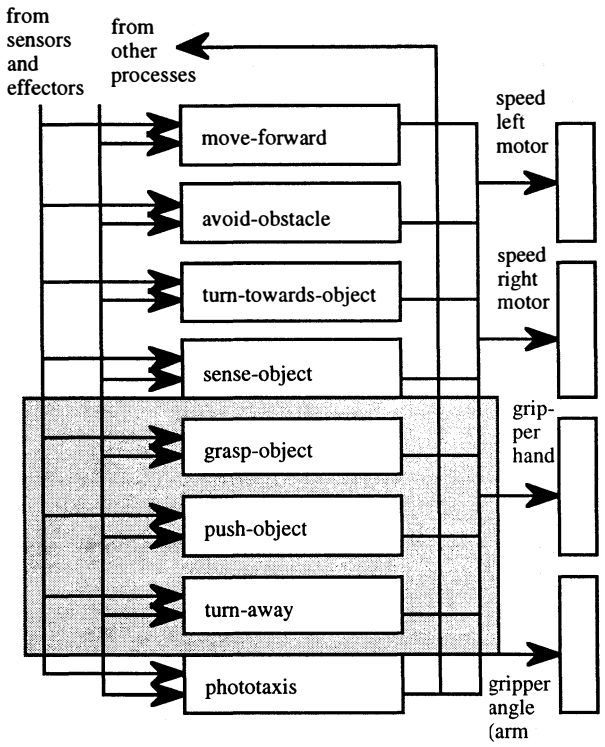


Figure 2: EBA architecture of the garbage-collecting robot. The shaded area is explained in more detail in figure 3.

As is well-known, reinforcement learning is very slow if it is based on random movements. In order to speed up learning, we increase the probability of successful behaviors like grasping or pushing by introducing a number of reflexes. For example there is a reflex for turning away from objects ("avoid-obstacle") and one for turning towards an object ("turn-towards-object"). The interaction of the two causes the agent to move along an object. There is an additional reflex that makes the agent sense the object ("sense-object"). Sensing is done by lowering the gripper over the object, trying to determine if it can be picked up or pushed. This reflex is triggered whenever the agent has been moving along an object for some time. The rationale behind this reflex is that if there is indeed an object, there is a good change that the subsequent behaviors will lead to success. In sum, the purpose of the reflexes is to bias the general-purpose reinforcement scheme.

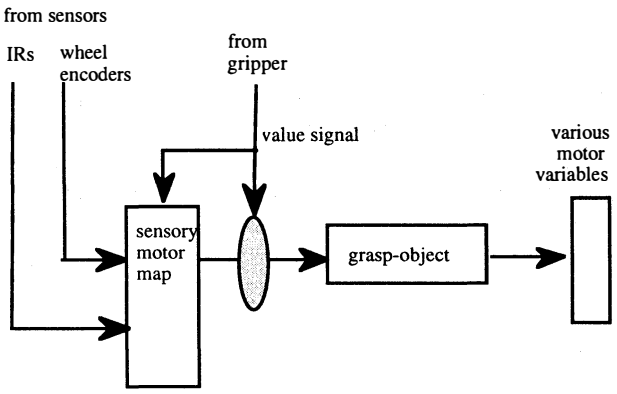


Figure 3: Illustration of the tuning of the various processes using the grasp process.

The following point is central to our argument. Rather than statically taking sensor images and applying algorithms to these images, the agent engages in a sensory-motor coordination which is caused by the reflexes. If the agent encounters an object, it will start moving along it. This in turn causes a correlation of sensor activation in different sensors, namely the wheel encoders and the IR-sensors. This correlation is induced by the agent's movement via the interaction of the agent with its environment. The correlation between the motor speeds (as measured by the wheel encoders) and the IR sensors comes about because, as we have seen, the agent moves around objects. Thus, a lateral IR sensor will be maximally active (whereas the others would typically be low), and the ratio of the values of the wheel encoders is constant (constant angular velocity). In other words, while the agent is engaged in a sensory-motor coordination, the dimensionality of the sensory-motor space is reduced. This makes it much easier for the reinforcement system to learn. In this scheme, learning only takes place, if a value signal is generated. In order to facilitate learning we have used an incremental sensory-motor map which we don't discuss any further, since it is not the main point here.

3.5 Results

Initially the agent has to move along an object for quite some time (about 60 steps of 150ms each) before the "sense-object" process takes over. After a few encounters with the various types of objects, it learns to perform the appropriate behavior after only about 10 to 20 steps without having to make use of the "sense-object" process any longer. We have performed extensive statistical analyses on the data [8]. The algorithms we have used are all incremental, meaning that new objects can be added in at any one time.

4 Design principles: evaluation

Let us now review the case study in terms of our design principles. *Principles concerning the kind of agent:* Our agent it is obviously embodied, and it is situated in that it learns from its own perspective. The purpose of the case study was to illustrate categorization, which is why the current version of the robot is not self-sufficient, and only has a limited degree of autonomy. However, the EBA-architecture allows the incorporation of additional processes, and thus provides the potential for self-sufficiency.

Principles concerning the properties of the agent: Let us mention the most important ones. The EBA-architecture, which consists of parallel processes, implements the design principle that intelligence is to be viewed as emergent from a large number of loosely coupled processes. This principle implies that there must be no central control. A similar point has been made by Brooks [2] in the context of the subsumption architecture.

The distinctions the agent makes over time (as can be seen in the agent´s behavior) between different types of objects are based on a process of sensory-motor coordination. Note that through this process, correlations between the two sensory systems are induced. The reduction in dimensionality achieved in this way makes learning tractable. This instantiates the principle that the agent should (a) engage in a kind of sensory-motor coordination, and (b) the sensory system, the motor system, and the neural substrate should be "ecologically balanced". This is in sharp contrast to most approaches which view perception as a pure information processing problem. This principle of "ecological balance" implies that it is not sensible to merely increase the sophistication of one part of the agent, without increasing the complexity of the others. If that were done, no interesting additional sensory-motor coordinations could take place.

The learning in our agent is value-based, meaning that the agent provides its own reinforcement. This is achieved by the agent´s value system. This kind of learning is a prerequisite for a situated agent. It works well in the agent because the learning problem has been simplified through the sensory-motor coordination.

Principles concerning the design heuristics: Let us just mention one. It states that the so-called "frame-of-reference" issue has to be taken carefully into account. For example, we say that the agent has learned to make distinctions, if—after some time—it consistently picks up small pegs and no longer tries to pick up large ones. Note that this characterization is entirely observer-based and does not resort to internal representations whatsoever. In this perspective, it would be a mistake to define learning in terms of internal representations.

5 Conclusions

We have illustrated a number of design principles for autonomous agents. The idea is that they should capture some of our intuitions and some of the vast but highly distributed knowledge that is around in various fields. Though not (yet) very systematic, we hope that they will lead to a lot of discussion, and provide a starting point for further productive theorizing

Acknowledgments

This research was supported in part by grant nb. 20-40581.94 of the Swiss National Science Foundation.

References

[1] Braitenberg, V. *Vehicles: experiments in synthetic psychology.* Cambridge, Mass.: MIT Press, 1984.

[2] Brooks, R.A. Intelligence without reason. *Proc. IJCAI-91,* pp. 569-595, 1991.

[3] Clancey, W.J. *Situated cognition.* Cambridge: Cambridge University Press (in preparation).

[4] Dewey, J. The reflex arc in psychology. Reprinted in J.J. McDermott (eds.). *The Philosophy of John Dewey.* Chicago, IL: The University of Chicago Press, pp. 136-148, 1981. (originally published 1896).

[5] Edelman, G.E. *Neural Darwinism. The theory of neuronal group selection.* New York: Basic Books, 1987.

[6] Milner, A.D., and Goodale, M.A. *The visual brain in action.* Oxford: Oxford University Press, 1995.

[7] Pfeifer, R. Cognition—perspectives from autonomous agents. *Robotics and Autonomous Systems*, **15**, pp. 47-70, 1995.

[8] Pfeifer, R., and Scheier, C. The metaphor of sensory-motor coordination in autonomous agent design. *Robotics and Autonomous Systems.* (in preparation)

[9] Pfeifer, R., and Verschure, P.F.M.J. Distributed adaptive control: a paradigm for designing autonomous agents. *Proceedings of the First European Conference on Artificial Life.* Cambridge, Mass.: MIT-Press, pp. 21-30, 1992.

[10] Scheier, C. and Pfeifer, R. Classification as sensori-motor coordination. In: *Proc. European Conference on Artificial Life, ECAL-95*, pp. 656-667, 1995.

[11] Thelen, E. and Smith, L. *A dynamic systems approach to the development of cognition and action.* Cambridge, Mass.: MIT Press, Bradford Books, 1994.

Artificial Passive Vision using High Speed Pixel Matching

H. Hanaizumi, T. Tsumashima
College of Engineering,
Hosei University
Koganei, Tokyo 184, JAPAN

S. Fujimura
Graduate School of Engineering,
The University of Tokyo
Bunkyo-ku, Tokyo 113, JAPAN

Abstract

A system and a method for artificial passive vision are proposed to measure shape of small object with high accuracy and high efficiency. For camera calibration, a turn table with some different length rods are used. A spiral matching algorithm is used for high speed matching of stereo pair images.

1 Introduction

We propose a system and a method for an artificial passive vision, which measures the shape of small object and to record it as a digital 3-D image [1]. In the vision, relative orientation, absolute one and matching of all the pixels in stereo pair images are essential. In our system, we use a turn table with some different length rods for both orientations. Stereo pair images are acquired by two TV cameras at a time. We observe the object twice; once for observing rods (normal image) and once for zooming in on the object (zoomed image). We use the former for determining the orientation and the latter for achieving high spatial resolution in observation the object. In matching process, we regard matching as image registration[2].

In this paper, we detail our system and describe the principle and the procedures of the method. Some results of the method applied to the head of a wooden statue are also shown.

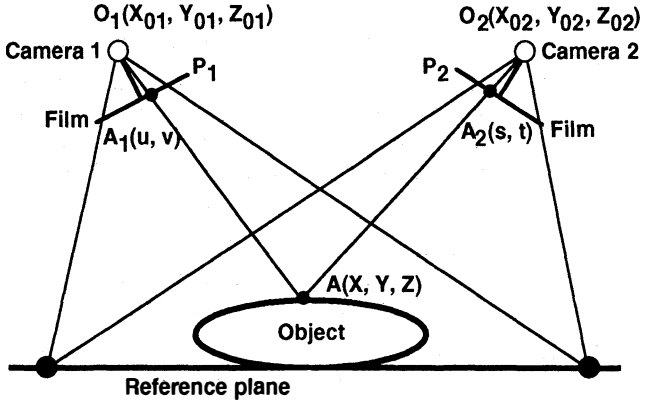


Fig. 1 Geometry of object and cameras.

2 Principle and Procedures

In our vision system, 3-D shape of an object is measured by following procedures : 1) data acquisition for orientation, 2) observation of the object with high spatial resolution, 3) high speed pixel matching with spiral scanning, and 4) reconstruction of the 3-D shape.

2.1 Data Acquisition and Shape Reconstruction

Figure 1 shows the geometry of an object and two cameras for observation. A point A(X, Y, Z) on the object is recorded as points $A_1(u, v)$ on a film P_1 and $A_2(s, t)$ on P_2, respectively, with points on the reference plane. The relations among these points are described by perspective transformations

$$\left.\begin{aligned} u &= \frac{c_1X + c_2Y + c_3Z + c_4}{c_9X + c_{10}Y + c_{11}Z + 1} \\ v &= \frac{c_5X + c_6Y + c_7Z + c_8}{c_9X + c_{10}Y + c_{11}Z + 1} \end{aligned}\right\}, \qquad (1)$$

and

$$\left.\begin{aligned} s &= \frac{d_1X + d_2Y + d_3Z + d_4}{d_9X + d_{10}Y + d_{11}Z + 1} \\ t &= \frac{d_5X + d_6Y + d_7Z + d_8}{d_9X + d_{10}Y + d_{11}Z + 1} \end{aligned}\right\}. \qquad (2)$$

Parameters c_i (i = 1~11) and d_i (i = 1~11) are estimated by using four or more reference points having known coordinates. In our system, the camera calibration is carried out by the determination of these parameters.

The coordinates of the object point A is retrieved from two stereo pair images as the solution of equations

$$\left.\begin{aligned}(c_1 - uc_9)X + (c_2 - uc_{10})Y + (c_3 - uc_{11})Z &= u - c_4\\ (c_5 - vc_9)X + (c_6 - vc_{10})Y + (c_7 - vc_{11})Z &= v - c_8\\ (d_1 - sd_9)X + (d_2 - sd_{10})Y + (d_3 - sd_{11})Z &= s - d_4\\ (d_5 - td_9)X + (d_6 - td_{10})Y + (d_7 - td_{11})Z &= t - d_8\end{aligned}\right\} \quad (3)$$

In order to achieve high spatial resolution, we get zoomed images from the same position, and same value of these parameters estimated above are used for retrieval. In this case, film points $A_1(u^*, v^*)$ and $A_2(s^*, t^*)$ in the zoomed images are described by

$$\left.\begin{aligned}u^* &= \frac{pc_1X + pc_2Y + pc_3Z + pc_4}{c_9X + c_{10}Y + c_{11}Z + 1}\\ v^* &= \frac{pc_5X + pc_6Y + pc_7Z + pc_8}{c_9X + c_{10}Y + c_{11}Z + 1}\end{aligned}\right\}, \quad (4)$$

and

$$\left.\begin{aligned}s^* &= \frac{qd_1X + qd_2Y + qd_3Z + qd_4}{d_9X + d_{10}Y + d_{11}Z + 1}\\ t^* &= \frac{qd_5X + qd_6Y + qd_7Z + qd_8}{d_9X + d_{10}Y + d_{11}Z + 1}\end{aligned}\right\}, \quad (5)$$

respectively, where p and q are zooming ratios.

2.2 Pixel Matching

We regard matching of all the pixels in stereo pair images as image registration. From the registration point of view, stereo pair images have locally large disparities due to parallax between them. In order to reduce processing time with keeping high accuracy, we adopt hierarchical processes : rough registration, pre-matching and matching with spiral scanning.

We refer to one of the stereo pair images as IMG1 and the other as IMG2. At first, we produce new image IMG3 by roughly registering IMG2 onto IMG1 so that we can easily search corresponding point pairs. In the rough registration, the difference of spatial resolution, parallel shift and rotation are removed. Next, we generate many lattice points on IMG1, and search their corresponding points on IMG3. This searching process is based on spatial correlation. We set a square mask on a lattice point in IMG1 and scan the same size mask in a search area in IMG3. We make density scatter diagram for pixels in both masks and assume a linear relation to it. The correlation coefficient is used as the index for searching. The addresses of selected corresponding points in IMG3 are transformed to those in IMG2.

The next is matching of all the pixels. Generally speaking, matching all pixels takes huge time. We adopt spiral scanning to achieve fast matching. The spiral scanning starts at one of the pairs at the lattice point in both IMG1 and IMG2. It proceeds spirally along the edge of the area which have already matched as shown in Fig. 2. We use a checkered mask for reducing the processing time for the correlation processing. Scanning masks spirally along the edge is effective in reduction of size of search area. We use 3 x 2 rectangular area as the search area on the assumption that the object has smooth shape and stereo pair image has less disparity in the line direction. There may occur miss-match during the spiral scanning. When miss-match occurs, we stop the current scanning and restart a new scanning at another pair of lattice point. In order to detect the miss-match, we produce a triangle net model to approximate the disparity between IMG1 and IMG2 as shown in Fig. 3. Matching results for lattice point pairs are used for the production of the rough surface model. We divide IMG1 into many triangles whose vertices are lattice points, and assume that disparity g due to parallax is represented by a linear function of position (x, y) as

$$g = \alpha x + \beta y + \gamma, \quad (6)$$

in a triangle. A miss-match is detected when the difference between disparity obtained by the matching and that by the net model exceeds a threshold. There may be remaining unmatched areas after finishing the spiral scanning for all the pairs of the lattice point. We interpolate these areas by using surface spline [3].

3 Experiments and Discussion

3.1 Measurement system

Figure 4 shows our measurement system. The system

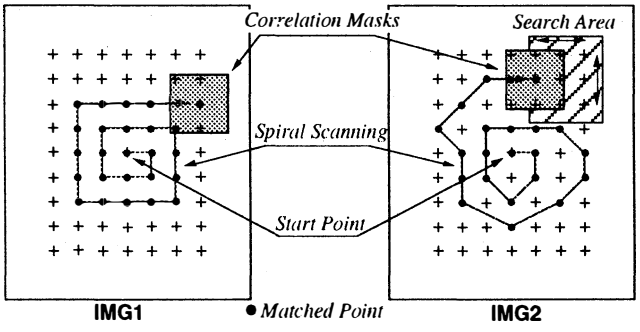


Fig. 2 Spiral scanning of correlation masks.

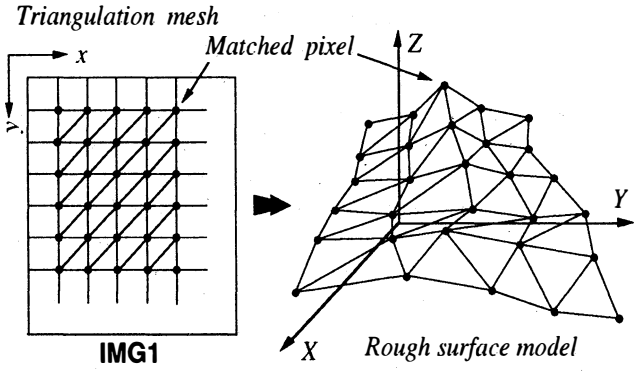


Fig. 3 Triangle net model for miss-match detection.

consists of a turn table (30 cm in diameter) and four kind of rods (25 cm, 20 cm, 15 cm and 10cm in length) with LED on the top. These rods are removable but their positions can be accurately determined. Rotation of the turn table and LED lightening are controlled by a computer. A TV camera is fixed at position about 1 m away from the object with look-down-angle of about 30 degrees, and is used for direct acquisition of image data with size of 640 x 480 pixels. The TV camera can observe the object with several zooming ratios. For each observation, we get rod image without zooming and object image with high zooming ratio. We estimate parameters for the perspective transformation from the rod image, and compensate them by the zooming ratio between rod and object images.

3.2 Experimental Results and Discussion

We measured 3-D shape of a wooden statue to confirm the validity of our method. Normal image and zoomed images were acquired for every 10 degrees of the rotation of the turn table. Figure 5 shows three examples of acquired images : left view (a) and (d), front view (b) and (e), and right view (c) and (f). We refer to images (d), (e) and (f) as IMG_A, IMG_B and IMG_C, respectively, in the following discussion. The zooming ratios were all 3.71. We obtained lattice point pairs between IMG_A and IMG_B as shown in Fig. 6, where space of the lattice was 30 pixels. We applied the spiral scanning algorithm to image pairs (IMG_A, IMG_B) and (IMG_B, IMG_C), respectively. While a conventional pixel-by-pixel matching process took 306 minutes to register them, each spiral scanning process took only 20 minutes (we used HP-735). In the pixel-by-pixel matching process, we used 31 x 31 pixel mask to calculate a correlation coefficient , and used the search area of 60 x 10 pixels.

Figure 7 (a) shows an example for the registration (pixel matching) result of IMG_A onto IMG_B. The result for IMG_B and IMG_C is also obtained. We reconstructed three

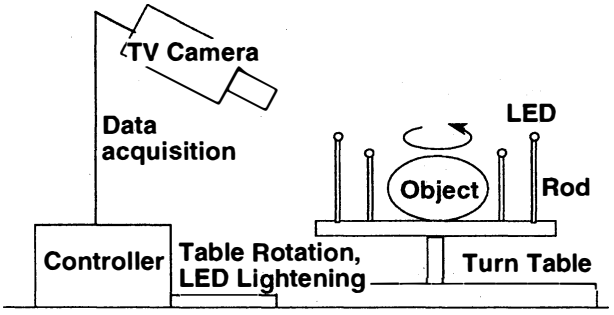


Fig. 4 Measurement system.

shapes of the wooden stature from results for image pairs (IMG_A, IMG_B), (IMG_B, IMG_C) and (IMG_A, IMG_B, IMG_C). For evaluating the performance, we compared distances among 6 test points as shown in Fig.7 (b). Table 1 indicates the evaluation results. From this evaluation, we see that root mean square error (RMSE) of the relative shape reconstruction is about 0.5 mm. Locations of the test point reconstructed from image pair (IMG_A, IMG_B) were different from those from (IMG_B, IMG_C), and these differences were lager than the RMSE above. This is due to insufficiency in estimating parameters for the perspective transformation.

4 Conclusions

We propose a system and a new method for matching all the pixels in stereo pair images are proposed for artificial passive vision. Three dimensional shape of the head of a wooden statue is measured with high spatial resolution. The spiral scanning algorithm in pixel matching achieves high speed processing.

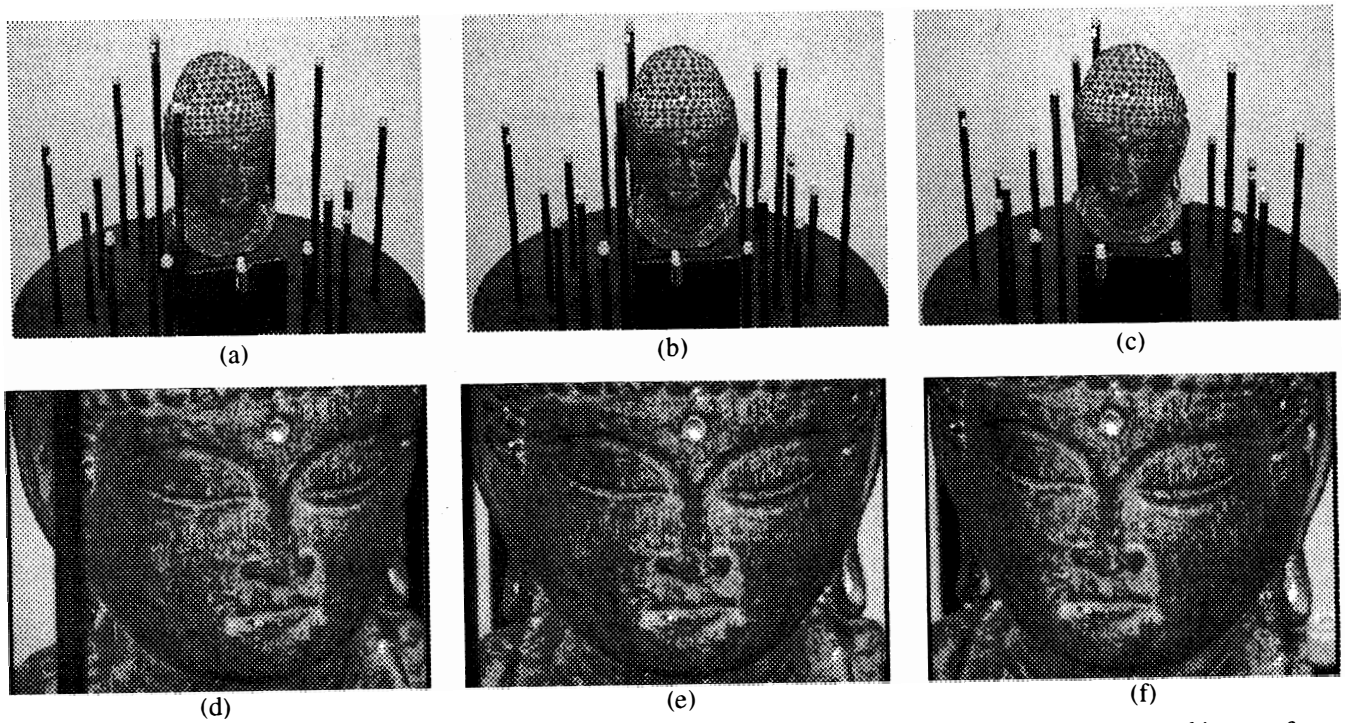
Improvement of accuracy in estimating parameters for perspective transformation and measurement of whole 3-D shape of an object are subjects for a future study.

References

[1] H. Hanaizumi, T. Tsumashima and S. Fujimura, "Application of Proximity Stereometry to Archeology - 3-D Shape Measurement of the Head of a Wooden Stature -", Proc. Close Range Techniques and Machine Vision, Vol. 30, No. 5, pp.150-154, 1994

[2] H. Hanaizumi, R. Takesaki and S. Fujimura, "An Automated Method for Estimating Terrain Height from SPOT Stereo Pair Images", Proc. Int. Geosci. Remote Sensing Symp., pp. 2167-2169, 1995

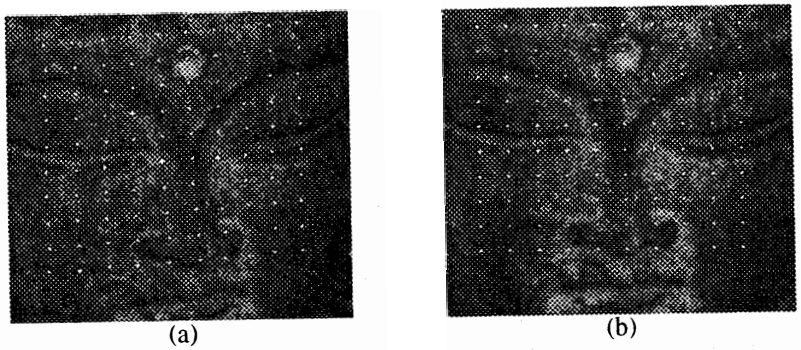
[3] R.L. Harder and R. N. Desmarais, "Interpolation using Surface Splines", J. AIRCRAFT, Vol. 9, No. 2, pp.189-191, 1972



(a) (b) (c)

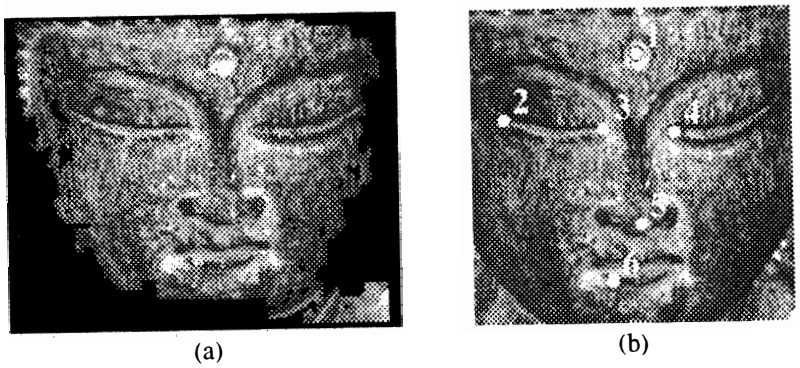
(d) (e) (f)

Fig. 5 Original images : normal images acquired from left (a), front (b), and right (c) directions, and zoomed images from left (d), front (e), and right (f) directions (we call images (d), (e) and (f) as IMG_A, IMG_B and IMG_C, respectively).



(a) (b)

Fig. 6 Lattice points on IMG_A (a) and IMG_B (b).



(a) (b)

Fig.7 Registrtaion (pixel mathcing) result of IMG_A onto IMG_B (a) and test points (b) for accuracy evaluation.

Table 1 Evaluation results of accuracy.

Point Pair	Point to point distance [mm]			
	True Value	Data Pair		
		A&B	B&C	A&B&C
1, 2	27.0	27.7	27.6	27.7
1, 3	14.0	13.7	13.9	13.8
1, 4	15.0	14.9	15.2	15.0
1, 5	28.0	28.3	28.6	28.4
1, 6	40.0	40.2	40.7	40.4
2, 3	18.5	18.7	18.5	18.6
2, 4	30.0	30.7	30.7	30.8
2, 5	30.0	30.2	30.3	30.3
2, 6	33.5	33.6	34.0	33.8
3, 4	13.0	12.6	12.7	12.7
3, 5	17.0	17.5	17.7	17.6
3, 6	27.0	27.8	28.0	27.9
4, 5	16.0	16.4	16.6	16.5
4, 6	28.0	28.7	29.0	28.8
5, 6	14.0	13.9	14.1	14.0
RMSE	-	0.45	0.58	0.51

A Fuzzy Adaptive Model of Emotion and Personality for an Autonomous Robot

Kenneth H. L. Ho
Bio-mimetic Control Research Centre,
RIKEN (The Institute of Physical and Chemical Research),
8-31 Rokuban 3-Chome, Atsuta-Ku, Nagoya, 456 JAPAN,
email: ho@nagoya.bmc.riken.go.jp,

Abstract

This paper proposes a fuzzy adaptive model of emotions for an autonomous mobile robot. We will argue in this paper that emotion is one of the most important attribute for an autonomous intelligent machine. If a robot has to survive in this unpredictable world, it has to adapt and to learn. We propose a simple fuzzy model of emotion for an autonomous mobile robot. The robot will learn its emotional behaviours through its own experience. Therefore, after a certain time the robot will start to express preferences of its behaviours based of its emotional model and its own experience. In our view, if the robot exhibits a preference of emotional behaviours, then it has acquired a personality.

1 Introduction

Today robots are used for mass production of products with similar features. Robots operate in factories where there is no human intervention or involvement. In the future robots have to work with people in an environment which is designed to be used by humans. Therefore, robots have to aware of its surroundings. They have to operate safely without harming themselves as well as others surrounding them. These robots are also likely to work outside of a factory, therefore they have to work in an uncertain and rapidly changing environment.

Current obstacle avoidance algorithms in robotics assume a static environment with predefined obstacles. Most solutions provide optimal paths. Unfortunately in reality, we seldom use an optimal solution for obstacle avoidance. If there is a big object running towards us, we will normally try to avoid it by getting away from it as much as possible instead of finding an optimal distance. Similarly if there is a child running after us, we will probably let the child run into us rather than to avoid him or her. Therefore, we choose different behaviours and strategies in different circumstances based on our own experience.

In this paper we will try to use a model of emotions to choose between different behaviours for our mobile robot under different circumstances. Our robot will have a set of external sensors and a set of internal sensors. It will use these sensors to evaluate its current position and its surrounding environment. It will try to perceive risks based on its own experience and modify its own mood or feeling, i.e. emotion. If it perceives dangers to a certain degree, it will shift its mood toward fear. By accumulating these effects, we can create different moods for different circumstances based on past experience. Different sets of behaviours are attached to these moods. When the robot requires instant decision on its action due to external or internal stimulus, e.g. low battery power or approaching obstacles, it will try to choose the set of behaviours which associate with its present feeling or mood.

2 Emotion - the missing link in an autonomous intelligent system

The motivation for this approach is based on the fact that emotions only exist on higher living organisms, e.g. humans and animals. They are created by billion years of evolution. Intelligent animals seem to be able to express and possess more complex emotions, for example humour is only expressed by human beings. Although emotions are normally regarded as irrational behaviours, they help us to survive and to evolve within our very uncertain and unpredictable world.

If we accept Darwin's evolution theory [2, 3], i.e. natural selection or the survival of the fittest, then we can see that emotion is not just a by-product of the development of intelligent species. If they are just a by-product or irrelevant, emotion would have been elminated by natural selection a long time ago. Using the same argument, we can see that if mathematical

logic is a basic attribute of an intelligent system, we should see more logical species within our own world. Since we cannot find any life forms which utilise mathematical logic, we can conjecture that mathematical logic may not be an important attribute in creating artificial intelligence.

3 Emotion and Behaviours

Modeling emotions has recently become fashionable within the AI and robotic research community. Marvin Minsky[7] outlined his idea of modeling emotion in an autonomous system. Mochida et al[5] and Pfeifer[9] tried to model emotion on a mobile robot in order to create more than one behaviour for a given environment. Gomi[8] tried to make emotion as an emergent property within the Patti Maes's [6] Action Selection Dynamics framework.

3.1 Fuzzy Behaviours and Instincts

The famous psychologist William James[4] suggested that instinct is different from Emotions. Instinct is similar to reflexes. In our view, behaviour-based robotics proposed by Brooks[1] can be treated as a model of instincts. Events are linked together with actions. Reflexes and instincts are parallel and they do not need logical analysis. Instincts can be higher level reflexes. Therefore, we can be able to think that emotions are higher level instincts.

In our model, different instincts, reflexes and behaviours are defined as fuzzy membership functions of the robot sensors. They are represented by fuzzy IF-THEN rules similar to one used by Tunstel[11, 12]. Unlike traditional robotic control algorithms which are mostly based on binary logic reasoning, our behaviours are governed by fuzzy if-then rules. We would like to use fuzzy reasoning to model emotions. The reason is that fuzzy logic is designed to model human imprecisions as well as vague concepts.

3.2 Modelling Emotions

Emotions can be treated as sets of basic instincts or behaviours. Emotion is our natural reaction when we are confronted with a complex environment. It is a built-in mechanism for us to react to situations which will require us to act upon immediately. It is our basic software which allows us to act on situation without analysis. It helps us to adapt to our ever changing environment. However, it does not provide a long term solution for our problems. That is why we sometimes view our emotional decision as irrational.

Characterisitc of emotional control:

- no reasoning
- short term
- interact between the external events and our own internal states of a being. One has to aware of oneself. Therefore, this is another argument for us to have internal sensors to evaluate our personal position given a set of external events.
- One also tries to give out signal to the others within ones surroundings. Therefore, one can project ones desire through emotions without communicating through language. Anger is a good example in which one will try to tell others that one is dangerous without wasting a lot of energy or effor to fight for what he or she wants.
- Emotional control exists in many animals which are less developed than us.

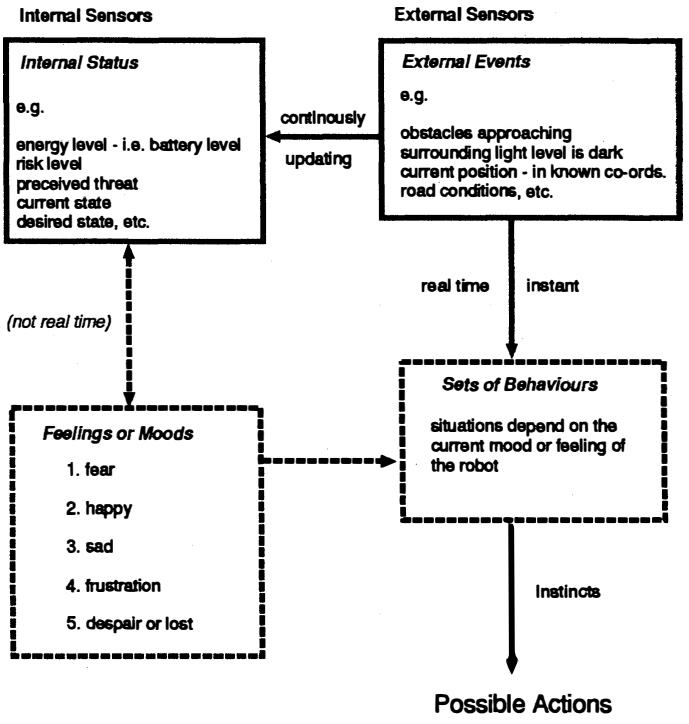


Figure 1: Modelling Emotions

In our model of emotion (see figure 1), we have a hierarchical structure of emotions. If instincts and reflexes can be grouped togther and can be categorised

into different groups, we can treat each of these groups as an emotional state, then each individual emotion has a set of behaviours which are linked directly with the external sensors of the robot. Therefore, emotional state "happy" will have a set of behaviours and "fear" will have another set of behaviours. At any one time, the robot will only associate with one emotional state, i.e. the robot will be either happy or in fear, although it is quite possible to extend our model and allow such emotional state to overlap with each others. Therefore, at any one time, a robot will have only one set of behaviours to choose from. If an external event trigger a reaction, the robot will then activate a behaviour from the current emotional state and ignore all other behaviours in other emotional states.

The emotional state of the robot can only be affected by the internal sensors. In our view, emotion is a manifestation of our internal sensors. Since internal sensors depends on changes of the external sensors, emotional states are indirectly connected to the external sensors (see figure 1).

An emotional state will change from one to the other depending on its internal sensors. It is actually a function of the status of its internal sensors over time. When certain sensors remain at the same status over a period of time, e.g. risk level is high and battery level is low, it will trigger a change in its emotional state. This is defined by using a set of fuzzy if-then rules. Therefore, certain external events may cause an accumulation of effects on these internal sensors which indirectly will cause a change in its emotional state.

These set of fuzzy if-then rules can then be learned by the robot through its own experience. If it finds that the action that it took causes undesirable effect, it will penalize the previous rule. Therefore, over time the robot will try to form its own preferences based on its experience. Different experiences will form different set of membership functions, hence different robots will form different personalities depending on their experiences.

It is similar to human beings, some people are always happy while other are more pessimistic. Therefore, one can learn to have different weight on ones emotional system. Some may put more weight on the fear emotion because one has had very bad experience while others may be more brave because he or she has more success on taking gambles. Therefore, one can define ones personality by modifying its intensity of ones emotional system.

This adaptation of our emotional system can be used in obstacles avoidance. If there is a fairly large object coming toward two robots. One robot has a optimistic personality, it will assume that the object will also detect the presence of the robot and hence no need to do anything. On the other hand, the robot with a pessimestic personality will assume that the large object may harm itself. Therefore, it will try to plan an avoidance path. In this case, it will try to be as far away as possible from the predicted position of the obstacle. This path planning is quite different from the normal optimal path planning procedures of most mobile robot, since optimal path is not necessarly the safest path.

4 Conclusion

In this paper we have proposed a fuzzy adaptive model of emotion for an autonomous mobile robot. We outlined that the robot will form different sets of emotions based on different experiences. We believe that an emergent property of a personality is exhibited by the robot. In the future, we would like to see more complex expression of emotions emerges from our robot, e.g. expression of affection or fear for people, etc.

References

[1] Brooks R.A., *A Robust Layered Control System for a Mobile Robot*, IEEE Transactions on Robotics and Automation, RA-2, No.1, p. 14-23, 1986

[2] Darwin C., *The Origin of Species*, John Murray, 1859

[3] Dawkins R., *The Selfish Gene*, 2nd Edition Oxford University Press, 1989

[4] James W., *The Principles of Psychology*, Vol.1,2, Henry Holt, 1890

[5] Mochida T., Ishiguro A., Aoki T., Uchikawa Y., *Behavior Arbitration for Autonomous Mobile Robots Using Emotion Mechanisms*, Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95), Pittsburgh USA, Vol.3 p.516-521, 1995

[6] Maes P., *A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature*, Proceedings of the 1st International Conference on Simulation of Adaptive Behavior, also in: Meyer J.A.,Wilson S.W., "From Animals to Animats", p.238-246, MIT Press, 1991

[7] Minsky M., *Autonomy and Emotion Machines*, IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS'93), Yokohama Japan, Plenary Session, 1993

[8] Gomi T., Ide K.I., *Emulation of Emotion Using Vision with Learning*, Proceedings of the IEEE International Workshop on Robot and Human Communication (RO-MAN'94), p.210-215, 1994

[9] Pfeifer R., *Emotion in Robot Design*, Proceedings of the IEEE International Workshop on Robot and Human Communication (RO-MAN'93), 1993

[10] Toda M., *The Urge Theory of Emotion and Cognition - English version*, School of Computer and Cognitive Sciences, Chukyo University, 101 Tokodate, Kaizu-cho, Toyota, 470-03, JAPAN 1994

[11] Tunstel E., *Coordination of Distributed Fuzzy Behaviors in Mobile Robot Control*, IEEE International Conference on Systems, Man and Cybernetics, vol.5, p.4009-4014, 1995

[12] Yanaru T., Hirotja T., Kimura N., *An Emotion-Processing System Based on Fuzzy Inference and its Subjective Objections*, International Journal of Approximate Reasoning, Vol.10, No.1, p.99-122, 1994

A Parallel Genetic Algorithm for Matching and Similarity Measure of Graphs

Yong Wang Naohiro Ishii
Department of Intelligence and Computer Science
Nagoya Institute of Technology
Gokiso-cho, syowa-ku, Nagoya, 466, Japan

Abstract

Graph matching and similarity measure of graphs have many applications to pattern recognition, machine vision in robotics and similarity-based approximate reasoning in artificial intelligence. This paper proposes a method of matching and similarity measure between two directed labeled graphs. We define the degree of similarity, the similar correspondence and the similar map which denotes the matching between two graphs. As a approximate computing method, we apply genetic algorithms(GA) to find the similar map and compute the DS between graphs. For speed-up, we make parallel implementations on almost steps of GA. We have implemented our method in a parallel C program and made a simulation on AP1000, a high parallel computer by Fujitsu. The simulation result shows our method is efficiently and usefully.

Key Words: *Graph Matching, Similarity Measures, Genetic Algorithms, Parallel Computing, similarity-based approximate reasoning.*

1 Introduction

Graph matching and similarity measure of graphs have many applications to pattern recognition, machine vision in robotics and similarity-based approximate reasoning in artificial intelligence. Obviously, graph matching and similarity measure of graphs are two close related problems. But many methods have studied only take one as the problem, and therefor can not fit to the other. For example, if the parts of two graphs are not isomorphic, then graph matching methods, many of them are based on subgraph isomorphism, will fail to make matching on the parts. But for similarity measure of graphs, an appropriate matching on the parts will be needed. Since these two problems can not be separated in many cases, we should study some methods for the both problems.

This paper proposes a method for both matching and similarity measure between two directed labeled graphs. We start from defining the correspondence and map from one graph to the other based on the same concepts in set theory. Then, we define the degree of similarity(denoted by DS) of graphs on a correspondence. Based on this, we define the DS, the similar correspondence and the similar map which denotes the matching between graphs.

It is clearly that the problem of computing the DS and the similar map between graphs is a kind of combinational optimization problem. As a approximate computing method, we apply genetic algorithm(GA) to our problem. For speed-up, we make parallel implementations for almost steps of GA.

In section 2, we give all definitions and theorems used in this paper. In section 3, we propose a sequential GA to computing the DS of graphs and find the similar map between graphs. Section 4 describes the parallel GA. Section 5 presents the simulation and discusses the results.

2 The DS and the Similar Correspondence between Graphs

In the definitions and theorems showed in this section, G denotes a directed labeled graph. V is the set of vertexes and E is the set of edges. The number of vertexes of graph G are denoted by $NN(G)$. The weight of a vertex v and an edge e are denoted by $W(v)$ and $W(e)$, respectively. Let $W(\phi) = 0$.

Definition 1 *The correspondence from graph G_t to G_s is denoted by $\mathcal{C}: G_t \to G_s$ and defined as follows:*
(1). $\forall v_t \in V_t$, v_t correspond to only one vertex v_s in G_s (denoted by $\mathcal{C}(v_t) = v_s$), or does not correspond to any vertex in G_s (denoted by $\mathcal{C}(v_t) = \phi$). And if $v_1, v_2 \in V_t$, $v_1 \neq v_2$, $\mathcal{C}(v_1) \neq \phi$, $\mathcal{C}(v_2) \neq \phi$ then $\mathcal{C}(v_1) \neq \mathcal{C}(v_2)$.
(2). The correspondence of edges is decided by the correspondence of vertexes, that is: $\forall e_t = (v_{t_1}, v_{t_2}) \in E_t$, if $\mathcal{C}(v_{t_1}) = v_{s_1} \neq \phi$, $\mathcal{C}(v_{t_2}) = v_{s_2} \neq \phi$ and $e_s =$

$(v_{s_1}, v_{s_2}) \in E_s$, *then* e_t *correspond to* e_s *(denoted by* $\mathcal{C}(e_t) = e_s$*), else* e_t *does not correspond to any edge in* G_s *(denoted by* $\mathcal{C}(e_t) = \phi$*).*

The inverse correspondence of the correspondence $\mathcal{C}$ is denoted by $\mathcal{C}^{-1}$.

Definition 2 *Let* $\mathcal{H}$ *be a correspondence from graph* G_t *to* G_s*. We call* $\mathcal{H}$ *the map from* G_t *to* G_s *if* $\forall v_t \in V_t$, $\mathcal{H}(v_t) \neq \phi$, *i.e.* $\mathcal{H}(v_t) \in V_s$*).*

Let $\mathcal{C}$ be a correspondence from graph G_t to G_s, and let $v_t \in V_t$, $v_s \in V_s$, $e_t \in E_t$, $e_s \in E_s$. The DS of vertex v_t and v_s on correspondence $\mathcal{C}$ is denoted by $DS_{\mathcal{C}}(v_t, v_s)$, and the DS of edge e_t and e_s on correspondence $\mathcal{C}$ is denoted by $DS_{\mathcal{C}}(e_t, e_s)$. They should be decided beforehand. For example, let $DS_{\mathcal{C}}(v_t, v_s) = 1$ if $v_t = v_s$, or 0 otherwise.

Definition 3 *Let* $\mathcal{C}$ *be a correspondence from graph* G_t *to* G_s*. The DS of* G_t *and* G_s *on* $\mathcal{C}$ *is denoted by* $DS_{\mathcal{C}}(G_t, G_s)$ *and defined as follows:* $DS_{\mathcal{C}}(G_t, G_s) = \frac{F_n + F_e}{M_n + M_e}$*. where*

$$F_n = \sum_{v_t \in V_t} \frac{W(v_t) + W(\mathcal{C}(v_t))}{2} DS_{\mathcal{C}}(v_t, \mathcal{C}(v_t))$$

$$F_e = \sum_{e_t \in E_t} \frac{W(e_t) + W(\mathcal{C}(e_t))}{2} DS_{\mathcal{C}}(e_t, \mathcal{C}(e_t))$$

$$M_n = \max\left(\sum_{v_t \in V_t} W(v_t), \sum_{v_s \in V_s} W(v_s) \right)$$

$$M_e = \max\left(\sum_{e_t \in E_t} W(e_t), \sum_{e_s \in E_s} W(e_s) \right)$$

Definition 4 *Let* $\mathcal{C}$ *be a correspondence from* G_t *to* G_s *and* $v_t \in V_t$*. The DS of graph* G_t *and* G_s *on* $\mathcal{C}$ *at vertex* v_t *is denoted by* $DS_{\mathcal{C}}(G_t, G_s)|_{v_t}$ *and defined as follows:*

$$DS_{\mathcal{C}}(G_t, G_s)|_{v_t} = DS_{\mathcal{C}}(G_t(v_t), G_s(\mathcal{C}(v_t)))$$

where $G(v)$ *is the subgraph of* G*, which consists of vertex* v *and the edges take* v *as their endpoints.*

Definition 5 *The DS of graph* G_t *and* G_s *is denoted by* $DS(G_t, G_s)$ *and defined as* $DS(G_t, G_s) = \max_{\mathcal{C}} DS_{\mathcal{C}}(G_t, G_s)$*, where* $\mathcal{C}$ *is a correspondence from* G_t *to* G_s*.*

If $DS_{\mathcal{C}_0}(G_t, G_s) = DS(G_t, G_s)$*, then* $\mathcal{C}_0$ *is called the similar correspondence from* G_t *to* G_s*, or the similar map from* G_t *to* G_s *if it is a map.*

Theorem 1 $DS_{\mathcal{C}}(G_t, G_s) = DS_{\mathcal{C}^{-1}}(G_s, G_t)$ *and* $DS(G_t, G_s) = DS(G_s, G_t)$*. where* $\mathcal{C}$ *is a correspondence from graph* G_t *to* G_s*.*

Theorem 2 *Let* $\mathcal{H}$ *and* $\mathcal{C}$ *be map and correspondence from graph* G_t *to* G_s*, respectively. For all* v *where* $v \in V_t$, $\mathcal{C}(v) \neq \phi$*, if* $\mathcal{C}(v) = \mathcal{H}(v)$ *then* $DS_{\mathcal{H}}(G_t, G_s) \geq DS_{\mathcal{C}}(G_t, G_s)$*.*

The above theorems can be derived easily from related definitions. It is also not difficult to derive the next theorem by the above theorems.

Theorem 3 *If* $NN(G_t) \leq NN(G_s)$ *then*

$$DS(G_t, G_s) = \max_{\mathcal{H}} DS_{\mathcal{H}}(G_t, G_s)$$

where $\mathcal{H}$ *is a map from graph* G_t *to* G_s*.*

Using definitions and theorems described in this section, we present the method of computing the similar correspondence and the DS between graphs in next subsection.

3 The Sequential GA

From theorem 1, we can assume $NN(G_t) \leq NN(G_s)$. By theorem 3, we only need to search maps to find the similar map and computing the DS between graphs on the similar map. It is clearly that this problem is a kind of combinational optimization problem. We apply GAs to this problem because they can find better solution fastly in combinational optimization problems.

In the following, we propose the genetic algorithm for our problem.

Let G_t and G_s denote two directed labeled graphs and assume $NN(G_t) \leq NN(G_s)$.

Coding maps from G_t to G_s: For every graph, we number all vertexes of that for distinguishing them. Each map from G_t to G_s is encoded by a string of numbers with length $NN(G_t)$. Each number in the string denotes a vertex of G_s, and its ordinal number denotes the corresponding vertex of G_t. For example, the string 23568 denotes that vertex 1, 2, 3, 4 and 5 of G_t correspond to 2, 3, 5, 6 and 8 of G_s, respectively.

Population model: We set the population size as $NN(G_t) + NN(G_s)$. The fitness of an individual is set as the DS between graphs on the map which is encoded by the individual (refer to definition 3). The fitness of

a gene in an individual is set as the DS between graphs on the map at the vertex of G_t correspond to the gene (refer to definition 4). The fitness of a population is set as the maximum fitness of individuals in the population.

Stopping condition: If the same fitnesses for $2(NN(G_t) + NN(G_s))$ populations running are obtained, then bring the genetic algorithm to a termination.

Initialization: The start population is initialized randomly under the following conditions:
(1). Each individual is made under the following condition: if there are the same labels for vertex v_t in G_t and vertex v_s in G_s, then let v_t correspond to v_s in the map encoded by the individual. By this way, it can be expected to find the similar map fastly.
(2). Do not make the identical individuals: It is for keeping diversity of individuals in a population. For doing this, if an individual which is equal to another was made, then mutation operates on it until it is different from all other individuals.

Selection: All individuals in the population are sorted by their fitnesses. Let $n = NN(G_t) + NN(G_s)$ which is the population size. In our method, all individuals are selected for mating. We perform crossover to the ith individual and the $(i+1)$th individual to generate $[(n+1)/2]$ individuals of the next population, where $i = 1, 2, \ldots, [(n+1)/2]$, then the individuals with higher fitnesses could be generated because the fitnesses of parents are higher. Also, we perform crossover to the jth individual and the $(n+1-j)$th individual to generate $[n/2]$ individuals of the next population, where $j = 1, 2, \ldots, [n/2]$, then the average fitness of individuals in the next population could be increased because the fitness of one parent is higher and the fitness of another is lower.

Crossover: Two selected individuals are called parent-A and parent-B, and assume the fitness of parent-A is higher or equal to that of parent-B. Every crossover results in one offspring. We present the method of crossover in following.
(1). Decide genes inherited from parents: In parent-A, Genes whose fitnesses are equal to or greater than that of corresponding genes of parent-B, are copyed to the offspring. The other genes in the offspring are inherited from parent-B. By this way, the genes with higher fitnesses in the individuals with higher fitnesses can be left to individuals in the next population.
(2). Avoid that there are the same genes in the offspring: Maybe there are the same genes in the offspring. By the definition of map from G_t to G_s, this situation must be avoided. If a gene inherited from parent-B is equal to one inherited from parent-A, then change this gene into one of genes which do not appear in the offspring, randomly.
(3). Let the fitness of the offspring be equal to or greater than the fitness of parent-A: If the offspring whose fitness is less than that of parent-A was generated by the way described in the above, then eliminating it and use parent-A as the offspring to leave to the next population. By this way, it is certain that the fitness of the next population will be equal or greater than that of the previous one, and the average of fitnesses of individuals in the next population will be equal or greater than that in the previous one.

Mutation: The probability of mutation is 10%. In our method, mutation means that replace a gene by a new gene not in the individual, or exchange it with another in the individual. The gene chosed for mutation will pick out from genes whose fitnesses are lower than that of the individual by the probability of 80%, or from the other genes by the probability of 20%.

Differentiation of the same individuals: If there are two individuals then mutation will operates on one until there are not same individuals in the population. In this case, the gene chosed for mutation will be decided randomly.

4 The Parallel GA

Since the offsprings are produced independently in each generation, the most steps of GAs are easily made parallel. The parallel implementation of the GA described in the above section is showed in the following. It consists of a host program and a cell program which is run parallel by a number of cells under control of the host.

Host Program

```
send population size p and graphs to all cells.
while (stopping condition not met) {
    receive fitnesses of all individuals from cells.
    if (stopping condition not met)
        send go-ahead message to cells.
    else
        send finish message to cells.
}
receive final population from cells.
```

Cell Program

```
receive p and graphs from host.
initialize the subpopulation whose size is p_i.
while (finish message is not received) {
  if (CellID = 0) send 0 to itself.
  while (go-ahead & finish message not received) {
    receive a message.
    case an individual is received:
      if (there is a same previous received one)
        mutate and sent it to related cell.
      else
        insert it to previous received individuals
        and send its fitness to host.
    case an number n is received:
      send n + p_i to the next cell.
      send individuals as parents to related cells.
    case a parent is received:
      if (all parents are received)
        implement crossover and mutation.
  }
}
send final subpopulation to host.
```

5 Simulation

We have implemented our method in a C program and made a simulation on AP1000, a high parallel computer with 32 cells by Fujitsu. For simulation, we made 1000 pairs of graphs. the number of vertexes in a graph is bonded from 5 to 50. each pair of graph G_t and G_s is made by the following:
(1). Making G_t randomly.
(2). Copying G_t to G_s. Now we get a map on which each vertex of G_t corresponds to the same one of G_s. we call it model-map.
(3). Revising G_s randomly with keeping the model-map. the revision number is equal or less than half of sum of numbers of vertexes and edges of G_t and decided randomly. the revision operators are deletion, insertion and substitution (includes exchange).

The DS between graphs on model-map is called model-DS between graphs. obviously, the model-map is equal or close to the similar map, especially for smaller number of revision. In the simulation, we compare DS_{GA} which denotes the DS between graphs obtained by GA to DS_{model} which denotes the model-DS between graphs. We divide 1000 pairs of graphs into groups by the difference $d = DS_{model} - DS_{GA}$, and evaluate numbers of groups. The simulation result shows that, there are 375 graph pairs for $d \in (-0.1, 0)$, 614 graph pairs for $d = 0$, 6 graph pairs for $d \in (0, 0.1)$ and 5 graph pairs for $d \in [0.1, 0.3)$. It means that, for about 99% graph pairs, the similar map and DS obtained by the GA is equal or very close to that we defined. The average, maximum and min minimum running time(CPU time) are 29.2998 seconds, 154.0580 seconds and 0.6266 seconds. These results shows our method is efficiently and usefully as a approximate computing method.

6 Conclusion

Graph matching, as applied to pattern recognition and computer vision, is an NP problem. This paper proposed a method of graph matching with similarity measure of graphs using GAs. our method is based on directed labeled graphs which has many applications such as knowledge representations and flowchart. It can easily be fitted to other graphs. Our method can also be applied to analogical reasoning and case-based reasoning when taking graphs as knowledge representations. The parallel implement described in this paper is only for reducing running time. We will consider to improve the quality of solutions in the next work. Using the method to build a real application system is also left as our future work.

References

[1] Almohamad, H. A. and Duffuaa, S. O., "Linear Programming Approach for th Weighted Graph Matching Problem." *IEEE Trans. on Pattern Analysis & Machine Intelligence*, Vol. 15, No. 5, pp. 522-525, 1993.

[2] H. Bunke and B. T. Messmer, "Similarity Measures for Structured Representations." Lecture Notes in AI 837, *Topics in Case-Based Reasoning*, Springer-Verlag, pp. 106–118, 1994.

[3] H. Kitano, *Genetic Algorithms*(in Japanese). Sangyou Tosyo, 1993.

[4] Richard E. Blake, "Partitioning Graph Matching with Constraints." *Pattern Recognition*, Vol. 27, No. 3, pp. 439-446, 1994.

[5] Robert D. B., Gareth J. and Peter W., "Matching Two-Dimensional Chemical Graphs Using Genetic Algorithms." *J. Chem. Inf. Comput. Sci.*, Vol. 34, No. 1, pp. 63-70, 1994.

[6] Samir W. M. and David E. G., "Parallel Recombinative Simulated Annealing: A Genetic Algorithm" *Parallel Computing*, 21, pp. 1-28, 1995.

Fine Motion Strategy for Skill-Based Manipulation

Akira Nakamura*
Takashi Suehiro**
Tsukasa Ogasawara*
Hideo Tsukune*

*Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan

**Real World Computing Partnership
1-6-1 Takezono, Tsukuba, Ibaraki, 305 Japan

Abstract

Generally, a manipulator task can be divided into several motion primitives called "skills". Skill-based motion planning is an effective way to execute a complicated task. When planning an assembly process, a technique of fine motion planning such as the backprojection method in configuration space is often used. This paper describes fine motion planning using a skill library, which consists of a pattern of trajectories of skill motions in configuration space.

Key words: manipulation skill, fine motion planning, backprojection, skill library

1 Introduction

In the future, robots will play an increasing role in our lives and will perform many tasks using an artificial brain instead of people. However, it is still difficult for robots to carry out even simple human tasks such as cleaning, cooking, dishwashing, handicraft and carpentry. In order to make robots in the home a reality, it is necessary to realize these tasks by using special techniques of manipulating robots. By analyzing human motions in tasks such as assembly and disassembly, movements were found to consist of several significant motion primitives [1],[2]. We called these "skills" and explained how most of the tasks of the manipulator can be composed of sequences of skills. That is, we demonstrated that robots can perform human tasks by using the concept of skill.

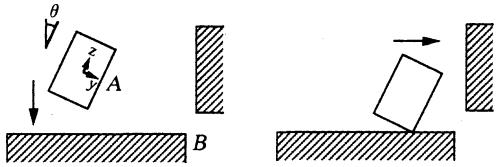
On the other hand, techniques of fine motion planning performed in configuration space are often used for planning assembly processes. In configuration space, an object is represented as a point to simplify the planning. Lozano-Perez et al. proposed the concept of pre-image, and carried out planning using generalized damping with uncertainty of the sensor, control and model [3]. Erdmann proposed the backprojection method, in which the goal region is projected in reverse using an error cone [4], which made it easier to obtain the reachable region to the goal than using the pre-image. Some methods of planning with rotation in configuration space have also been proposed [5],[6]. However, there is no method using the concept manipulation skills.

In this paper, we propose a fine motion planning for various tasks composed of several manipulation skills, using a skill library which consists of patterns of trajectories of skill motion in configuration space. A skill sequence composing the task and the initial position and orientation from which the object can reach the goal can be obtained by this method.

In the next section, we explain the concept of manipulation skills. The representation of skills in configuration space is shown in section 3. Construction of the skill library represented by trajectories in configuration space and the composition of a task are explained in section 4. Two examples, a peg-in-hole task and unfastening of a bolt using a wrench, show the effectiveness of our method in section 5.

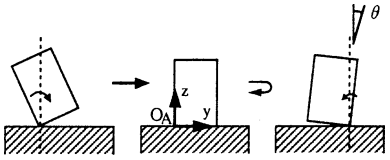
2 Manipulation Skills

In the execution of a manipulation task, its motion can be divided into several motion primitives, each of which has a particular target state. We call these primitives "skills", and in each primitive there is a transition from one state to another state [1],[2]. Most manipulation tasks can be realized by a combination of skills. A set of skill primitives is regarded as one component of the control hierarchy of manipulation. The skill layer is placed between the servo layer and the task



(a) move-to-touch$_f$ skill (b) move-to-touch$_c$ skill

Fig.1 Move-to-touch skill



(a) (b) (c)

Fig.2 Rotate-to-level skill

program layer, so it is possible to program a task as a sequence of skills, independent of the control of the servo layer.

Skills in which the contact states vary at assembly and disassembly tasks are particularly significant. In this paper, we consider three skills of "move-to-touch", "rotate-to-level" and "rotate-to-insert" which play an important part in such tasks. We consider skill motions in two-dimensional environments.

(1) "Move-to-touch" Skill

The move-to-touch skill means the transition from "free" to "vertex-to-face" contact between a grasped object and another object (Fig.1(a)). The similar transition of keeping the contact in a different direction of motion is also included in this skill (Fig.1(b)). These are represented by the "move-to-touch$_f$" skill and the "move-to-touch$_c$" skill, respectively. These skills are carried out in velocity control mode, and the contact is detected by the increased resistance in the direction of motion.

(2) "Rotate-to-level" Skill

The rotate-to-level skill means the transition from "vertex-to-face" contact to "edge-to-face" contact (Fig.2). The accomplishment of this skill is detected by an abrupt change of the instantaneous center position.

(3) "Rotate-to-insert" Skill

In an insertion task, it is generally difficult to achieve directly the state of Fig.3(d) in the case of a small clearance. First, the transition from the state of Fig.3(a) to the state of Fig.3(b) is performed by using the move-to-touch$_f$ skill, and the state of Fig.3(c) is achieved by using the move-to-touch$_c$ skill. Then, the state of Fig.3(d) is accomplished by gradually raising the object while keeping the contact state of Fig.3(c). The rotate-to-insert skill means this motion of rotating the object obliquely into the hole in order to carry out the insertion. The accomplishment of this skill is also detected by an abrupt change of the instantaneous center position. In this paper, we assume that the rotate-to-insert skill includes the press motion to the goal of an insertion task (Fig.3(e)). Therefore, this skill is continued until the resistance in the direction of pressing increases sharply.

3 Skills in Configuration Space

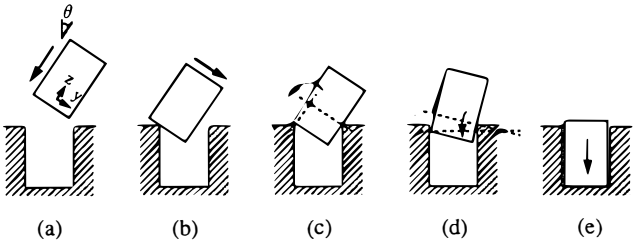
In this section, we consider the trajectory of skill motions in configuration space.

(1) "Move-to-touch" Skill

In configuration space, the trajectories of the object by the move-to-touch$_f$ skill and the move-to-touch$_c$ skill are drawn in Fig.4(a). Considering the uncertainty of control of the manipulator with respect to the move-to-touch$_f$ skill, the trajectory using the control uncertainty cone is drawn in Fig.4(b).

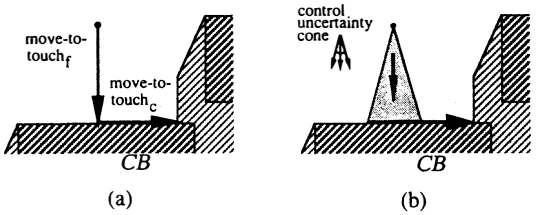
(2) "Rotate-to-level" Skill

For simplicity, we suppose that the reference point O_A is a vertex in contact with the surface in advance (Fig.2). Then, the position of O_A on the YZ-plane in



(a) (b) (c) (d) (e)

Fig.3 Move-to-touch and rotate-to-insert skills



(a) (b)

Fig.4 Move-to-touch skill in C-space

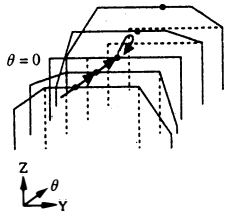


Fig.5 Rotate-to-level skill in C-space

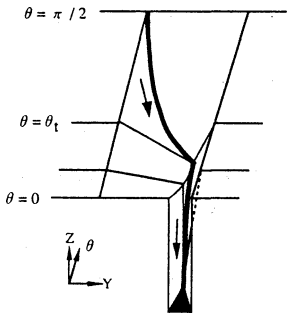


Fig.6 Rotate-to-insert skill in C-space

configuration space is kept constant, even if the orientation θ of the object varies within a negative value by the rotate-to-level skill (Fig.5). On the other hand, after the orientation $\theta = 0$ where edge-to-face contact is just achieved, the transition of the position of O_A occurs. However, in the ideal case, the motion of the manipulator stops and this trajectory in the configuration space terminates at this boundary $\theta = 0$, so this unnecessary transition does not occur. Therefore, we consider the ideal trajectory without overhang, though the manipulator goes beyond this in practice because of the state detection based on the change of the instantaneous center position.

(3) "Rotate-to-insert" Skill

In configuration space, the trajectory of the object by the rotate-to-insert skill is shown in Fig.6. The transfer motion of a vertex from Fig.3(c) to Fig.3(d) is done at the orientation $\theta = \theta_t$ in Fig.6, where the phase of C-obstacle *CB* changes, and passing this point can be detected by a change of the instantaneous center position.

4 Construction of Skill Library and Composition of Task

First, the skill library is constructed in advance. The trajectories drawn in configuration space are derived for several skill primitives such as the above-mentioned skills. A skill library which consists of skill primitives expressed by trajectories in configuration space is thus constructed.

Next, the skill sequence required to perform the instructed task is derived. First, some skill which can be performed in configuration obstacles is selected, and a backprojection is drawn by projecting the trajectory from the goal in the reverse direction. Second, regarding the derived region as a subgoal, another backprojection is drawn. Repeating this procedure, a skill sequence composing the task and initial position and orientation from which the object can reach the goal can be obtained.

5 Example

(1) Peg-in-Hole Task

This example shown in Fig.7 is similar to the insertion task described in (3) in section 2, and we will show the initial positions and orientations from which the object is guaranteed to reach the goal by the instructed sequence of skill primitives. First, C-obstacles *CB* in $\theta = [0, \pi/2)$ are derived as shown in Fig.8, and the back trajectories (Fig.9(b)) from the goal (Fig.9(a)) by a practicable rotate-to-insert skill are obtained. Secondly, regarding the region of Fig.9(b) as a subgoal, the backprojection by a practicable move-to-touch$_c$ skill of the transition in the y-direction while pressing in the

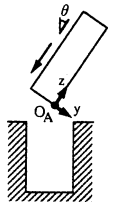


Fig.7 Peg-in-hole

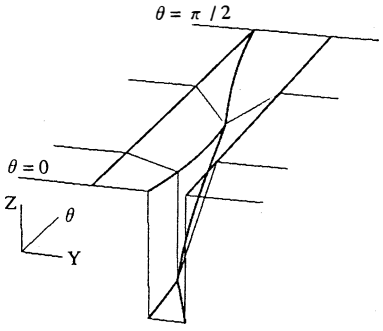


Fig.8 C-obstacles

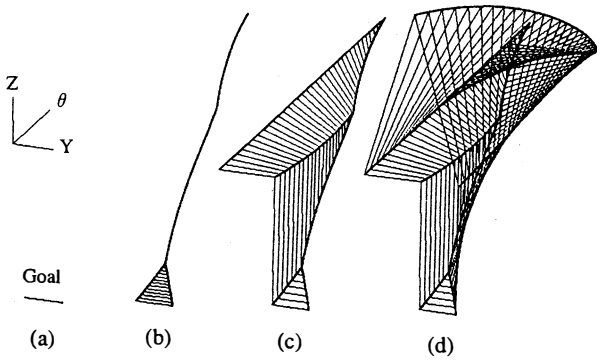


Fig.9 Backprojection in each skill

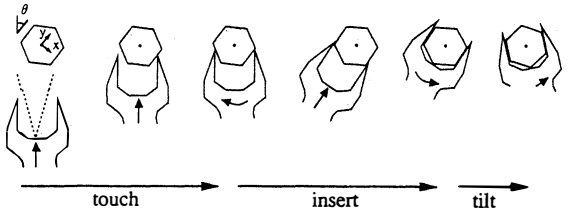


Fig.10 Unfastening of a bolt by a wrench

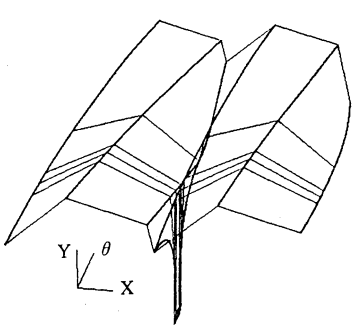


Fig.11 C-obstacles

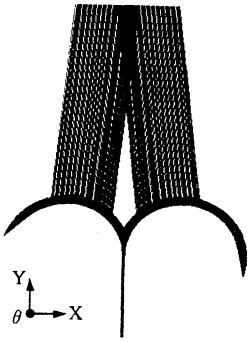


Fig.12 Backprojection

-z-direction can be derived (Fig.9(c)). Next, the backprojection from this region by a practicable move-to-touch$_f$ skill of the transition in the -z-direction can be derived (Fig.9(d)). Although the area of backprojection becomes larger as the value of θ becomes smaller, too small a value θ is undesirable for the reason of (3) in section 2. The initial position and orientation are decided by considering this equilibrium.

(2) Unfastening Task of a Bolt Using a Wrench

Let us consider unfastening a bolt using a wrench (Fig.10). In this example, we deal with the stages of touch and insertion. We suppose that a bolt is a movable object and a wrench is a fixed obstacle, opposite to the real world. Assuming that the reference point O_A is the center position of the bolt, the C-obstacles CB in the range of angle of $\pi/3$ are shown in Fig.11 considering a symmetry of bolt. Fig.12 shows a collection of the boundaries of the backprojections from the goal by the rotate-to-insert skill and move-to-touch skill with control error, and the radial blank region (to be exact, its inverse transformed region on position and orientation) means the appropriate initial position. Generally, since a bolt is small for a range sensor, it is difficult to measure the angle of the bolt around the axis of rotation, even though the position of the axis of rotation is known. However, even under such unfavorable conditions, this task composed of two skills succeeds if it starts from the initial region.

6 Conclusion

We have shown a fine motion planning of tasks with skills using a skill library which consists of skill primitives represented as patterned trajectories in configuration space. The procedure of multi-step planning and its computability have been mainly researched for the backprojection method used as a fine motion strategy in this paper. However, there has been no research on backprojection for the skill-based technique. The method proposed here uses not only straight-line motions but also rotary motions with skill. Therefore, general and skillful planning can be derived, and manipulation tasks closer to human tasks can be achieved. In future, we will study the derivation of a more appropriate initial position and orientation for reaching the goal, the precise treatment of errors on control, sensor and model, and reinforcement of the skill library.

Reference

[1] T. Suehiro and K. Takase, "Skill Based Manipulation System," (in Japanese), J. of Robotics Society of Japan, Vol.8, No.5, pp.551-562, Oct. 1990.

[2] T. Hasegawa, T.Suehiro and K. Takase, "A Model-Based Manipulation System with Skill-Based Execution in Unstructured Environment," '91 Int. Conf. on Advanced Robotics, pp.970-975, Pisa, Italy, June 1991.

[3] T. Lozano-Perez, M. T. Mason and R. H. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots," Int. J. Robotics Research, Vol.3, No.1, pp.3-24, 1984.

[4] M. Erdmann, "Using Backprojections for Fine Motion Planning with Uncertainty," Int. J. Robotics Research, Vol.5, No.1, pp.19-45, 1986.

[5] B. R. Donald, "Planning Multi-Step Error Detection and Recovery Strategies," '88 Int. Conf. on Robotics and Automation, pp.892-897, Philadelphia, USA, April 1988.

[6] R. C. Brost, "Computing Metric and Topological Properties of Configuration-Space Obstacles," '89 Int. Conf. on Robotics and Automation, pp.170-176, Scottsdale, USA, May 1989.

An Emergence of Fuzzy Control Rules for Mobile Robots using DNA Coding Method

Tomohiro Yoshikawa, Takeshi Furuhashi, Yoshiki Uchikawa
Department of Information Electronics, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01, Japan
Tel.+81-52-789-2793 Fax. +81-52-789-3166
E-mail: yoshi@bioele.nuee.nagoya-u.ac.jp

Abstract

This paper presents a new coding method based on biological DNA. A mechanism of development from the artificial DNA is also presented in this paper. This mechanism realizes flexible representation of fuzzy rules. The artificial DNA is composed of four kinds of bases. The proposed DNA allows redundancy and overlaps of genes. Effective fuzzy rules for mobile robots are emerged through chasing and avoiding operations.

I. INTRODUCTION

Genetic Algorithms (GA)[1][2] have been widely studied and applied to many problems. This paper proposes a new coding method based on biological DNA and a mechanism of development from the artificial DNA. This paper calls this coding method the "DNA coding method". The GA based on a biologically motivated model was proposed by Willfried Wienholt[3]. This model took into account gene expression which involves translation of nucleotide sequences of DNA into amino acid sequences. This method, however, is devised to solve parameter optimization problems. The DNA coding method and a mechanism of development from the artificial DNA proposed in this paper are suitable for knowledge representation. Complex knowledge are emerged from the simple coding. The length of the proposed DNA chromosome is variable and it is easy to insert and delete parts of the chromosome. GAs based on this DNA coding method are capable of selecting appropriate sets of input/output variables and proper number of rules. The DNA chromosome has a redundancy, and it can be translated flexibly.

This paper studies emergence of effective fuzzy rules using the proposed DNA coding method. These fuzzy rules are used to control mobile robots which play chasing and avoiding. Effective fuzzy rules of a robot are emerged through interactions with the other robots.

II. DNA CODING METHOD

Fig.1(a) shows a flow from biological DNA to cells. The biological DNA consists of nucleotides which have four bases, Adenine(A), Guanine(G), Cytosine(C), Thymine(T)[4][5]. Most of these bases in the top figure in Fig.1(a) are not used for the synthesis of proteins. A messenger RNA (mRNA), which has many unused parts, is first synthesized from the DNA. Then the unused parts are cut out. This operation is a splicing. After this splicing has occurred, the mRNA is completed. Three successive bases called codons are allocated sequentially in the mRNA. These codons are the codes for amino acids. 64 kinds of codons correspond to 20 kinds of amino acids. This allocation of amino acid makes proteins, and proteins make up cells.

Fig.1(b) shows the proposed DNA coding method and the flow of development to sets of fuzzy rules. This figure shows a correspondence of the proposed method to the biological development. A chromosome consists of combinations of four bases, A, G, C, T. The chromosome has many redundant parts, and after a splicing, the mRNA is completed. This process is similar to that shown in Fig.1(a). The codons in Fig.1(b) correspond to amino acids. Unlike the biological amino acid, each artificial amino acid has several meanings, and the meanings of a gene is determined by the combination of the amino acids. An amino acid can be translated as an input variable or a form of membership function, and so on. A sequence of amino acids makes a fuzzy rule. The DNA chromosome makes up sets of fuzzy rules for controlling a mobile robot.

Fig.2 shows an example of the DNA chromosome and its translation mechanism. In this figure a gene starts from the start codon ATG, and ends at the end codon TAG, and codons in the gene are translated into amino acids: Tyr, Thr, ・・・. Each amino acid has its own role for the problem.

By the proposed mechanism of development from the DNA, the starting point can be shifted from a base to another and some genes overlapping on other genes can be translated.

Each overlapping gene plays an important role. Fig.3 shows this overlapped representation. In this figure, GENE5 in addition to GENE3 and GENE4 can be read from the DNA chromosome. This chromosome has redundancy and also compresses information by overlaps of genes.

Fig.4 shows examples of crossover and mutation. Fig4(a) is an example of one point crossover. Right hand sides from the crossover points are exchanged and new GENE2', 4', 5' are generated. By this method, depending on the crossover points, genes can be drastically changed. There is no constraint on the crossover points. Fig4(b) shows an example of mutation. One base indicated in the figure is changed from T to G. As a result, the GENE1 is changed to GENE1'. By this change, the start codon ATG is newly generated and new GENE7 is generated.

The DNA coding method has the following features:

(a)Flexible representation of knowledge.

(b)The coding is redundant and overlapped.

(c)The length of the chromosome is variable.

(d)No constraint on crossover points.

III. PROBLEM FORMULATION FOR KNOWLEDGE ACQUISITION

Two different types of mobile robots play chasing and avoiding in the area of 2.33m wide and 3m long surrounded by walls. There are several foods in the area for the avoiding robot. The radius of the chasing robot is 150mm, and that of the avoiding robot is 100mm. The chasing robot has eight ultra-sonic sensors (seven in the front, and one in the rear). These sensors can measure the distances between obstacles and themselves in the range of 200mm to 1700mm. The chasing robot must acquire rules to catch the other robot. The avoiding robot has twelve infrared sensors to see around. These sensors can measure limited distance less than 350mm. Therefore the avoiding robot cannot recognize the approaching robot until the enemy comes near to it. The avoiding robot must find fuzzy rules not to be caught by the other robot. Each robot has a chromosome containing a set of fuzzy rules. The fuzzy rules steer and accelerate/deccelerete the robot to chase/avoid the other robot and stay away from the walls.

The robot which reaches or avoids the other robot receives more payoffs from the environment. Considering these payoffs as fitness values, the genetic operators are applied to the chromosomes and the fuzzy rules are evolved.

IV. APPLICATION OF DNA CODING METHOD

The way of development from DNA chromosome to the fuzzy rules and the way to find effective rules are described in this section.

4.1 Representation of rules

The candidates for the input variables of each robot are the detected values of sensors D and the robot's velocity V, and those for the output variables are the steering angle u, and its velocity V. The chromosome has sets of fuzzy rules which are represented by IF~ THEN~ rules. The chromosome determines combinations of input/output variables and membership functions of each fuzzy rules. The central position x_c and the width σ of the membership functions are also encoded into the chromosome.

In the biological DNA, a gene starts from the start codon ATG, and ends at the end codon TAA, TAG or TGA. In this paper, a gene also starts from the start codon ATG which corresponds to IF. The end codon is not definitely determined. A gene consists of the codons between IF codon and some related codons succeeding to THEN codon. Reading from the top of the DNA chromosome, translation to a fuzzy rule starts upon finding the start codon ATG. As described in Section II, overlaps of genes is allowed in the DNA chromosome. After reading a fuzzy rule, re-reading is restarted from the second base of the IF-codon and a new IF codon is sought. Fig.5 shows an example of the DNA chromosome and genes (fuzzy rules). In this figure, bases are read from the head of the DNA chromosome, and if the start codon ATG is found, a fuzzy rule starts from this part. In this example, the next codon GCT is Alanine, and Alanine here means that the input variable is sensor. The sequence of Alanine, Serine, Leucine means that this sensor is No.0 sensor (input variable is D_0). The next part of Glycine, Cysteine determines the form of membership functions, the central position x_c and the width σ, for the input D_0. The next codon GCC makes also Alanine, and this Alanine here means AND. Like this, each amino acid has several meanings, and one meanings is selected based on the position in the gene. In this example, GENE2 in addition to GENE1 can be read from the DNA chromosome.

4.2 Genetic operators

There are seven chasing robots and seven avoiding robots

in one generation. Each robot, either chasing or avoiding, is tested twice by randomly choosing its opponents from the seven counterparts. Each robot has initial payoffs E_1. Each test ends when either of the two confronting robots crashes the wall or the chasing robot catches the other or a certain amount of time has passed. The payoffs for fitness values of the chromosomes are given as follows:

If the robot crashes the wall, the robot loses payoffs E_w ;

If the chasing robot can reach the avoiding robot within a fixed time, the chasing robot receives payoffs E_c, conversely the avoiding robot loses payoffs E_a ;

If the chasing robot cannot reach the avoiding robot in a fixed time, the chasing robot loses payoffs E_m ;

If the avoiding robot can reach the foods, the avoiding robot gains payoffs E_f per a food.

After these tests are done, the genetic operations are applied to the chromosomes of the robots by regarding the payoffs of each robot as its fitness value. One chromosome of chasing robot which has the smallest payoffs is deleted. Two chromosomes which are selected from the remaining six robots are reproduced, and one-point crossover is applied to them. One of the two new chromosomes is selected randomly for a chasing robot. The chromosome of avoiding robot is newly generated in the same way. There is no constraint on the crossover points as described in section II. The mutation operator is also applied to the newly generated chromosomes. The mutation operation can be done simply by changing the bases. This mutation is done to each base at a rate of P_m. The payoffs of all robots are reset at E_1 again. After another simulation for the new generation is done, the genetic operations are applied again. These steps are repeated, and fuzzy rules which control the robots to chase or avoid the other robot and to avoid the wall are expected to be evolved.

V. SIMULATION

Simulations were done. The length of each initial DNA chromosome was 500. The payoffs E_1, E_w, E_c, E_a, E_m,, E_f were 0, 20, 50, 20, 20, 5, respectively. The probability of the mutation P_m was 0.05. The chasing robot which could not reach the avoiding robot within 30 seconds lost payoffs E_m. The genetic operations were applied to the chromosomes of chasing and avoiding robots for 100 generations alternately.

The effectiveness of the proposed DNA coding method and the mechanism of development from the artificial DNA was examined. The following two cases were compared: (i)redundancy and overlapping, (ii)no redundancy nor overlapping. In the case of (ii), no start codon is used. The DNA is translated from its head. Fig.6 shows the average of all payoffs of 7 chasing robots during the 3000-5000th generation. The solid line is the case (i) and the dotted line is the case (ii). Each is the average of 30 trials. In a trial, the simulation from the initial generation to the 5000th generation was done. Since the genetic operations were applied to the chromosomes of chasing and avoiding robots for 100 generations alternately, the payoffs fluctuated periodically. The case which has redundancy and overlapping worked well. An example of emergence of effective control rules of chasing and avoiding robots is shown in Fig.7. This was the case where the avoiding robot acquired very effective rules. The avoiding robot was running along the wall. The chasing robot could measure only the distance using the ultra-sonic sensors. Therefore, the avoiding robot had, so to speak, protective coloring. The avoiding robot could avoid the chasing robot within 30 seconds.

VI.CONCLUSIONS

This paper proposed a new coding method, DNA coding method, and the mechanism of development from the artificial DNA. This DNA coding method is suitable for knowledge representation. The DNA chromosome has a redundancy, and allows overlappings of genes. This paper applied the proposed DNA coding method to emerge of effective fuzzy rules based on chasing and avoiding actions of mobile robots. Further work is the experiments of real mobile robots using this DNA coding method.

REFERENCES

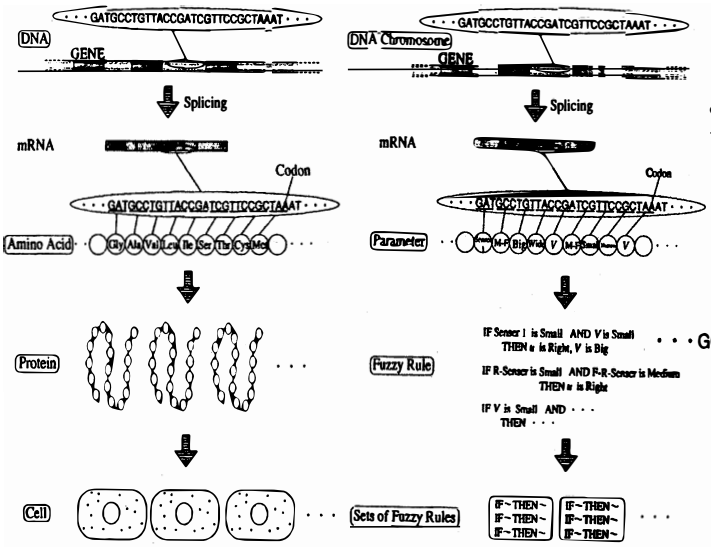
[1]D.E.Goldberg, "*Genetic Algorithm in Search* ", Optimization and Machine Learning, Addison Wesley (1989)

[2]L.Davis(Editor), "*Handbook of Genetic Algorithm* ", Van Nostrand Reynold (1989)

[3]W.Wienholt, "*A Refined Genetic Algorithm for Parameter Optimization Problems* ", Proceedings of The Fifth International Conference on Genetic Algorithms (1993)

[4]B.Albers and others, "*Molecular Biology of the Cell* ", Garland Publishing (1994)

[5]A.Kornberg, "*DNA Synthesis* ", W.H.Freeman and Company (1974)



(a)Biological Way (b)This Method

Fig.1 Flows of development from DNA Chromosome

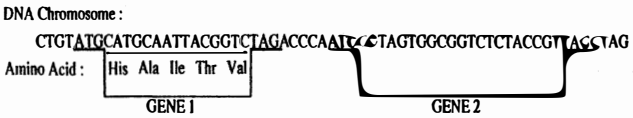


Fig.2 Example of a Chromosome and Translation Mechanism

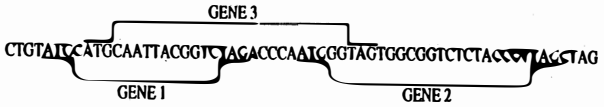
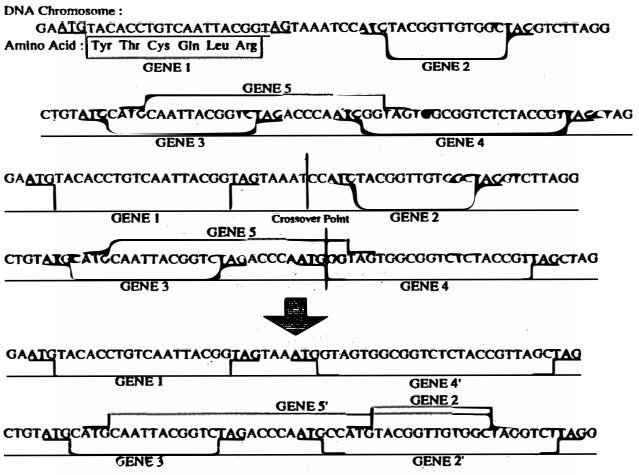
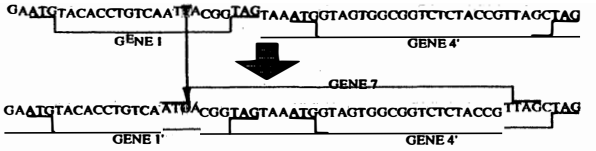


Fig.3 Overlapping of Genes



(a)Crossover

Fig.4 Crossover and Mutation



(b)Mutation

Fig.4 Crossover and Mutation

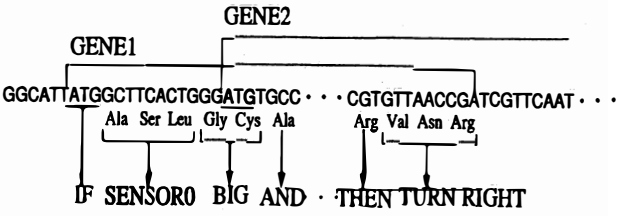


Fig.5 Example of Chromosome and Genes

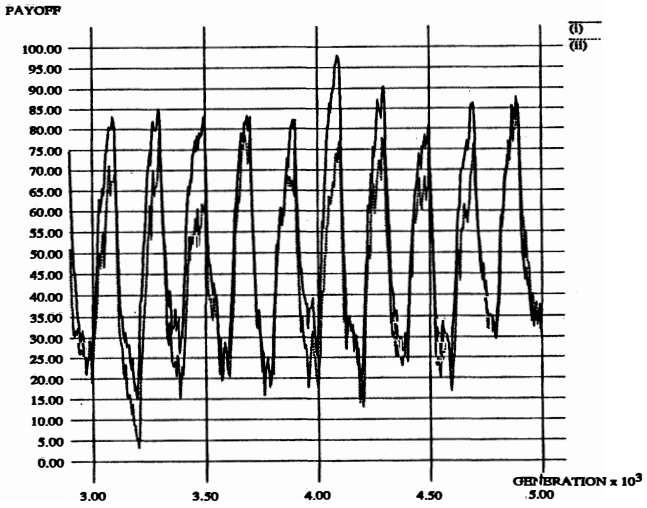


Fig.6 Paypffs of chasing robots (30 trials)

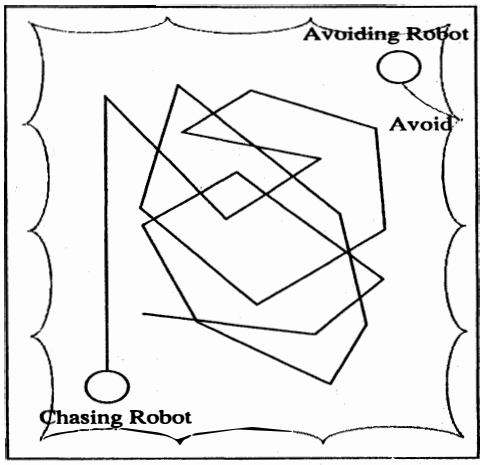


Fig.7 The movement of each robots

A Fuzzy Classifier System for Evolutionary Learning of Robot Behaviors

Yasushi Iwakoshi*, Takeshi Furuhashi**, Yoshiki Uchikawa
*: Chubu Hitachi Electric Co. Ltd.
**: Dept. of Information Electronics, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-01, Japan
Tel. +81-52-789-2792, Fax. +81-52-789-3166
Email furuhashi@nuee.nagoya-u.ac.jp

Abstracts: This paper presents an evolutionary learning of robot behaviors by Fuzzy Classifier System(FCS). The FCS introduces a fuzzy rule base and a fuzzy inference system in place of the rule base and the production system of the Classifier System(CS). The FCS has a feature in that it is robust to environmental changes. The FCS is applied to behavioral learning of a mobile robot for evolving fuzzy control rules. Simulations are done under eight different conditions. Robustness of the acquired fuzzy rules is compared to that of the rules obtained by the CS.

1. INTRODUCTION

Our problem is whether a robot, having a short-sighted sensors and no maps, can reach a goal within a limited number of steps without crashing into walls. Onuma and Hoshino have studied this problem by using the Classifier System(CS)[1]. They pointed out that the CS was not robust to environmental changes.

Valenzuela-Rendon proposed Fuzzy Classifier System (FCS)[2] by introducing a fuzzy rule base and a fuzzy inference system in place of the rule base and the production system of the CS, respectively. The FCS can handle continuous variables. The FCS was, however, applied to an approximation of a single input-single output function in [2]. Studies on apportionment of credits to fuzzy rules for describing the multi-input systems have not been done. The authors have studied application of the FCS to knowledge acquisition of large scale systems[3 - 5]. A new method for apportionment of credits to the fuzzy rules were proposed in [4] and [5]. By this method, fuzzy rules which describe knowledge in complex multi-input/output systems were made possible to be found.

This paper studies an evolutionary learning of robot behaviors by the FCS and examines robustness of the FCS to changing environment. The FCS is applied to find fuzzy control rules for driving a mobile robot. Simulations incorporating experimental conditions are done. Robustness of the found fuzzy rules is compared to that of the production rules acquired by the CS.

2. MOBILE ROBOT

The robot used for experiments is shown in Fig.1. The robot is a micro-mouse on which eight infrared sensors are installed. The allocation of the sensors is shown in Fig.2. The infrared sensors s_1 - s_8 detects the distance between an obstacle and the sensor itself within the range of 5 - 35 cm. The robot is movable controlled by its own CPU.

Fig.1 Exterior of robot

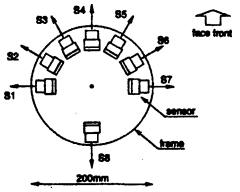


Fig.2 Allocation of sensors

3. FUZZY CLASSIFIER SYSTEM

Figure 3 shows the configuration of the FCS The FCS consists of a fuzzy inference system, an apportionment of credit system, a fuzzy rule base, and a rule generation mechanism.

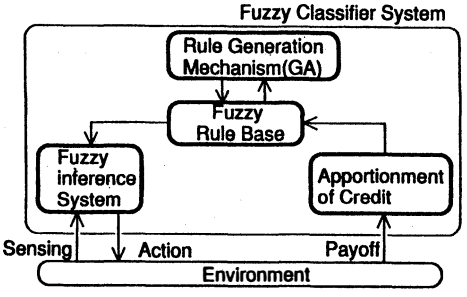


Fig.3 Configuration of Fuzzy Classifier System

3.1 Fuzzy rule base

This fuzzy rule base has n fuzzy rules. The input variables of each rule are the detected values of infrared sensors s_1 - s_8. The antecedent part of each rule consists of eight loci and each locus has one of the five labels, small (S), medium small (MS), medium (M), medium big (MB), big (B), of membership functions shown in Fig. 4 (a). The output is the steering angle u of the robot. The locus for this output also has one of the five labels of the membership functions in Fig. 4 (b).

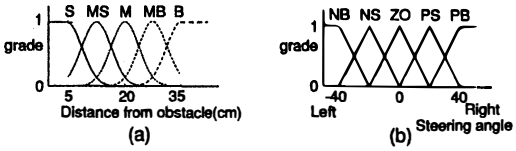


Fig.4 Membership functions

3.2 Fuzzy inference system

The fuzzy inference system senses the environment by the eight infrared sensors s_1 - s_8 and drives the robot. This system generates the command of the steering angle u for the drive system of the robot using the fuzzy rules in the fuzzy rule base. The inference method used in this system is product-sum-center of gravity method. The control of the robot is done until the robot reaches the goal or it collides with the wall. The payoffs are given to the FCS for the result of the control.

3.3 Apportionment of credit system

The apportionment of credit system delivers credits to the fuzzy rules in the fuzzy rule base. The credits to each rule is in proportion to the payoffs from the environment and the degrees of contributions of the rules to the results of control. The credit α of a fuzzy rule is updated as follows:

(a) When the robot reaches the goal,

$$\alpha = \alpha + \sum_{i=0}^{n_g-1} \omega_{k_g-i} \times 1000 \times \left| u_{k_g-i} \right| \qquad (1)$$

where i is the sampling sequence of the controller, n_g is the number of steps from the start to the goal, k_g is the sampling sequence at the time of the goal, ω_x is truth value of the fuzzy rule at x, and u_x is the defuzzified value of the output of the rule at x, i. e. the central position of the membership function in the consequent portion.

(b) When the robot reaches one of the sub-goals in the maze,

$$\alpha = \alpha + \sum_{i=0}^{n_{sg}-1} \omega_{k_{sg}-i} \times 100 \times \left| u_{k_{sg}-i} \right| \qquad (2)$$

where k_{sg} is the sampling sequence at the time to reach the sub-goal and n_{sg} is the number of steps to the sub-goal from a previous sub-goal.

(c) When the robot collides with the wall,

$$\alpha = \alpha - \sum_{i=0}^{n_f-1} \omega_{k_f-i} \times 1000 \times \left| u_{k_f-i} \right| \qquad (3)$$

where k_f is the sampling sequence at the time of collision and n_f is the number of steps to consider the contributing rules to the failure.

3.4 Rule generation mechanism

Fuzzy rules are selected and reproduced using the genetic algorithm (GA). The FCS controls the robot in n_e mazes. Then the GA operation is applied to the fuzzy rules with the accumulated credits.

n_{sel} rules with the least credits α are changed. This change to each rule is done by crossover or mutation selected randomly.

One point crossover operation is applied to a rule in the n_{sel} rules and a reproduced chromosome from remaining n - n_{sel} rules. One of the two generated rules is randomly chosen to replace the rule in the n_{sel} rules. If the labels of membership functions in the antecedent are the same as those of one of the existing rules, this rule will be deleted. The crossover operation is repeated until a rule with a new antecedent part is generated.

When the mutation operation is selected, the labels of membership functions in the antecedent of one of the n_{sel} rules is changed. This mutation operation is also applied to generate a rule with a new antecedent part.

Another mutation operation, which is to change the labels of the membership functions in the consequent portions is applied to the n rules with a probability of p_m.

After these genetic operations are applied, the credits of the n rules are set at zero. The rules at the next generation are now generated and the control of the robot is resumed.

4. SIMULATIONS

Simulations were done under n_e = 8 different conditions as shown in Fig. 5. Each maze was 5 m wide and 5 m long, and divided into 25 sub-areas. The center of every border of the sub-areas was the

sub-goal. The actual experimental conditions including the robot in Fig. 2 and 3 and the environment were incorporated into the simulations. The default direction of the robot at the start point was parallel to the wall. We changed the starting direction randomly within the range of $\pm$ 30 degrees from the default direction at every trial. The fuzzy rule base has n = 100 rules. The numbers of steps to consider the contributing rules n_f was 2. The probability of mutation in the consequent portion of the fuzzy rules p_m was 0.02. The number of rules to be screened out by the selection operation n_{sel} was 10.

By the genetic operations, the FCS could find fuzzy rules which were able to control the robot to reach the goal under all the conditions without colliding with the wall. The tracks of the robot were shown in Fig. 6.

The robustness of the acquired fuzzy rules were compared with those obtained by the CS. Table 1 shows the conditions for the comparison. Production rules were in the rule base of the CS. Each input space were divided crisply as shown in the table. When the distance from an obstacle to the sensor was less than 20 cm, it was detected as near. When the distance was larger than 20 cm, it was far. * meant "don't care of obstacle". The steering angle u was discretized in 5 different angles. The CS could also find rules to control the robot to reach the goal. Fig. 7 shows the mazes used for evaluating the acquired rules by the FCS and the CS. The GA operations were stopped. Ten start points were selected as indicated by the numbers in Fig. 7. At each start point, both the rules of the FCS and the CS were tested 360 times, respectively. The initial direction of the robot was rotated 1 degree at every test. Table 2 shows the success rates at each start point. The rules acquired by the FCS were better than those by the CS. Fig. 8 shows an example of track of the robot controlled by the rules of the FCS starting from the number 8 start point. The robot went along the left side wall.

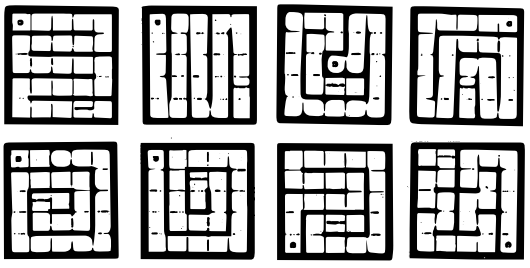


Fig.5 Mazes for learning

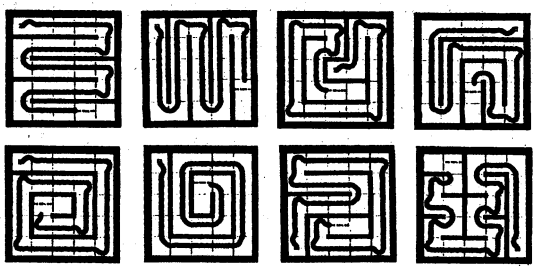
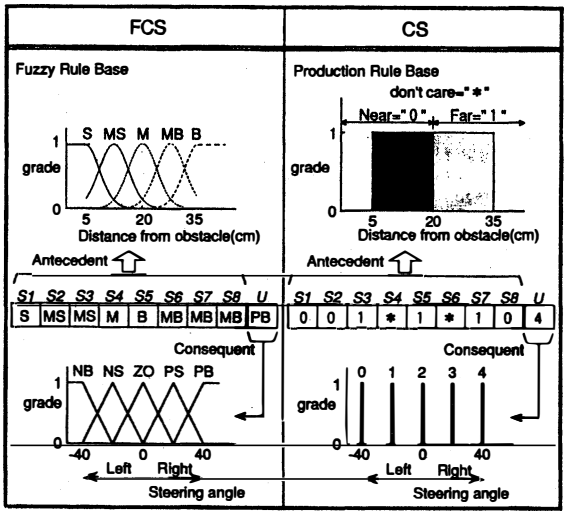


Fig.6 Tracks of robot

Table1 Conditions of FCS and CS



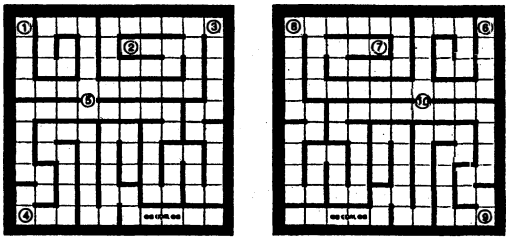


Fig.7 Mazes for evaluation

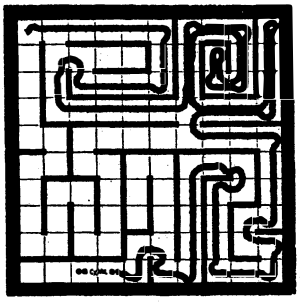


Fig.8 Example of result

Table2 Success rate of rules by FCS and CS in unknown mazes

		Success rate	
		FCS	CS
Start point	①	9%	2%
	②	0%	0%
	③	18%	3%
	④	3%	3%
	⑤	7%	0%
	⑥	22%	22%
	⑦	0%	0%
	⑧	14%	6%
	⑨	31%	19%
	⑩	11%	11%
Average		11.5%	6.6%

5. CONCLUSIONS

This paper presented an evolutionary learning of robot behaviors by Fuzzy Classifier System(FCS). The FCS has a feature in that it is robust to environmental changes. Simulations were done under eight different conditions. Robustness of the acquired fuzzy rules was better than that of the rules obtained by the CS.

The authors are now carrying experiments. The results will be presented at the symposium. The authors also study further on the representation of fuzzy control rules and the apportionment of credit system for acquiring more robust fuzzy rules.

REFERENCES

[1] K. Onuma and T. Hoshino, "Evolutionary Learning of Robot Behaviors by Classifier Systems", JSAI Technical Report, pp. 9-14 (1994)

[2] M. Valenzuela-Rendon, "The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables", Proceedings of the fourth International Conference on Genetic Algorithm, pp.346-353 (1991)

[3] T. Furuhashi, K. Nakaoka, K. Morikawa, Y. Uchikawa, "Controlling Excessive Fuzziness in a Fuzzy Classifier System", Proceedings of the fifth International Conference on Genetic Algorithm, p.635 (1993)

[4] T. Furuhashi, K. Nakaoka, Y. Uchikawa, "A Study on Knowledge Finding Using Fuzzy Classifier Systems", J. of Japan Society of Fuzzy Theory and Systems, vol.7, no.4, pp.839-848 (1995)

[5] K. Nakaoka, T. Furuhashi, Y. Uchikawa, "A Study on Apportionment of Credits of Fuzzy Classifier System for Knowledge Acquisition of Large Scale Systems", Proceedings of Third IEEE International Conference on Fuzzy Systems, pp.1797-1800 (1994)

An Optimization Method of a Combinatorial Optimization Problem using a Replicator Equation

Kazuo Tsuchiya

Dept. of Aeronautics and Astronautics
Advanced School of Engineering
Kyoto University
Kyoto, Japan 606-01

Katsuyoshi Tsujita
Yutaka Inoue
Dept. of Mechanical Engineering for
Computer Controlled Machinery
Faculty of Engineering
Osaka University
Suita, Japan 565

Abstract

This paper deals with an optimization method of a combinatorial optimization problem by using replicator equations. The growth rates are designed in terms of the performance index and the constraints posed on the problem. The replicator equations show the successive bifurcations ; Solutions satisfying the constraints become stable successively by increasing the control parameter and solutions with high performance become stable earlier than solutions with low performance. Then, sub optimum solutions are obtained if the control parameter is set to be as small as possible. The characteristics of the bifurcations and the performances of optimization of the equations are examined analytically and numerically.

1 Introduction

An optimization problem is a basic problem in the engineering science. A path planning is one example of the optimization problems in robotics. Among optimization problems, it is well known that a combinatorial optimization problem is one of the hard problems [1]. A lot of algorithmic methods have been proposed, and recently, optimization methods based on dynamical systems have been proposed. The methods based on dynamical systems have some advantages, one of which is to realize a parallel algorithm easily. The method proposed by Hopfield et al. [2] is one example. These methods, in the most case, utilize a class of dynamical system, i.e., a gradient system; First, a potential function is designed in terms of a performance index and constraints and then, a dynamical equations are derived by the use of the potential function. Sub optimum solutions of the optimization problem are obtained as steady state solutions. These methods can search good sub optimum solutions, but the performance largely depends on the values of the parameters involved in the potential functions and there is no guiding rule to determine the values of the parameters [3]. One method to overcome the difficulty is a simulated annealing method [4] [5]. This method tunes the constraints posed on the problem gradually by varying the parameters in the modal and obtains good sub optimum solutions. A simulated annealing method has been first applied to a stochastic dynamical system and then applied to a deterministic dynamical system as a mean field annealing method. A mean field annealing method has been applied to an optimization problem and obtained good sub optimum solutions.

Here, we will propose an optimization method using a replicator equation. A replicator system is a system composed of reproductive elements of which growth rates are functions of populations of the elements [6]. When all the growth rates are always negative, the replicator equation has a trivial solution that populations of all the elements are zero. On the other hand, by tuning the growth rates appropriately, various patterns can be formed through bifurcations. An optimization method can be constructed using replicator equations ; The growth rates are designed in terms of the performance index and constraints posed on the problem. The equations show successive bifurcations ; By increasing a parameter (control parameter) in the equations, solutions with high performance become stable earlier than solutions with low performance. Then, sub optimum solutions may be obtained if the control parameter is set to be as small as possible. The characteristics of bifurcation and the performances of optimization of this class of dynamical system are examined by applying to a Travelling Salesman Problem.

2 Analysis

The Travelling Salesman Problem is a problem which finds the shortest path starting from a city, travelling all other cities once and coming back to the city. The problem is formulated as follows ; Define state variables u_{ij} ($u_{ij} \geq 0$) where $u_{ij} = 1$ if city i is visited at jth visit. The variables u_{ij} obey the following equations

$$\dot{u}_{ij} = \left\{ 1 - u_{ij}^2 - \frac{\alpha_0}{2} \left(\sum_{i' \neq i}^{N} u_{i'j}^2 + \sum_{j' \neq j}^{N} u_{ij'}^2 \right) - \alpha_1 \sum_{i'} l_{ii'} u_{i'j\pm1}^2 \right\} u_{ij} \quad (1)$$

where $l_{ii'}$ is a distance between city i and i'. The second term in the right hand side of Eq. (1) represents the constraint that each city must be visited once and only one city must be visited at a time. On the other hand, the third term represents the performance of the travel, that is, the length of the path of the travel. In the case where the parameter α_1 is equal to zero, Eq. (1) has the following solutions

Sol. A

$$u_{ij} = u_0 \quad (i,j = 1, \cdots, N)$$
$$u_0 = \frac{1}{1 + \alpha_0 (N-1)}$$

Sol. B

$$\begin{cases} u_{ij} = 1 & (i = i_0, j = j_0) \\ u_{ij} = 0 & (i = i_0, j \neq j_0) \\ (i \neq i_0, j = j_0) \end{cases} \quad (2)$$

While Sol. A is an uniform solution which does not satisfy the constraint, Sol. B is a solution which satisfies the constraint. Sufficient conditions of stability of Sol. A and Sol. B are given as follows ;

$$\begin{aligned} &\text{Sol. A;} \quad 0 < \alpha_0 \leq 1 \\ &\text{Sol. B;} \quad 1 \leq \alpha_0 \end{aligned} \quad (3)$$

When parameter α_0 is increased, the stable solution of Eq. (1) changes from Sol. A to Sol. B through a bifurcation at the point where $\alpha_0 = 1$. When the parameter α_1 is set to be a small positive value, Sol. A and B are modified as follows ;

Sol. A

$$u_{ij} = (u_0 + \Delta u_{ij})$$

where variable Δu_{ij} satisfies the following equation

$$2\Delta u_{ij} + \alpha_0 \left(\sum_{i' \neq i} \Delta u_{i'j} + \sum_{j' \neq j} \Delta u_{ij'} \right) = -\alpha_1 u_0 \sum_{i'} l_{ii'}$$

Sol. B

$$\begin{cases} u_{ij} = 1 - \frac{\alpha_1}{2} \left(l_{ii'(j-1)} + l_{ii'(j+1)} \right) & (i = i_0, j = j_0) \\ u_{ij} = 0 & (i \neq i_0, j = j_0) \\ & (i = i_0, j \neq j_0) \end{cases} \quad (4)$$

Sufficient conditions of stability of Sol. A and B are given by

$$\begin{aligned} &\text{Sol. A} \quad 0 < \alpha_0 \leq \alpha_{0A} \\ &\text{Sol. B} \quad \alpha_{0B} \leq \alpha_0 \end{aligned}$$

where

$$\begin{aligned} \alpha_{0A} &= 1 + \Delta\alpha_{0A} \\ \Delta\alpha_{0A} &= \frac{\alpha_1}{N} \sum_{i,i'} l_{ii'} \\ \alpha_{0B} &= 1 + \Delta\alpha_{0B} \\ \Delta\alpha_{0B} &= \frac{2\alpha_1}{N} \left(L - \frac{1}{N} \sum_{i,i'} l_{ii'} \right) \end{aligned} \quad (5)$$

L is a length of the path of the travel corresponding to the solution. When the parameter α_0 is increased, a solution of Sol. A become unstable near the point where $\alpha_0 = \alpha_{0A}$ while a solution of Sol. B become stable near the point where $\alpha_0 = \alpha_{0B}$. Since the parameter α_{0B} is a function of L, the length of the path of the travel corresponding to the solution, a solution of Sol. B with a shorter path become stable earlier than a solution with a longer path. Then, sub optimum solutions may be obtained if the parameter α_0 is set to be as small as possible.

3 Simulation Results

Numerical studies are carried out to a Travelling Salesman Problem with 51 cities. The locations of the cities and its optimum path are shown in Fig. 1.

3.1 Characteristics of Bifurcation

Parameter S_1 is defined as follows.

$$\begin{aligned} S_1 &= -\sum_j \sum_i p_{ij} \ln p_{ij} \\ p_{ij} &= < \frac{u_{ij}^2}{\sum_i u_{ij}^2} > \end{aligned} \quad (6)$$

where $< * >$ means the averaged value of $*$ over a set of initial values. Parameters S_1 has a large value in the region where Sol. A is stable, and has the minimum value in the region where Sol. B is stable. So, parameter S_1 shows the characteristics of a bifurcation of Eq. (1). Numerical results are shown in Figs. 2 and 3. Fig. 2 shows parameter S_1 as a function of $\Delta\alpha_0$ in the case where $\alpha_1 = 0.1$. Figure 3 shows parameter S_1 in $\Delta\alpha_0$ and α_1 plane. Parameter S_1 is shown to change rapidly near the point $\Delta\alpha_{0B}$. opt. the point where the solution with the optimum path become stable.

3.2 Performances of Optimization

Stability condition (5) indicates that sub optimum solutions may be obtained when parameter $\Delta\alpha_0$ is set to be as small as possible. Figure 4 shows the performance of optimization as a function of $\Delta\alpha_0$. Case (a) and (b) are the cases where $\alpha_1 = 0.01$ and 0.1, respectively. The performance of optimization is gradually improved as parameter $\Delta\alpha_0$ become small. On the other hand, the values of parameter α_1 has little effect on the performance of optimization. Figure 5 shows numbers of iteration required to obtain a steady state solution. The numbers of iteration increase rapidly when the value of $\Delta\alpha_0$ become small. Then, a lower limit of the value of $\Delta\alpha_0$ is limited by the CPU time to obtain a steady state solution, and this, in turn, restricts the performance of optimization of this method.

4 Summary

This paper deals with an optimization method of a combinatorial optimization problem by using a class of dynamical system, replicator equations. The dynamical system shows successive bifurcations ; By increasing the control parameter, the solutions which satisfy the constraints posed on the problem become stable successively according to the values of the performance index. Then, sub optimum solutions may be obtained if the control parameter is set to be as small as possible. The characteristics of bifurcation and the performances of optimization of the dynamical system are examined by analytical methods and numerical simulations. Based on the results obtained in this paper, to establish an optimization algorithm with parameter tuning will be remained in a future work.

References

[1] M.Mesard, G.Parisi and M.A.Virasoro, Spin Glass Theory and Beyond, World Scientific, Singapore, 1987.

[2] J.J.Hopfield and D.W.Tank, "Neural Computation of Decisions in Optimization Problems," Biological Cybernetics, Vol. 52, pp. 141-152, 1985.

[3] G.V.Wilson and G.S.Pawley, "On the stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank," Biological Cybernetics, Vol. 58, pp. 63-70, 1988.

[4] S.Kinkpatrick, C.D.Gelatt Jr. and M.P.Vecchi, " Optimization by Simulated Annealing," Science, Vol. 220, pp. 671-680, 1983.

[5] D.E.Van den Bout and T.K.Miller, "Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing," Biological Cybernetics, Vol. 62, pp. 129-139, 1989.

[6] A.S.Mikhailov, Foundations of Synergetics I, Springer, Berlin, 1990.

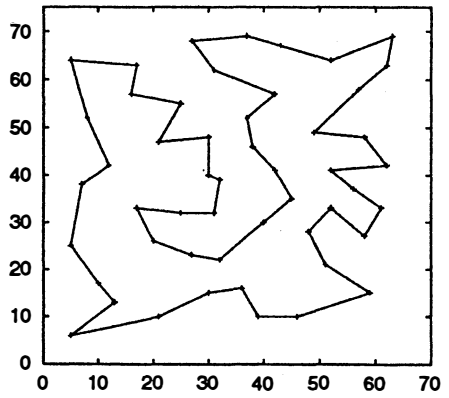


Fig. 1 The locations of 51 cities and the optimum path

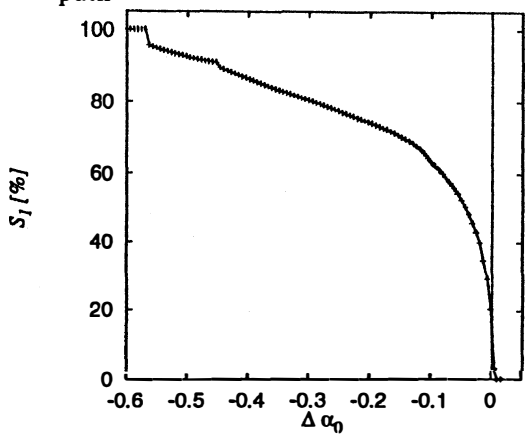


Fig. 2 Parameter S_1 as a function of $\Delta\alpha_0$
% : $(S_1/S_{1max}) \times 100$ $(\alpha_1 = 0.1)$

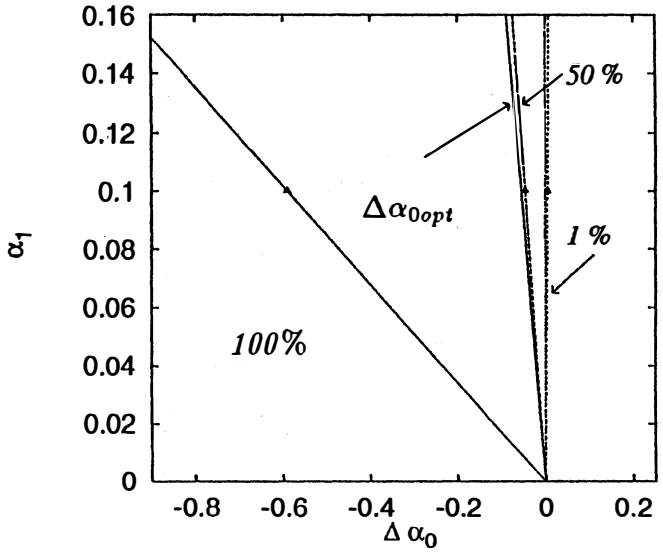


Fig. 3 Parameter S_1 in $\Delta\alpha_0$ and α_1 plane
% : $(S_1/S_{1max}) \times 100$

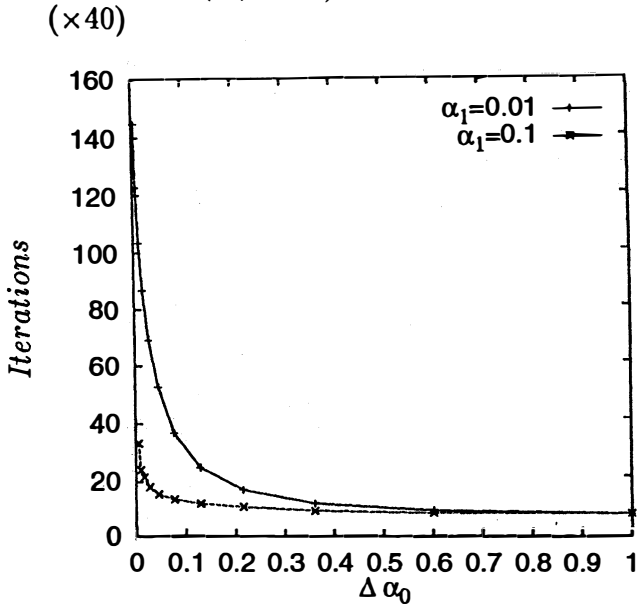


Fig. 5 Numbers of iteration

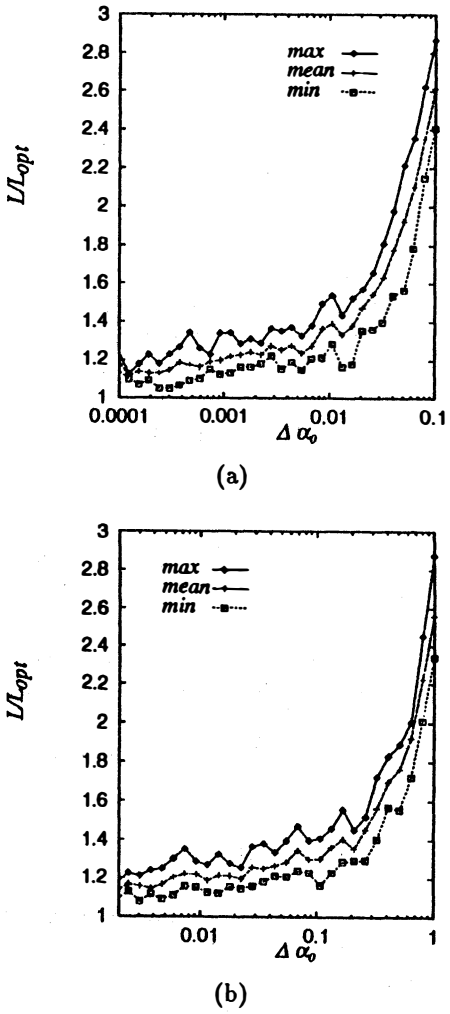


Fig. 4 L/L_{opt} as a function of $\Delta\alpha_0$
(a) : $\alpha_1 = 0.01$, (b) : $\alpha_1 = 0.1$
L_{opt} : length of the optimum path

An Advanced Simulation Scheme for Electric Railway Power Systems Based on Artificial Life Approach

T.Jinzenji and M.Sasaki
Electrical Engineering Department, Faculty of Engineering
Tohoku University
Aramaki Aza Aoba, Aobaku, Sendai 980-77, Japan

Abstract

This paper describes a simulation scheme for DC electric railway power systems taking train dynamic behaviors into consideration. This simulation is useful to determine the rated capacities of new DC substation equipment. In the present simulation, train operating conditions are assumed. Therefore, simulation results are quite different from the field data when a fleet behaves like lives under crush load circumstances. Two factors determine the behaviors. One is the passengers' dynamic flow at every station, which causes unexpected delay. The other is the railroad signal, which keeps trains apart in an appropriate interval. These behaviors must be introduced into the simulation. Typical simulation result of the advanced simulation scheme shows that the feeder current variations of the simulations are similar to field data.

Keywords: Electric train, Train dynamic behavior, DC system simulation, Life

1 Introduction

Over the past few decades, DC electric railway power system simulations consisted from three steps, train run curve simulation, diagram editing and DC power simulation.

The first step had been based on railroad conditions, train performance, number of passengers and scheduled time, assuming that the pantograph voltage will be kept constant. In the second step, a diagram had been edited according to the typical train run curves and the time table. The diagram assumes that the train runs will be repeated in a specified time.

In the last few years, three simulation steps are joined to make sure that the simulation and the field data provide same results(see [1]). However, little attention has been given to the motormen and passengers. Even if the trains try to follow up on a diagram during the crush load circumstances, the train operations are restricted physically and controlled by the railway signals.

The field data of DC substation are often different from the simulation results, especially in the rainy morning, as shown in Fig.1.

This paper deals with an advanced simulation scheme for DC electric railway power systems considering how trains behave in a fleet. They behave some types of patterns like a life. We introduce these behaviors to simulations for DC electric railway power system.

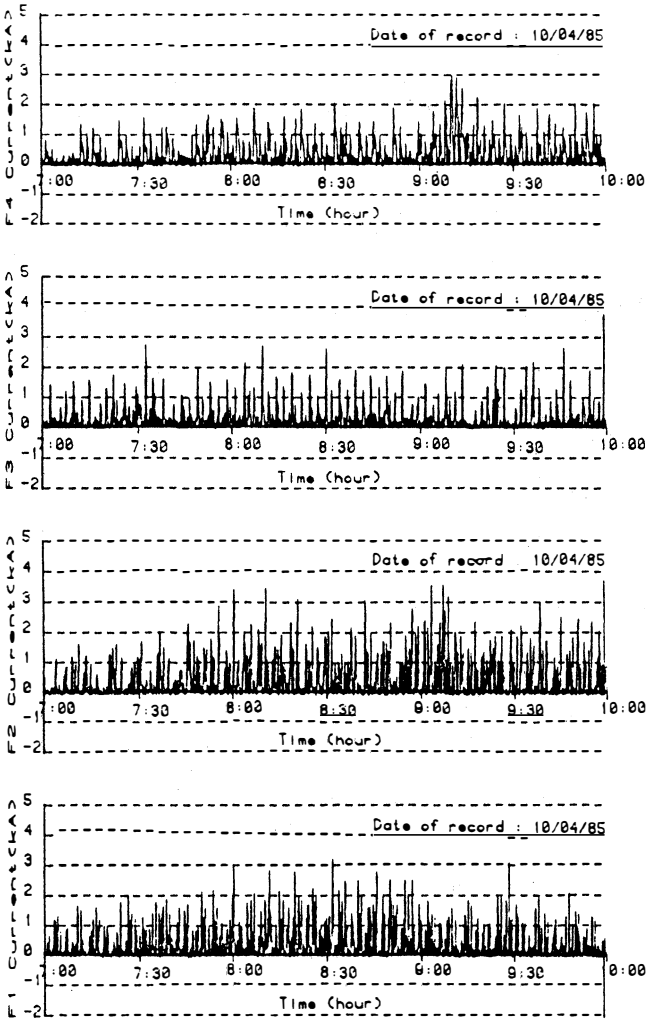


Fig.1. Field data of current of substation.

2 Electric Railway Power Systems

Fig.2 shows DC electric railway power systems. DC substations locate within five to ten kilometers and are connected in parallel. They convert AC supply voltage into DC. They supply DC power to trains in up and down lines at up and down directions in parallel. Fig.3 shows the equivalent circuit of DC electric railway power systems on one side ,one section of substations, where source of voltage means substation voltage, source of current means train current. Solving this equivalent circuit leads to following equations. Since there are actually up and down lines on up and down directions, whole current value of substation is calculated by adding four current values.

$$I_{s1} = \{V_{s1} - V_{s2} + \sum_{i=1}^{n} r(L - x_i) I_{Li}\} / rL \quad (1)$$

$$I_{s2} = \{V_{s2} - V_{s1} + \sum_{i=1}^{n} r x_i I_{Li}\} / rL \quad (2)$$

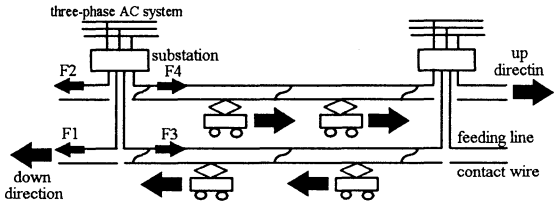


Fig.2. DC electric railway power system.

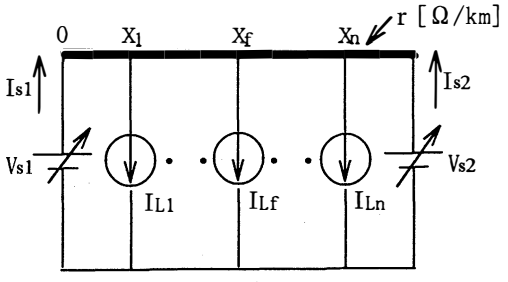


Fig.3. Equivalent circuit.

3 Train behavior

3.1 Restrictions

Train operations are restricted physically. So if it can always get constant voltage, it can move regularly. Even if many trains run at the same time and they can't get constant voltage, we can estimate these actions by solving train motion equations.

However, trains are restricted by signal conditions and passengers, too.

As for signal condition, a block system is used. Fig.4 shows a block system. There are some block sections in a line. To keep trains safe, there is a block signal before each section. If there is a train in this section, it indicates red signal and tells following motorman not to enter this section. If we put too many sections in a line, there can be many trains in the line, but block signals are almost red and the velocities of trains get slow as a whole.

In respect of passengers, they cause late from an expected schedule, especially in the morning rush hour. The numbers of passengers go on increasing from about 8:00 to 9:00, (generally Japanese office opens at 9:00) and decrease gradually after 9:00.

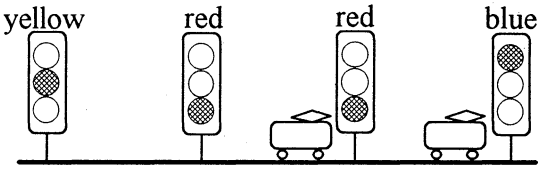


Fig.4. Block system.

To take these conditions into account, train runs assume following restrictions.

a) A train will be controlled by signals (signal condition),

b) A train can be late from an expected schedule (train diagram conditions),

c) The numbers of passengers are established corresponding to time zone.

The trains will recover time delay as early as possible while it is late from a diagram.

3.2 Train behavior

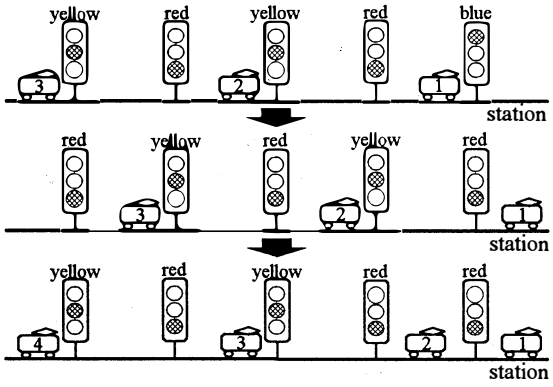
Broadly speaking, there are three types of behaviors of the fleet.

First, for the time zone that there is a great increase in the number of passengers, stoppage time at substations gets increasing. Because they spend a lot of time in getting on and off, which shorten the distance from the stopping train to the next running one. Accordingly each distance of trains will get short (see Fig.5(a)).

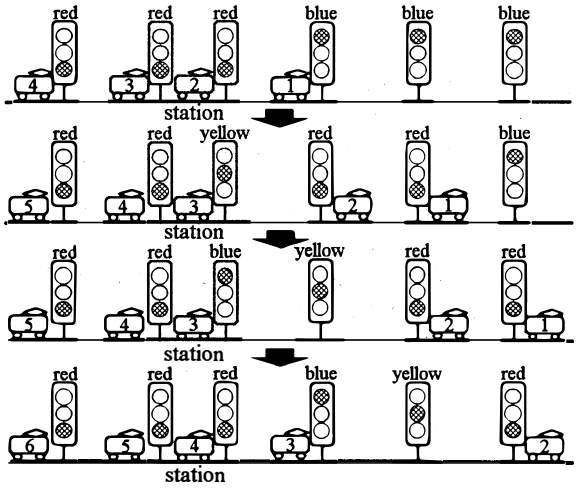
Secondly, for the time zone that there is a slight increase in the number of passengers, the distances of trains are fluctuating. The reason is that the forward train stops long time because there are many passengers in the station, while

the following one stops schedule time because there aren't so many passengers that cause time loss. In this way, these behaviors will be repeated (see Fig.5(b)).

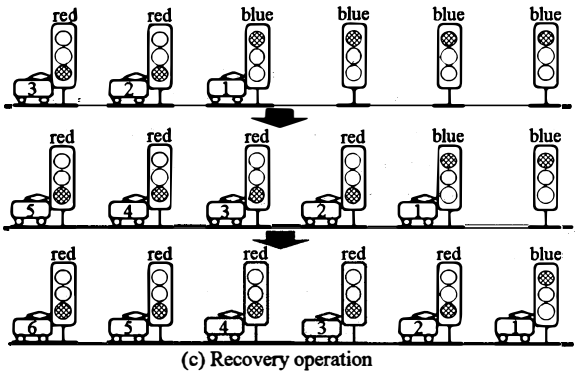
Thirdly, for the time zone that there are less than the number of passengers that cause no time loss. If trains are late from the expected schedule, they all together tend to recover time as early as possible without coasting so far as signals permit. The distances of trains will keep short(see Fig.5(c)).



(a) Narrowing distances between trains.



(b) Trains in clusters.



(c) Recovery operation

Fig.5. Behaviors of trains in a fleet.

To make sure the behaviors of trains in a fleet under these circumstances, we established a simple model for train running.

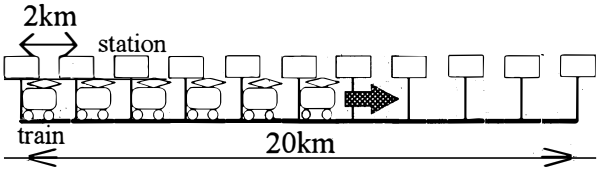


Fig.6. Simple model for train running.

Fig.6 shows a typical location about stations and trains. The stations locate as intervals of two kilometers in parallel. Six trains start all together for the same direction from each station. At this simulation, we change the number of persons which is related to delay at the station.

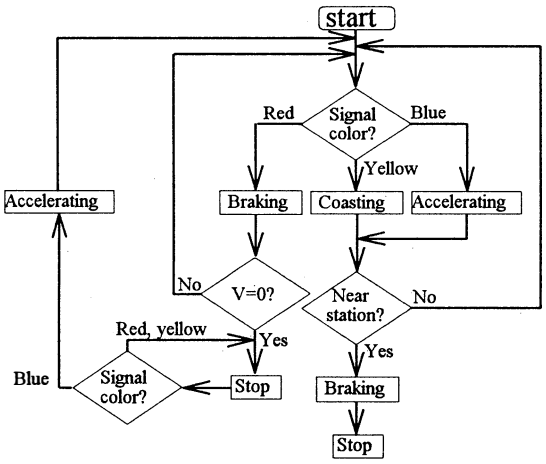
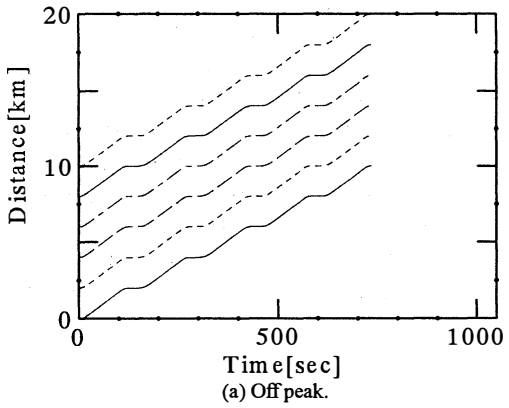


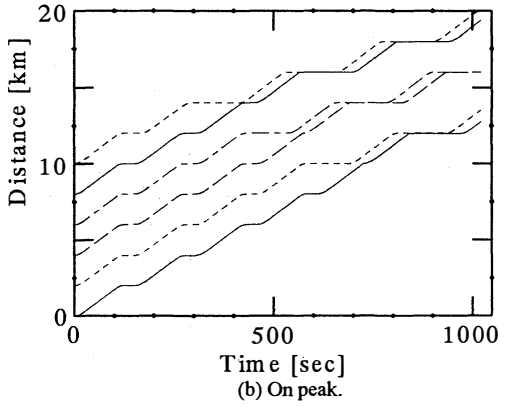
Fig.7. Flow chart of train operating mode.

Fig.7 shows the flow chart of train operating mode by signals from a station to the next one. The results are presented in Fig.8. We can find behaviors of trains like a life.

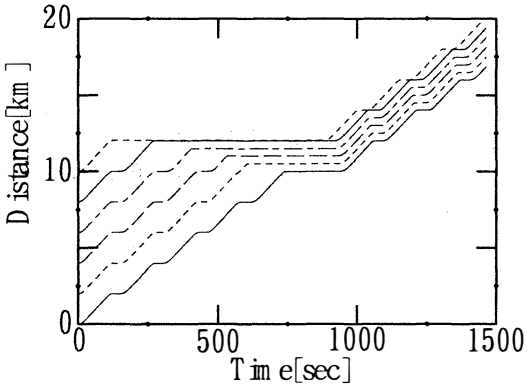
Fig.8(a) shows the scheduled running pattern which is off peak. There is a small number of persons and trains which stop scheduled time every station. The trains almost keep same intervals. Fig.8(b) shows the train runnings which is on peak. Fig.8(c) shows that there is a long stop at the first train and recovers again.



(a) Off peak.



(b) On peak.



(c)Long stop and recovery.

Fig.8. Time-distance curve of trains.

4 Practical simulation

4.1 Conditions and results

Table I shows the simulating conditions under signal conditions. Table II shows increase rate of persons per a second corresponding to time zone. Until a train comes persons increase up to 600. The delay time is define as:

$$T = \begin{cases} 0 & (P \leq 200) \\ (P - 200)/10 & (P > 200) \end{cases}$$

T : Delay time

P : Number of person at station

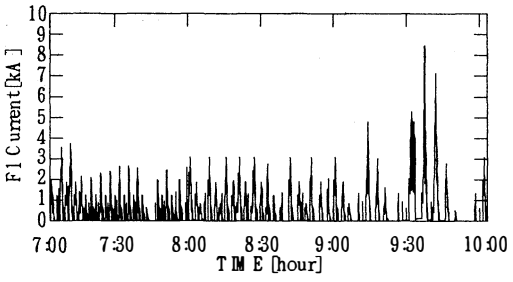
Table I Simulating conditions

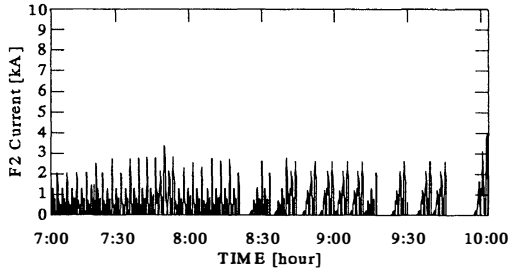
conditions	contents
railroad conditions	model:Toei Asakusa line(18.3km) number of substations:5 number of stations:20 line resistance:0.03Ω/km
train performance	5000series resistor control maximum speed:70km/h
operating conditions	train formation:6M
	train weight:290.4t (crush load 200%)
	diagram:5minute headways
computer conditions	sampling time: 1sec simulating time: 3hour

Table II
Establishment of number of persons

	time zone					
station	7:00~7:30	7:30~8:00	8:00~8:30	8:30~9:00	9:00~9:30	9:30~10:00
ordinary station	6	7	8	7	4	3
junction station	8	9	10	9	5	4

unit [person/sec]





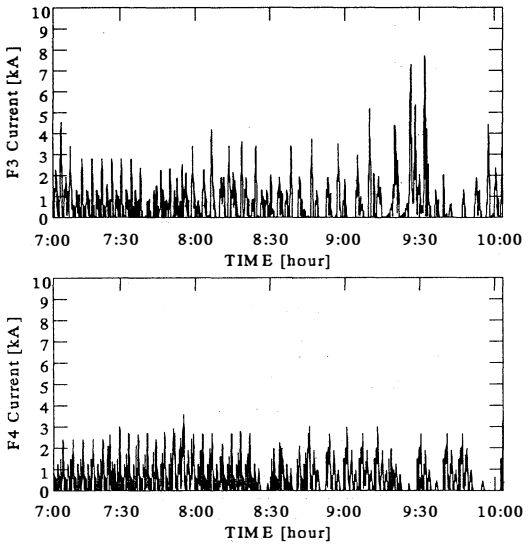


Fig.9. Current of substation.

The results are shown in Fig.9.

4.2 Discussion

Compared with the field data, we can say that the simulation results have a characteristic similar to the field data. For example, there are peak current after 9:00. It means that trains run as fast as possible to recover time in this time zone. But there are some different points between two figures.

First, the simulation current is higher than the field data on the whole, especially peak current after 9:00. It is for this reason that the numbers of persons are roughly established corresponding to time zone. And the equation which is defined as delay time due to persons in getting on and off may cause an error in calculation.

Secondly, when a train accelerates, much current flows in a train. So when trains accelerate crowed together at the same time, there is a steep increase of the current of substations. Though a driving interval is estimated, the load current is practically dispersed because of the other factors except stoppage time.

5 Conclusion

The behaviors of trains in a fleet are restricted by signal and the passengers' flow. Under these circumstances, trains behave like a life. They seem to have a defined will, but their behaviors are restricted by some factors. By recognizing their behaviors we can calculate current values of substation precisely.

Most important of course is the determination of the human factor which is fed back from the field data in our simulation. The human factor should be given according to the artificial life practice in the future.

References

[1]R. Takagi and S.Sone, "Precisely Fixed Start-to-Stop Time Simulation of DC Railway Power Feeding Systems,"Transaction of IEE Japan, Vol.115-C, No.8, pp.975-983,1995.

Adaptive Block Dividing and Qunatization for Image Compression using Artificial Breedings

†H.Hiramatsu,††Y.Kurematsu and ††S.Kitamura
†Department of Computer and Systems Engineering Kobe University, Kobe, 657
††College of Industrial Technology,Amagasaki,667

Abstract

This paper describes a method of adaptive blocking and quantization of a digital image compression based on the genetic algorithms. In conventional compression techniques, an original image is divided into small blocks and the discrete cosine transform(DCT) is applied on each block. The resulting coefficient is then quantized by the quantization table and coded for transmission. Subblocks have same size and the quantization table is generated by experience or experimentally, so that the feature of image is not reflected. We have applied the genetic algorithm and artificial breeding techniques for both problems. In this method, elements of the quantization table and the size and position are regarded as a genotype. Fitness of each genotype is examined by subjective evaluation. It is shown that better quantization tables and block dividing methods matching to original images can be obtained.

1 Introduction

Global computer networks have become widespread in recent years, and researches of multimedia, which integrates images, sounds, text and so on, have become an important subject. Especially, digital image data is most important media, and the high performance of image compression techniques have been pursued for fast transmission and low cost storage. For this purpose, many techniques have been proposed, for instance, the discrete cosine transform(DCT) and the vector quantization.
The international standard for digital video and still-image compression and decompression was developed by the Joint Photographic Experts Group(JPEG)[1]. This algorithm uses the DCT of 8×8 block size, a quantization table and Huffman coding. But this transform coding technique has a problem of deterioration of image quality such as blocking and mosquito noises in reconstructed images.[2] There are two methods for improving this situation. The first is an adaptive quantization taking into account human's visual characteristics.[2] The another is an adaptive blocking (*i.e.* variable block size) which emphasizes various features of a given image.
In this paper, we propose a method for generating a quantization table and dividing of images by the genetic algorithm. In this procedure, the artificial breeding technique was used by taking into account human's subjectivity (or sensibility) for reserving the features of original images in the reconstructed images.
It is shown that better quantization tables and block dividing method matching to the features of original image can be obtained. Noises of reconstructed images were decreased, and the quality of reconstructed image was better than that of the JPEG algorithms, with high compression rate.

2 Techniques of Digital Image Compression

A diagram of image compression systems is shown in Figure 1. An image is divided into subblocks of size $N \times N$ and 2-*Dim* DCT is applied for each block $f(x,y)$ to produce the DCT coefficient block $F(u,v)$ of same size [1][2].

$$F(u,v) = \frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot \cos\left[\frac{\pi u(2x+1)}{2N}\right]\cdot\cos\left[\frac{\pi v(2y+1)}{2N}\right] \quad (1)$$

where

$$C(i) = \begin{cases} \frac{1}{\sqrt{2}} & (i=0) \\ 1 & (i\neq 0) \end{cases}$$

The resulting coefficient array $F(u,v)$ is then quantized by a quantization table and coded for transmission.

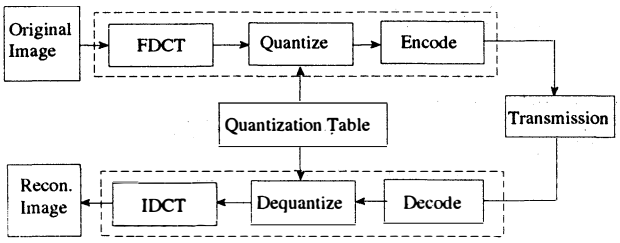


Figure 1: DCT block diagram

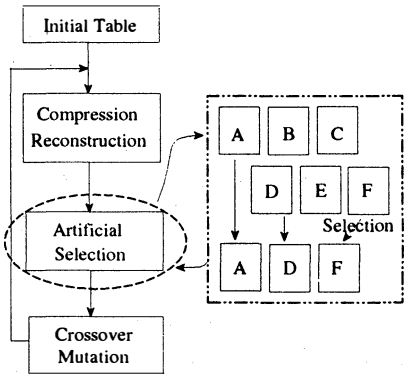


Figure 2: Flow of generating the quantization table

3 Generation of Quantization Table by Artificial Breedings

Normally, the performance of image compression techniques is evaluated by the quality of reconstructed image and compression rate, where quality of images has been evaluated by objective (or numerical) values. However, the goodness of images depends much on human's subjectivity. This problem has been left open, since we did not have an appropriate method to take into account the human's subjectivity.
The quality of reconstructed images is dependent on the quantization table. In most researches in this field, they have been generated by heuristic or experimental method. In the following, we propose a method for generation of a quantization table by taking into account human's subjective evaluation as well as the numerical evaluation.

3.1 Generating the quantization table

We define first an individual with a chromosome, as a genotype in the GA, consisting of elements of a quantization table. We apply the JPEG compression procedure with this quantization table for a given image. This process was repeated for all individuals in the population. Next, we select better individuals from reconstructed images (*i.e.*phenotype) by the subjective evaluation as well as numerical values of the compression rate and the number of subblocks. Finally, the population in the next generation is generated from the selected individuals by the genetic algorithm operations. The flow of this algorithm is shown in Figure 2. The followings are important parameters:

Population Number of individuals was 6 for easiness in artificial selection.

Genotype A qunatization table is transformed from 2-*Dim* array to 1-*Dim* array in the binary form. The array obtained is a genotype (Figure 3).

Genetic operations As reproduction operations, the single-point crossover and the mutation were used.

Fitness This is not numerical but operator's subjectivity. The operator selects better 3 images from reconstructed 6 images.

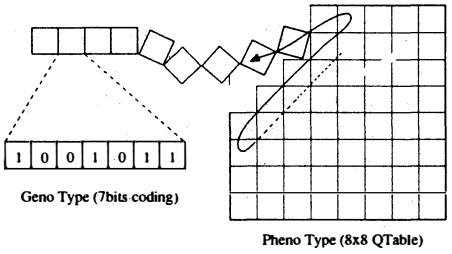


Figure 3: Coding of quantization table

3.2 Results of Simulations

The algorithm was repeated for 100 generations. Figure 4 shows a reconstructed image at the first generation, and Figure 5 at 100th generation. The mean square error(MSE) and the compression rate of both images are shown in Table 1.

Figure 4: Image at Initial Generation

Figure 5: Image at 100th Generation

Table 1: MSE and Compression rate

	MSE	Compression rate
Image at first gen.	140.4	9.1
Image at 100th gen.	47.7	8.9

The blocking and mosquito noises were obviously reduced in Figure 5. Also the values of MSE were improved with almost same compression rate. These results show an extensibility of the proposed method for more general images.

4 Adaptive blocking

Generally in the transform encoding method, an original image is divided by $N \times N$ arrays of pixels with a constant N. This method is simple, however, neglects various features in the image. The situation is improved by variable block size methods.[4],[5] In the following, we propose a method for optimal block sizing by applying the GA.

4.1 Adaptive block dividing

In this method, the size and position of each subblock are defined as a genotype. An original image is divided into subblocks according to this genotype. The image coding techniques are applied for each block. The fitness function was defined in terms of MSE, bitlength(*i.e.* compression rate) and the number of subblocks for a reconstructed image(*i.e.*pheno type). The followings are important parameters:

Genotype The number and position of subblocks are coded as shown in Figure 6.

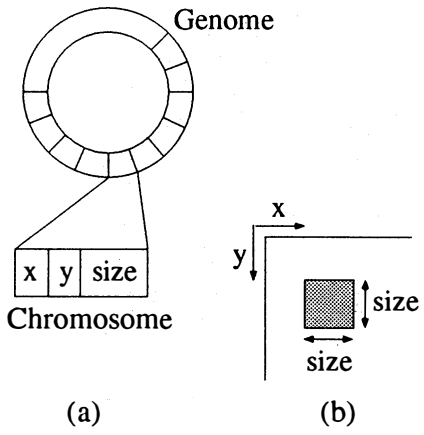


Figure 6: Definition of genotype

Block Each subblock is a square with $4\times4, 8\times8, 16\times16, 24\times24, 32\times32$ or 64×64 pixels.

Genetic Operations We used two-points crossover and the mutation.

Fitness The fitness function for nth individual is defined by

$$f(n) = w_m M + w_b B + w_s S \qquad (2)$$

where M, B and S denote the scores of MSE, bitlength and the number of subblocks, and w_m, w_b and w_s are weighting constants.

4.2 Results of Simulations

Figure 7 shows an image at the first generation, and Figure 8 that at 500th generation.

Figure 7: Image at first generation

Figure 8: Image at 500th generation

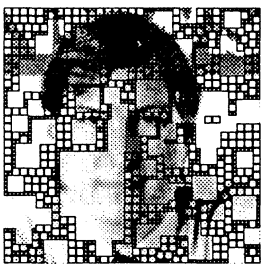


Figure 9: Subblocks at first generation

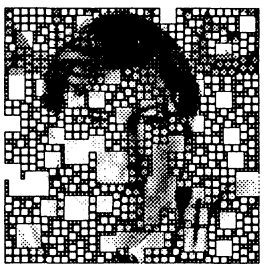


Figure 10: Subblocks at 500th generation

The distribution of subblocks corresponding to Figures 7 and 8 are shown in Figures 9 and 10. In Figure 10, we can observe that small subblocks are allocated near the edges in the image and large subblocks at the homogeneous areas. This shows the effectiveness of the proposed method.

5 Conclusion

Generating a quantization table and an adaptive blocking method by the genetic algorithm were proposed. It showed better and reasonable results as an effective data compression method. As shown here, the algorithm based on the evolutionary computation is rather simple, while it can enable us to take into account human's subjectivity (or sensibility) for evaluation of the features of images.

References

[1] H.Yasuda: International standard of multimedia encoding, pp.1-47, Maruzen 1991

[2] K.R.Rao and P.Yip: Discrete Cosine Transform, Academic Press 1990.

[3] Rafael C.Gonzalez and Paul Wintz: Digital Image Processing, Addison-Wesley 1977

[4] Arie Heiman,Variable Block-Size Transform Image Coder, IEEE Transaction on Communications,Vol.38,No.11,November 1990

[5] S.Okada: A Study on Block Dividing Method for Variable Blocksize DCT Coding, The Transaction of the Institute of Electronics, Information and Communication Engineers, Vol.J75-D-II, No.8 pp.1386-1396, August 1992.

Evolution of Locally Defined Learning Rules and Their Coordination in Feedforward Neural Networks

Hajime Murao* Shinzo Kitamura†
Computer and Systems Engineering,
Kobe University,
Kobe, 657

Abstract

This paper shows how an evolutionary process can coordinate locally defined learning rules in a neural network. Each of these rules is defined as a function of only local information for the corresponding neuron. Learning rules are exposed to the selective pressure based on the fitness values which are calculated from errors between teaching signals and output signals of the network. That is, if the fitness value of a learning rule becomes less than that of an appropriate number of learning rules, this rule is replaced by new one. As a result, the efficient learning rules survive in the network. The locally defined learning rules coordinate each other and generate the macroscopic adaptive behavior of a neural network.

1 Introduction

Learning is one of the most fundamental processes of organisms' adaptation. To utilize this adaptive behavior, various artificial neural networks (ANNs) and associated learning methods have been proposed. These methods are based on physiological aspects; changes of the connective strength between neurons play an important role during the learning of organisms. The methods calculate the changes of connective strength in an ANN by using errors between teaching signals and output signals of the ANN. Currently, physiology can not explain how the changes of connective strength coordinate each other and generate macroscopic adaptive behavior of the brain. Practical learning methods for huge and complex ANNs are not currently available, either.

Evolution is another process of organisms' adaptation. The structure and the functions of the brain are both gained as a result of evolution. Recently, to cope with huge and complex ANNs, a relatively large number of studies on combinations of evolutionary computing and ANNs have been reported[1]. In most case, genetic algorithms have been used to adapt the connective strength in a neural network, or to find an architecture for a neural network. These methods have yielded interesting results on the optimization of network design. Much interesting work has been done on the evolution of learning[2][3]. However, only parameters in the learning function of ANNs have been optimized.

In this paper, we apply genetic programming (GP) to evolve learning rules in an ANN. First, we outline a general framework to describe a learning rule which only use local information of the corresponding neuron. Then, GP is employed to evolve learning rules and their coordination suitable for supervised learning of layered ANNs. Learning rules are encoded into genomes and exposed to the GP operation.

*murao@kobe-u.ac.jp
†kitamura@kobe-u.ac.jp

2 Framework for Evolution of Learning in ANNs

2.1 Neuron Model for Our Simulations

In this section, we specify the neuron model which is used in our ANNs. Figure 1 shows the signal flow of neuron i. An output signal $y^i(t)$ of neuron i at time t is given by

$$u^i(t) = \sum_j w^i_j(t) x^i_j(t) \quad (1)$$

$$y^i(t) = f(u^i(t)) \quad (2)$$

where $u^i(t)$ is an internal variable of the neuron, which is given as a weighted sum of input signal $x^i_j(t)$ and connective strength $w^i_j(t)$. We use a sigmoid function as f defined by

$$f(x) = \frac{1}{1+e^{-x}} \quad (3)$$

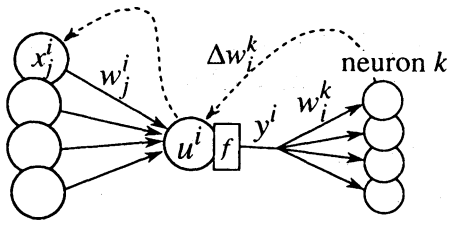


Figure 1: Neuron i calculates its output signal $y^i(t)$ at time t from the internal variable $u^i(t)$, which is a weighted sum of input signals $x_j^i(t)$ and connective strength $w_j^i(t)$. $\Delta w_i^k(t)$ is a change of connective strength between neuron i and a downstream neuron k.

2.2 Framework for Learning Rule

We define a learning rule for neuron i as a function of only local variables for the neuron,

$$\Delta w_j^i(t) = g^i(y^i(t), u^i(t), x_j^i(t), w_j^i(t), \delta^i(t)) \quad (4)$$

where $\delta^i(t)$ is defined by

$$\delta^i(t) = \sum_k \Delta w_i^k(t) \quad (5)$$

$\Delta w_i^k(t)$ is a change of connective strength between neuron i and following neuron k. When neuron i is an output neuron, we put a teaching neuron $k(= i)$ next to the neuron i. The teaching neuron k is employed for generating $\Delta w_i^k(t) = t^i(t)$ or $\Delta w_i^k(t) = t^i(t) - y^i(t)$.

2.3 Genetic Coding of the Learning Rule

Now, we specify how the GP is employed to evolve learning rules that have been represented in Section 2.2.

A learning rule of a neuron is encoded into a genome which is constituted by binary trees. Figure 2 shows a genome and corresponding learning rule. Internal nodes of the trees correspond to functions (i.e. operations) and external nodes correspond to terminals (i.e. input signal). We define the set of functions F and the set of terminals T as follows;

$$F = \{+, -, /, *\} \quad (6)$$

$$T = \{y^i, u^i, x_j^i, w_j^i, \delta^i, 2^{l-3}, -2^{l-3}, S+1, S+2, S+3\} \quad (l = 0, 1, ..., 4) \quad (7)$$

We include connecitve terminals $S+1$, $S+2$ and $S+3$ in the terminal set. When a connective terminal in a genome is evaluated, it evaluates another tree which constitutes the same genome and returns the result.

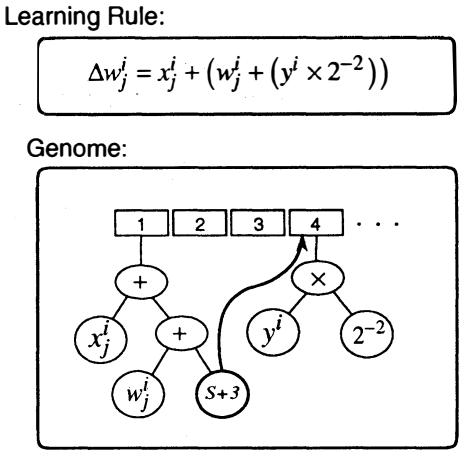


Figure 2: A genome and corresponding learning rule. Internal nodes of the tree corespond to functions, external nodes correspond to terminals.

2.4 Fitness Value of the Genome

To evolve coordination among the learning rules, we have to specify fitness values for the learning rules based on the success of an ANN in producing correct output signals. The fitness values are given by

$$fitness = 100 \times \left(1 - \frac{E}{N}\right) \quad (8)$$

where E is a total number of errors of an ANN for training data sets, N is the number of training data sets. As each output signal of the ANN ranges from 0 to 1, Eq.(8) yields a fitness value between 0 and 100. A fitness value is calculated for an ANN and is given to all learning rules in the ANN. In our simulation, the fitness value is given to all neurons which constitute the ANN.

2.5 Genetic Operations

We use three genetic operations to modify genomes: reproduction, recombination and mutation. The reproduction operation removes a genome, for which the fitness value is less than that of appropriate number of neurons, and generate new one by recombination. A number of neurons which determine a threshold for the fitness values is given by multiplying a population of genomes and a reproduction rate P_r.

The recombination operation creats new offspring genomes by exchanging sub-trees between randomly chosen two parental genomes.

The mutation operation creats new genome by replacing a node with one of the functions and the terminals, which is randomly chosen from the set F and

T. The mutation operation has a probability of P_m for each node.

3 Evolution of Learning in Single-Layered ANNs

3.1 The ANN Architecture

We use single-layered ANNs for our simulation. The ANN consists of 5 input neurons and 1 output neuron, which is described in Figure 3.

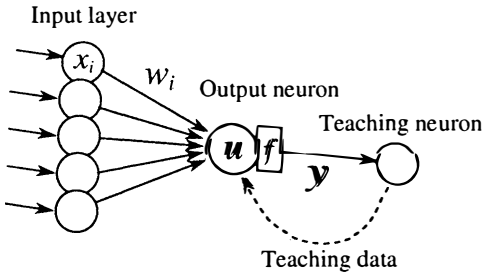


Figure 3: Fully-connected single-layered feedforward ANN. Whole network is trained by the learning rule of the output neuron.

Input neurons make no changes on input signals but transmit them to an output neuron. The neuron model described in Section 2.1 is used for the output neuron. A teaching neuron transmits the teaching signals as δ for output neuron. The learning rule of the output neuron calculates changes of connective strength between input neurons and the output neuron. Whole network is trained by the learning rule of the output neuron.

3.2 Evaluation of Fitness

Fitness values of each ANN is evaluated as follows:

1. Create training data sets. Each data consists of 5 input signals and randomly chosen 1 teaching signals. As our ANN has 5 input neurons, we can prepare N training data sets out of 32.
2. Create a single-layered ANN, for which each connective strength is initialized randomly between -1 and 1.
3. For a number of epochs (typically 10), the ANN is learned trough the training data sets. We propagate input signals trough the ANN and employ to yield output signal for each training data set. Then, the connective strength in the ANN are adjusted according to the learning rule of output neuron.
4. Calculate a fitness value from errors between teaching signals and output signals according to Eq. (8).

3.3 Results of Simulations

We trained a population of 100 single-layered ANNs through 5000 generation, where reproduction rate $P_r = 0.1$ and mutation rate $P_m = 0.001$. Figure 4 shows resulting fitness values and fitness values for testing data sets.

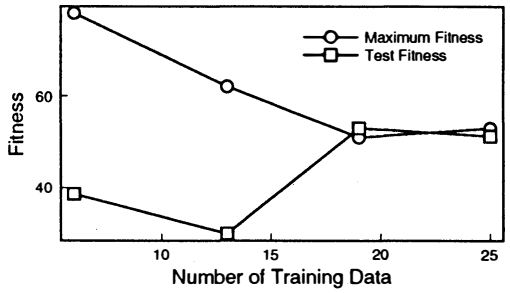


Figure 4: Resulting fitness value for various number of training data sets. "Maximum fitness" shows the greatest fitness value for the training data sets. "Test fitness" shows a fitness value for testing data sets, which is given by an ANN that gives the greatest fitness value for training.

Testing data sets consists of 32 testing data set, for which we determine randomly chosen teaching signal for each input signals. As a number of training data sets increasing, fitness values for testing data sets also increased. This means that the diversity of the training environment is important to acquire a learning rule that is efficient for different environments.

As a result of the simulation for $N = 19$, we observed a version of the well-known delta rule (or Widrow-Hoff rule),

$$\Delta w_i = y \cdot (t - y) \cdot x_i \tag{9}$$

We could also find versions of the delta rule for other N, but they yielded less fitness values than the most efficient rules. Figure 5 shows evolution of fitness values for $N = 19$.

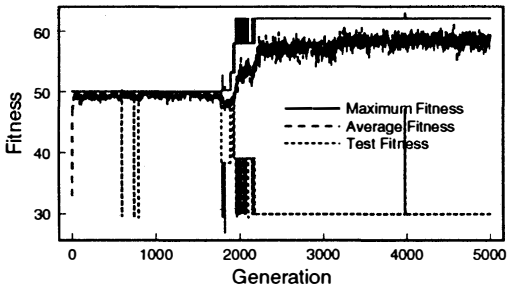


Figure 5: Evolution of fitness values for single-layered ANNs ($N = 19$).

4 Further Experiments with Multi-Layered ANNs

4.1 The Architecutre

Furthermore, we experiment with three-layered ANNs. The ANN consists of 5 input neurons and 1 output neuron, which is described in Figure 6.

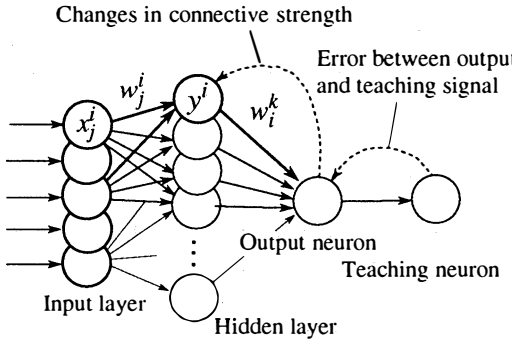


Figure 6: Fully-connected 3-layered feedforward ANN.

Input neurons work as "fan out neuron" that transmit input signals to hidden neurons only. The neuron model described in Section 2.1 is used for the hidden neurons and the output neuron. A teaching neuron transmits errors between teaching signals and output signals as *delta* for output neuron. Connective strength between input layer and hidden layer is updated by learning rules of hidden neurons. Connective strength between hidden layer and output neuron is also updated by learning rules of output neurons.

4.2 Results of Simulations

We trained a population of 100 multi-layered ANNs through 5000 generation, where a number of hidden neurons is 5, reproduction rate $P_r = 0.1$ and mutation rate $P_m = 0.001$. Fitness values of each neuron which consists ANNs is calculated by Eq. (8). Figure 7 shows evolution of fitness values for $N = 19$. Maximum fitness values fluctuate but are fairly good over 55 points.

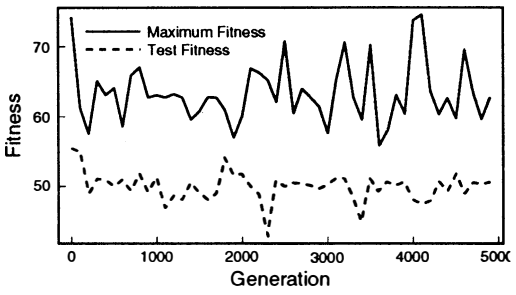


Figure 7: Evolution of fitness values for three-layered ANNs ($N = 19$).

5 Discussion and Future Direction

We have proposed a framework of the evolution of learning rules in ANNs. Through the numerical simulations on single-layered ANNs, we have shown an ability of our method to evolve efficient learning including the delta rule.

Furthermore, we have shown fairly good adaptive behaviors by three-layered ANNs, though they are trained only by locally defined learning rules. It seems that the coordination of the learning rules is evolved. We should analyze what kind of learning rules are evolved in three-layered ANNs.

We think the evolution of learning rules and their coordination are depend on how to distribute fitness values on neurons. Future improvements concentrate on distributing a fitness value calculated from ANN's behavior to each neuron which consists a multi-layered ANN.

References

[1] John R. Koza and James P. Rice, Genetic Generaion of Both The Weights and Architecture for A Neural Network, IJCNN-91-Seattle, Vol.II, pp.397-404, 1991

[2] David J. Chalmers, The Evolution of Learning: An Experiment in Genetic Connectionism, *Proc. of the 1990 Connectionist Models Summer School*, San Mateo

[3] Takashi Sumikawa, Studies on Acquiring of Learning Rules of Neural Networks, Master Thesis of Graduate Instrumental Engineering, Kobe University, 1994

A Self-Organization Learning Algorithm for Visuo-Motor Coordination in Unstructured Environments

H. B. Zha T. Onitsuka T. Nagata

Department of Computer Science and Communication Engineering
Faculty of Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812 JAPAN

Abstract

We present a new method for coordinating a robot system where a manipulator controller is directly connected to a stereo camera system. We try to avoid the calibration process between the manipulator and the vision system by dividing the whole nonlinear transformation into local linear transformations. The linearization is achieved by a learning process based on a Kohonen-style self-organization network. We give some modification to the learning process in two aspects. Experimental results are reported to show the properties of our method and its applicability.

Keywords : self-organization, visuo-motor coordination

1 Introduction

Many researches on robots equipped with some kind of sensors have shown the effectivenesses of use of sensory data. It may be still poor in accuracy and reliability if we have no appropriate methods to have them make quick reactions to the sensor signals. It is, therefore, very important to develop efficient techniques for coordinating robot motions according to sensor signals.

A manipulator coordination based on a camera systems is usually referred to as visuo-motor coordination. The visuo-motor coordination problem is defined in a more specific form how to make an end effector of the manipulator move to a target pose, which is designated through a camera system. Essentially, the control should be carried out by performing a nonlinear transformation from an arbitrary pose in its image coordinates to the joint-angle space of the manipulator. One of conventional methods to perform such transformation is to depend on a precise calibration of the camera system and the manipulator. Even calibrated by time-consuming procedures, it may be still poor in accuracy and reliability according to utilized instruments. Moreover, the system based on precise calibration is also very vulnerable to changes in the working environment.

There has existed some researches for the visuo-motor coordination with a large degree of adaptation capability on the basis of neural network techniques. In the coordination of the end effector with stationary cameras, Kuperstein [3] reported a method for mapping stereo disparities to robot joint angles directly by a nonlinear network. However, for initializing the fixed map it is necessary to give more a priori knowledges about input information and the learned map is hence not able to deal with homogeneous inputs. Walter *et al.* [4] evolved the idea into the dynamic mapping resolution based on a self-organizing feature map proposed by Kohonen [1] [2]. However, to achieve an accurate mapping, a large number of learning steps should be used. As a result, it will take large computational costs.

In this paper, we present a new method for coordinating a robot system where a manipulator controller is directly connected to a stereo camera system. Clearly, the relationships between the set of visual data and joint angles relating pose of the end effector are nonlinear. In our method, this whole nonlinear transformation is decomposed into local linear transformations and the linearization map is recorded by the manipulator controller. The linearization is achieved by a learning process based on a Kohonen-style self-organization network.

We give some modification to the learning process in the following two aspects: The first is the reduction of huge computational costs required in implementation practically; The second is the investigations about the distributions of the neuron cells in more complex situations. Experimental results are reported to show the properties of our method and its applicability.

2 Visuo-motor coordination

A typical make-up of a visuo-motor coordination system is shown in Fig. 1. In general, learning of the coordination is carried out by iterating the following steps: 1) For one designated target in random, joint angles are set so that the end effector of the manipulator moves to a nearby pose of the target; 2) Visual information about the end effector is extracted by processing two images obtained by the stereo camera system; 3) The visuo-motor coordination is modified in order to decrease the error between the resultant pose

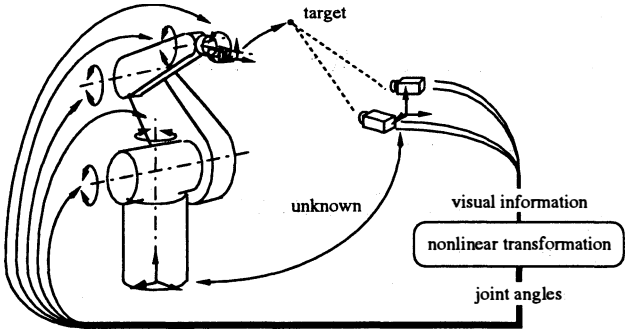


Fig. 1: Visuo-motor coordination

and the target. Essentially, the joint angles have to be associated directly with the visual information on the end effector poses.

The input and output signals we choose are described as follows. First, input is given as each two-dimensional image coordinates of the target which is identified in the two camera images. Next, parameters are essential to control the manipulator as output, which correspond to each of the joint angles. Consequently, the visuo-motor coordination becomes a problem of how to identify the nonlinear transformation from an input vector $\boldsymbol{u}_t$ to joint angles $\vec{\boldsymbol{\theta}}$, which is represented by eq. (1).

$$\boldsymbol{u}_t \subseteq \boldsymbol{R}^n \;\mapsto\; \vec{\boldsymbol{\theta}} \subseteq \boldsymbol{R}^m \tag{1}$$

3 Self-organization network for visuo-motor coordination

The nonlinear transformation, shown in section 2, is divided into local linear transformations on the self-organization network. At first, input space (that is, vector space formed by the two camera image coordinates) is quantized into several cells, each of them associated with a neuron. Then, each of the neurons is modeled with parameters that have been described in previous section, and stores three elements as follows: 1) Reference vector $\boldsymbol{w}$ which corresponds to input space and indicates the center coordinates of a cell; 2) Joint-angle vector $\vec{\boldsymbol{\theta}}$ which makes the end effector move to the pose as represented by the reference vector; 3) Jacobian matrix $\boldsymbol{A}$ which indicates the first derivatives of the joint-angle vector with respect to the reference vector. As the result of the self-organization based on the following learning rule, these relationships are described as

$$\vec{\boldsymbol{\Theta}}(\boldsymbol{u}_t) = \vec{\boldsymbol{\theta}} + \boldsymbol{A} \cdot (\boldsymbol{u}_t - \boldsymbol{w}) \tag{2}$$

where target vector $\boldsymbol{u}_t$ is derived by a random motion of a manipulator in each learning step and $\vec{\boldsymbol{\Theta}}(\boldsymbol{u}_t)$ is the locally and linearly transformed control signals defined inside a topological neighborhood around the target vector. For any cell under consideration, its neighborhood is defined by a neuron ordering procedure. Given the reference vector of the cell, the other neurons are ordered by the Euclidean norm between this reference vector and those computed by eq. (2) for all of them. The learning rule has following characteristics owing to the introduction of the topological neighborhood. The neighboring units charge analogous patterns, that is, the analogy of input patterns is reflected to the closeness on the position of responding output units.

Fig. 2 illustrates the correction of each element by applying the following learning rules. The length of the arrow on the solid line denotes the correction value depending on the neuron order.

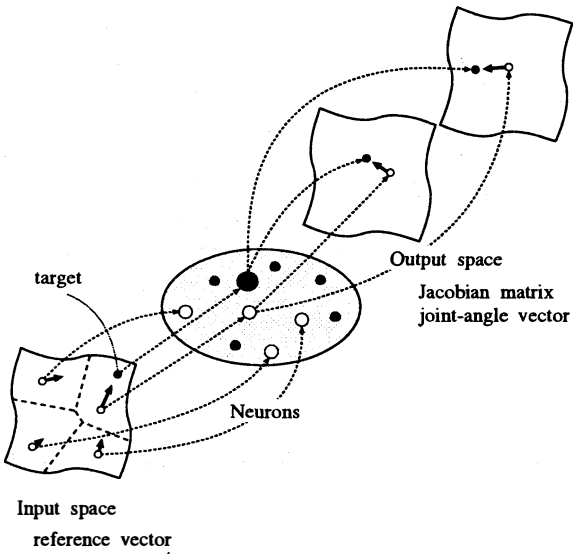


Fig. 2: Neuron activity changes in the self-organization network

The three elements given above should be modified in order to decrease the error between the resultant visual feedback signals and the target vector. Here, we use a self-organization network with N neurons with the number determined on the basis of the system complexity.

The learning rule (as described in detail in [5]) for the k th neuron ($k = 0, \cdots, N\text{-}1$) is as

$$\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k + \varepsilon_{in} \cdot \mathrm{g}_{in}(k) \cdot (\boldsymbol{u}_t - \boldsymbol{w}_k) \tag{3}$$

$$\vec{\boldsymbol{\theta}}_k \leftarrow \vec{\boldsymbol{\theta}}_k + \varepsilon_{out} \cdot \mathrm{g}_{out}(k) \cdot \boldsymbol{\Delta}\vec{\boldsymbol{\theta}}_k \tag{4}$$

$$\boldsymbol{A}_k \leftarrow \boldsymbol{A}_k + \varepsilon_{out} \cdot \mathrm{g}_{out}(k) \cdot \boldsymbol{\Delta}\boldsymbol{A}_k, \tag{5}$$

where ε is certain positive parameter, and the subscripts *in* and *out* indicate parameters belonging to input and output spaces, respectively. $\mathrm{g}(k)$ is the Gauss-type function of the k th neuron and is used to weight

each neuron on the basis of the topological neighborhood around the target vector. Here, ε and the neighborhood size decay as the learning step increases.

4 Modification to learning process

In this section, we will describe some modification to the learning process.

4.1 Network initialization based on simulation

In practical implementation of the learning algorithm, the number of the learning steps should be reduced in consideration of time-consuming manipulator motion and image processing.

To achieve a more effective implementation, the whole learning process is made up of a simulation stage and then a real manipulation stage. The simulation stage, which is carried out on a computer on the basis of some rough measured data on the coordination system, aims to provide a better initialization for the real network. This strategy decreases the number of required manipulator motion and image processing considerably.

4.2 Introduction of repulsive forces

In a case of unstructured environment where obstacles may exist, we suppose that the representation of internal models differs from those in the environment. This is caused by constraint that the degree of freedom of the manipulator decays. For the reason, we introduce the following two forces into the original learning algorithm. One is the attractive force which acts so that the neurons are closer to target. The other is the repulsive force which acts so that the neurons are farther to obstacles.

The repulsive force F_k is defined on the basis of the distance r_k between the reference vector of kth neuron and the obstacle. It is given as

$$F_k \propto \frac{1}{{r_k}^2}. \tag{6}$$

5 Experiments

In the following, we present experimental results obtained by applying the self-organization networks.

5.1 Local linear maps

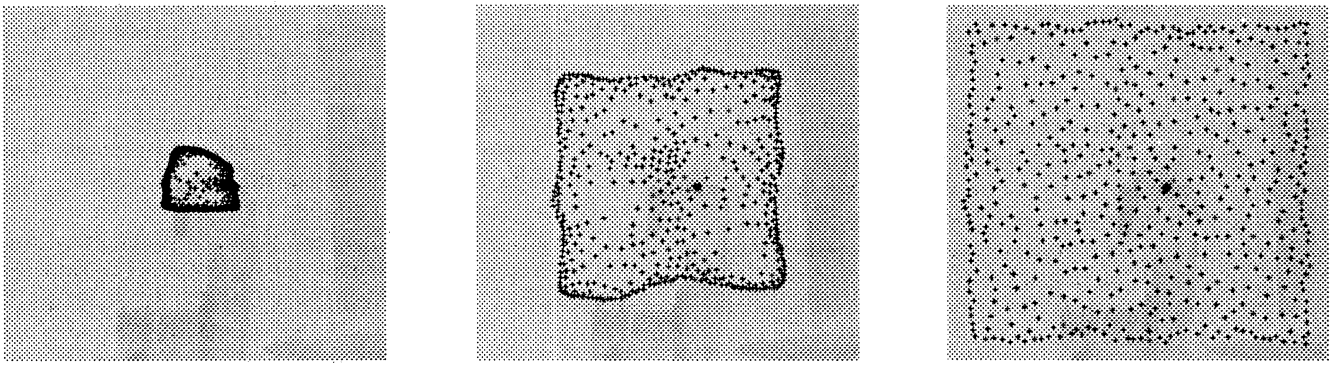
The state of local linear maps is illustrated by the distribution of neurons. We perform the visualization by projecting the reference vectors which correspond to input space onto one camera image plane.

Fig. 3 shows changes of the local linear maps with respect to learning steps. Through the learning process in the simulation stage (a) and then (b), the final homogeneous distributions in the real manipulation stage (c) enable us to fulfill the linear transformation in the divided space, as described in eq. (2).

5.2 Positioning performance and adaptation capability

Fig. 4 presents positioning accuracy versus the number of the learning steps. The positioning accuracy is calculated by the Euclidean norm in 3-D space between the target position and the resultant position for the end effector.

As shown in this figure, the learning process with initial setting given by a simulation stage, approximates the nonlinear transformation accurately, which is represented by the solid curve. Moreover, in the result illustrated by the dashed curve, a structural change by pulling the manipulator away from the camera by 100 mm after 60 learning steps is occured. In spite of such



(a) (b) (c)

Fig. 3: Changes of neuron's distributions along with the learning steps

environmental change, the system can be adapted quickly to it and almost recover to the previous accuracy.

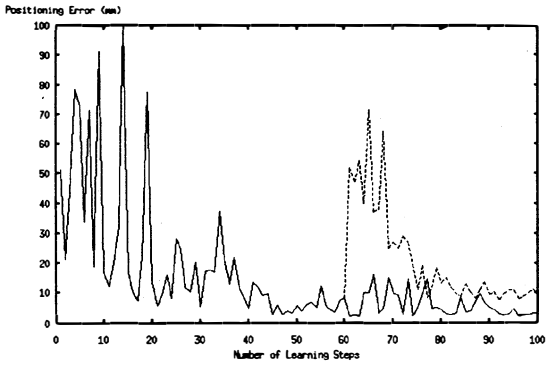


Fig. 4: Positioning performance

5.3 Effectiveness of the Simulated Networks

We initialize the whole network with a simulated network to reduce the implementation time. Hence, we give a comparison of it with a network which is initialized by random values.

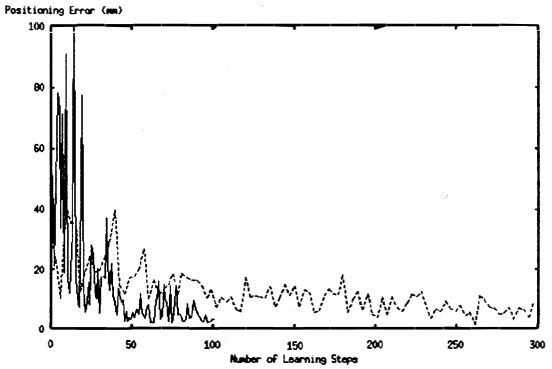


Fig. 5: Performance for different initializations

Fig. 5 shows the error changes for the position learning corresponding to the two different networks. The results of the learning with a simulation stage are represented by the solid curve, while those without the simulation stage by the dashed curve. It is shown that the convergence rate is remarkably improved by the simulation stage.

5.4 Property based on the repulsive forces

We present a result of simulation in more complex situation where a circular obstacle exist.

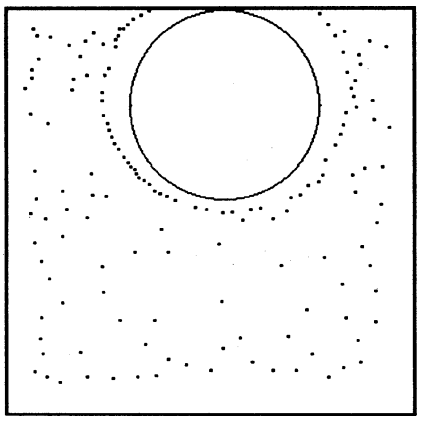


Fig. 6: Distribution of neurons according to the working space

Fig. 6 shows local linear maps similar to the result in § 5.1. Non-uniformity of the distribution indicates that a neighborhood differs inside which each neuron acquired a local linear transformation.

6 Conclusions

By applying a two-stages learning algorithm, the system can approximate the joint angles of a manipulator by local linear maps in faster implementation time. The adaptation capability with respect to a sudden change in the environment is also proven. Moreover, by applying the algorithm into which the repulsive forces are introduced, we confirm that the distribution of the neuron cells is closely connected with the working space.

Our future task will be oriented toward a visuo-motor coordination in real situations where some obstacles exist.

References

[1] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biol. Cybern.*, 43, pp. 59-69, 1982.

[2] T. Kohonen, "Analysis of a simple self-organizing process", *Biol. Cybern.*, 44, pp. 135-140, 1982.

[3] M. Kuperstein, "INFANT neural controller for adaptive sensory-motor coordination", *Neural Networks*, Vol. 9, pp. 131-145, 1991.

[4] J. A. Walter and K. J. Schulten, "Implementation of self-organizing neural networks for visuo-motor control of an industrial robot", *IEEE Trans. Neural Networks*, Vol. 4, No. 1, pp. 86-95, 1993.

[5] H. B. Zha, T. Onitsuka, and T. Nagata, "Self-organization based visuo-motor coordination for a real camera and manipulator system", *in Proc. IEEE Conf. Syst., Man, and Cybern.*, pp. 3322-3327, 1995.

Hardware Evolution - a real "life on the silicon"

Hitoshi Hemmi
Evolutinary Systems Department
ATR Human Information Processing
Research Laboratories
Kyoto 619-02, Japan

Katunori Shimohara
Evolutinary Systems Department
ATR Human Information Processing
Research Laboratories
Kyoto 619-02, Japan

Abstract

A hardware evolutionary system that automatically changes HDL (Hardware Description Language) specifications of digital circuits and adapts them to applied specific problems is proposed. Outputs from the system can be converted to real electronic circuits instead of the conceptual data structure in computer memory. Such individual circuits can be seen as an artificial life that behaves at electoronics speed: true "life on the silicon".

1 Introduction

Hardware evolution and evolvable hardware will become the most important components for complex adaptive systems. Despite long tenacious research efforts, creating flexible autonomous agents is still a tough problem for engineers. Thinking about the amazing creativity of nature, however, evolvable hardware that mimicks natural evolution can very likely to create such an agent. Furthermore, such agents are expected to consist of a huge number of components; a software only technique solely is not feasible. Massive parallelism of hardware are indispensable to cope with such hugeness.

We have been constructing a hardware evolutionary system [1, 2] named AdAM (Adaptive Architecture Methodology). This system uses a special purpose programing language, HDL (Hardware Description Language), and evolves HDL-programs. Section 2 gives an overview of this system.

The main technique used in this system is Production Genetic Algorithms (PGAs)[1], which skillfully change the HDL programs and makes their evolvement possible. Section 3 briefly reviews this technique.

Section 4 is devoted to experiments and results of applying the AdAM system to some application problems.

Section 5 explain the future work.

In work related to the AdAM project, Higuchi et al.[3] uses a FPGA (Field Programmable Gate Array), looks at the architecture bits of the FPGAs as chromosomes and evolves them using Genetic Algorithms.

The main difference between Higuchi's method and AdAM is that the former aims to evolve circuit structures while the latter aims to evolve circuit behaviors expressed as HDL programs.

2 The AdAM system

The AdAM system uses LSI (Large Scale Integrated circuit)-CAD (Computer Aided Design) system and add automatic-al evolutionary mechanism to the system. Figure 1 shows an overview of the AdAM system.

This system provides a program development process. In the process, production rules translated from HDL grammar and structured chromosomes are used to automatically generate HDL source programs. The generated HDL programs are simulated along with some application problems by using a behavior simulation tool. The results are then evaluated as to how well each HDL-program (i.e., digital circuit) fits the problem. After the evaluation results are obtained, some transformations are performed on the development process to improve circuit performance. The generated HDL-programs are converted into circuit schematics, FPGA (Field Programmable Gate Array) configuration data, or LSI mask patterns using a netlist synthesizer and other CAD tools. In this sense, the HDL programs can be considered to correspond to hardware.

3 Production Genetic Algorithms

This section briefly reviews the Production Genetic Algorithms (PGAs) [1].

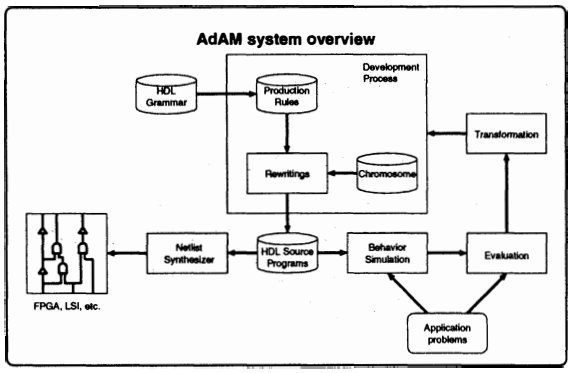


Figure 1: The AdAM system

One significant drawback of HDL as a hardware evolution tool is that HDL is too vulnerable to the change caused by evolutionary operations. The main objective of PGAs is to apply evolutionary operations safely to the HDL programs. We will explain PGAs along with Fig. 1.

3.1 Production rules

Figure 2 shows production rules corresponding to a HDL grammar. In Fig. 1, HDL programs are developed by a rewriting process using these production rules.

```
(r0.0)  module      → K_MOD name list_comp list_pin list_action
(r1.0)  name        → K_NAME
(r2.0)  list_comp   → comp
(r2.1)  list_comp   → list_comp comp
(r3.0)  list_pin    → empty
(r3.1)  list_pin    → list_pin pin
(r4.0)  list_action → action
(r4.1)  list_action → par_action
(r4.2)  list_action → cond_action
                :
(r7.0)  action      → action1
(r7.1)  action      → action action1
(r7.2)  action      → action action2
(r8.0)  action1     → register
(r9.0)  action2     → memory
                :
(r20.0) comp        → K_INPUT input_name
(r20.1) comp        → K_OUTPUT output_name
(r20.2) comp        → K_BIDIRECT bus_name
(r20.3) comp        → K_INSTRIN inst_name
                :
```

Figure 2: Production Rules (HDL grammar)
The numbers following to 'r' are production numbers. For example, 'r4.1' indicates a rule having category number 4 and sub-number 1

3.2 Chromosomes

In the development process in Fig. 1, chromosomes in Fig. 3 are used to determine the production rules applied to each symbol.

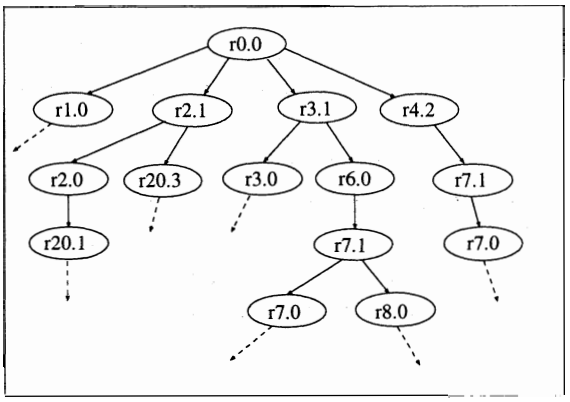


Figure 3: Chromosomes

3.3 Genetic operations

The AdAM system has five genetic operations. Here, we explain three of those.

Crossover

In tree-structured chromosomes, crossover can be operated between subtrees for root nodes having identical category rules; that is, two rules are applied to the same symbol.

Figure 4 shows a crossover example.

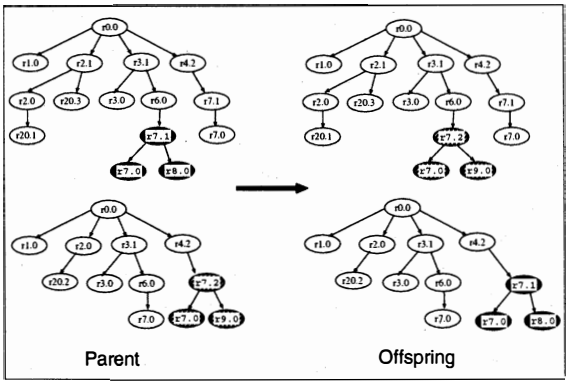


Figure 4: Crossover

Mutation

Mutation acts on a node of a chromosome and changes the label to index another rule with the same category as the previous rule.

Gene duplication

This operation is related to recursive rules. There is a rule that generates its left hand symbol among its right hand symbol sequence. In a chromosome, such a rule can appear repeatedly in a line. Gene duplication is an operation that copies a node and its subtree block, and arranges them in a line. This is a neutral operation; that is, the HDL program does not change its functionality.

This operation is useful when combined with mutation. When mutation occurs in either subtree block, the HDL program gets a new functionality, but its past functionalities are preserved.

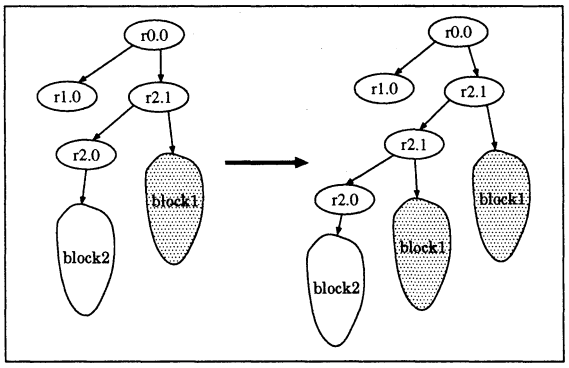


Figure 5: Gene duplication

A copy of block1 is inserted between block1 and block2

4 Experiments

This section describes some experiments and their results, applying the AdAM system to behavior control tasks that concern artificial ant problems.

The artificial ant problems are tasks to make an artificial ant on toroidal lattice space eat food adequately arranged on the space as much as possible. Jefferson et al. [4] succeeded in making an ant follow a trail called the "John Muir Trail" (Fig. 6) by using a Finite State Automaton and an Artificial Neural Network. In Fig. 6, the black cells indicate the food's existence. Similarly, Koza did this on the "Santa Fe Trail" with GP [5].

Our target is to generate a digital circuit capable of controlling an ant on the John Muir Trail. One input and two outputs are sent to the circuit. The ant only gets one sensory input from the cell in front of it, whether or not the food exists on the cell. The two outputs are decoded and used to determine four actions of the ant: go straight, turn left, turn right, or do nothing.

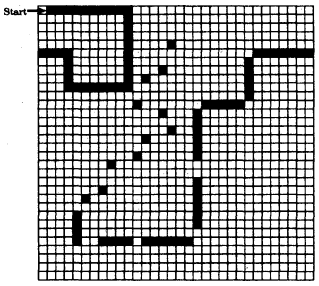


Figure 6: John Muir Trail

We set three different situations for this problem and did experiments to each case. The result of basic one was reported in previous paper ([1]). Here, we explain two others.

Experimental conditions were same in all cases: population size was 200, crossover rate was 50% per individual, mutation rate was 0.5%, gene duplication rate was 2%, and deletion rate was 1% (deletion is a operation roughly the opposite of gene duplication).

In all cases, the fitness evaluation waw given by the following formula:

$$performance = score + (time\_limit - time\_steps)$$

Score is the number of pieces of food (maximum 89) the ant ate within the limited time (time_limit = 350). Time_step is the number of steps that the ant took. If the time limit was reached before the ant ate all the food, then time_step equals 350.

Case A

In this experiment, each circuit consists of two modules. Each module has one output, and these outputs are combined to make two outputs of whole circuit. One input to the whole circuit is distributed to both modules of the circuit. Each module is treated as belonging to different species and evolving in different populations. This is an experiment for a coevolution model.

Case B

The HDL we use has a lot of built-in operators that express logical and arithmetic operations. In this experiment, the production rules are modified to inhibit some operators form appearing in the HDL programs. This experiment tests the independentness of this technique to the particular HDL.

Figure 7 graphically show the results for case A and case B. Each graph shows average fitness of the population and fitness of best individual.

In the left graph of Fig. 7, at the 144th generation, a circuit with a perfect score appears. This circuit traverses the entire trail in 314 time steps. At the 270th generation, a circuit that takes 266 time steps to traverse the trail appears. The two modules of this circuit both have 4 control states. Here, two circuit modules of whole circuit belonging different species were co-evolved and cooperatively control one artificial ant. Figure 8 shows part of schematic diagram of the best circuit.

In the right graph of Fig. 7, at the 103rd generation, a circuit with a perfect score, that takes 300 time steps to traverse the trail appears. At the 381th generation, a circuit which takes 282 time steps appears.

This result implies that the success of the AdAM method is not sensitivity dependent on a particular programing language.

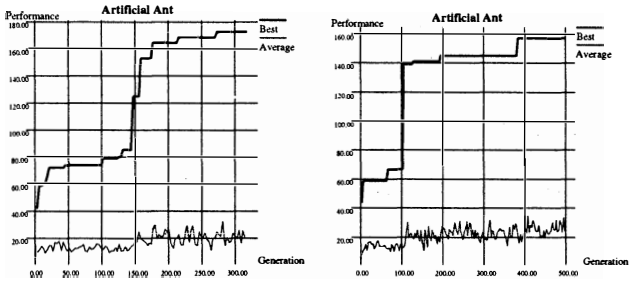


Figure 7: Artificial Ants

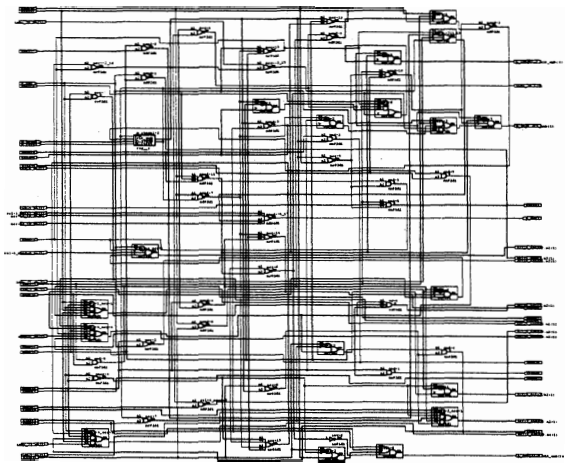


Figure 8: Part of schematic diagram of circuit obtained in case A

5 Future work

In biological nature, the development processes differs from species to species. We plan to modify the system so that some set of rewriting rules that are used in generation of HDL-specifications are obtained as the result of evolution from a simple set of rewriting rules; each set of rewriting rules corresponds to one "species".

We also plan to implement this evolutionary sysytem on hardware. This will increase strikingly the system's operation speed. The result will be a real time evolutionary system. We will no longer have to wait to see the progress of evolution.

The system mentioned here can be used to develop controllers of robots and animats. The system itself may be used as controllers of some kind of interactive Artificial Life (A-Life) art works, such as an automatic music composers or a general simulator of A-Life phenomena.

References

[1] Jun'ichi Mizoguchi, Hitoshi Hemmi, and Katsunori Shimohara. Production genetic algorithms for automated hardware design through an evolutionary process. In *IEEE Conference on Evolutionary Computation*, 1994.

[2] Hitoshi Hemmi, Jun'ichi Mizoguchi, and Katsunori Shimohara. Development and evolution of hardware behaviors. In Rodney Brooks and Pattie Maes, editors, *Artificial Life IV*. MIT Press, 1994.

[3] Higuchi Tetsuya, Iwata Masaya, Kajitani Isamu, Iba Hitoshi, Hirao Yuji, Furuya Tatsumi, and Bernard Manderick. Evolvable hardware and its applications to pattern recognition and fault-tolerant systems. In *Towards Evolvable Hardware*, Lecture Notes in Computer Science. Springer-Verlag, 1996.

[4] David Jefferson et al. Evolution as a theme in artificial life; the genesys/tracker system. In *Artificial Life II*, pages 549–576. Addison-Wesley, 1992.

[5] John R. Koza. *Genetic Programming: On the programming of computers by means of natural selection.* The MIT Press, 1992.

Experimental Study on Autonomous Mobile Robot Acquiring Optimal Action to Avoid Moving Obstacles

**Takeshi AOKI, *Toshiaki OKA, *Soichiro HAYAKAWA, *Tatsuya SUZUKI, *Shigeru OKUMA

*School of Electric Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01, JAPAN
TEL: +81-52-789-2778, FAX: +81-52-789-3140
Email: oka,suzuki,okuma@okuma.nuee.nagoya-u.ac.jp

**Nagoya Municipal Industrial Research Institute
3-4-41, 6-ban, Atsuta-ku, Nagoya, 456, JAPAN
TEL&FAX: +81-52-654-9932
Email: aoki@okuma.nuee.nagoya-u.ac.jp

Abstract

The principal aim of this study is to show how an autonomous mobile robot can acquire the optimal action to avoid moving multiobstacles through the interaction with the real world. In this paper, we propose a new architecture using the hierarchical fuzzy rules, fuzzy evaluation system and learning automaton. By using our proposed method, the robot autonomously acquires the fine behavior how to move to the goal avoiding moving obstacles using the steering and velocity control inputs, simultaneously. Also we show the experimental results to confirm the feasibility of our method.

1. Introduction

In this paper, we propose a new action selection technique based on the framework of learning automata[1] so that an mobile robot can autonomously acquire the optimal action using the steering and velocity control inputs, simultaneously. Our proposed system consists of 6 modules explained in the following sections. Basic steering and velocity control inputs are decided based on the fuzzy logic in the hierarchical fuzzy module[2] and fine adjustment or combination of these control inputs are acquired by the reinforcement learning algorithm in the action learning module. This is because the combination rules between two control inputs can hardly designed in advance. In the learning module, each rule describing how to combine or adjust the control inputs is regarded as an action, and several actions are prepared in advance. Then using this learning method, the robot can autonomously acquire the optimal action suitable for each situation the robot faces through the interaction with the real world.

The reinforcement learning method on the framework of the learning automata used in this study has the property called "absolute expediency", which means that action selection probabilities are guaranteed mathematically to converge under the unknown static random environment. Moreover, This learning algorithm is executable in real time because of its simplicity.

In case of using learning methods like the reinforce learning ones, we also have to consider how to evaluate the performance. The world always changes and the performance index is inclined to be complicated. In our proposed system, we construct the evaluating function using the fuzzy logic. This is because the fuzzy logic can reflect the designer's experiences or intuitions explicitly and evaluate the performance easily.

Using our proposed system, the robot performs the learning cycle "sense-select-act-evaluate-learn(update)" in the real world. Finally, the robot autonomously acquires the optimal action selection suitable for the real world. In the following section, the details of our system are explained.

2. Proposed System

2.1. System for Single Obstalce

2.1.1 Architecture

In this subsection, we state the outline of our proposed system for single obstacle. The proposed architecture is shown in Figure 1. The system consists of 5 modules. The role of each module is as follows.

1)***Sensing Module***: This module senses the environment around the robot and calculate the relative positions and velocity vectors between the robot and obstacle.
2)***Hierarchical Fuzzy Rule Module***: This module decides the steering and velocity control inputs for each obstacle avoidance, respectively. The advantage of this hierarchical algorithm is that the number of fuzzy rules can be reduced without degrading control performance. Figure 2 shows the structure of hierarchical fuzzy rules, the definition of the fuzzy parameters and the relations between inputs and outputs. The details of this structure is shown in [2].
3)***Action Generating SubModule***: This module selects one action (rule) among actions by the roulette method based on the action probabilities explained later in order

to combine the steering control input with velocity control input based on the selected action for each obstacle. In this module, there are four action selections

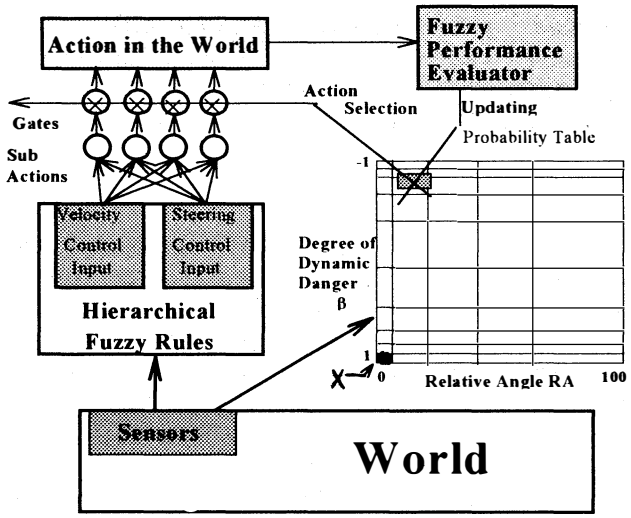


Figure 1. Proposed System for Single Obstacle

Table 1. Evaluation Fuzzy Rules for Single Obstacle

		Parameter ② Change of Between Robot and Goal		
		Negative	Zero	Positive
Parameter ① $\lvert\beta(n)\rvert - \lvert\beta(n-1)\rvert$	Negative Big	Positive Big	Positive Medium	Positive Medium
	Negative Small	Positive Small	Positive Small	Zero
	Positive Small	Negative Small	Negative Small	Negative Medium
	Positive Big	Negative Medium	Negative Big	Negative Big

(e.g.:Combine 50% of the steering control input for obstacle avoidance with 50% of it for goal, and use the velocity control input for obstacle avoidance)

4)***Fuzzy Performance Evaluation Module***: This module evaluates how good or bad the selected action (rule) is by using the fuzzy logic. The fuzzy evaluation rules for single obstacle are shown in Table 1. Using this table, the fuzzy evaluation module evaluates the robot action.

5)***Action Learning Module***: This module updates and memorizes the action probabilities in the element of the table based on the evaluation value from the evaluation module by the reinforcement learning method in Learning Automata.

2.1.2 Probability Table

The probability table for single obstacle is designed as the two-dimensional finite space. The environment around the robot is mapped into this table. In other words, each element of the table reveals the situation that the robot faces. The dimensions of the SubModule Table represent the dynamic danger β[3] and the relative angle RA shown in Figure 1[2]. For example, the element X in Figure 1 means that the obstacle is dangerous in the sense of dynamic danger and approaches the robot from the front.

2.1.3 Learning Scheme

It is difficult for a designer to specify the selection rule of the action in advance because the designer can not get all the information about the real world in advance. One of the most promising method to overcome this problem is adopting the algorithm based on the framework of Learning automata[1]. Using this reinforcement method, the robot can autonomously acquire the good action in any situation through the interaction with the real world. As the reinforcement scheme, we use the PLr-i algorithm[1]. The PLr-i method updates the action probabilities based on the binary value of evaluation. When the robot action is good, that is, the output from the fuzzy performance evaluation module described before is positive, the action probabilities are updated as follows.

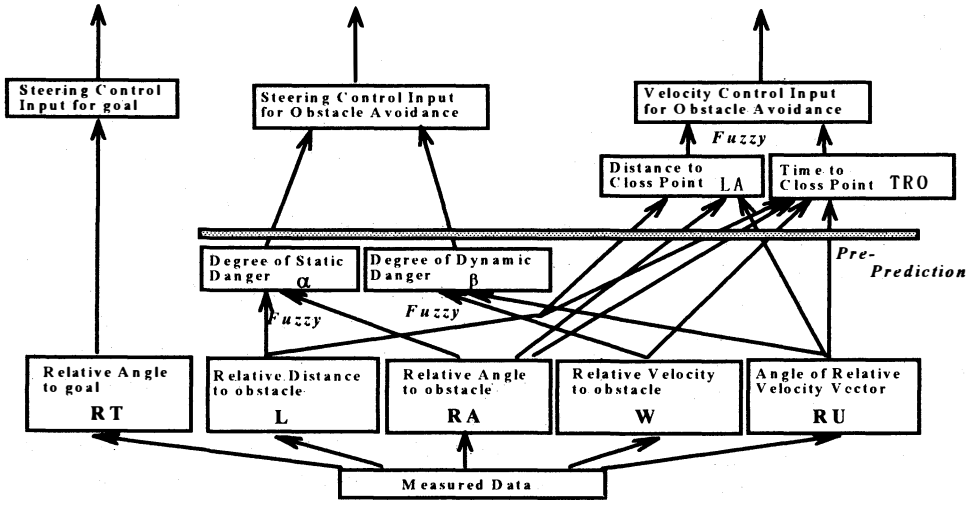


Figure 2. Hierarchical Fuzzy Rules

PLr - i Scheme

$$\begin{cases} p_i(n+1) = p_i(n) + c[1 - p_i(n)] & \text{if } a(n) = a_i \text{ and } b(n) = 1 \\ p_j(n+1) = (1-c)p_j(n) & \text{for } j \neq i \\ p_i(n+1) = p_i(n) & \text{for all i, if } b(n) = 0 \end{cases}$$

where c is learning parameter, a_i and a(n) denote the ith action and selected action at the nth stage, respectively. Pi(n) denotes the probability of action i at the nth stage and b(n) is the output of the fuzzy evaluation module at the nth stage. When the robot action is bad, that is, the output from the evaluator is not positive, the action probabilities are not changed.

Each element of the probability table has four probabilities which correspond to four actions. The initial value of each probability is 1/4. After choosing an action according to the probabilities of the element which reveals the situation of the robot, the robot updates the probabilities of the element with the evaluation value using PLr-i method. The main idea of the PLr-i algorithms is that the absolute expediency in the unknown random world is guaranteed[1]. This is due to the operation of "inaction" in case that the evaluation is not good.

2.2 Extended System for Multiple Obstalces

2.2.1 Architecture

In this subsection, we state the outline of the extended system for multiple obstacles. The extended architecture is shown in Figure 3. The system consists of 6 modules. ***Sensing Module, Hierarchical Fuzzy Rule Module and Action Generating SubModule*** work the same as those of the system for single obstacle. The role of other modules is explained in the following subsection.

2.2.2 Role of Modules in Meta Level

Action Generating MetaModule:This module also selects one action (rule) among actions based on the action probabilities in order to combine the steering control inputs for each obstacle with weights based on the selected action. This module has 4 actions which means the number of obstacles to be paid attention to . Note that the Action Generating SubModule is assumed to be learned enough in advance.

Fuzzy Performance Evaluation Module: This module evaluates how good or bad the selected action (rule) of the metamodule is by using the fuzzy logic. The fuzzy evaluation rules are like ones in Table 1.

Action Learning Module:This module updates and memorizes the action probabilities in the element of the table for the action generating metamodule based on the evaluation value from the evaluation module by the

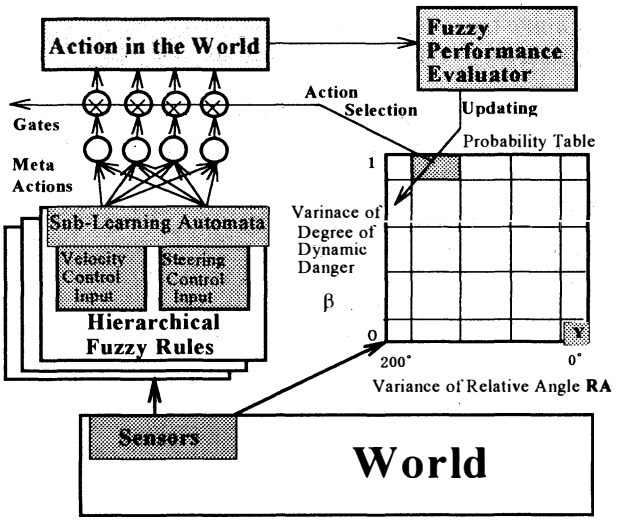


Figure 3. Extended System for Multiple Obstacles

reinforcement learning method in Learning Automata. The probability table for multiple obstacles is also designed as the two-dimensional finite space. The environment around the robot is mapped into this table. In other words, each element of the table reveals the situation that the robot faces. This table for the metamodule shown in Figure 3 represent the variance of the dynamic danger β and the variance of the relative angle RA. The variance of the parameter β indicates a kind of the distribution of the dangerous moving obstacles. For example, the element Y of the table in Figure 3 means that all the obstacle are equally dangerous in the sense of dynamic danger and that the obstacles are located at the same direction.

3. Experimental Study

In this section, the experimental results are shown to confirm the feasibility of our proposed system. The assumptions made in the experiments are as follows.

Obstacle

1)The number of obstacles is three.
2)The start points are set at random in the front side of the robot at every run.
3)The moving directions are set so as to approach the robot
4)The Obstacles move straight with constant velocity. The velocities are set at random up to the initial velocity of the robot at every run.

Robot.

5) The start point, moving direction and initial velocity is same at every run.
6)The action probabilities are updated at each time when

the robot performs one action.
7)PLr-i algorithm is used as the learning method.

The mobile robot and three obstacles used in this experiment are Khepera made by EPFL. The robot is controlled by a NEC PC9801BA Computer (CPU:Intel 486DX2 40MHz) via a RS232C line by means of a cable. This cable is also used to supply electrical power. As for the sensor, CCD camera is used to get the information of the environment around the robot. The camera is set at the height of 2m from the ground in order to watch 2.5x3m area. The camera is also connected with the computer. All processes such as picture reading and analyzing, decision making and motor control are performed with on-line process by the computer. The experimental setup is depicted in Figure 4. Learning of the probability tables are performed by simulation in advance. Then using those tables, the fesibility of our method is tested with real robots .Before learning, the action probabilities in the Action Generating MetaModule are not well-tuned. In this case, the robot often chooses irrelevant actions and bumps into the obstacles. On the contrary, after learning, the robot has acquired the intelligence of action selection based on the action probabilities. In Figure 5, the robot chooses fine actions and inhibits unnecessary actions and the robot can move to the goal avoiding moving obstacles. Figure6 shows the probability history of the element Y of the action probability table in Figure 3. This figure indicates that the robot has acquired action rule that robot regards obstacles as one group and pays attention to just the most dangerous obstacle in the situation corresponding to element Y.

4.Conclusion

We propose a new action selection technique for moving obstacles avoidance using hierarchical fuzzy rules, fuzzy evaluation system and learning automata through the interaction with the real world. By using our proposed system, an mobile robot autonomously acquires the fine behaviors how to move to the goal avoiding moving obstacles using the steering and velocity control inputs, simultaneously. We show the feasibility of our proposed system with experimental results.

References

[1]K.Narendra et.al.,"LEARNING AUTOMATA", Prentice-Hall. Inc.,1989

[2]T.Aoki et.al,"Motion planning for Multiple Obstacles Avoidance of Autonomous Mobile Robot Using Hierarchical Fuzzy Rules," 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp.265-271.

[3]Y.Maeda et.al,"Collision Avoidance Control among Moving Obstacles for a Mobile Robot on the Fuzzy Reasoning," J.of the Robotics Society of Japan, Vol6, No.6, pp:518-522, 1988.

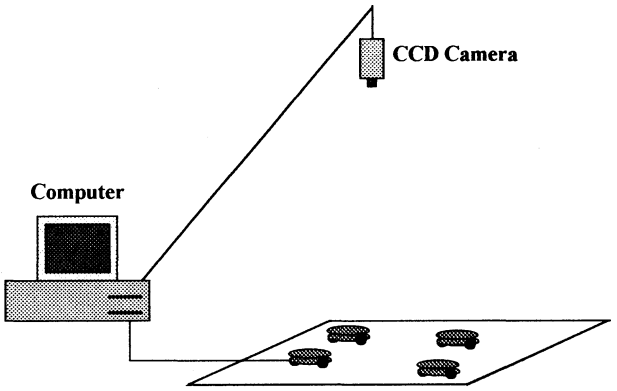


Figure 4 Experimental Setup

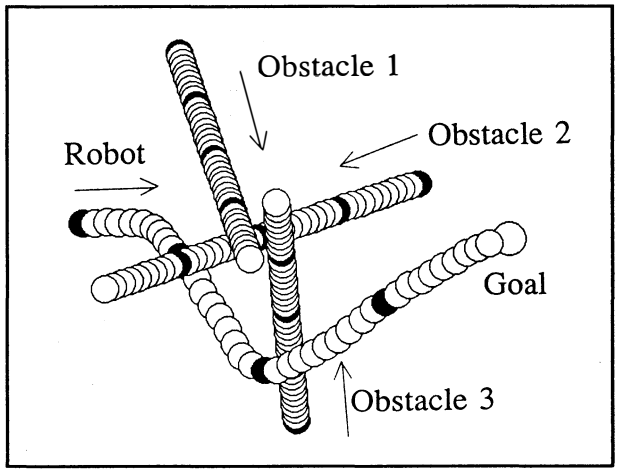


Figure 5 Experimental Result after Learning

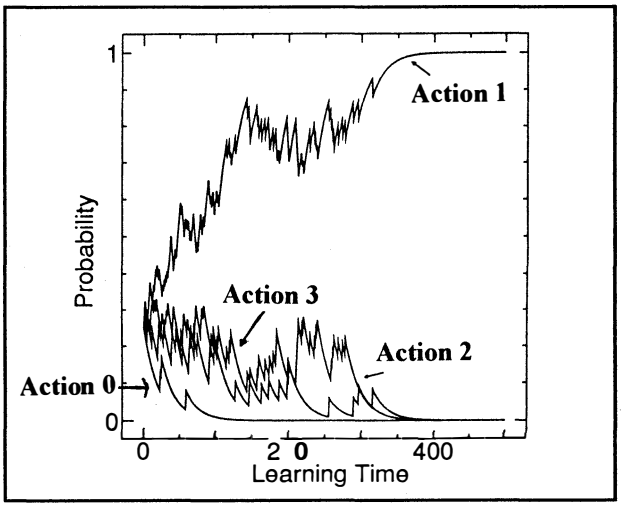


Figure 6 Action Probability History of Element Y of Action Probability Table in Figure 3

Auto Navigation on the Wheel Chair

Hongbo WANG* Chul-Ung KANG*
Takakazu ISHIMATSU* Tsumoru OCHIAI**

* Department of Mechanical Systems Engineering, Nagasaki University , Bunkyo-machi, Nagasaki 852, JAPAN
**Ube Technical College, Nishihon-machi, Ube, Yamaguchi, JAPAN

Abstract

The paper presents a wheel chair system with the capability of self-localization and obstacle avoidance. Firstly, the approaches of landmark recognition and the self-localization of the wheel chair are described. Then, the principle of the obstacle avoidance using laser range finder is described. Finally, the total system of the wheel chair is introduced and a navigation experiment is given. Experimental results indicate the effectiveness of our system.

Keywords: Auto-navigation, Wheel chair, Landmark, Image Processing, Laser range finder, Obstacle avoidance

1 Introduction

The commercial electric wheel chair has been widely used by handicapped persons suffering from lower extremity amputations. However, it is difficult for handicapped persons, having both upper and lower extremity amputations, to drive a wheel chair. For this reason, an automatic wheel chair with the capability of self-localization and obstacle avoidance has been developed in our laboratory so that daily tasks can be coped with easily.

Our goal is that the handicapped persons can use the wheel chair to move from one position to another position in an indoor environment avoiding both static and dynamic obstacles. This paper describes an auto-navigational wheel chair system with the capability of self-localization and obstacle avoidance.

The capability of self-localization is critical to reliable performance of the wheel chair. There are many methods that can determine the self-localization of the wheel chair. One simple and reliable method is by providing landmark features of the workspace as external reference sources. Several researchers have approached the self-localization problem of mobile robots by employing "standard marks" [6, 7]. The key idea is to use special marks that include a wealth of geometric information under perspective projection such that the camera location can be easily computed from the image of the guide-marks. In our research, we have chosen "ceiling lights" as landmarks because they can be easily detected due to the high contrast between the light and the ceiling surface and do not require special installation. Dulimarta [2] used the ceiling light as landmark. However, the ceiling light was used only for light counter. We use the ceiling light for landmark to perform the self-localization of the wheel chair.

Fig. 1. Wheel chair with self-localization and obstacle avoidance

Our desire is to enable the wheel chair to navigate from one position to the desired position without any human assist. It is assumed that this wheel chair is used only inside of one building whose geometry are known in advance. In order to realize the auto-navigation in the building, our wheel chair has a geomagnetic azimuth sensor and a vision sensor to detect ceiling lights. While the geomagnetic azimuth sensor gives data about the azimuth angle, the output data are often affected by the environment. Therefore, our system employed the ceiling lights as landmarks to guide the wheel chair with enough reliability. Of course, the position and the posture of every ceiling light are assumed to be known in advance. In the case that the ceiling lights are not detected, the wheel chair navigates with the information of the geomagnetic azimuth sensor and rotating angle of both wheels.

When the wheel chair navigates automatically, it is unavoidable to encounter the static and dynamic obstacles. The obstacle avoidance is one of the fundamental requirements for an auto wheel chair to execute its tasks. Several studies have been done for the obstacle avoidance problem of mobile robot by the use of ultrasonic [3, 4], and visual sensors [5, 8]. Among the sensors used for mobile robots, visual sensors probably provide the richest source of useful information about the surrounding. However, visual sensors are also the ones that are the most computational expensive. One possible solution for the problems of vision, is to employ specialized hardware for navigation, such as laser range finder, sonars, and inertial navigation system. In our system described here, we employ the original laser range finder to realize the obstacle avoidance of a wheel chair.

Our wheel chair is equipped with two CCD cameras. The camera facing up is used for detecting landmarks and other camera facing in front in combination with a slit laser sensor is used for detecting the obstacles as shown in Fig 1.

In section 2, the approach of landmark detection and recognition and the self-localization of the wheel chair are described. In section 3, the laser range finder is described. The total system of the wheel chair and experimental results of auto navigation are described in section 4. Finally, we give the conclusion of the paper.

2 Landmark Recognition and Self-Localization

In our experimental environment, the "ceiling lights" are two fluorescent lights that are installed parallel. When the wheel chair navigates, the camera may acquires some unnecessary images as a result of wall light, the sunlight from window and another light sources. From all images, the useful image as landmark can be retrieved by checking the area, boundary, distance and position of every image. Considering the shape of fluorescent light, the procedure is described as follows.

(1) Image signals from the CCD TV camera are converted into binary image signals.

(2) Profiles of white cluster in the image (Run-Length Data) are extracted from the binary signals by a logic circuit constructed by FPGA (Field Programmable Gate Array: XC4010).

(3) By the labeling program, we obtain the geometrical data (area, length, width and the position) of every white cluster.

(4) Considering the area, length , width and the distance from the neighboring white cluster, we can select the white clusters corresponding to the images of fluorescent lights from the other clusters.

The contour of two lights extracted using the above method is shown in Fig. 2. Using the contour points of the light, the central line of the light can be obtained. From the cross points of the central line and image contour, the raster coordinates P_i (i=1,2,3,4) of the light ends can be obtained [1].

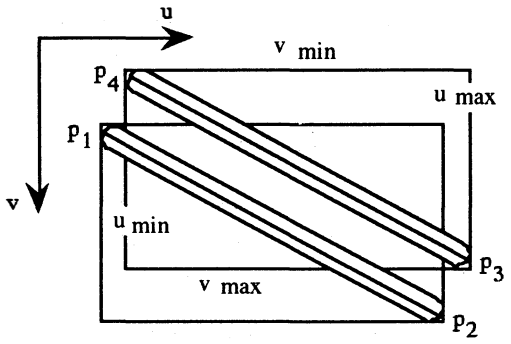


Fig. 2. Image of ceiling Light

On the basis of the raster coordinate of the ceiling light, we have the vectors $p_i=(u_i\ v_i\ f)$ (i=1,2,3,4) in camera coordinate system. The position vectors q_i (i=1,2,3,4) of light ends relative to camera coordinate system can be determined using the three-dimensional image processing method. From perspective transformation, the relation between q_i and p_i can be expressed as

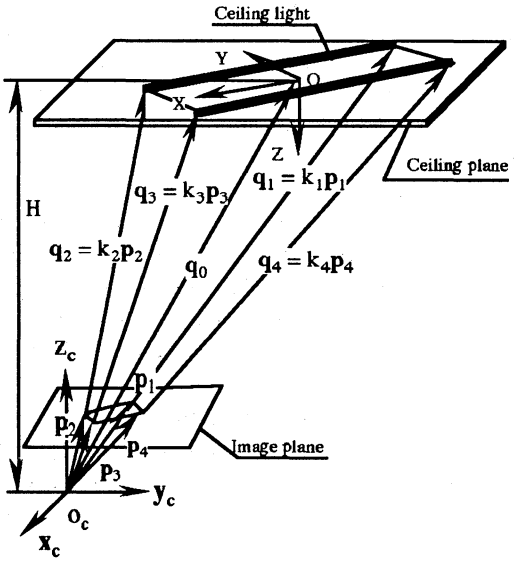


Fig. 3. Relation between camera and light system

$$q_i=k_ip_i(i=1,2...4) \quad (1)$$

where $q_i=(x_i\ y_i\ z_i)$ indicates the position vector of light end in camera coordinate system, k_i is the proportion coefficient

between two vectors. Since the two light are parallel and the lengths of the two light are equal, following vector equation can be obtained

$$k_1p_1-k_2p_2=k_4p_4-k_3p_3 \quad (2)$$

From above equation, the proportion coefficient k_1 can be derived as follows [1]

$$k_1=\frac{|q_2-q_1|}{\sqrt{p_1^2-2c_{21}(p_1\cdot p_2)+c_{21}^2p_2^2}} \quad (3)$$

where $c_{21}=p_3\cdot(p_4\times p_1) / p_3\cdot(p_4\times p_2)$ and $|q_2-q_1|$ is equal to the length of the light. Similar to k_1, the k_2, k_3 and k_4 can be obtained. From the equation (1), the q_i can be calculated.

After the q_i is obtained, the relation between the coordinate systems of the light and the camera can be established. Considering that the light is installed on the ceiling of the building, we take the light coordinate system as the base coordinate system. The base system is set as shown in Fig. 3. The origin O of the base system is chosen at the middle point of points q_1 and q_3, that is, $q_o=(q_3+q_1)/2$. X-axis is along with the direction from q_1 to q_2; Z-axis is along the normal line of the plane that is determined by three points q_1, q_2 and q_3. The Y-axis can be determined by right hand rule.

After the base system is determined, the transformation matrix [A] of the base coordinate system relative to camera system can be obtained. Therefore, the position and orientation of the wheel chair in base coordinate system can be determined easily [1].

3 Laser Range Finder

In order to detect the obstacles on the way, a compact laser range finder is developed. The principle is based on the slit-ray projection method. As you see in Fig. 4, a slit-ray is projected from a laser emitter. The slit-ray image is detected by a CCD TV camera and the image signal is processed by a logic circuit constructed with FPGA. Two examples of slit-ray images are shown in Fig. 5(a) and Fig. 5(b). If some obstacle or strong undulation is on the way, the slit-ray deviates from straight line. Otherwise, the slit-ray should draw a straight line on the image. Based on this idea, the logic circuit detects the most bright point in every horizontal scanning which corresponds to the slit-ray image. Furthermore, the logic circuit examine if the slit-ray is inside the allowable range or not. If the slit-ray is not inside the allowable region, the vertical coordinates where allowable region is violated are transmitted to main computer. Therefore, the logic circuit has two output signal lines. One output signal is used to inform the controller whether the way is safe or not. The other output signal is used to inform the controller where the obstacles exist via serial signals.

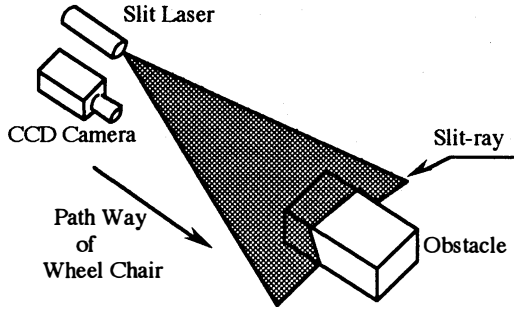


Fig. 4. Principle of laser range finder using slit-ray

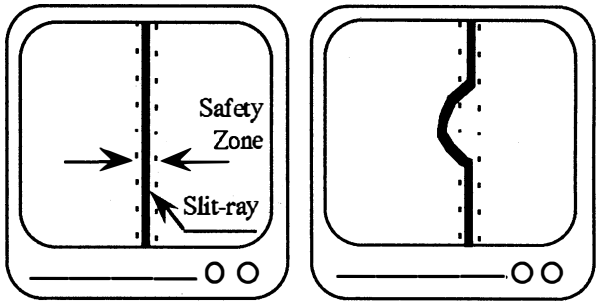


Fig. 5. Slit-ray image

Due to the processing by a logic circuit, all the processing of data are executed without any time delay. Therefore, the laser range finder tests the way 60 times per a second. One problem of the above sensing system occurs in the case that external light is stronger than the laser light. In order to cope with this problem, our laser range finder has another function. Under the condition that the slit-ray is often affected by the external lights, the laser light is turned on and off synchronous with the video signal. The frequency of the switching is 30 Hz, and the logic circuit executes subtraction between the sequential two images. Due to this subtraction, the effects of external lights are reduced remarkably. Using the subtracted image, the existence of the obstacles on the way is examined. When we use this function. The laser range finder tests the way 30 times per a second.

4 Total System and Experimental Results

The total navigation system of the wheel chair is shown in Fig. 6. One of the two CCD cameras and a slit laser

sensor are employed to detect obstacles and another camera to recognize landmarks. Two rotary encoders are installed at two drive wheels respectively to measure the distance which the wheel chair has traveled. The resolution of counter is 40 degrees. To know the direction of the wheel chair, the geomagnetic azimuth instrument is employed. The sensor can measure two vertical geomagnetic components on horizontal. From the output analogue volts of the geomagnetic azimuth instrument, the direction angle can be calculated after A/D transformation.

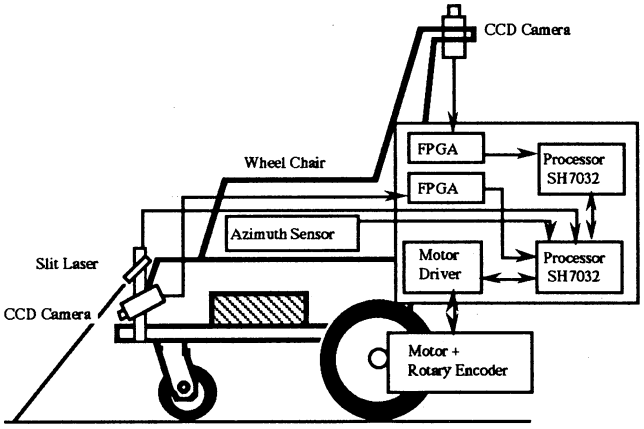


Fig. 6. Navigation system of wheel chair

We had an auto-navigation test in a room where ten ceiling lights are installed. The navigation map of the wheel chair is shown in Fig. 7. We have tested the navigation of the wheel chair 20 times along the navigation course specified with the solid line in Fig. 7. Our wheel chair is able to locate its final destination with a maximum position error of 0.35 meters and a maximum orientation error of 17 degrees. During this experiment, we put one obstacle on the way. Our wheel chair succeeded to avoid the obstacle with 70 cm apart from the static and dynamic obstacle.

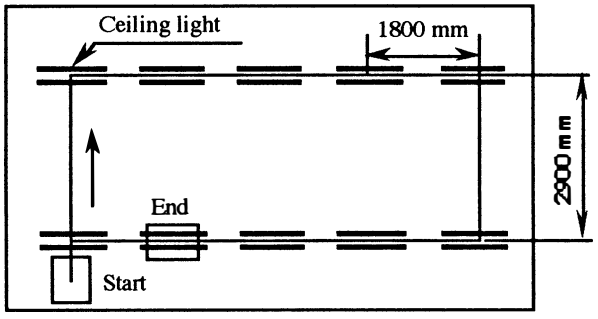


Fig. 7. Navigation map of wheel chair

5 Conclusion

In this paper, we present an automatic wheel chair system with the capability of self-localization and obstacle avoidance. The ceiling light is chosen as landmark to guide the navigation of the wheel chair. The laser range finder is used to realize the obstacle avoidance. Due to the employment of FPGA and RISC-type CPU (SH7032), the control system and the laser range finder could be constructed in compact body. Furthermore, the laser range finder could examine the safety of the path at video rate (60Hz). We had experiments of the navigation of the wheel chair in a room. The experimental results indicate the effectiveness of the approach we presented here.

References

[1] H. B. Wang, S. Tanaka, C. Kang and T. Ishimatsu, "Computer control of wheel chair by using landmarks", *in Proc. of 10th Korea Automatic Control Conference* - international program, pp 388-341, Seoul, 1995

[2] H. S. Dulimarta and A. K. Jain, "Mobile robot localization in indoor environment," *in Proc. of third int. Conf. Automat, Robotics, and Computer Vision,* vol. 3, pp. 2204-2208, Singapore, 1994.

[3] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobil robots," *IEEE Trans. on Robotics and Automation,* vol. 7, no. 3, pp. 278-288, 1991.

[4] J. L. Crowley, " World modeling and position estimation for a mobile robot using ultrasonic Ranging," in Proc. *IEEE int. Conf. Robotics Automat, Scottsdale,* AZ, pp. 674-680, 1989.

[5] K.weber, S. Venkatesh and D. Kieronska, "Insect based navigation and its application to the autonomous control of mobile robots," *in Proc. of third int. Conf. Automat, Robotics, and Computer Vision,* vol. 2, pp. 1228-1232, Singapore, 1994.

[6] M. R. Kabuka and A. E. Arenas, " Position verification of a mobile robot using standard pattern," *IEEE Trans. on Robotics and Automation,* vol. 3, no. 6, pp. 129-142, 1989.

[7] S . Y. Chen and W. H. Tsai, "Determination of a robot locations by common object shapes," *IEEE Trans. on Robotics and Automation,* vol. 7, no. 1, pp. 149-156, 1991.

[8] Y. Yagi, S. Kawato, and S. Tsuji, "Real-time ominidirectional image sensor(COPIS) for vision-guided navigation," *IEEE Trans. on Robotics and Automation,* vol. 10, no. 1, pp. 11-22, 1994.

Vision Chip Architecture with Light Adaptation Mechanism

H. Kobayashi † T. Matsumoto †† T. Yagi ††† T. Tanaka ††
† Teratec Corp. Tokyo 180 Japan †† Waseda University, Tokyo 169 Japan
††† Kyushu Institute of Technology, Iizuka 820 Japan

Abstract

Light adaptive algorithms/architectures are proposed for regularization vision chips. The adaptation mechanisms allow the regularization parameters to change in an adaptive manner in accordance with the light intensity of given images. This is achieved by adaptively changing the conductance values associated with massively parallel resistive networks. The algorithms/architectures are inspired by the adaptation mechanisms of the horizontal cells in the lower vertebrate retina.

1 Introduction

Vision chips generally refer to massively parallel arrays of simple analog circuits together with parallel array sensors. Many of the chip architectures/algorithms are motivated by visual information processing mechanisms found in vertebrate retina while some of them are derived from the Tikhonov regularization theory. This paper proposes regularization vision chip architectures which incorporate **light adaptation** and enable one to change the **filter width** of a $\nabla^2 G$-like filter in accordance with the input light intensity. Two adaptation architectures are proposed: global and local. In the former, global light intensity information controls a particular parameter in the information processing cells (pixels) altogether, while in the latter, local light intensity information regulates a parameter in each individual pixel. CMOS circuits are also proposed to implement the adaptation mechanisms. The problem is formulated in terms of the regularization theory, while the adaptation algorithm has been inspired by the adaptation mechanism of the horizontal cells in the lower vertebrate retina.

2 Early Vision Problems via Regularization

When a solution to an operator equation,

$$Av = d, \quad v \in X, \quad d \in Y, \tag{1}$$

loses existence or uniqueness or continuity in d, equation (1) is called ill-posed. The Tikhonov regularization [1] converts (1) into a family of minimization problems:

$$G(v, d, \lambda) = ||Av - d||^2 + \lambda \Omega(v), \quad \lambda > 0 \tag{2}$$

where $||\cdot||$ denotes a norm, $\Omega : X \rightarrow R$ is continuous and strictly convex. If $Av^* = d^*$, then under reasonable conditions, (2) regularizes (1) in the sense that for any ϵ-neighborhood $N_\epsilon(v^*)$of v^*, there is a δ-neighborhood $N_\delta(d^*)$ of d^* such that if $d \in N_\delta(d^*)$, and if $\lambda(\delta) > 0$ is appropriate, then there is a unique $v(d, \lambda(\delta)) \in N_\delta(v^*)$ which minimizes (2). It is argued in [2] that many of the early vision problems (edge detection, stereo, optical flow, etc.) can be formulated as Tikhonov regularization problems.

Let $\mathbf{v} := (v_1, v_2, ..., v_n) \in R^n$ and replace the derivative operations by the difference operations, e.g., $(\frac{dv}{dx})(x) \rightarrow v_k - v_{k-1}$, $(\frac{d^2v}{dx^2})(x) \rightarrow v_{k+1} - 2v_k + v_{k-1}$ which can be put in a vector form as $(\frac{dv}{dx})(\cdot) \rightarrow \mathbf{Dv}$, $(\frac{d^2v}{dx^2})(\cdot) \rightarrow \mathbf{Lv}$ respectively, where $\mathbf{D}$ and $\mathbf{L}$ are appropriate matrices. Therefore, the regularization problem on R^n corresponding to (2) calls for the minimization of

$$G(\mathbf{v}, \mathbf{d}) = ||\mathbf{Av}-\mathbf{d}||^2 + \sum_{r=1}^{P} \begin{cases} \lambda_r ||\mathbf{L}^{\frac{r}{2}}\mathbf{v}||^2 & r\text{: even} \\ \lambda_r ||\mathbf{D}\mathbf{L}^{\frac{r-1}{2}}\mathbf{v}||^2 & r\text{: odd} \end{cases} \tag{3}$$

where $\mathbf{A}$ is now a map between finite dimensional spaces and $\lambda_r := \lambda C_r$ are called the **regularization parameters**. Differentiating (3) with respect to $\mathbf{v}$ and setting it to zero, one has

$$\frac{1}{2}\frac{\partial G}{\partial \mathbf{v}} = \mathbf{A}^T(\mathbf{Av} - \mathbf{d}) + \sum_{r=1}^{P}(-1)^r \lambda_r \mathbf{L}^r \mathbf{v} = \mathbf{0}. \tag{4}$$

If $\lambda_r \neq 0$, the solution (4) contains the $\mathbf{L}^r\mathbf{v}$ term, which corresponds to the presence of the $d^{2r}v/dx^{2r}$ term in the infinite dimensional case. In this paper $\mathbf{A} = \mathbf{1}$, the identity map.

Let us explain how (4) can be naturally mapped into parallel resistive networks. Consider the case $P =$

1, and (4) reads $v_k - d_k - \lambda_1(v_{k-1} + v_{k+1} - 2v_k) = 0$, which is naturally mapped into a parallel resistive network where each node k is induced by a current source u_k, and connected with g_0 to ground and nodes k+1 and k-1 with g_1 with $\lambda_1 = g_1/g_0$, $d_k = u_k/g_0$ [3].

3 Light Adaptive Architectures

A. Double-Layer Network : In all the vision chip architectures that we know of, λ_r's are **fixed**. Our proposed architectures make λ_r variable so that adaptation can be incorporated. We will consider the minimization of (3) where λ_r is $\lambda_r(\mathbf{d})$, and among many possible adaptive networks, the smoothing-contrast enhancement filter network [4] has probably one of the most interesting structures suited for this adaptation. Let us first state the following fact proved in [5].

Fact Consider the double-layer network in Fig.1. (i) The second layer voltage distribution v_k^2 solves the second order regularization problem with $\lambda_1 = g_{s1}/g_{m1} + g_{s2}/g_{m2}$, $\lambda_2 = g_{s1}g_{s2}/g_{m1}g_{m2}$, $d_k = T_1/g_{m1}g_{m2}u_k$. (ii) The first layer voltage distribution v_k^1 solves the first order regularization problem with $\lambda_1' = g_{s1}/g_{m1}$, $d_k' = u_k/g_{m1}$. (iii) The difference $x_k := v_k^1 - v_k^2$ enhances contrast of u_k after smoothing.

Fig.2 shows the responses x_k to a narrow "slit" located at the center with two different g_{s2}'s, whereas other parameters are fixed. We see that the network response naturally approximates the $\nabla^2 G$ filter, and with different g_{s2}-values, one can have different "filter width". There is one feature associated with our network; **a larger g_{s2} results in a higher filter gain.** This is very natural if g_{s2} increases when the environment is darker, which is exactly what happens in vertebrate retina.

B. Global Adaptation Network : Consider the double-layer network in Fig.1, where g_{s2} has an adaptation mechanism described by

$$g_{s2}(\mathbf{u}) := 1/(G\sum_k u_k), \quad G > 0. \tag{5}$$

Then (i) the second layer voltage distribution v_k^2 solves the second order regularization problem with $\lambda_1(\mathbf{u}) = g_{s1}/g_{m1} + g_{s2}(\mathbf{u})/g_{m2}$, $\lambda_2(\mathbf{u}) = g_{s1}g_{s2}(\mathbf{u})/(g_{m1}g_{m2})$, $\lambda_2(\mathbf{u})/\lambda_1(\mathbf{u}) = 1/(g_{m1}/g_{s1} + g_{m2}G(\sum_k u_k))$, $d_k = T_1 u_k/(g_{m1}g_{m2})$. Statements (ii) and (iii) of Fact are still valid.

When the total input current $\sum_k u_k$ gets larger (the environment is **light**), g_{s2} and the ratio $\lambda_2(\mathbf{u})/\lambda_1(\mathbf{u})$ decrease. This means that when $\sum_k u_k$ is large, the emphasis of the network on the second order derivative decreases, which has rather interesting implications. Suppose that $u_k = u_k^0 + \xi_k$, where u_k^0 is the noiseless image while ξ_k stands for noise with zero mean. $\sum_k u_k$ large means that effect of noise is less significant than when $\sum_k u_k$ is smaller. Thus when $\sum_k u_k$ is smaller, noise is more significant and the network putsmore emphasis on the second order derivative penalty. Fig.3 show the network responses for a rectangular input image with the Gaussian white noise. Fig.3 (a) shows the network response x_k without adaptation. A dramatic effect is discernible when the g_{s2}-adaptation (5) is incorporated. Observe that while Fig.3 (a) gives no information about the edges of the original object, Fig.3 (b) which is the network response with the g_{s2}-adaptation given by (5), correctly identifies the edge of the original image by its zero crossings. Fig.4 gives the responses of the networks for a slit input.

C. Local Adaptation Network : If

$$g_{s2(k,k+1)} := 1/L(v_k^1 + v_{k+1}^1), \quad L > 0, \tag{6}$$

the second-layer horizontal conductance $g_{s2(k,k+1)}$ between nodes k and $k+1$ is inversely proportional to the sum of the first-layer voltages v_k^1 and v_{k+1}^1. Fig.5 (a) is a rectangular input while Fig.5 (b) compares the response incorporating the local adaptation (6) with those responses without adaptations. One can see that where the input intensity is high, the response with (6) is closer to that with $1/g_{s2} = 5M\Omega$, and where the intensity is low, the adapted response behaves similarly to the one with $1/g_{s2} = 500k\Omega$. Therefore with (6) contrast is even more enhanced where interesting difference exists.

4 Physiological Findings on Retina

This section describes a neural adaptation system found in the lower vertebrate retina. In the eye, light passes through the transparent retina to reach the photoreceptor which transduces light into an electrical signal. The electrical signal is transmitted to the second order neurons, which are the horizontal cell and the bipolar cell. These three types of neurons interact via chemical synapses in the outerplexiform layer. Previous physiological studies revealed that neighboring horizontal cells are coupled electrically and possess a large receptive field which sometimes covers almost the entire retina [6, 7]. It was also revealed that neighboring photoreceptors are also coupled electrically but the size of receptive field is much smaller than that of horizontal cell [8]. In the present study, the network of photoreceptor and horizontal cell is decsribed by the equivalent electrical circuit shown in Fig.1. Each

photoreceptor is represented by g_{m1} and each horizontal cell by g_{m2}. g_{s1} and g_{s2} represent the coupling resistance of photoreceptor and horizontal cell, respectively. The significance of electrical coupling is thought to be relevant to the noise reduction. When cells are electrically coupled, the current generated in a single cell spreads into neighboring cells, and the signal to noise ratio can be improved when the image has an appropriate size [9, 10]. However, the electrical coupling blurs the image in return.

The bipolar cell is the first neuron which exhibits a $\nabla^2 G$-like receptive field in the vertebrate visual system. The response to a stimulus placed in the center region antagonizes the one placed in the surround region. The center response is mediated by the direct input from photoreceptor and antagonistic surround response by the horizontal cell. The $\nabla^2 G$-like receptive field is able to perform a smoothing and contrast-enhancement of the input image simultaneously [11].

There is another type of neuron which modulates the signal processing in the outer plexiform layer. The interplexiform cell (IP cell) is a centrifugal neuron innervating to the horizontal cell [12]. It has been demostrated that IP cell reduces the receptive field size of horizontal cell by increasing the resistance g_{s2} [10]. The function of IP cell is not yet well understood. Considering previous physiological observations [13], however, we hypothesize that the IP cell *adaptively controls the receptive field size of the horizontal cell* according to the signal to noise ratio of the image. If the intrinsic noise is constant regardless of the adaptation level of the retina, the relative magnitude of noise to signal is small in the daytime since the light intensity of signal image is high. In that situation, the size of the bipolar cell receptive field is to be reduced to gain the spatial resolution. This adaptation is likely to be carried out by the IP cell in the retinal neural circuit and the architecture of the vision chip shown here was inspired by these physiological studies.

5 Analog CMOS Circuits for Adaptation

An analog CMOS implementation of the network in Fig.1 without adaptation has been reported in [4] (Fig.6). The photosensor is realized as a phototransistor and its current is converted to a voltage which is fed into the first layer network, and the node voltage of the first layer network is applied to the node of the second layer network. This implements the Thévenin equivalent of the current sources in Fig.1. The node voltage of the first layer network is subtracted from that of the second layer network, and it is read out through analog switches.

Fig.7 shows a circuit which realizes the global adaptation network. The voltage v_k^0 is first converted into current I_k by the V-I converter so that I_k is proportional to v_k^0, and the summed current I is given by $I := \sum_k I_k \propto \sum_k v_k^0$. I is fed into the bias voltage generator which produces a bias voltage v_c so that the g_{s2} value is inversely proportional to I. Also Fig.8 shows a block diagram of the local adaptation network.

References

[1] A. Tikhonov, "Solution of Incorrectly Formulated Problems and the Regularization Method," Soviet Math. Dokl. 1035-1038 (1963).

[2] T. Poggio, et.al., "Computational Vision and Regularization Theory," *Nature*, 314-319 (1985).

[3] C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley (1989).

[4] T. Matsumoto, et.al., "A Second Order Regularization Vision Chip for Smoothing-Contrast Enhancement," *IJCNN*, Beijing (1992).

[5] H. Kobayashi, et.al., "Image Processing Regularization Filters on Layered Architecture," *Neural Networks*, 327-350 (1993).

[6] K-I. Naka, et.al., "S-Potential from Luminosity Units in the Retina of the Fish (*Cyprinidae*)," *J. Physiol.*, 437-461 (1967).

[7] T. Yagi, "Interaction Between the Soma and the Axon Terminal of Retinal Horizontal Cells in *Cyprenus Carpio*," *J. Physiol.*, 121-135 (1986).

[8] D. Baylor, et.al., "Receptive Fields of Cones in the Retina of the Turtle," *J. Physiol.*, 256-294 (1971).

[9] T. Lamb, et.al., "The Relation Between Intracellular Coupling and Electrical Noise in Turtle Photoreceptors," *J. Physiol.*, 256-286 (1976).

[10] T. Teranishi, et.al., "Dopamine Modulates S-Potential Amplitude and Dye-Coupling between External Horizontal Cells in Carp Retina," *Nature*, 234-246 (1983).

[11] D.Marr, et.al., "Theory of edge detection," *Proc. of Royal Society London*, 187-217 (1980).

[12] J. Dowling, et.al., "Synaptic Organization of the Dopaminergic Neurons in the Rabbit Retina," *J. Comparative Neurology*, 203-220 (1978).

[13] T. Shigematsu, et.al., "Effects of Dopamine on Spatial Properties of Horizontal Cells in the Retina of the Goldfish," *Neurosci. Res. Supple.*, s69-s80 (1988).

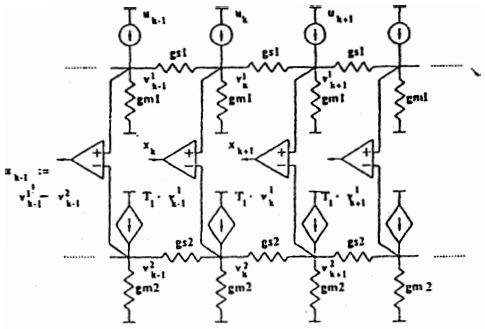


Fig.1 : The double layer network realizing a $\nabla^2 G$-like filter.

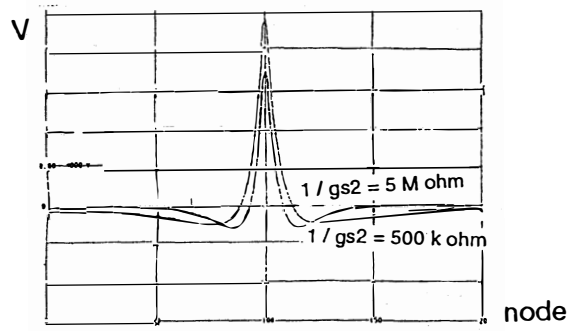


Fig.2 : Responses of the double layer network with $1/g_{s2} = 5M\Omega$ and $1/g_{s2} = 500k\Omega$. for a slit input.

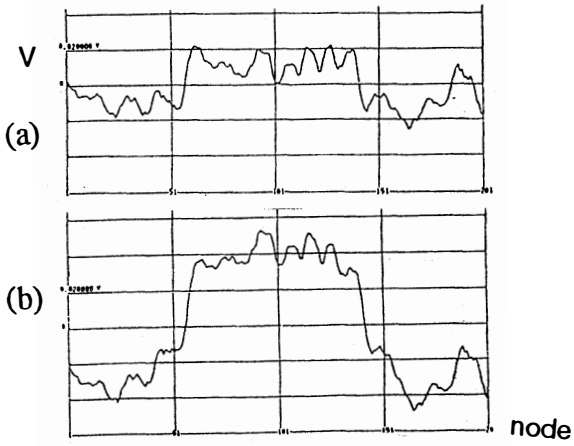


Fig.3 : Responses of the network in Fig.1
(a) with $1/g_{s2} = 5M\Omega$ (no adaptation)
(b) with $G = 1.0 \times 10^{13}$ (adaptation).

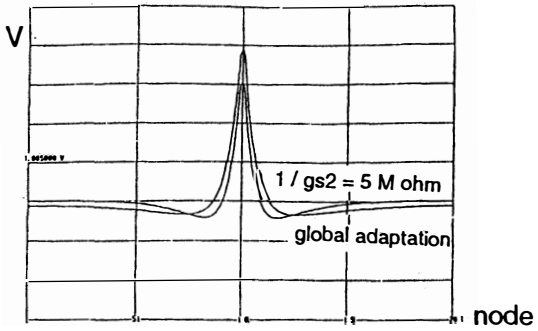


Fig.4 : Responses of the networks in Fig.1 with $1/g_{s2} = 5M\Omega$ (no adaptation) and $G = 2.0 \times 10^{15}$ (adaptation) for a slit input.

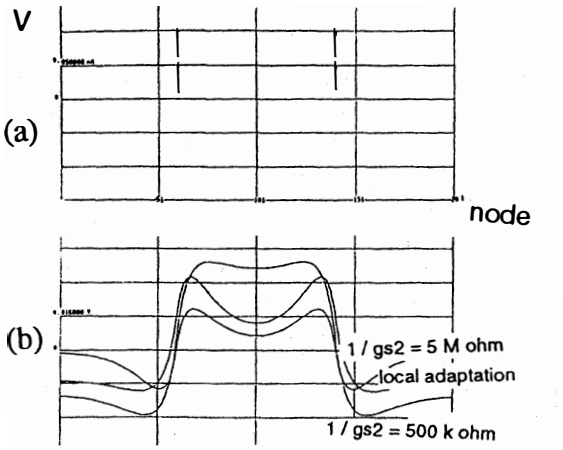


Fig.5 : Response of the locally adaptive network.
(a) A rectangular input image.
(b) Responses with $1/g_{s2} = 5M\Omega$, $1/g_{s2} = 500k\Omega$ (no adaptation) and $L = 2 \times 10^7$ (local adaptation).

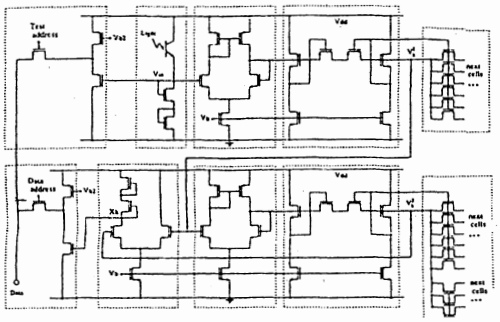


Fig.6 : Unit cell circuit of a double layered network [4].

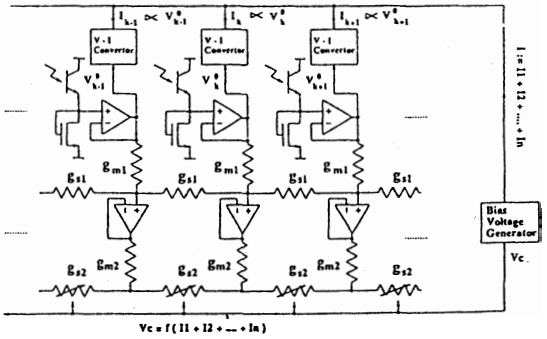


Fig.7: Block diagram of the global adaptive network.

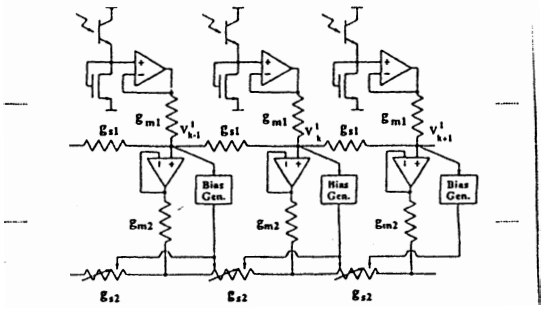


Fig.8 : Block diagram of the local adaptive network.

Aspects of Non-Cartesian Robotics

Takashi Gomi
Applied AI Systems, Inc.
340 March Road, Suite 600
Kanata, Ontario, Canada K2K 2E4
Tel. +1 613 592 3030; Fax. +1 613 592 2333
email: gomi@AppliedAI.com

Abstract

A new trend in robotics that began with the introduction of behavior-based artificial intelligence open up possibilities of creating very fast, flexible, robust, efficient, animate, and economic robots using methods much simpler than conventional robot development. These robots are fundamentally "open" in their design, and hence are amenable to incremental or evolutionary review of their functionality. Robots with such characteristics may be called non-Cartesian robots because of the strong underlying differences in philosophical background from conventional robotics [1]. Most efforts to describe and formalize the cause and theoretical aspects of those desirable characteristics are still speculative, and at best tentative. However, after almost a decade of R&D in the field, the effort for the application of a new form of robots is progressing steadfastly, some almost reaching the realm of a practical real world application. Using examples, this paper examines each of the desirable characteristics of robots implemented on behavior-based or non-Cartesian principles .

1. Introduction

The behavior-based approach to create robots and robot-like automata marks the debut of what is now referred to as non-Cartesian robotics. This approach, which began in the mid-1980's, produces robot motions which are faster and more flexible than those produced using traditional robot control methods and achieves similar functions with code which is up to one-thousandth the size of that using conventional methods. The most important aspect of behavior-based control is not in computer hardware or software but in the architecture. The crucial requirement is a framework which allows execution in parallel of short, simple processes developed in hardware, software, or their mixture, called *competence.* The *competence* structure implemented on a specific robot is non-hierarchical and there is no "main routine" or "control box". Self-organization among these mini-processes results in an emergent global behavior of the system as a whole. Execution time of each *competence* is very short, allowing pseudo-parallel execution of *competences* using even the simplest of conventional von Neumann processors. Because the global behavior of the robot is dependent on emergent properties of many mini-processes the robots demonstrate graceful degradation in the event of failure, rather than abrupt halting of the entire system as seen in conventional robots and computer systems.

Subsumption Architecture (SA) begun by R. Brooks of MIT [2] [3] [4] [5] is the most successful realization of behavior-based AI principles in robotics to date. An SA robot typically has several to several dozen hand-coded or automatically generated *competences.* Each *competence* takes in sensory inputs from one or more sensors and generates output to one or more actuators through which actions are generated in response to a specific situation.

D. Floreano and F. Mondada successfully achieved an evolutionary system using the Khepera robot, a small robot 5.5 cm in diameter and 3 cm tall [6]. It is connected to a workstation through an RS232 cable which uplinks sensor inputs and downlinks motor signals generated at the workstation to drive Khepera's two differential motors. The input layer of a two-layer feed-forward neural network is connected to Khepera's 6 forward and 2 backward-facing active infrared proximity sensors. One hundred individual control programs (neural networks) per generation are sequentially executed. Each network learns a set of *competences* in the form of input pattern/output signal pairs governed by weights in the network. A standard variant genetic algorithm with fitness scaling, roulette wheel selection, one point crossover, and high degrees of mutation is used. Individuals who evolved successfully navigated around a circular track with a smooth path and constant direction. Other similar achievements in recent years in this new field are also seen in [9] [10].

2. Characteristics of Non-Cartesian Robots

Figure 1 shows the serial nature of processing and the hierarchical decomposition of the procedural structure typical for a Cartesian intelligent robot. The insistence on a hierarchical module structure and hence the propagation of a fault which may originate at any module at any level upward through the hierarchy, and the lack of uniformity in module

structure are all characteristic of the Cartesian control system evidenced here.

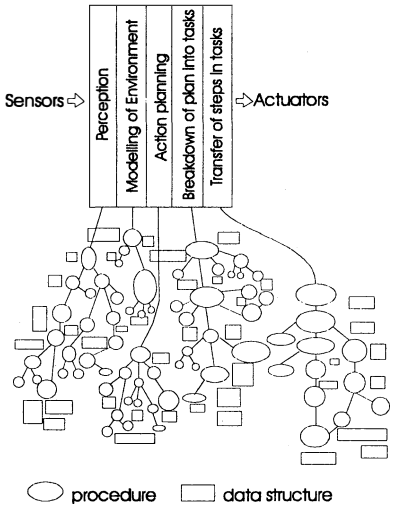


Figure 1 Structure of Cartesian intelligent robot system

This is contrasted by Figure 2 which depicts the module structure for a typical non-Cartesian behavior-based robot. Here there is relative uniformity of the modules (*competences)*, even distribution of the modules, and no static master-slave relationships, no explicit functional or signal connections among the modules, and no center of control. Although it is not obvious from the two diagrams, there is also a large difference in the size of code required (often 1,000 to 1 difference) when implemented in software and an estimated 100 to 1 difference in the number of circuit components when implemented in hardware.

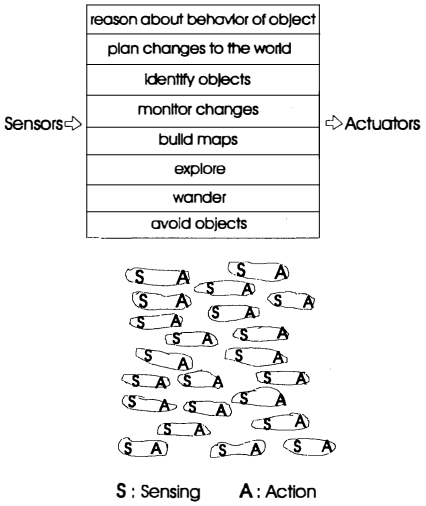


Figure 2 Non-Cartesian intelligent robot system

Some of the differences in characteristics of Cartesian robots and non-Cartesian robots are summarized in Table 1.

Table 1 Comparison of Cartesian and non-Cartesian intelligent robots

Aspects	*Cartesian intelligent robots*	*Non-Cartesian intelligent robots*
Nature of operation	**Control**	**Autonomy**
Identification of entities	**Definitions** - Defining action, position, movement, task, work, and operational environment	**No definitions** - Position, motion, even effective spheres in operational environment are *emergent*
Representation of entities in robot	**Models** to document and visualize the **defined** entities	**No models** - instead, a robot acquires or generates a collection of *competences* to cope with reality
Monitoring of robot's actions	**Measurements** - positions, velocity, momentum, mass, force, all need to be measured continuously	**No measurements**. Humans as an intelligent agent, for example, do not measure parameters in their execution of actions
Nature of processing	Continuous **computation** to verify compliance to the **definitions** and **models** and to carry out **measurements**	**No computation**. It is not **computation** that mobilizes actuators of highly intelligent agents but reactive signals

Generation of action sequence	**Planning** - tasks and a detailed sequence of actions in each task need to be planned in advance and in explicit terms, and monitored through **measurements** during execution	**Emergent plans**. Most serious real world tasks are too complex to be explicitly planned. Only simple task sequences (agenda) would be given. Action sequences *emerge* as a result of interactions among **goals, situations** in the operational environment, and robot's internal **conditions**
Method of obtaining "intelligence"	**Algorithm** prepared in accordance with a *plan* is executed and "intelligence" is assumed to be generated as a result	**No algorithm** -intelligence *emerges* as a result of self-organization among *competence* processes running in parallel
Mode of execution	**Centralized control** or implied **central control**	**No centralization nor control.** Spontaneous and asynchronous invocation of independent *competence* modules
Distribution of action generation mechanism	Overt and/or covert **singularity**	**No singularity** - *competences* are fully distributed throughout system
Method of managing levels of abstraction	Explicit static functional **hierarchy** designed and implemented in control structure	**No** predefined **hierarchy** - hierarchical structures may emerge dynamically during execution but none permanent

3. Significant Behaviors Generated by Non-Cartesian Robots

The observed unique features of non-Cartesian robots are:

- Speed and agility: as a mobile and a stationary robot, this new approach achieves unprecedented speed of action;
- Flexibility: the robots have the ability to rapidly adjust their response to changing constraints;
- Robustness: the robots persistently resist external and internal factors attacking their wellness, allowing a true graceful degradation;
- Efficiency: the robots exhibit considerable code, material, energy, and size/weight efficiency;
- Evolvability: it is drastically easier to apply evolutionary computational methods to non-Cartesian robots than to conventional counterparts;
- Naturalness: highly animate motions with naturalistic accelerations and non-uniform reactions can be easily generated;
- Economy: smaller hardware, software development, and testing costs for similar functions; and,
- Affordability: the robots can be implemented by those with less formal training and experience in robotics.

The theories and mechanisms that are responsible for these features are currently being studied by a range of researchers. Existing theoretical attempts, some in robotics and others in Computer Science, AI and their neighboring fields include the ones shown below:

- COG [7] Generation of human-like consciousness by artificial means through a fully *situated* and *embodied* robot.
- *Emergent Computation* [8] A relatively new theory on computation which takes a number of independent, simple task-achieving computational modules and executes them in parallel in a fully distributed manner to observe the emergence of intelligence.
- *Selectionism* [9] Each *competence* is decomposed into behavior systems which in turn have a dynamic relationship between a set of sensory and action pairs and a set of internal quantities.
- *Fungus Eaters* [10] An autonomous agent cannot simply go after its predisposed tasks but also must look after other functions such as eating.
- *Enaction* [11] There are common principles between sensory (perception) and motor (action) systems such that how the perceiver is embodied affects its actions.
- *PerAc* [12] Emphasizes the importance of perception-action pairs *'sans processing'* as a foundation for intelligent actions.
- *Collectionism* (C. Langton) High level structure *emerges* to govern the operation of the system through self-organization of low level reactive functions.

The following list points out some of the examples of non-Cartesian robot's behavioral characteristics which open up their application in heretofore difficult fields such as service applications.

1) Swift movement at very high efficiency

Since non-Cartesian robots do not execute predefined algorithms to generate their motions, it can be made highly reactive. This ability to activate their actuators much more rapidly and spontaneously allow them to respond to changing situations detected through sensors more appropriately.

2) Robust recovery from confinement
   Confinement is a situation in which a robot's movements towards achieving its goal is undesirably restricted. By providing a robot with the necessary set of *competences* to break loose from such a situation, allowing them be invoked autonomously through sensor signals, the robot can dislodge itself from confinement.
3) Flexibility of emergent trajectory
   Since actuators are under the direct influence of sensor signals, and there is no explicit path planned for their motion nor algorithm to generate it, the trajectory of the robot's actuator is *emergent*, as in humans. It is up to sensor inputs and a specific *competence* associated with them to adjust the trajectory in accordance with a given situation.
4) Persistent behaviors
   Robots can be built to automatically repeat behaviors in difficulties towards achievement of a goal.
5) Flexible adaptation to changing goals
   The goal of a robot can also be changed and adapted to respond to changing situations.
6) Emotive factor
   Robot's movement can be easily modified by emotion-like internal state changes.

4. Discussion

As described above, non-Cartesian robots have a number of significant performance characteristics which are unique and appear useful in a new range of robot applications. These robots are capable of producing movements which have not been achieved by most conventional robot control methods. This inspired us to believe non-Cartesian robotics will be highly suitable for building robots which conduct tasks that are commonly found in service applications such as cleaning, transportation, dynamic interaction with humans and other robots, explorations, underwater activities, and tasks which are generally dangerous to humans. The most significant aspect of non-Cartesian robots which have these promising as well as interesting characteristics is that these task-achieving behaviors are realizable with a very limited amount of resources. This simplicity is due to the dependence of non-Cartesian robots on *emergent* functions generated from executions of *competences* either implemented in computer software or hardware, rather than direct and explicit execution of instructions. This is a stark contrast to the way motions of robots are generated in conventional robotics where the movement of the robot or robot parts is the direct result of the execution of explicit algorithms implemented in either computer program or electronic circuity.

References

[1] T. Gomi, "Non-Cartesian Robotics", *Special Issue on Biorobotics in Robotics and Autonomous Systems*, Elsevier Science, to be published in January 1996.

[2] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, RA-2, March 1986.

[3] R.A. Brooks, "New Approaches to Robotics", in *Science*, Volume 253, 13 September 1991.

[4] R.A. Brooks, "Intelligence Without Reason", *Proceedings, International Joint Conference on Artificial Intelligence (IJCAI'91)*, Sydney, Australia, August 1991.

[5] R.A. Brooks, "Intelligence Without Reason", revised version in *The Artificial Life Route to Artificial Intelligence*, edited by Luc Steels and Rodney Brooks, Lawrence Erlbaum Associates, Publishers, 1995.

[6] D. Floreano and F. Mondada, "Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot", *Proceedings, Third Conference on Simulation of Adaptive Behavior (SAB'94)*, edited by D. Cliff, P. Husbands, J. Meyer, S.W. Wilson, MIT Press, Bradford Books, 1994.

[7] R.A. Brooks and L.A. Stein, "Building Brains for Bodies" *MIT A.I. Memo No. 1439*, August 1993.

[8] S. Forrest, "Emergent Computation" in *Emergent Computation*, The MIT Press, Cambridge, MA, 1991.

[9] L. Steels, "A Case Study in the Behavior-Oriented Design of an Autonomous Agent", *Proceedings, Third Conference on Simulation of Adaptive Behavior (SAB'94)*, edited by D. Cliff, P. Husbands, J. Meyer, S.W. Wilson, MIT Press, Bradford Books, 1994.

[10] R. Pfeifer, B.Goetz, C. Scheier, "Building Fungus Eaters: Accounting for Self-Sufficiency", Submitted to *Artificial Life IV, Fourth International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge, MA, July 1994.

[11] F.J. Varela, "The Re-Enchantment of the Concrete", in *The Artificial Life Route to Artificial Intelligence, Building Embodied, Situated* Agents, edited by Luc Steels and Rodney Brooks, Lawrence Erlbaum Associates, Publishers, 1995.

[12] J.D. Nicoud, Introduction at *From Perception to Action*, Lausanne, Switzerland, September 1994.

Artificial Retina Chips

Kazuo Kyuma and Jun Ohta
Advanced Technology R&D Center, Mitsubishi Electric Corporation
8-1-1 Tsukaguchi-Honmachi, Amagasaki-city, Hyogo, 661 Japan
(TEL:81-6-497-7078, FAX:81-6-497-7288, e-mail:kyuma@qua.crl.melco.co.jp)

Abstract

We have developed a novel type of image sensor called an artificial retina which can simultaneously sense and process images. Device structure, fundamental performance, operation principle of image processing, processing functions, and applications are described.

1. Introduction

Recognition and understanding of real world images is one of the most important information processing technologies for the multimedia systems of 21st century. Such images contain a very large amount of information which present image processing systems cannot analyze in real time. This is because the present systems separate image sensing and image processing ; images are sensed by camera and processed by computer. Consequently performance is limited by slow camera frame rate and low transmission rate between camera and computer. By contrast, human beings can execute very difficult tasks in real time. For example, we can instantly recall the names of the people by glancing at their faces, and we can analyze the motion information by selectively detecting the moving object. This is due to the inherent parallelism of our visual system, particularly of our retina, which combines image sensing and processing functions.

The artificial retinas defined in this paper are devices that can simultaneously sense and process images. To date, several kinds of related devices have been developed, such as vision chips, focal plane processors and silicon retinas. However, the specialized functions of the devices reported so far precluded them from use in large-scale industrial or commercial applications. Therefore, devices with more on-chip functionality are required. The most profound distinction of our artificial retinas compared to other devices is its on-chip functional flexibility.

2. Structure, Performance

Figure 1 shows our artificial retina which consists of a 2-D (256×256) variable sensitivity photodetection cell (VSPC) array, a random scanner for sensitivity control, and a multiplexer for output. These circuits are integrated on a Si substrate using 1-μm standard MOS technology. The VSPC's contains a pn photodiode (PD) and a transistor circuit, which functions are to set the detection sensitivity of the VSPC to one of the three values, (1,0,-1), by applying the control voltage to the random scanner, and to accumulate the photo-carriers to increase the sensitivity.

The performance of the fabricated artificial retina is summarized in Table 1. The size of each VSPC is 35×26 μm^2 and the total size of the chip is 12×12mm^2. The profound feature compared to other image sensors is its functional flexibility, that is, on-chip image processing capability, wide dynamic range (38dB), and variable frame rate (1-1000Hz). Figure 2 shows a microphotograph of the artificial retina.

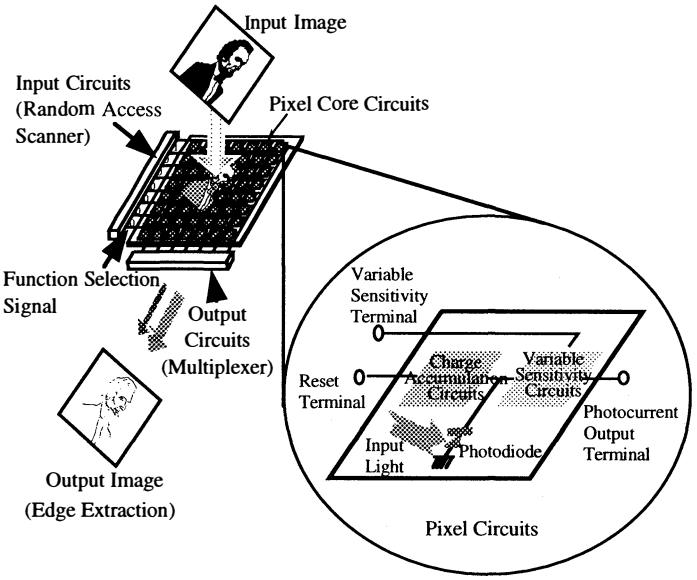
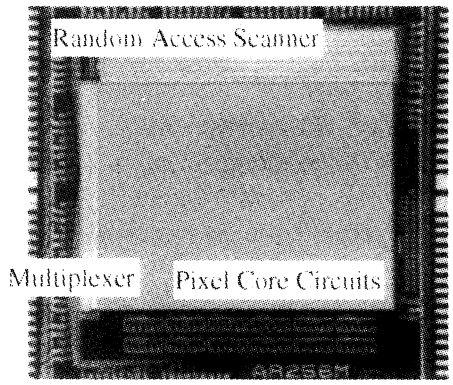


Fig.1: Image processing in the artificial retina as a matrix-vector multiplication. The product (vector J) of the input image (matrix W) with the sensitivity control voltage (vector S) is obtained through multiplexer in semi-parallel operation.

Table 1: Characteristics of the artificial retina.

Pixel Number	256 x 256
Pixel Size	35μm x 26 μm
Chip Size	12mm x 12mm
Sensitivity	0.20 μA/lx・msec
Dynamic Range	More than 40dB
Frame Rate	variable: 1Hz～1000Hz
Imager Type	MOS Type
Functions	Image Sensing, Edge Extraction, Noise Elimination, Pattern Matching, Attention Focus, Random Access, 2D→1D Projection



(12mm×12mm)

Fig.2: Microphotograph of the artificial retina with 256×256 variable sensitivity photodetection cell (VSPC) array.

3. Operation Principle, Processing Functions

The on-chip image processing is based on optical matrix-vector multiplication. In Fig.1, the input image is directly projected onto the chip as the weight matrix W. All VSPC's have one electrode (detection sensitivity electrode) connected along rows, yielding the sensitivity control vector S. Thus the VSPC sensitivities are set to one of the three values (1,0,-1) at each row. On the other hand, the output electrode is connected along columns, yielding an output photocurrent vector product J=WS. The remaining electrode (reset electrode) is used to reset the accumulated photo-carriers.

The significant feature of our artificial retina over other devices is the variety of processing that can be achieved through simply changing the control voltage pattern S. Several examples of processing are summarized in Fig.3, including TV-camera like image sensing, edge extraction, image smoothing, pattern matching, image compression/recognition. Let us describe three examples.

The first example is the edge extraction operation in Fig.3(b). The sensitivities of two adjacent VSPC rows are set to+1 and −1, respectively, while all other sensitivities are set to 0. In this case, the output current is proportional to the difference in light intensities of the two active rows. By shifting the control voltage pattern cyclically (0,+1,−1,0,0,...0), the horizontal edges of the input image are extracted. Thus the system operates in a time-sequential and semi-parallel operation mode. The processing time, which is equal to the frame rate, is 1ms at fastest. The experimental results of the edge extraction is depicted in Fig.4(b), together with the image taken by the TV-camera like image sensing mode (Fig.4(a)). These experiments are made by the camera system with the artificial retina (see Fig.5), under the conditions of 8msec frame rate and 15mW/cm^2 illumination power.

The second example is a random access function (see

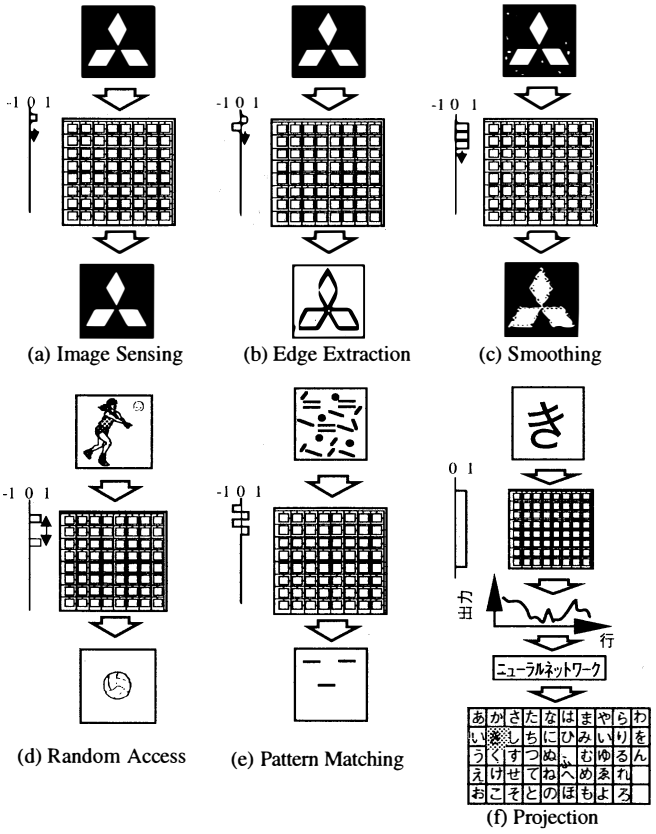


Fig.3: Examples of image processing options in the artificial retina.

(a)TV-camera like image sensing mode: all VSPC rows are successively scanned with a single polarity voltage, one at a time.

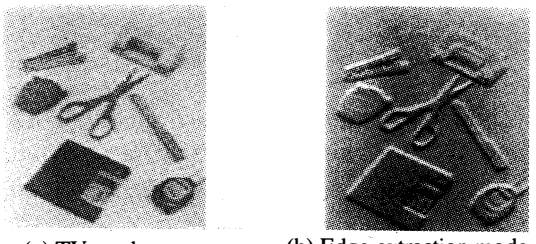
(b)Edge extraction mode: two adjacent rows are addressed with opposite polarity.

(c)Smoothing mode: voltage sequence such as (1,1,1) is used to suppress noise and smooth out edges.

(d)Random access mode: a part of the rows is selectively scanned to take out a partial image projected onto the retina chip.

(e)Pattern matching mode: the target item is detected by using it as the scanned control signal.

(f)Projection mode: all rows are simultaneously addressed with the same polarity, and the retina output (projection) is then fed into a neural network for classification.



(a) TV mode (b) Edge extraction mode

Fig.4: Experimental photographs of edge extraction mode (b), together with the image by the TV-camera like image sensing mode (a).

Fig.5: Photograph of the artificial retina camera system.

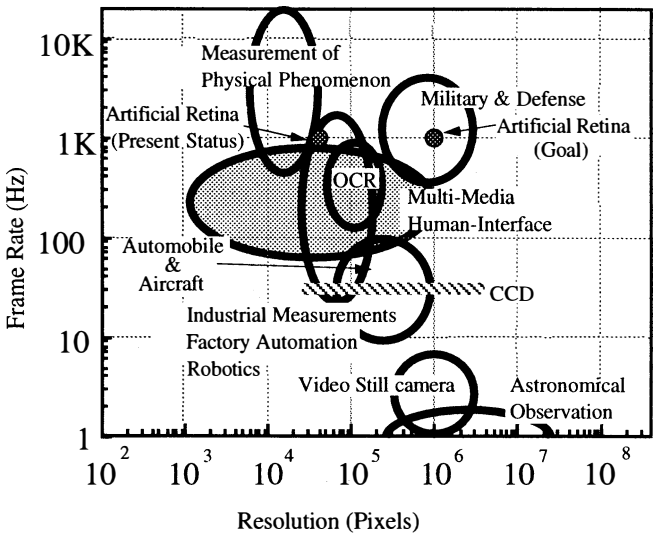


Fig.7: Performance of the artificial retina and potential application areas.

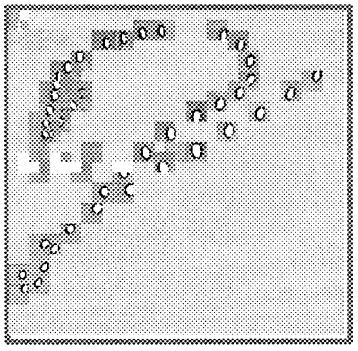


Fig.6: Experimental photograph of light spot tracking.

Fig.3 (d)) where only a part of the image projected onto the VSPC array (for example, a partial image of attention projected onto the 16×16 cells) is scanned by the random access scanner, and output by the multiplexer. The experimental result of the tracking of the moving object (light spot), which uses the random access function described above, is shown in Fig.6.

The third example is an image recognition system using a neural network as a post-processor of the artificial retina output. As shown in Fig.3(f), all VSPC's are set to a uniform sensitivity of one. The photocurrents (proportional to light intensity) in each VSPD column are summed up such that the output current vector, J, constitutes a vertical projection of the input image. The output signal J is a compressed representation of the input image W and fed into the neural network for recognition. We have applied this system to the optical character recognition, and succeeded in the recognition of the full set of 1945 Japanese Joyo Kanji characters with almost 100% recognition rate.

4. Applications

The application fields of artificial retinas include, as shown in Fig.7, man-machine interface for multimedia systems, automotive and avionics sensors, factory automation and robotics, video still camera, and defense applications. Particularly interesting use is a man-machine interface in multimedia systems where it can replace the present keyboard with gestures and/or body actions. Another interesting use is an artificial eye in moving vehicles where it can assist the driver by recognizing road markers. With its particularly fast response, the artificial retina could prevent collisions by sensing obstacles and approaching vehicles.

5. Conclusions (Towards New Generation Artificial Retinas)

Artificial retinas execute multiple, easily programmable forms of image processing with inherently high speed and parallelism. A new generation of high resolution retinas that incorporate memory devices and/or light emitting devices will further processing capabilities.

(References)

1) E.Lange et al.,IEEE MICRO, 14,6, p.29,1994.
2) K.Kyuma et al. nature,372,6502,p.197, 1994.

Distributed and Autonomous Sensing based on Immune Network

Yoshiteru Ishida
Graduate School of Information Science,
Nara Institute of Science and Technology,
Ikoma, Nara, 630-01 Japan

Abstract

A distributed and autonomous sensor network is proposed based on the informational feature of immune network; recognition of non-self by distributed and dynamically interacting units, recognition by simple comparison with units themselves, dynamical propagation of activation that would lead to system level recognition, and memory embedded as stable equilibrium states in the dynamical network.

The network is explained by an illustrative examle of eight coin puzzle: a balance must be used only three times to identify one coin with weight different from seven other coins. Our network also uses dynamical structure network rather than fixed structures used in neural networks. Simulations show that non-self (different coin in the eight coin puzzle, sensor/process fault in the monitoring example) will be identified by dynamically propagating activation through the network.

1 Introduction

Theory of immune network proposed by Jerne [1] provides the network view that B-cells are mutually and dynamically connected by antigen antibody interaction. Not only antigen but also antibody generated by B-cells will act as an angigen against the other B-cells, thus presupposing *internal image* of the antigen. Both antigen and its internal image activate the same specific type of B-cell. This network view would implicitly suggest that the immune memory is somehow embedded in the network; when disturbed by antigen, the network in an equilibrium will move to other equilibrium point, thus memorized the encounter with the antigen.

After the proposal of immune network (idiotypic network) theory by Jerne[1], systems approach have been made extensively to characterize immune response by dynamical models [2]. These dynamical models essentially describe population dynamics of antigen, antibody, and specific immune related cells in a similar manner to the models describing population dynamics of species in mathematical ecology.

Although the systems approach may explain some immunological phenomena by associating characteristics of immune response with interactions among antigen, antibody and B-cells, fundamental problems such as mechanism of immune memory still remained; whether the memory can be attributed to local cells (long-lived cell) or to the network of cells.

Several models based on informational feature of immune systems have been also studied [3, 4]. Our models have been developed [5] in a similar line to these models; rather than modeling immune systems, extracting important informational character for the present information systems such as robots, computers and computer network. Ultimate goal of our research is to develop a relation science that can deal with dynamical networks where the population of units, topology of network structure, and the mode/function of interactions may change.

Features of immune systems we have tried to use may be the following: recognition of non-self by distributed and dynamically interacting units, recognition by simple comparison with units themselves, dynamical propagation of activation that would lead to system level recognition, and memory embedded as stable equilibrium states in the dynamical network.

The immune network architecture we adopted can be understood with the following weighing metaphor. There are roughly two methods of weighing objects: one is using scale that require a sophisticated central

[1] Theory of immune network by Jerne is not much mentioned recently in immunology after more practical view of interleukin network (which is complex network involving not only B-cells but T-cells, MHC molecules; not only antibody but interleukin that do not correspond to specific antigen) becomes dominant. Nevertheless, it may give an important insight for information processing to attain a specific pattern recognition: self/non-self discrimination.

processing mechanism that maps weight to the information of number, another is using balance and many types of balance weights. The former is often accurate and efficient, once the mechanism of mapping is devised. However, the latter method of *object-against-object* offers a distributed and robust way of weighing. The network view of immune systems can be understood with this metaphor; regarding many types of balance weights as recognizing agent (immune related cells such as B-cell and T-cell that a receptor with specific antigen) and action of weighing by balance as recognition by paratope and epitope with spatial complementarity. The result of recognition is used to activate other recognizing agent, similarly to the fact that the result of balance is used to determine more appropriate balance weight against the target object. This *object-against-object* weighing is more robust, since the weighing mechanism is simple comparison and that the information is distributed into many balance weight. Although this metaphor is rough[2], it seems to reflect many important informational features of immune network such as specificity, diversity and control by activation-propagation.

Section ?? discusses

2 Sensor Network Based on the Concept of Immune Network

Based on the three features of immune network above, we proposed the dyanamical network which would solve self/nonself discrimination.

$$\begin{aligned} dr_i(t)/dt &= f(\{R_j(t)\},\{r_i(t)\}) \\ R_i(t) &= \frac{1}{1+\exp(-r_i(t))} \end{aligned}$$

In this network, not only state of units (i.e. $r_i(t)$ or $R_i(t)$) but also the connection between units are subject to change as presented in the next section. We have proposed several variations of this model, and analyzed their properties. To show that the network would have an emergent property, consider the problem of identifying nonself in the network shown in Fig. 1. Each unit decides the identity of other units; outputs $+/-$ depending on the target unit is self/non-self, respectively. However, non-self units may not follow this rule. State variable (r_i and its normalization R_i) indicating the identity of units are assigned to each unit; $R_i = 1$ for self $R_i = 0$ for non-self. Dynamical system is constructed by associating the time derivative of the state variable with state variables of other units connected by the evaluation chain.

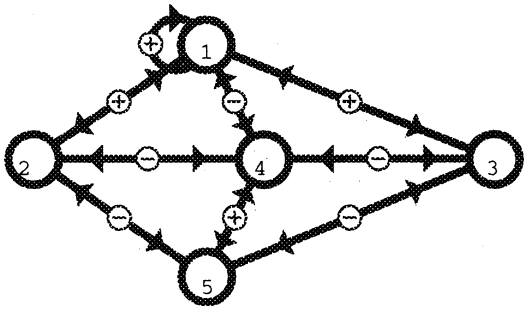


Figure 1: An example of evaluation chain

A possible association corresponding to the evaluation chain would be the following dynamical system:

$$\begin{aligned} dr_1(t)/dt &= -4R_4(t) - r_1(t) \\ dr_2(t)/dt &= -4R_4(t) - 4R_5(t) - r_2(t) \\ dr_3(t)/dt &= -4R_4(t) - 4R_5(t) - r_3(t) \\ dr_4(t)/dt &= -4R_1(t) - 4R_2(t) - 4R_3(t) - r_4(t) \\ dr_5(t)/dt &= -4R_2(t) - 4R_3(t) - r_5(t) \end{aligned}$$

When the units 4 and 5 are nonself, we have the test result as shown in Fig. 1. Simple voting at each unit does not work, since three units 2, 3 and 5 are all evaluated as nonself by two other units and hence cannot be ranked in terms of the possibility of nonself. However, the above dynamical model would solve the problem [7].

3 The Eight Coin Problem

In this section, we elaborate the model so that a set of units can test other set of units and that interactions are also dynamically activated.

The eight coin problem is the famous puzzle: how many times a balance must be used to identify the one coin whose weight is different from other seven coins. To solve the puzzle, let eight be labeled by $1,2,3,4,5,6,7,8$ and grouped into: $\{1,2,3\}, \{4,5,6\}, \{1,4\}, \{2,5\}$. Then the coin with different weight will be identified using balance only three

[2] This metaphor would be closer to immune network if recognizing unit (balance weight equiped with balance) can reproduce its clone with mutation to enhance elimination capability and affinity. In this paper, however, we do not persue the adaptive behavior by adaptive change of the population of specific agent mentioned reproduction and mutation above. These features are discussed in the immune algorithm we proposed elsewehre [6].

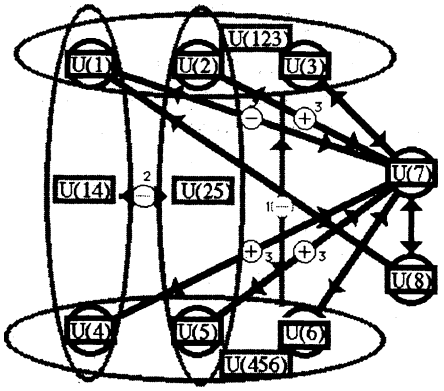


Figure 2: Network Expressing the Eight Coin Problem when the Coin 1 is different

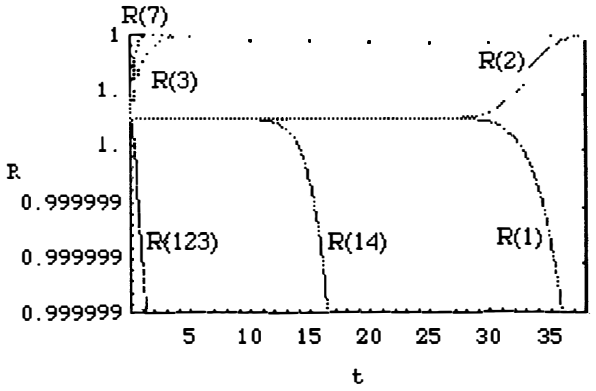


Figure 3: Simulation for the Eight Coin Problem when the Coin 1 is different

times among these groups and coins. Let $U(ij...k)$ indicate the total weight of all the coin in the set $\{i, j, .., k\}$. First compare the weight between $U(123)$ and $U(456)$. When their weights are not equal then the coin with different weight is included in these sets, subsequently narrowing the set by comparing $U(14)$ and $U(25)$. Otherwise, $U(7)$ and $U(8)$ will be compared. The third comparison will be performed based on the results of these two comparisons. Thus, control of the sequence of comparisons allow the gradual narrowing of the subset including the coin with different weight and small number of comparison. This puzzle motivates our network that activates comparison based on the state of other units.

$dr_i(t)/dt = \sum_j w_{ji} R_j + \sum_i w_{ij} R_j - \frac{1}{2} \sum_i (w_{ij} + 1)$

where $w_{ij} = \alpha_{i,j}(t) T_{ij}$ and T_{ij} is comparison result between $U(i)$ and $U(j)$: $+1$ for equal and -1 for not equal, thus $T_{ij} = T_{ji}$ for this example. $\alpha_{ij}(t)$ (also assumed to be symmetrical in this example) is activation level dependent upon the state of the units that activate or inactivate the link from the unit i to j. For the simulation, the following is used.

$$
\begin{aligned}
\alpha_{123,456}(t) &= 1 \\
\alpha_{14,25}(t) &= 1 - (R_{123}(t) + R_{456}(t))/2 \\
\alpha_{1,8}(t) &= 1 - (R_7(t) + R_8(t))/2 \\
\alpha_{1,7}(t) &= 1 - (R_7(t) + R_8(t) + R_{14}(t) + R_{25}(t))/4 \\
\alpha_{2,7}(t) &= 1 - (R_{14}(t) + R_{25}(t))/2 \\
\alpha_{3,7}(t) &= (R_{14}(t) + R_{25}(t))/2 \\
\alpha_{7,8}(t) &= (R_{123}(t) + R_{456}(t))/2 \\
\alpha_{6,7}(t) &= (R_{14}(t) + R_{25}(t))/2 \\
\alpha_{4,7}(t) &= 1 - (R_{14}(t) + R_{25}(t))/2 \\
\alpha_{5,7}(t) &= 1 - (R_{14}(t) + R_{25}(t))/2
\end{aligned}
$$

In this modeling, we use the binary information whether two weights are equal or not, hence the interaction between units is symmetric; $\alpha_{ij}(t)T_{ij} = \alpha_{ji}(t)T_{ji}$ Fig. 2 shows the network solving the eight coin puzzle when the coin 1 is different. Comparison associated with +(-) indicates that these weights are equal(different) and the number beside the sign is the order when of the test activation. Comparisons without +/- are those not activated. Fig. 3 shows the simulation result for this case. It is known that the subset including the coin with different weight is subsequently narrowed down until the coin 1 is identified finally. This would capture not only structurally dynamical feature of immune net, but also its cooperative interaction for identifying nonself.

4 Application to Flow Monitoring Problem

Consider the example of flows shown in Fig. 4. The constraints of this flow follows:

$$
\begin{aligned}
F(2) + F(4) &= F(3) \\
F(1) + F(2) + F(4) &= F(5) \\
F(1) + F(3) &= F(4) + F(6) \\
F(1) + F(3) &= F(5) \\
F(5) &= F(6) + F(4)
\end{aligned}
$$

Let U(i) indicate flow lost after the sensor i. Checking the balance of each equation can identify whether the flow lost occurs after the sensors of the left handsinde of the equation. When the balance of the equation

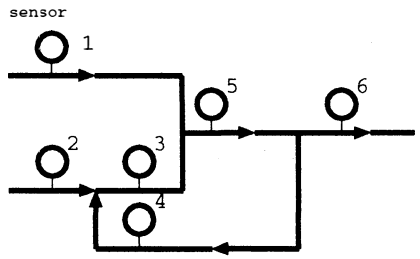


Figure 4: An Example of Flow Network

$F(1) + F(3) = F(4) + F(6)$ hold, then the balance of $F(1)+F(2)+F(4) = F(5)$ and $F(2)+F(4) = F(3)$ will be checked. Otherwise, the balance of $F(1) + F(3) = F(5)$ and $F(5) = F(6) + F(4)$ will be checked. Fig. 5 shows the test activation order and their results when flow after sensror 4 is lost. Fig. 6 shows the simulation result for the case.

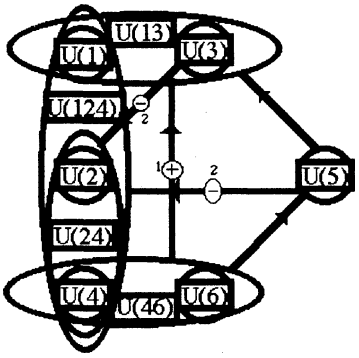


Figure 5: Network for the Flow Monitoring Problem

$$\alpha_{46,13}(t) = 1, \quad \alpha_{5,124}(t) = R_{13}(t), \quad \alpha_{3,24}(t) = R_{13}(t)$$

$$\alpha_{5,13}(t) = 1 - R_{13}(t), \quad \alpha_{46,5}(t) = 1 - R_{13}(t)$$

5 Conclusion

The distributed and autonomous sensor network we proposed is the first step towards self-organizing, learning, and evolving network found in immune networks. The advantage of this network is that it can synthesize the system level interpretation by cooperation of sensors, rather than sensing each value independently, as explained in the examples of the eight coin puzzle and the flow monitoring problem.

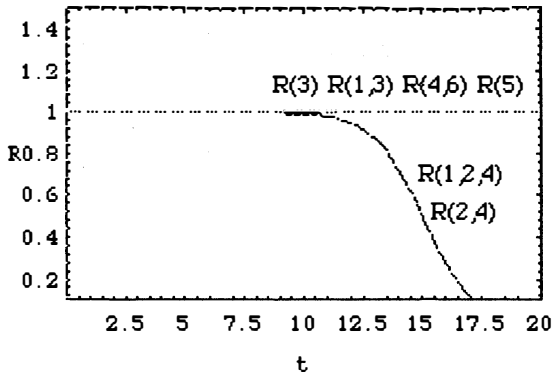


Figure 6: Simulation for the Flow Monitoring Problem

Application to the sensor integration of robot would lead to more life-like system. Enormous recognition capability of human may require not only conventional learning technology, but also self-organizing and evolving network of sensing found in immune systems.

For further research, aunotnomy of unit should be enhanced so that they can form groups and activate other links and units based on the environment.

References

[1] N. K. Jerne, "The Immune System" *Sci. Am.* Vol. 229, No.1, pp. 52-60, 1973.

[2] R. R. Mohler, C. Bruni, and A. Gandolfi, "A Systems Approach to Immunology", *Proc. of the IEEE*, Vol. 68, No. 8, pp. 964-990, August, 1980.

[3] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The Immune Systems, Adaptation, and Machine Learning" *Physica,* 22D, 187, 1986.

[4] H. Atlan and I. R. Cohen, Eds, *Theories of Immune Networks*, Springer-Verlag, 1989.

[5] Y. Ishida, " Fully Distributed Diagnosis by PDP Learning Algorithm: Towards Immune Network PDP Model",Proc. of IJCNN 90,San Diego, 1990.

[6] Y. Ishida and N. Adachi, "An Immunological Algorithm and Its Application to Disturbance Rejection," NAIST-IS-95030, 1995.

[7] Y. Ishida and F. Myzessyn, "An Immune Network Approach for Sensor-Based Diagnosis with Sensor Faults," NAIST-IS-95027,1995.

Artificial Active Vision Using Spatial Coding by M-Sequence

Sadao Fujimura and Senya Kiyasu
Department of Mathematical Engineering and Information Physics,
Graduate School of Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113, JAPAN

Abstract

Methods for measuring the 3-D shape of objects using spatial coding by M-sequence are described. Using a micro property of an M-sequence, the pattern of a part of M-sequence observed is placed on the original M-sequence. This enables us to correspond a point in an image of an object to a point in the other image of the object. The correspondence is used to measure the 3-D shape of objects with diffuse surfaces as well as those with specular surfaces.

For diffuse objects, a grid pattern coded by an M-sequence is projected onto an object and observed by a TV camera. The position of each point on the object is determined by triangulation.

For specular objects, the plane light source is coded by an M-array which is a two-dimensional array made by an M-sequence. The array pattern reflected by a specular object is observed by a TV camera. The position of each point on the object is determined by using geometry of reflection.

Key Words: computer vision, triangulation, M-array, diffuse objects, specular objects

1 Introduction

Recognition of objects using vision is one of the most important task for intelligent robots. Binocular stereo is a typical technique of passive measurement. However, there are many difficulties in determining correspondence between two images and it requires a lot of time for processing. Slit light or spot light projection methods [1] are often used for active measurement. However, they require special equipments and take rather long time for scanning.

Spatial coding techniques [2], which also belong to the active methods, do not require scanning. Most of them, however, require to process many images. This also takes a lot of time. Here we present two methods for measuring the 3-D shape of objects using spatial coding by M-sequence (maximum-length sequence), one objects with diffuse surfaces and the other for those with specular surfaces.

2 Spatial Coding by M-Sequence

An M-sequence [3] a_i of degree n, generated by a rule is a binary sequence of length $N=2^n-1$ specified by a primitive polynomial.

Although M-sequence is usually used as a random sequence (macro property), we use a micro property of M-sequence "window property": that is, if a window of size n is slid along the sequence, each of the 2^n-1 nonzero binary n-tuples is seen through the window once and only once when the end of the sequence is connected to the other end of the sequence. We project a grid pattern onto a diffuse surface object. The pattern is encoded by black and gray stripes according to the M-sequence.

The width of an observed stripe changes according to the orientation of surfaces, on which the stripe is projected. Therefore the number of stripes is not determined if the same level of stripes are neighboring. We put a white stripe "return-to-zero code" between every black and/or gray stripe as shown in Fig. 1.

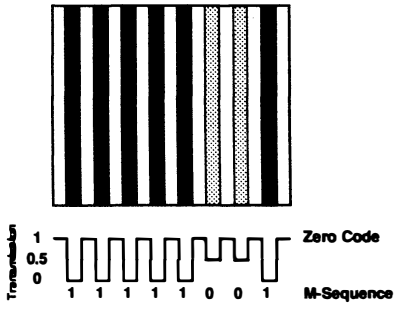


Fig. 1. M-sequence coded grid pattern

Thus three level coding is a key in this method. Each stripe is recognized correctly by quantizing the image of TV camera into three levels.

The M-array is a two-dimensional artificial random pattern constructed from an M-sequence. We take a number $N=2^{k_1k_2}-1$ such that $N_1(=2^{k_1}-1)$ and N_2 $(=N/N_1)$ are prime to each other and greater than 1. Then the M-array $b(j,k)$ of size $N_1\times N_2$ is obtained by

$$b(i \bmod N_1,\ i \bmod N_2) = a_i, \qquad (i=1,\cdots,N) \quad (1)$$

where a_i is an M-sequence. This means that the M-sequence fills an $N_1\times N_2$ array by writing a_i down the main diagonal and continuing from the opposite side whenever an edge is reached.

The M-array has also the "window property": that is, if a prescribed size $(k_1\times k_2)$ of window is slid over the array, each of the $2^{k_1k_2}-1$ possible nonzero arrays is seen through the window once and only once when the end of the array is considered to be connected to the other side of the array.

We encode the plane light source by circles on the white background assigned by gray and black according to the M-array. This is also three level coding. Light source of size $(N_1+k_1-1)\times(N_2+k_2-1)$ is constructed from the M-array of size $N_1\times N_2$. In the 18×18 binary pattern shown in Fig. 2, every 4×2 binary pattern appears just once.

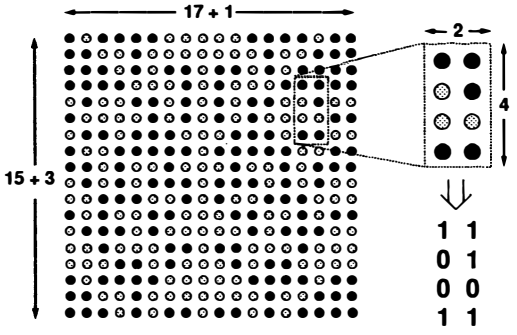


Fig. 2. An M-array coded light source and the window property

This property is used to match the pattern of a part of M-sequence (M-array) with a part of the same sequence (array). Thus, each point on the pattern is uniquely positioned by using the regularity of the M-sequence or M-array. This enables us to correspond a point on an image of an object to a point on the other image of the object. The correspondence is used to measure the 3-D shape of objects with diffuse surfaces as well as those with specular surfaces.

3 Measurement of the 3-D Shape of Diffuse Objects

3.1 Principle of Measurement

For diffuse objects, a grid pattern coded by an M-sequence is projected and observed by a TV camera as shown in Fig. 3. The space is coded by an M-sequence, and divided into stripe areas, where (X_C, Y_C, Z_C) is the center of the camera lens, (X_P, Z_P) is the center of the projector lens, l_C and l_P are the distance of image and projector plane from each lens, respectively. The angles of each lens axis are denoted by θ_0 and Φ_0.

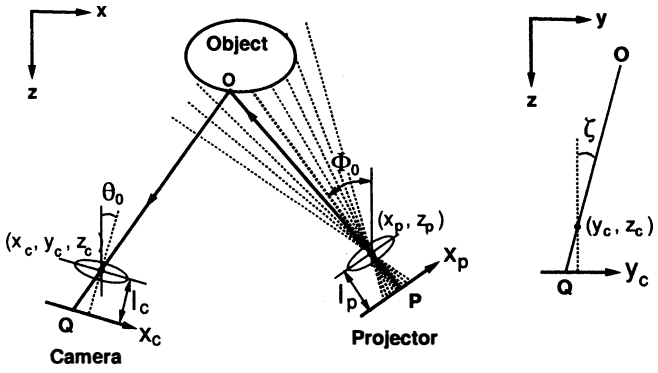


Fig. 3. System configuration for grid pattern projection

When the position of a point $P(x_P)$ on the projector's plane and that of a point $Q(x_c, y_c)$ on the image plane corresponding to the object point O are known, the coordinates of $O(X,Y,Z)$ are given by,

$$\begin{aligned} X &= X_c - Z'\tan(\theta+\theta_0), \\ Y &= Y_c - Z'\tan\xi, \\ Z &= Z_c - Z', \end{aligned} \qquad (2)$$

where

$$Z' = \frac{X_P - X_C + (Z_C - Z_P)\tan(\phi+\Phi_0)}{\tan(\phi+\Phi_0) - \tan(\theta+\theta_0)}, \quad (3)$$

$$\tan\theta = x_C/l_C, \quad \tan\xi = y_C/l_C, \quad \tan\phi = x_P/l_P,$$

and y axis of projector's plan is parallel to the Y axis.

To calculate the coordinates of O, we have to know the correspondence between P and Q. They are determined by the property of M-sequence. First, the image is normalized by using the image of uniform light source. Next, the image is quantized into three levels. Then a part of the sequence observed is matched

with a part of the M-sequence. By using the result of matching, the correspondence of the points P and Q are determined and the 3-D shape of the object is reconstructed by calculating the coordinates of the point O.

3.2 Experimental Results

Three objects (box, cylinder, bowl) are used for experiment. The distance from the object to a camera was about 100cm, and the distance from the camera to the projector was about 30cm. The angle between the axis of the camera and the axis of the projector is about 17 degrees. The effective size of the image acquired by the CCD camera was 127 × 256 pixels.

The result of reconstruction is shown in Fig. 4. Surface of three objects were reconstructed correctly. The accuracy of measurement was estimated to about 1cm for the objects set about 100cm apart from the camera.

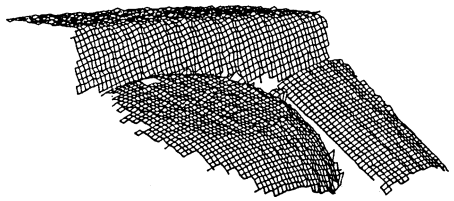


Fig. 4. Result of 3-D shape reconstruction of diffuse objects (box, cylinder, bowl)

4 Measurement of the 3-D Shape of Specular Polyhedrons

4.1 Principle of Measurement

For specular object, an image of a light source reflected on a specular surface is acquired by a TV camera.

Specular objects in an acquired image reflect the patterns on the light source plane. Using the window property, we can determine the position on the light source plane where the detected pattern comes from.

First, the image of coded light source is quantized into three levels after the brightness is normalized using the image of uniform light source. The black level means '1', and the gray level '0', respectively. The white level shows the background.

The point on the light source plane is found where the pattern matches the "window" pattern.

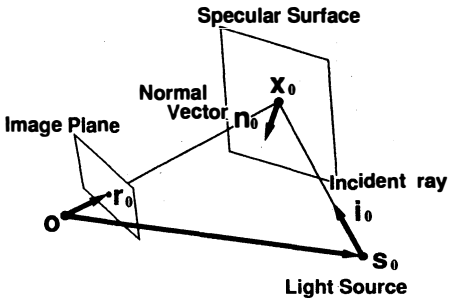


Fig. 5. Geometry of observation

We have determined the direction of the reflected ray and the corresponding point of the light source. However, they are not enough to determine the three-dimensional shape of the object,. We assume that the object is a polyhedron so that the local surface of the object is a plane.

Figure 5 illustrates the geometry of vectors we define, where $\mathbf{o}$ is the center of the camera, $\mathbf{x}_0$ the reflecting point on the specular surface, $\mathbf{r}_0$ and $\mathbf{s}_0$ the corresponding point on the image plane and the light source plane, respectively, and $\mathbf{i}_0$ a unit vector of the incident ray. The vector $\mathbf{r}_0$ indicates the direction of the reflected ray. As the normal vector $\mathbf{n}_0$ at point $\mathbf{x}_0$ lies on the plane spanned by $\mathbf{r}_0$ and $\mathbf{s}_0$, an equation

$$(\mathbf{r}_0 \times \mathbf{s}_0, \mathbf{n}_0) = 0 \tag{4}$$

holds, where (,) and × mean inner and outer products of vectors, respectively. If we take another point $\mathbf{x}_1$ on the surface close to $\mathbf{x}_0$, we obtain another equation

$$(\mathbf{r}_1 \times \mathbf{s}_1, \mathbf{n}_1) = 0, \tag{5}$$

If the points $\mathbf{x}_0$ and $\mathbf{x}_1$ lie on the same plane, the normal vectors $\mathbf{n}_0$ and $\mathbf{n}_1$ have the same direction. Thus $\mathbf{n}_0$ is given by

$$\mathbf{n}_0 = \frac{(\mathbf{r}_0 \times \mathbf{s}_0) \times (\mathbf{r}_1 \times \mathbf{s}_1)}{|(\mathbf{r}_0 \times \mathbf{s}_0) \times (\mathbf{r}_1 \times \mathbf{s}_1)|}. \tag{6}$$

In practice we determine the normal vector which minimizes the value of

$$\sum_{i=0}^{8} (\mathbf{r}_i \times \mathbf{s}_i, \mathbf{n}_0)^2 \tag{7}$$

to reduce the effect of measurement error, where $\mathbf{s}_i$ ($i = 1, \cdots, 8$) are the 8 nearest points around $\mathbf{s}_0$ on the light source plane and $\mathbf{r}_i$ ($i = 1, \cdots, 8$) the corresponding points on the image plane.

After the direction of $\mathbf{n}_0$ is determined, the position on the specular surface $\mathbf{x}_0$ can be determined by

using the condition that the normal vector $\mathbf{n}_0$ bisect the angle between the incident and the reflected rays. The vector of incident ray $\mathbf{i}_0$ is expressed by

$$\mathbf{i}_0 = \frac{2\,(\mathbf{r}_0, \mathbf{n}_0)\mathbf{n}_0 - \mathbf{r}_0}{|\mathbf{r}_0|}. \tag{8}$$

The point on the specular surface is written in two ways:

$$\mathbf{x}_0 = t\,\mathbf{r}_0 \tag{9}$$

and

$$\mathbf{x}_0 = u\,\mathbf{i}_0 + \mathbf{s}_0, \tag{10}$$

where t and u are real-valued parameters. From eqs. (9) and (10), we obtain the equation

$$t\,\mathbf{r}_0 \times \mathbf{i}_0 = \mathbf{s}_0 \times \mathbf{i}_0. \tag{11}$$

Thus, t is written as

$$t = \frac{(\mathbf{s}_0 \times \mathbf{i}_0, \mathbf{r}_0 \times \mathbf{i}_0)}{(\mathbf{r}_0 \times \mathbf{i}_0, \mathbf{r}_0 \times \mathbf{i}_0)}. \tag{12}$$

From eqs. (8), (9) and (12), we can determine point $\mathbf{x}_0$ on the specular surface.

4.2 Experimental Results

A specular polyhedron composed of three specular planes was used for the experiment. Each side of the object is about 10cm long, and the distance from the object to the camera was about 50cm, and that to the light source plane about 15cm. Images were acquired by a high-resolution CCD camera which provides a digitized image of 1024 × 1024 pixels.

The result of reconstruction is shown in Fig. 6. Calculated points on a surface are connected by lines. The dashed lines show the regression planes obtained from the calculated points. It is seen that each plane of the polyhedron is reconstructed correctly.

In order to estimate the precision of this measurement, one specular plane was measured 6 times with varying orientation. The mean residual is less than 1mm.

5 Conclusions

We have described two methods for measuring the three-dimensional shape of diffuse objects and specular objects as well using spatial coding by M-sequence. The shape of diffuse objects and specular polyhedron were reconstructed correctly. The precision of measurement was estimated to be about 1cm for diffuse objects about 100cm apart from a camera, and 1mm for specular objects about 50cm apart from a camera. Improvement of accuracy for the measurement of diffuse objects by using a camera with much higher resolution, and extension of the algorithm for specular objects to curved specular surfaces are the subjects for a future study.

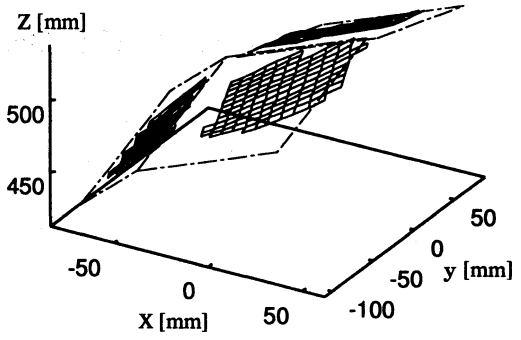


Fig. 6. Result of 3-D shape reconstruction of a specular polyhedron

References

[1] Y.Shirai, "Recognition of Polyhedrons with a Range Finder", *Pattern Recognition*, Vol.4, pp.243-250, 1972.

[2] S.Yonezawa and Y.Tamamura, "Coded Grating Method for Measuring Three Dimensional Objects", *Trans. IECE Japan*, Vol.J61-D, pp.411-418, 1978.

[3] F.J.MacWilliams and N.J.A.Sloane, "Pseudo-Random Sequences and Arrays", *Proc. IEEE*, Vol.64, pp.1715-1729, 1976.

[4] X.Ye and S.Fujimura, "Measurement of 3-D Object by Using M-Sequence Grid Projection", *Trans. Soc. Instrum. Control Eng.*, Vol.25, pp.644-650, 1989.

[5] S.Kiyasu, H.Hoshino, K.Yano and S.Fujimura, "Measurement of the 3-D Shape of Specular Polyhedrons Using an M-Array Coded Light Source", *IEEE Trans. Instrumentation and Measurement*, Vol.44, pp.775-778, 1995.

Cooperative Behavior of Interacting Robots

Ken Sugawara and Masaki Sano
Research Institute of Electrical Communication
Tohoku University
Sendai,980-77 Japan

Abstract

This paper studies the relation between the number of robots and the efficiency of group through the task to gather pucks in the field. In this study, we assumed a robot which has a very simple structure to clarify the behavior and the efficiency of group. The robot had a driving system, a light and some sensors. They could have interactions by using their lights and sensors. The effectiveness of group behavior was studied under various (homogeneous, localized) puck distributions. To evaluate the efficiency of group behavior, we examined the relation between the job time and the number of robots, and the relation between the interaction period and the efficiency of group. We found that a simple interaction between robots improves efficiency of group considerably compared with independent individuals.

1 Introduction

Insects such as ants and bees establish flexible and well-organized societies. It seems that each individual does simple work, but their society is very complex and has subtle functionality [1] [2] . This is because they have an interaction with each other and functions are self-organized in their society through cooperative behavior. It suggests that we may be able to synthesize complex systems just like the ant societies by coordinating interactions between elements even if the function of each element is simple. From this point of view, some experimental researches have been made by using a group of autonomous robots [3] [4] [5] . It seems that the robot systems which were used in these studies were on a large scale. The purpose of our study is to investigate the behavior and the efficiency of active elements which have the simple architecture when they act as a group. So we studied the behavior of the elements with the simplest structure at first. In this study, we assume a mobile robot as an active element. The robot has a driving system, a light and some sensors and operates autonomously [6] . To evaluate the efficiency of these active elements, we chose a task to collect pucks that were dispersed in a limited field.

2 Experiment

We made some robots with simple architecture and made experiments to collect pucks.

2.1 Materials and Methods

The shape of robot that was used in this experiment is circular and its diameter is 12cm. It is driven by a pair of DC motors. This robot has two fixed arms (Figure 1). There is a mechanical switch at the tip of each arm. When the switch is turned on, the motor on the opposite side rotates reversely: If the left switch touches something, the right motor rotates reversely. As a result, robot can avoid colliding boundary walls and other robots. There is a micro switch between the arms, which detects the puck. The robot has a pair of light sensors. These sensors are used to lead the robot to the center of the field. Magnet switches for detecting the center of the field are placed at the bottom. A light switch placed on the top is a power switch for main circuit of robot. We can initiate the robots maneuvering simultaneously by turning on/off the room light. The battery for the robot is attached by magnets. The battery can be changed easily without disturbing its direction.

The robot has two modes: search mode and return mode. Search mode is to move straight until it meets boundary walls or other robots. When the puck sensor catches a puck, the robot is in return mode. Return mode is to move toward the light at the center of the field using their light sensors. When it arrives at the center, the robot goes back and turns and changes its mode to search mode.

The field for this experiment was 190x190cm. Boundary has a wall. Light and magnet are placed

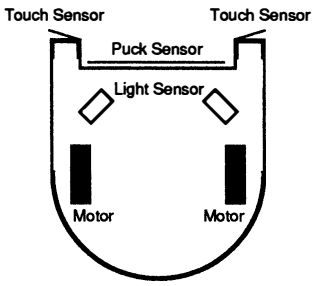


Figure 1: Schematic drawing of the robot. It has touch sensors on its arms to avoid obstacles, a puck sensor between arms and a magnet sensor at the bottom.

near the center for robots to find out the center of the field. We will call the center of the field "home." Pucks were placed homogeneously in the field. Total number of pucks was 32. The size of puck was 2.5cm in diameter and 5cm in height. Each robot could go back to home by the light, so the experiments were made in the dark room.

2.2 Result

Figure 2 shows the time evolution of collected pucks. Each plots are the average of three trials. From this figure, we know that the completion times of the task by one, two and three robots were 874 sec, 502 sec and 353 sec, respectively.

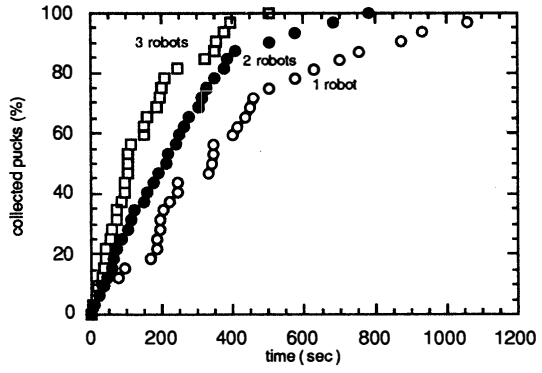


Figure 2: Time evolution of collected pucks for experiments involving one to three robots.

3 Simulation

To confirm the behavior of robots in various situations, we simulated their movements on computer. We assumed three types of distributions of pucks: homogeneous, localized in 25% area of the field and localized in 1% area of the field (Figure 3).

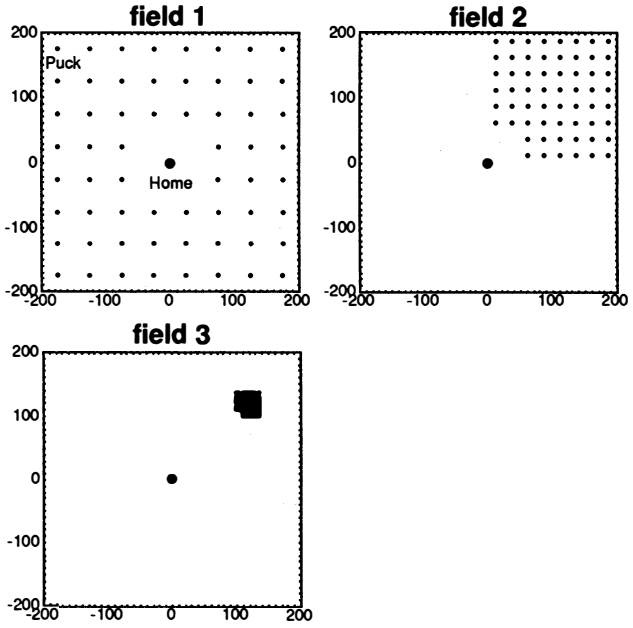


Figure 3: The fields which were used in computer simulation. field 1: homogeneous, field 2: localized in 25% area of the field, field 3: localized in 1% area of the field.

We also assumed simple interaction between each robot. In this study, interaction is expressed as follows. The robot which meets a puck stays there and turns on its light for a fixed period. We call the period as "interaction period." Other robots which have no puck react to this light and change their direction to the light. In this case, we can control the strength of the interaction by changing its interaction period.

Figure 4 shows the relation between the number of robots and completion time of tasks. In this figure, time means calculated step and 1 step corresponds to 0.1 sec. Each data point represents the average of 20 trials. In this study we defined the completion time such that 90% of all pucks were collected to home. Figure 4(a) shows the results for robots without interaction and Figure 4(b) shows the case with interaction.

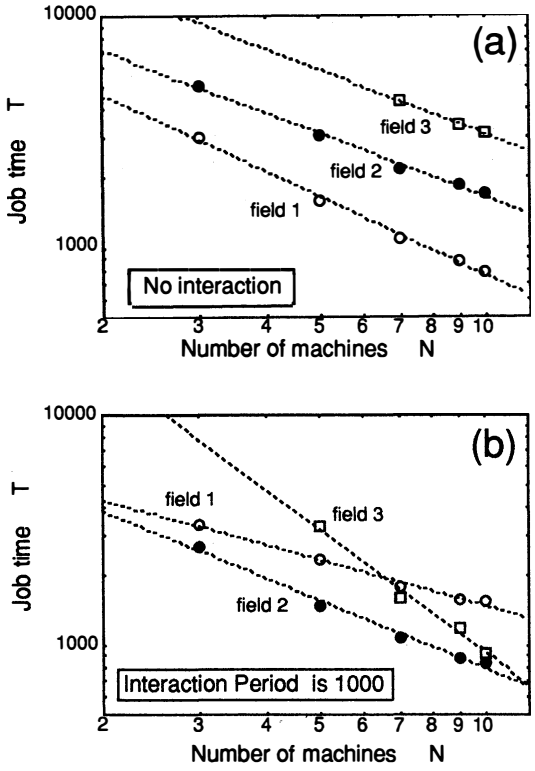


Figure 4: Relation between the number of machines and the completion time. (a):The results for robots without interaction. (b):The results for the case with interaction.

Note that the completion time T and the number of robots N have power law relation in Figure 4.

$$T \sim N^{\beta}$$

The relation between interaction period and exponent β is shown in Figure 5. β -1(dotted line in figure 5) means that the completion time and number of robots are inversely proportional which is expected if each robot work independently. If the exponent β is less than -1, it implies that the group is more effective for the task.

4 Discussion and Conclusion

We showed the relation between the number of robots and efficiency of group through the task to gather pucks in the field. If there is no interaction between each robot, their working ability is in proportion to a number of robots. But if there are interactions,

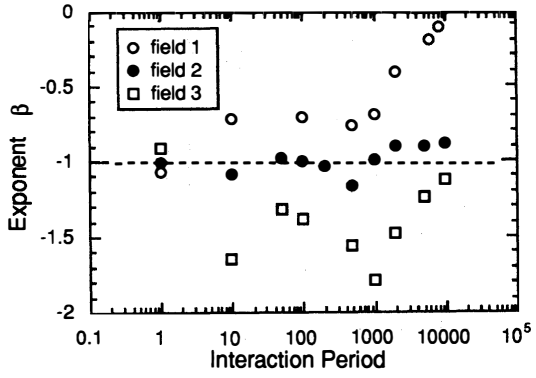


Figure 5: The relation between interaction period and exponent of the power law. In this figure, exponent -1 means that the completion time and the number of machines are inversely proportional. The case that the exponent is less than -1 implies the group is more effective for the task.

their efficiency as a group depends on the situation of field. In case of the field which is extremely localized, robots can improve their working ability by because of their interactions. We evaluated the efficiency of group quantitatively. Power law exponents for job time as a function of number of robots changes from -1 to -1.8 when the optimal interaction period is chosen. It implies that a simple interaction can improve the efficiency of the whole society drastically as the number of elements is increased.

In this paper, we mainly studied relation between the number of robots and efficiency of group for the task to gather pucks in the field. The interaction period and the efficiency of group is shown in figure 5. If there is no interaction, each robot collect pucks solitarily. As a result, the group is slightly effective in field1 (homogeneous) compared with in field3 (localized). And its completion time is inversely proportional to the number of robots. In other words, (completion time)*(number of robots) = constant. On the other hand, if there is an interaction, the efficiency of group depends on both the situation of field and interaction time. In case of homogeneous field, the longer the interaction period is, the worse the efficiency of group is. When pucks are localized, interaction period is more effective. However too long interaction period lowers the effectiveness. It is because the interaction period prevents robots to carry pucks. Therefore there exists optimal values for interaction period. Making balance between interacting and independent period

is essential to synthesize efficient cooperative behavior.

Acknowledgements

We would like to thank Y. Sawada, Y. Hayakawa and T. Mizuguchi for valuable comments and discussions.

References

[1] R.A. Brooks, "A Robust Layered Control System For A Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp. 14-23, 1986.

[2] R.A. Brooks, "Intelligence Without Representation", *Artificial Intelligence*, Vol. 47, pp. 139-160, 1991.

[3] R. Beckers, O.E. Holland and J.L. Deneubourg, "From Local Actions to Global Tasks: Stigmergy and Collective Robotics", *Artificial Life IV*, edited by R.A. Brooks and P. Maes, pp. 181-189, The MIT Press, 1994.

[4] M.J. Mataric, Ph.D Thesis, MIT, 1994.

[5] Z.D. Wang, E. Nakano and T. Matsukawa, "Cooperating Multiple Bhavior-Based Robots for Object Manipulation", *Proc. 1994 Int. Conf. on Distributed Autonomous Robotics Systems(DARS'94)*, pp. 371-382, 1994.

[6] V. Braitenberg, *Vehicles: Experiments in synthetic psychology*, Friedr, 1987.

Classification of Landsat TM Data Using Neural Network Approaches

Yoshiyuki Kawata
Environmental Information Research Lab.
Kanazawa Institute of Technology
Ogiga-oka 7-1, Nonoichi-machi
Ishikawa, 921 Japan

Yumi Yashima
Pasco Corporation
Akasaka 7-10-20
Minato-Ku, Tokyo,
03 Japan

Abstract

Although statistical classification algorithms are the most commonly used in remote sensing, the neaural network approaches to classify multispectral remote sensing images has been investigated recently (Benediktsson et al., 1990). In this study we have examined several types of input signatures in a 3 layer neural network model, using Landsat TM (Thematic Mapper) data with six spectral channels. We found that the classification accuracy of Landsat TM image by neural network using spectral data as input signatures is as good as that by the statistical maximum likelihood method. Furthermore, a significant improvement by about 10 % in classification accuracy was obtained by the neural network using coocurrence matrix components as input signatures, when compared with the accuacy by the maximum likelihood classifier.

1 Introduction

Six out of TM's seven spectral channels are used as the input signatures, because the ground resolution in channel 6 is different from those in other six channels. The total number of land-cover classes was assumed to be thirteen, and they are listed in Table I. The study site is Kanazawa area in Japan, where is characterized by a blend of aqueous, urban, rural, and ecological land-cover features associated with rugged terrains. Ground cover conditions in this area were identified by a joint work of the Geographical Survey Institute of Japan and the Ishikawa Research Laboratory for Public Health and Environment, with the aid of the land-use map, aerial photographs, and actual examination of the ground sites. These ground truth data were used for choosing class samples as the training data, as well as for the verification data of the classification accuracy. We selected 32 ground control points within the study site image to compute the geometric transformation coefficients by using a least square method and the registration error was thus achieved to be less than 1 pixel. As for the resampling process of the geometrically corrected image, the nearest neibouring method was adopted.

2 Neural Network Classification

In this study, we assumed a three layer neural network model, consisting of 6 , 15 and 13 elements in the input layer, the hidden layer and output layer, respectively. Six elements in the input layer can accept the corresponding TM's 6 spectral channel signatures. Thirteen elements in the out put layer correspond to 13 land-cover classes. The Sigmoid function was chosen for a semilinear activation function. The appropriate slope value of Sigmoid function was assumed to be 0.8 and the appropriate number of elements in the hidden layer was 15. These values were found from a view point of the classification accuracy and computational efficiency by the analysis of learning process experiments using the training data. Nine different types of training signatures for the input layer were considered here: 1) Gray level signature of a randomly sampled pixel , 2) Modes , 3) Mean (μ) , 4) Median , 5) a set of Mode, μ , and Median, 6) a set of μ, $\mu-\sigma/2$, and $\mu+\sigma/2$, 7) a set of μ, $\mu-\sigma/3$, and $\mu+\sigma/3$, and 8) a set of μ, $\mu-2\sigma/3$, and $\mu+2\sigma/3$. We denote a standard deviation as σ. We made classification experiments for such 9 input signature types using the above

neural network model. The average classification accuracies for nine input types are listed in Table II, together with the accuracy by the conventional maximum likelihood method. We found that the best classification accuracy is obtained when the set of μ, $\mu-2\sigma/3$, and $\mu+2\sigma/3$ is used as input signatures. The accuracy in this case is about 53 %, which is better by about 5 % than in the case of the maximum likelihood method. The calssification maps based on the maximum likelihood classifier and the neural network using the input set of μ, $\mu-2\sigma/3$, and $\mu+2\sigma/3$ are shown in Fig. 1 and 2.

3 Inclusion of Spatial Information

We also considered to use the spatial information on the image for the input signatures in the neural network model. The spatial gray level distribution of the image can be represented by computing components of a normalized co-occurence matrix C with a size of n x n where n is a number of gray levels. Each component of this matrix corresponds to a probability , P_δ (i,j), that points with gray level i occur relative to points with gray level j, with a displacement of $\delta = (r, \theta)$, and (i, j = 0,1,.., n-1). The schematic diagram of this displacement is shown in Fig.3. In this study a moving mask of 5x5 pixels size was used to compute P_δ (i,j). It is also necessary to requantized original gray levels into a few levels in order to keep the size of C manageable. The experiments of the neural net classification into 5 major land cover classes, namely, aqueous, urban, rice field, forest and others were performed for the gray levels in the range of 16, 32, 64, and 128. We found that the best classification accuracy is obtained in the case of 32 gray levels. Assuming a simple three layer neural network shown in Fig.4, the classification accuracy was found to be about 78.3 % by using co-occurence matrix elements as the input signatures, much better than the results (about 68.4 %) by either the neural network using the input set of μ, $\mu-2\sigma/3$, and $\mu+2\sigma/3$, or the maximum likelihood method (see Table III). We should also note that Kushardono et al. (1994) obtained an even better classification accuracy of 84 % in the neural network using a similar co-occurence matrix approach for a different study site. The resulting classification map based on the co-occurence matrix elements is shown in Fig.5. The corresponding classification map by the maximum likelihood classifier is shown in Fig.6. Although many of misclassified pixels are seen in the forest areas in Fig.6, they are correctly classified in Fig.5. We notice that many fine structures in Fig.6 are lost in Fig.5 due to masking operations in computing co-occurence matrices.

4 Conclusions

This study came to the conclusions as follows:

1) We found that the classification accuracy of Landsat TM image by neural network using spectral data as input signatures is as good as that by the statistical maximum likelihood method.

2) Furthermore, a significant improvement by about 10 % in classification accuracy was obtained by the neural network using coocurrence matrix components as input signatures, when compared with the accuacy by the maximum likelihood classifier. This is because the spatial information can be taken into accout by introducing the co-occurence matrix in the neural network.

Reference:

1. J.A. Benediktsson, O.H. Swain, and O.K.Ersoy, "Neural Network Approarches Versus Statistical Methods in Classification of Multisource Remote Sensing Data", IEEE Trans. on GRS, vol.28, no.4, pp.540-551, 1990.

2. D. Kushardono, K. Fukue, H. Shimoda, and T. Sakata, "Characteristics of Neural Network in the Spatial Classification based on Co-occurence Matrix", Proc. of Japan Society of Photogrammetry Annual Meeting, pp.115-118, 1994.

Table I. Category Classes

1. Highly densed urban area	8. Meadow area
2. Urban area	9. Deciduous forest
3. Residential area	10. Mixed forest
4. Highway	11. Coniferous forest
5. Sand area	12. River
6. Rice field	13. Sea
7. Fields	

Table II. Average Classification Accuracy.

Input Signature	Average Accuracy(%)
1) Random sampling	34.9
2) Mode	48.8
3) μ	51.4
4) Median	51.6
5) Mode,μ, Median	53.2
6) μ, $\mu-\sigma/2$, $\mu+\sigma/2$	53.2
7) μ, $\mu-\sigma/3$, $\mu+\sigma/3$	51.8
8) μ, $\mu-2\sigma/3$, $\mu+2\sigma/3$	53.5
Max. Likelihood	48.0

Table III. Average Classification Accuracy in 5 classes (%)

co-occurence matrix	78.3
Max. Likelihood	68.4

Fig. 1. The classification map by the maximum likelihood classifier in the case of 13 classes.

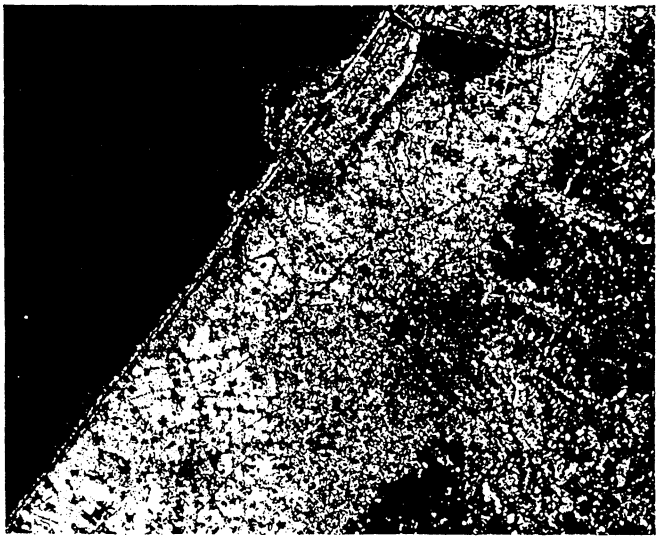


Fig. 2. The same as in Fig.1, except it is made by the neural neywork with a set of μ, $\mu-2\sigma/3$, $\mu+2\sigma/3$.

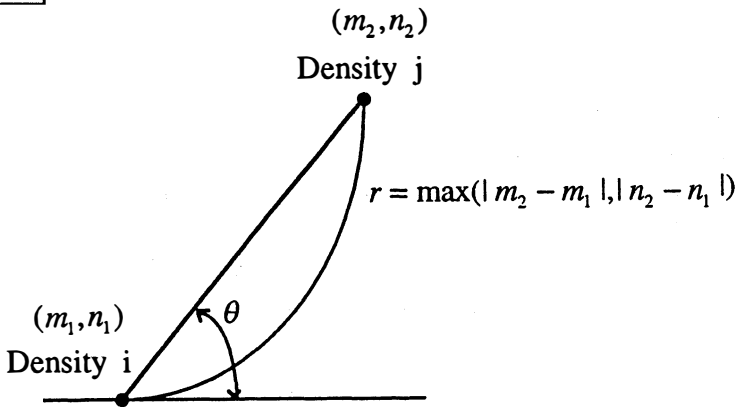


Fig. 3. The schematic diagram of a displacement, $\delta = (r, \theta)$.

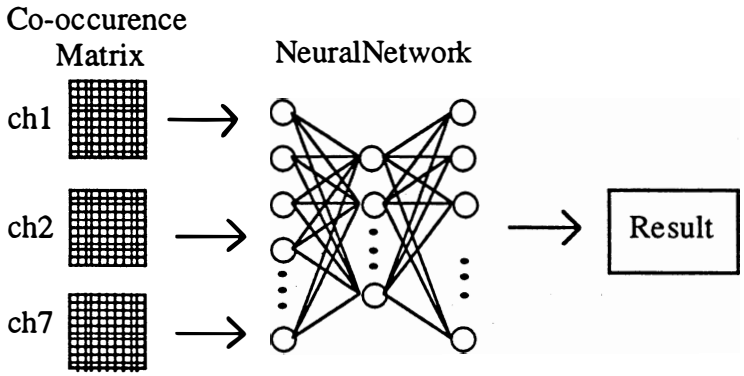


Fig.4. The neural network model using co-occurence matrix.

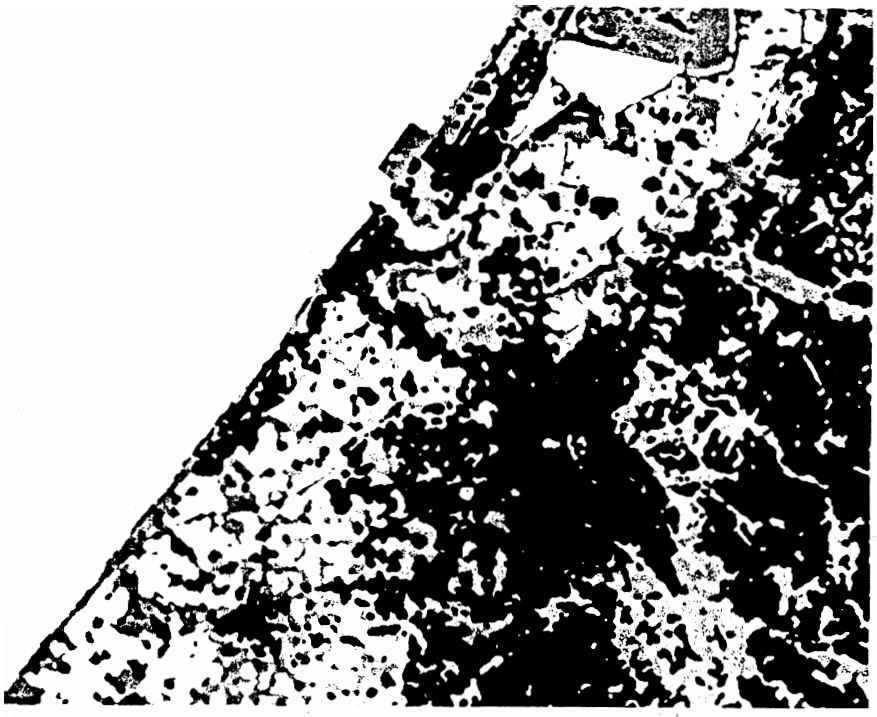


Fig. 5. The classification map by the neural network using co-occurence matrix in the case of 5 classes.

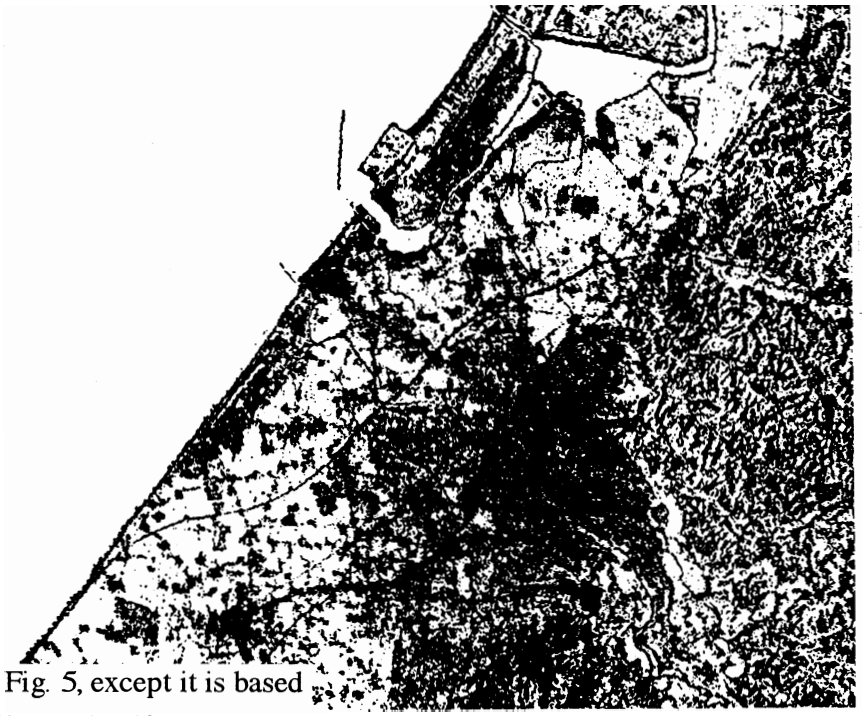


Fig. 6. The same as in Fig. 5, except it is based on the maximum likelihood classifier.

A Control Design of Robotics Using the Genetic Algorithm

T. Tsuji T. Hachino
Dept. of Electrical Eng.
Kyushu Institute of Tech.
1-1 Sensui-cho, Tobata-ku
Kitakyushu, 804 Japan

R. Oguro N. Umeda
Research Laboratory
Yaskawa Electric Co.
2346 Fujita, Yahatanishi-ku
Kitakyushu, 806 Japan

H. Takata
Dept. of Electri. and Electro. Eng.
Kagoshima University
1-21-40 Korimoto
Kagoshima, 890 Japan

Abstract

This paper presents a new and practical method for a control design of a robotic system. In general, actuators in robotic systems are set with gears whose characteristics are elastic. Since a state feedback type digital controller is usually used for such a robotic system, the design of the feedback gain of the controller is important, because undesirable vibration or overshoot in responses occur for high gains. So that desirable response, an output of a reference model, is designed initially and the feedback gains are determined so as for the response to coincide this desirable response which is an optimization problem. The gradient method works to some extent, but it takes long time to get a satisfactory result. Thus we apply the genetic algorithm(GA) for this nonlinear optimization problem, and as a result, the GA gives very first convergence. The obtained gains are useful for applications. Results of simulation are given.

1 Introduction

A control of a robot arm with a flexible joint is investigated in some papers for its undesirable vibration of the arm [1]. Consider a two link robot with reduction gears, a two mass system with an elastic joint. Then for the application purpose, we have the 10 dimensional system and obtain a nonlinear optimization problem for optimal gain determination. Since now we assume that an observer can detect all states required, a state feedback controller can surely prevent this undesirable vibration. However, although there are many methods to determine feedback gains such as an optimal regulator or a pole placement method, it is another problem to determine a desirable response. That is, if we have so many gains that we can not predict the response for each combination of values of gains. In this paper, the performance index introduced in the next section is seemed to be unusual and requested from the field. Minimizing this performance index by the feedback coefficient is an optimization problem. In this problem, the gradient method did not work well, while the GA is found to show better convergence. Some simulation results are shown to give the appropriateness of our proposal.

2 Problem Formulation

Consider a two link robot with reduction gears which are elastic joints, then we have

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{1}$$

where the contents of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are given in Appendix, and

$$\boldsymbol{x} = \Big[\int \theta_{ml} dt \;\; \theta_{ml} \;\; \dot{\theta}_{ml} \;\; \theta_{sl} \;\; \dot{\theta}_{sl} \;\; \int \theta_{mu} dt \;\; \theta_{mu} \;\; \dot{\theta}_{mu} \;\; \theta_{su} \;\; \dot{\theta}_{su} \Big]^T$$

$$\boldsymbol{u} = \begin{bmatrix} u_l & u_u \end{bmatrix}^T \qquad \boldsymbol{y} = \begin{bmatrix} \theta_{ll} & \theta_{lu} \end{bmatrix}^T$$

where θ_{ml} and θ_{mu} are the angular positions of the motor of the lower and upper links respectively. θ_{sl} and θ_{su} are the torsional angles in the gears at the lower and upper motors. The outputs θ_{ll} and θ_{lu} are the angles of the lower and upper links respectively and satisfy the following equations:

$$\theta_{sl} = N^{-1}(\theta_{ml} - \theta_{ll}) \;, \quad \theta_{su} = N^{-1}(\theta_{mu} - \theta_{lu}) \tag{2}$$

where θ_m is the output of the reference model commonly for θ_{ml} and θ_{ll} given by the follows:

$$\frac{d}{dt}\begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\omega_n \end{bmatrix}\begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}\theta_{mref} \tag{3}$$

where θ_{mref} is the input for the model.
From the field, the following performance index is considered.

$$J = h_1 \parallel \theta_m - \theta_{ll} \parallel + h_2 \parallel \theta_m - \theta_{lu} \parallel + h_3 \parallel \dot{\theta}_{sl} \parallel / \mid \dot{\theta}_{sl} \mid_{max} + h_4 \parallel \dot{\theta}_{su} \parallel / \mid \dot{\theta}_{su} \mid_{max} \tag{4}$$

where in general $\parallel a(t) \parallel$ means :

$$\parallel a(t) \parallel = \sqrt{\sum_{k=1}^{n} a^2(t_k)}$$

for suitable sampling points t_k and the interval count n, and h_1, h_2, h_3, and h_4 is weighting factors.

The problem is to determine the feedback gains, K_{vl}, K_{pl}, K_{il}, K_{sl}, K_{sDl}, K_{vu}, K_{pu}, K_{iu}, K_{su}, and K_{sDu} which optimize J, where the control is given by

$$u = -K_g x + K_p v \tag{5}$$

where

$$K_g = \begin{bmatrix} K_{vl}K_{pl}K_{il} & K_{vl}(K_{pl}+K_{il}) & K_{vl} & K_{sl} & K_{sDl} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{vu}K_{pu}K_{iu} & K_{vu}(K_{pu}+K_{iu}) \\ & & & & & 0 & 0 \\ & & & & & K_{vu} & K_{su} \end{bmatrix} \quad \begin{matrix} 0 \\ K_{sDu} \end{matrix}$$

$$K_p = \begin{bmatrix} K_{vl}K_{pl}K_{il} & K_{vl}K_{pl} \\ K_{vu}K_{pu}K_{iu} & K_{vu}K_{pu} \end{bmatrix}$$

$$v = \begin{bmatrix} \int \theta_{mlref} dt \\ \theta_{mlref} \end{bmatrix}$$

Moreover, we claim that $K_{il} = 0.5K_{pl}$ and $K_{iu} = 0.5K_{pu}$ from the field.

3 Optimization by the GA

In what follows, we write $K_1 \sim K_8$ instead of K_{vl}, K_{pl}, K_{sl}, K_{sDl}, K_{vu}, K_{pu}, K_{su}, and K_{sDu} respectively for convenience. There are several methods to execute the GA. It is difficult to say which method is the best. The uniform resolution may be one of the important factors. The probabilities of the crossover and mutation are also thought to be the important factors. Here, we set the necessary values, functions and probabilities, by trial and error and empirically. Initially, a gene is constructed by allocating a r_i bits string for each gain as follows:

$$\overbrace{\underbrace{1001\cdots1}_{K_1}\ \underbrace{0100\cdots0}_{K_2}\ \cdots\cdots\ \underbrace{0100\cdots0}_{K_8}}^{\sum_{i=1}^{8} r_i \text{ bits}}$$

The GA is executed as folows:

step 1: Initial population

Generate an initial population of M genes randomly.

step 2: Decoding

Decode the gene by:

$$K_i = \frac{K_{imax} - K_{imin}}{2^{r_i} - 1} \mathcal{K}_i + K_{imin} \tag{6}$$

where $[K_{imin}, K_{imax}]$ is the preset range of K_i and $\mathcal{K}_i$ is the decimal value of the binary representation of K_i.

step 3: Fitness value calculation

Since we have M sets of gains ($K_1 \sim K_8$), we can compute x(t) by solving (1) substituted by (5), so that we have M values of the performance indices denoted by $J_i (i = 1, 2, \cdots, M)$.

Here let the fitness values

$$F_i = 1/J_i \quad (i = 1, 2, \cdots M).$$

step 4: Reproduction

The reproduction is implemented as linear search through roulette wheel slots weighted in proportion to the fitness value of the individual gene. Practically, the linear fitness scaling mentioned below is utilized in order to avoid the undesirable premature convergence. Let the minimum, average and maximum values of $F_i (i = 1, 2, \cdots M)$ to be F_{min}, F_{avg} and F_{max} respectively. Then the scaled fitness values $F_i' \ (i = 1, 2, \cdots, M)$ are calculated as follows:

1) In case of $F_{min} + F_{max} - 2F_{avg} \geq 0$

$$F_i' = \frac{F_{avg}}{F_{max} - F_{avg}} (F_i + F_{max} - 2F_{avg}) \quad (i = 1, 2, \cdots M) \tag{7}$$

2) In case of $F_{min} + F_{max} - 2F_{avg} < 0$

$$F_i' = \frac{F_{avg}}{F_{avg} - F_{min}} (F_i - F_{min}) \quad (i = 1, 2, \cdots M) \tag{8}$$

The reproduction is carried out by using this scaled fitness values $F_i' (i = 1, 2, \cdots M)$. Namely each individual gene is reproduced with the probability of

$$F_i' / \sum_{j=1}^{M} F_j' \quad (i = 1, 2, \cdots, M).$$

step 5: Crossover

Pick up two genes randomly and decide whether or not to cross them over according to the crossover probability P_c. If the crossover is required, exchange strings in the genes at a crossing position. The crossing position is chosen randomly. P_c is usually chosen greater than 50%.

step 6: Mutation

Alter each bit (0 or 1) of strings in the genes according to the mutation probability P_m. P_m is generally designed to be quite low, for example less than few percent.

step 7: Repetition

step 2 $\sim$ **step 6** are repeated from generation to generation so that the fitness value of the population increases until the termination citerion is satisfied. The elitest preserving strategy which guarantees that the individual gene having the best fitness value survives in the nest generation is introduced in this algorithm.

Finally at the termination of this algorithm the control parameters are determined by the gene with the best fitness value over all the past generations.

4 Illustrative example

For the reference model (3), we put $\omega_n = \sqrt{1600}$, and weighting factors in the performance index (4) are chosen to be $h_1 = h_2 = 1, h_3 = h_4 = 0.05$.

The design parameters of the GA are given as fillows:

(1) population size $M = 30$

(2) bit length of each gene $l = 120$

(3) crossover probability $P_c = 0.8$

(4) mutation probability $P_m = 0.025$

The algorithm is terminated when ε becomes less than 0.04 in our case where

$$\varepsilon = \frac{\| \theta_m - \theta_{ll} \| + \| \theta_m - \theta_{lu} \|}{2 \| \theta_{mref} \|} . \qquad (9)$$

The detail about the other parameters are shown in Table 1. The results are shown in Table 2.

Fig.1 shows the step responses of the lower arm position of the reference model(dotted line) and the control model(solid line) for the best gains in the initial generation. Fig.2 shows the same responses but after convergence of the GA. Fig.3 shows the change of the best fitness value for each generation. From this figure, it seems that there are some local extrema, although it is difficult to confirm this.

5 Conclusions

In this paper a control design method of two link robot manipulator using the GA has been presented. The GA shows at least quicker convergence to a better solution than the gradient method does (about 1/16).

Table 1. The search ranges and r_i of the control parameters

link	control parameter	search range	r_i [bit]
l	K_{vl}	$[0, 300]$	10
	K_{pl}	$[0, 100]$	10
	K_{il}	$0.5K_{pl}$	–
	K_{sl}	$[10^3, 10^6]$	25
	K_{sDl}	$\pm[10^2, 10^4]$	15
u	K_{vu}	$[0, 300]$	10
	K_{pu}	$[0, 100]$	10
	K_{iu}	$0.5K_{pu}$	–
	K_{su}	$[10^3, 10^6]$	25
	K_{sDu}	$\pm[10^2, 10^4]$	15

Table 2. Determination result of the control parameters

link	control parameter	determination result
l	K_{vl}	1.8182×10^2
	K_{pl}	1.8768×10^1
	K_{il}	9.3840
	K_{sl}	7.0110×10^5
	K_{sDl}	-9.9323×10^3
u	K_{vu}	1.3812×10^2
	K_{pu}	1.7595×10^1
	K_{iu}	8.7975
	K_{su}	9.5370×10^5
	K_{sDu}	-5.4052×10^2

References

[1] M. Matsuda, T. Murakami, K. Ohnishi, "Vibration Suppression and Disturbance Rejection Control of a Flexible Link Arm", *IECON'95*, pp1260-1265, (1995).

[2] J.E.Lansberry, L.Wozniak, and D.E.Goldberg, "Optimal Hydrogenerator Governor Tuning with a Genetic Algorithm", *Trans. on Energy Conversion*, Vol.7, No.4, pp623-630 (1992)

[3] B.Porter and A.H. Jones, "Genetic Tuning of Digital PID Controllers", *Electronics Letters*, Vol.28, No.9, pp843-844 (1992)

Appendix

$$A = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & -1.5 & -1.8070 \cdot 10^{5} & -1.7037 \cdot 10^{2} \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & -9.8039 \cdot 10^{-3} & -5.2308 \cdot 10^{3} & -4.9318 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.8673 \cdot 10^{-13} & 1.7605 \cdot 10^{-16} \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.8673 \cdot 10^{-13} & 1.4286 \cdot 10^{-16} \\ 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & -1.5 & -1.8070 \cdot 10^{5} & -1.3824 \cdot 10^{2} \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & -9.8039 \cdot 10^{-3} & -7.3318 \cdot 10^{3} & -5.6091 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1.0 & 0 \\ 0 & 0 \\ 6.5359 \cdot 10^{-3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1.0 \\ 0 & 0 \\ 0 & 6.5359 \cdot 10^{-3} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & -N & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -N & 0 \end{bmatrix}$$

$(N = 160)$

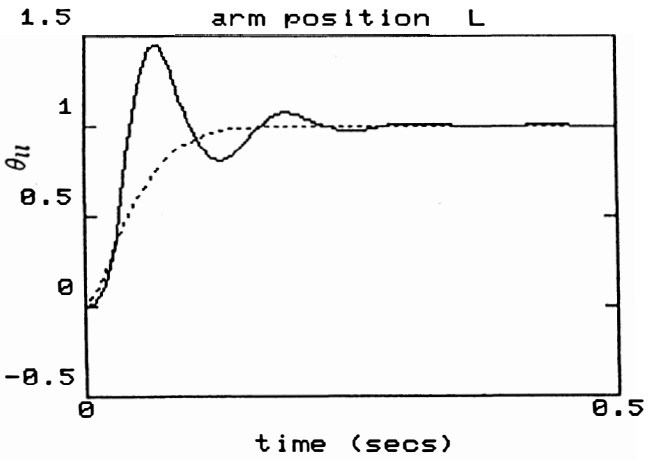


Fig.1 Step reponses of the lower arm position of the reference model(dotted line) and the controlled model(solid line) for gains in the best gene of the intial generation.

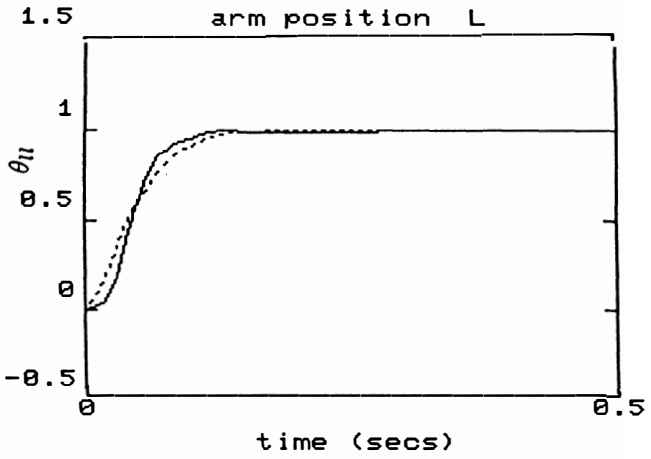


Fig.2 Step response as the same as Fig.1 but after convergence of the GA.

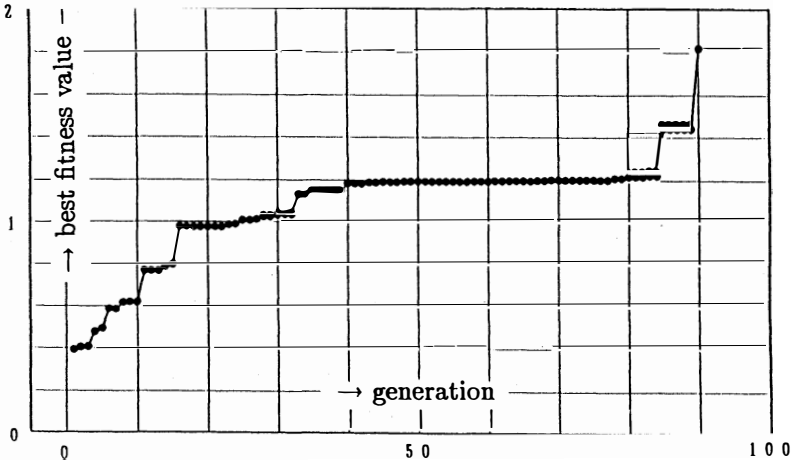


Fig.3 Maximum fitness values for each generation.

Development of Practical Control Method For an Intelligent Mobile Vehicle

M. Sugisaka[1], X. Wang[1] and J. J. Lee[2]

[1]Oita University, Oita 870-11 Japan

[2]Korea Advanced Institute of Science and Technology, Taejon 305-701 Korea

Abstract

In this paper recent research into operating an intelligent mobile vehicle(MV), developed in Prof. Sugisaka's laboratory at Oita University Japan, is presented. A complex control procedure, having a two-input and two-output fuzzy controller as kernel, is used. The inputs to the fuzzy controller are provided by a CCD camera which transforms information on spacial objects by using an image processor. The MV is able to track specified guideline types, search for object in space and recognize a stimulant traffic signal. Results of laboratory testing are presented.

1. Introduction

The results of latest research on a mobile vehicle (MV), eveloped in Prof. Sugisaka's research laborotoary [1], are presented. Although a number of control strategies may be applied to operate a MV, achieving good control effect remains a difficult problem.

In this work a fuzzy controller, consisting of two inputs and two outputs, is the kernel of the MV control system responsible for steering and driving the vehicle. The fuzzy logic inference rules are of the IF~THEN form with twenty-eight rules used in the experimental studies. The inputs to the fuzzy controller are provided by a visual controller, which obtains data from a CCD video camera. The outputs are used to operate the driving speed and steering angle of the MV. The visual controller is able to recognize objects and this acts as the eye of the MV. If necessary, another controller, called the search controller, may be activated thus enabling the CCD video camera, which can rotate freely both horizontally and vertically, to search for an object or identify a traffic signal.

Many factors are considered in the overall design, including the sensitivity and area of search of the visual sensor, the harmonious relationship between driving and steering, the nonlinearities of the performing unit, the friction and inertia.

2. Structure of Mobile Vehicle

The MV has four-coupled structure with dimensions of 1.5m×0.7m×0.5m and weights approximately 150kg. Its two rear wheels can rotate frontwards and backwards about the axle shaft and are driven by a 24V DC motor using cross helical gears. Besides free forwards and backwards rotations without power, the MV's front wheels are driven by a 24V DC stepping motor for motion to the left or right, with transmission being via gear wheels. Two 6V DC motors are employed to rotate the the CCD camera in the horizontal and vertical directions. The location of the front wheels and camera are detected independently by three encoders.

3. Control System

The MV control system is shown in Figure 1 and comprises a fuzzy controller, a visual controller and a searching controller.

3.1 Fuzzy controller

A good introduction to fuzzy control is given in references [2] and [3]. The five fuzzy sets NB, NM, ZE, PM and PB are used to span the universes of discourse of each of the two inputs, Edx={edx}, Edy={edy}, and the four fuzzy sets S, M, MB, B are used to span the universes of discourse of each of the two outputs, Del={del}, Spe={spe}. The membership functions are shown in Figure 2. The linguistic rules are implemented as the twenty-eight sets of fuzzy conditional statements of the form:

R1: If edx is NB and edy is S
then del is NB and spe is S;
ELSE R2: If edx is NM and edy is B
then del is NB and spe is M;
....
ELSE R15: If edx is ZE and edy is M
then del is ZE and spe is B;
....
ELSE R28: If edx is PB and edy is S
then del is PB and spe is M.

where

P=positive,N=negative, ZE=zero,
B=big, M=medium, S=small;
edx, edy is the center of gravity of the
guideline type after image processing;
del, is the steering angle and
spe is the driving speed.

It should be noted that the shape of the spacial object will continuously change following image processing [4]. The role of the fuzzy controller is therefore to identify these changes and make the correct decision on both the driving speed and angle of steering.

3.2. Visual controller

By means of the camera rotating upwards the MV is capable of recognizing a traffic signal located at a height of more than 1.5m. On seeing a red signal the MV will stop and remain stationary until the light disappears. The camera checks every seven seconds and if there is no change within 28 seconds the total system will be stopped.

3.3 Searching controller

The searching controller is used to rotate the CCD camera either horizontlly or vertically thus expanding significantly its field of vision.

4. Experimential Results

The outputs of the fuzzy controller are shown in Figure 3. The results of laboratory tests are shown in Figure 4 and are divided into two parts: Fistly, the MV runs along the guideline type AMC (MD cancealed). At point B, the searching controller may be initiated to rotate the camera to the left enabling the MV to run along BC. Secondly, the MV runs along the guideline type CMD (MA cancealed). At points S and P, searching controller is likely to be initiated to rotate the camera to the right thus enabling the MV to run along MPD. Throughout each of these runs if the MV identifies a red traffic signal it stops. If will then wait for up to 28 seconds, checking the signal every 7 seconds. If the signal disappears within this time then the MV will continue its run, otherwise the complete system is stopped.

The maximum speed of the MV is about 2.0m/s.

5. Conclusions

A complex controller incorporating a fuzzy controller as kernel has been successfully applied to enable a mobile vehicle to follow specfied guideline type runs. Work is continuing on improving the capability of the camera/image processing stage in order to account for problems such as luminosity changes and objects of similar colour. Both neurocomputers and artificial life technologies [5] will also be applied to the MV control system to make the mobile vehicle recognize and track moving objects and avoid obstacles using advanced intelligence.

References

[1] M. Sugisaka, et al., Running Test of a Mobile Vehicle, *Proceeding of First Asian Control Conference*, Vol.1, Tokyo, 439/442, 1994.

[2] Y.S. Li, J. Li, *Theory and Application of Fuzzy Control*, Industry Press, B.J. 1992.

[3] M. Madan, K. Rommohan , R. Ronald , *Advances in Fuzzy Set Theory and Applications*, North-Holland Publishing Company, 1990.

[4] Y. Anzai, *Pattern Recognition and Machine Learning*, Academic Press, 1992.

[5] D. Wasserman, *Neural Computing Theory and Practice*, Van Nostrand Reinold Press, 1989.

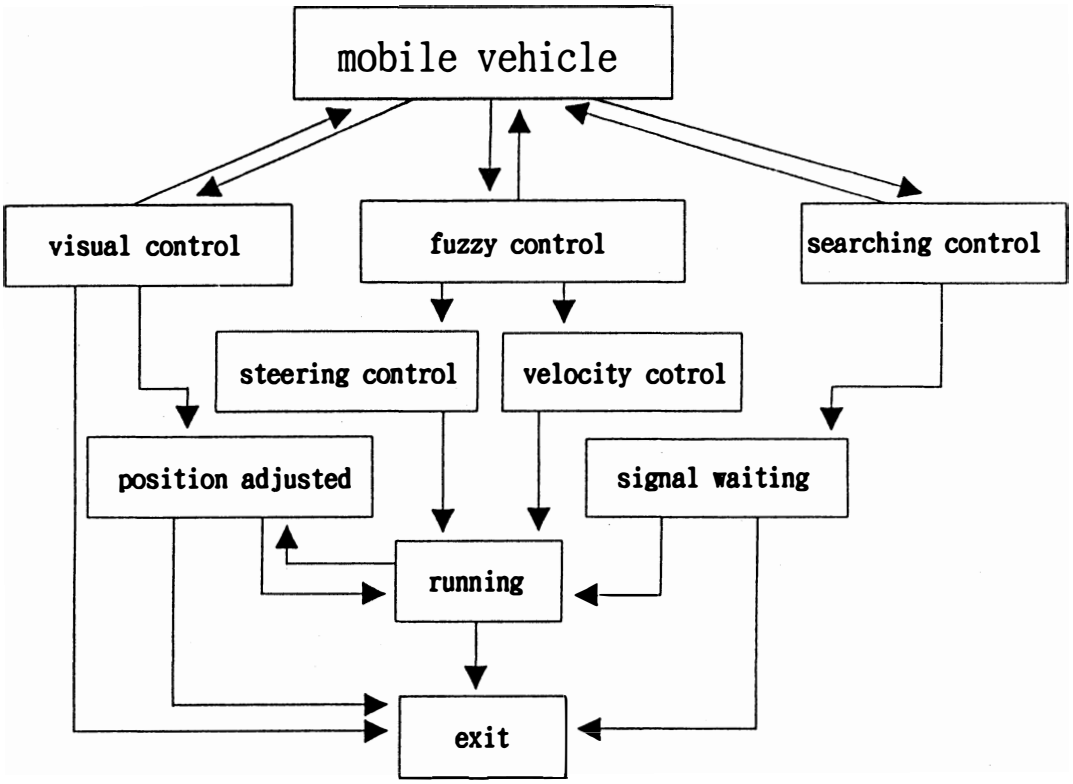


Fig. 1 The block diagram of MV control system

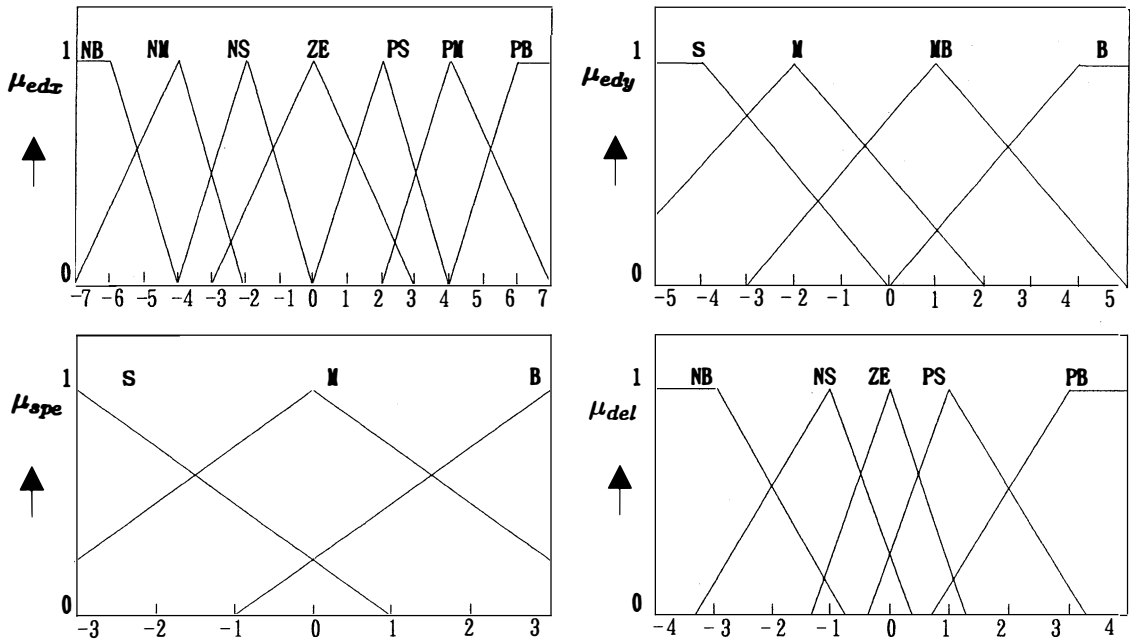


Fig. 2 The membership functions of edx, edy, spe, del

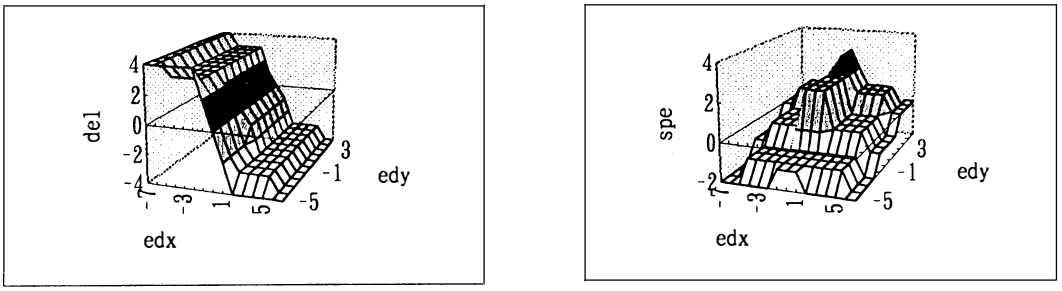


Fig. 3 The outputs of the fuzzy controller

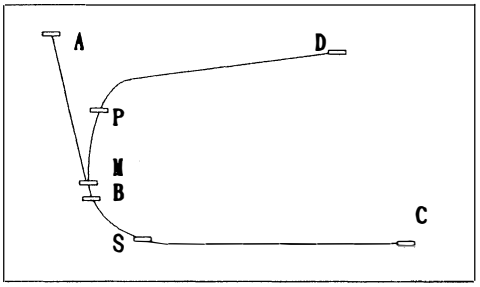


Fig. 4 The running ways of MV

Fuzzy-Logic Approach to Industrial Control Problems

D J G James
Control Theory & Applications Centre
Coventry University
COVENTRY CV1 5FB

K J Burnham
Control Theory & Applications Centre
Coventry University
COVENTRY CV1 5FB

Abstract

Increasing demands for improved profitability and product quality, together with a growing awareness of the effects of pollution on the environment, is forcing manufacturers to closely examine their process operations. As a consequence there is currently significant research and development activity aimed at improving control system strategies. In recent years, there has been renewed interest in fuzzy rule-based control strategies and this paper considers some potential industrial applications of such strategies.

1 Introduction

Manually tuned three-term proportional-integral-derivative (PID) control strategies are widely used in practice and provide satisfactory performance for many applications. The disadvantage of such a controller is that it is required to be re-tuned/de-tuned for each new operating point/range and schemes for overcoming this shortfall have included auto-tuning and linear self-tuning strategies. However, when applied to industrial systems exhibiting nonlinear characteristics such schemes often prove to be inadequate and frequently underperform the basic PID controller. In recognition of this research is undertaken, in the Control Theory and Applications Centre at Coventry University, into the design of nonlinear adaptive control strategies. Particular emphasis being placed on bilinear model based self-tuning control [6] [2] and fuzzy rule-based control [5] [3] [8] strategies.

In the area of bilinear self-tuning control, where the system model is able to capture and replicate input dependent plant dynamics and steady-state gain, nonlinear adaptive controllers have been successfully developed for high temperature heating plant applications [6]. The bilinear systems modelling approach developed is also applicable to a wide range of engineering, biomedical and socio-economic systems, see for example [9] [1]. Whilst the bilinear STC approach has distinct advantages when applied to systems for which local bilinearisation holds, the requirement for *a priori* knowledge of model structure is its main limiting factor.

One of the main advantages of fuzzy rule-based approaches, over the nonlinear model based approaches, is the apparent immunity of the approach to lack of *a priori* knowledge of model structure. Rather the equivalent model is embedded within the knowledge base of the fuzzy controller. It therefore does not suffer due to lack of precise structural knowledge of the system nonlinearities and as a consequence allows for more flexibility in its design. A good introduction to fuzzy control is found in [10].

2 Industrial applications

In this section two industrial applications are considered, which reflect the role of fuzzy methodologies within both control algorithms and supervisory/decision making schemes. In the control case the application is that of speed control in engine testing dynamometers and in the decision making case a fuzzy methodology is used to optimize the start up of a combined cycle power generation plant.

2.1 Engine testing dynamometer

An engine testing dynamometer is a torque absorbing device which is coupled directly to the drive shaft of the engine under test. In this particular case consideration is given to the hydraulic dynamometer which, due to its wide range of applicability, is extensively used in practice. Essentially such a dynamometer operates on the flow of water through its working compartments with the output flow being the controlled variable. It exhibits severe nonlinear behaviour in its speed/torque characteristics which makes it not amenable for effective implementation of model based adaptive control methodologies. Current practice is to

use conventional PID controllers and, since such controllers are tuned to provide a compromise across the full operating range of the system, there is scope for designing improved controllers provided means can be found to accommodate the nonlinear characteristics within the control algorithm. Recognising the limitations of the model based methodologies the effectiveness of fuzzy controllers has been investigated [4].

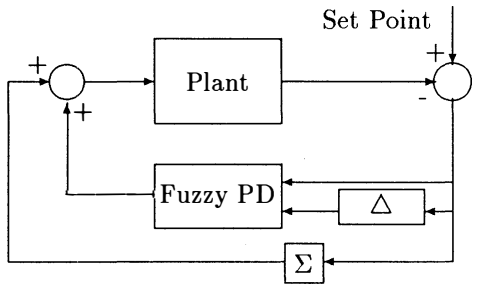


Figure 1: A fuzzy PID controller

Initially a controller, incorporating a fuzzy PD component and a conventional integrator term, which emulates a well-tuned PID controller is developed; this is shown schematically in Figure 1. The product form of the t-norm and the centre of gravity approach to defuzzification are adopted and triangular membership functions chosen for the two linguistic variables error $e(t)$ and change of error $\triangle e(t)$. The resulting fuzzy controller surface is shown in Figure 2. Whilst in simulation studies this replicates a conventional PID controller, in practice computational delays in the fuzzy approach lead to a reduced performance. However, the fuzzy controller provides greater flexibility than the conventional PID controller in that the rule-base can be adjusted to accommodate an element of scheduling of the gains of the proportional and derivative terms. For example, recognising the need to increase the gains at the extremities and to lower the gains in the well controlled regions, the rule-base can be adjusted to give the control surface of Figure 3. When implemented on the dynamometer this updated fuzzy controller is superior to the conventional PID controller in some aspects, particularly in the steady-state mode, but computational delays remain a problem in the transient phase. For systems, such as the hydraulic dynamometer, which exhibit fast dynamics necessitating the need for rapid sampling rate, computational aspects of fuzzy control can be a problem. Alternative fuzzification strategies are being investigated as a possible way of overcoming this problem.

An alternative to the fuzzy PID approach is to adopt a fuzzy model-based approach. In the case of the

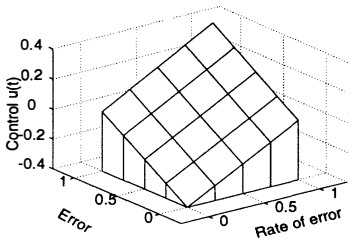


Figure 2: A linear PID control surface

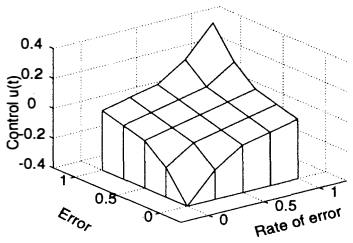


Figure 3: A non linear fuzzy control surface

hydraulic dynamometer behaviour is essentially first order within local operating regions, where it can be modelled by equations of the form

$$y(t) = -ay(t-1) + bu(t-1)$$

where $y(t)$ and $u(t)$ represent the speed and valve position respectively. Thus, the first step is to identify such linear models throughout the full operating range of the dynamometer; this initially being done off-line to produce a nominal model surface. This nominal model surface is then updated using fuzzy logic, with $u(t-1)$ and $y(t-1)$ as the antecedent linguistic variables and incorporating equally spaced triangular membership functions coupled with the product form of the t-norm. A typical resulting model surface is shown in Figure 4. In order to accommodate the variable dead time associated with the dynamometer the fuzzy PID controller, previously developed, is combined within a Smith predictor scheme as shown in Figure 5 which incorporates the fuzzy model surface. Initial testing on a laboratory based fluid level system, having similar characteristics to the dynamometer, have been very encouraging and indicate potential improvement over conventional model based self-tuning strategies when the estimated delay is realistic. Work is continuing on improving the controller through on-line tracking of the dead time.

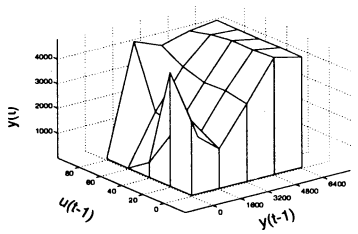


Figure 4: A typical model surface

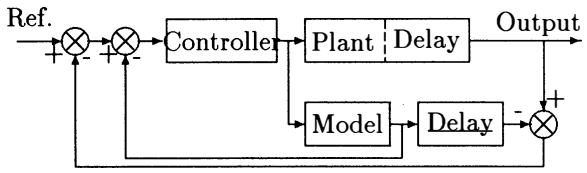


Figure 5: A Smith predictor control scheme

2.2 Combined cycle power generation plant

In order to improve thermal and economic efficiency combined cycle power generation plant (CCP) make use of a combination of both gas and steam turbines. Such plants operate by utilising the waste heat from the gas turbine to form steam, through heat exchange in a heat recovery steam generator. The steam may then be used in a steam turbine. Full details of CCP theory may be found in [7]. The increased efficiency of CCP leads to important design and operation considerations, in particular plant start up. At present start up regimes are calculated prior to commissioning and hence may not be optimal. The start up regimes essentially consist of the speed rates (ie. the ramp rates at which to increase the speed of the rotor to its rated speed), and load rates, (ie. the rates at which load may be ramped onto the plant until full rated output is reached).

One of the most important considerations is the level of thermal stress experienced by those sections of the plant which are subjected to high temperature variations, particularly during start up and shut down. Whilst the thermal stress problem exists in all steam plant special care needs to be taken with CCP. This is due to the difference in timescales with the start up regimes of gas and steam turbines; the former may well take less than fifteen minutes to achieve full load, whilst the latter may well require two hours or more in the same situation. As a result, full steam conditions may be experienced by the steam turbine in a shorter period of time than in conventional plant. It is in recognition of such problems, coupled with the need for a more sophisticated approach, that prompted the research into the design of a supervisory fuzzy logic decision making controller (FSC).

As thermal stresses are unavoidable maximum limits are imposed for each start up and shut down to ensure that the plant design life is met. Thermal stress equations are nonlinear and the important variables include steam temperature, steam pressure and/or their rates of change, heat transfer coefficient between steam and rotor surface and the dimensions of the rotor. The aim of the supervisory controller is to complement the existing scheme by manipulating the load rate set points so as to minimise start up times whilst adhering to (ie. not violating) thermal stress constraints. In order to achieve this the supervisory strategy makes use of the combined heuristic features of model based prediction. A supervisory load rate controller for a CCP has to incorporate several distinct features. Whilst it is considered to be important to minimise start up time, it is imperative that the maximum thermal stress limits are respected otherwise severe detrimental effects on plant life expectancy may result. Other requirements are that the load may only be increased or held at the same level throughout the loading phase and the load rate may only be switched between a set of discrete predefined rates. The present start up regimes are open loop in nature, with respect to thermal stress, and involve switching between the predefined load rates at given outputs of the plant.

In the proposed scheme the inputs to the supervisory controller are thermal stress and its rate of change giving as output a 'condition value' [11] which triggers a decision on the load rate. The condition value takes a value on the interval $[-1, 1]$ and will lie in one of three decision making regions, (i) $[T_1, 1]$, (ii) $[T_2, T_1]$ and (iii) $[-1, T_2]$; where T_1 and T_2 are thresholds with $1 > T_1 > 0$ and $0 > T_2 > -1$. These decision intervals correspond to regions where the supervisory load rate controller will (i) increase the load rate to the next in the sequence, (ii) keep the load rate the same and (iii) decrease the load rate to the next in the sequence. Two other features that are included to reduce the effect of noise are (i) load rates are only altered if the condition value lies in a decision region for m consecutive samples and (ii) the fuzzy logic strategy should aim to achieve a maximum stress level j percent below the maximum stress limit; m, j, T_1 and T_2 being user defined parameters. A schematic representation of the controller operation is illustrated in Figure 6.

Due to the complexity of the problem the feasibil-

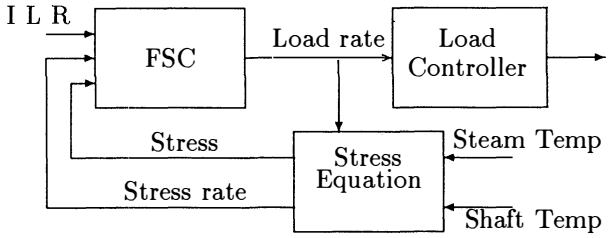


Figure 6: Supervisory control strategy

ity of adopting a fuzzy approach has been investigated. Thermal stress and its rate of change are taken as the antecedent linguistic variables with values being determined from a thermal stress equation which relates temperature to stress. The output or consequence determine the condition value which trigger decisions on load rate changes. Initial results are again encouraging and indicate the potential for significant improvement in terms of reduced start up time [13]. Further developments include adaptation of the threshold values with load and use of radial basis function neural networks to classify the different start up load rates [12].

2.3 Conclusions

The paper considers two applications where fuzzy methodologies may be advantageous in comparison to model based strategies for control and decision making as industrial plant. The results to date are encouraging indicating the potential for implementation of fuzzy methodologies in practice. In systems exhibiting fast dynamics computational delay remains a problem for further investigation. The work on fuzzy modelling is at an early stage of development and work is ongoing to link the control and model surfaces coupled with alternative control strategies to the error driven PID.

Acknowledgements

The authors gratefully acknowledge the work of their research students John Dunlop and Chris Smith who carried out much of the work described in this paper.

References

[1] C Bruni, G Di-Pillo, G Koch, 'Bilinear systems: an appealing class of 'nearly linear' systems in theory and applications', IEEE Trans Aut Control, 19, 334-348, 1974.

[2] K J Burnham, K J Disdell, D J G James, C A Smith, 'Developments in industrial applications of self-tuning', J Control Engineering Practice, 3, 9, 1265-1276, 1995.

[3] J A Dunlop, Intelligent Fuzzy Controller for Engine Testing Systems, PhD Thesis, Coventry University, Coventry, 1995.

[4] J A Dunlop, K J Burnham, D J G James, P J King, 'A self-tuning scaling method for fuzzy control', Proc 3rd IEEE Conf on Control Applications, 683-687, Strathclyde, Glasgow, 1994.

[5] J A Dunlop, K J Burnham, D J G James, P J King, 'Application of fuzzy logic based self-tuning control, Proc IEEE/IAS Int Conf on Industrial Automation and Control, 109-114, Hyderabad, 1995.

[6] S G Goodhart, K J Burnham, D J G James, 'Bilinear self-tuning control of a high temperature heat treatment plant', IEE Proc D Control Theory and Applications, 141, 1, 12-18, 1994.

[7] R H Hehlhofer, Combined-cycle Gas and Steam Turbine Power Plants, Prentice-Hall, 1990.

[8] P J King, K J Burnham, D J G James, A combined rule-based and model-based adaptive control scheme', Proc IEE Control '94 1484-1489, IEE Publication 389, London, 1994.

[9] R R Mohler, Bilinear Control Process, Academic Press, New York, 1973.

[10] W Pedryz, Fuzzy Control and Fuzzy Systems (2nd Edn), Research Studies Press, Tauton, UK, 1993.

[11] C A Smith, K J Burnham, P A L Ham, D J G James, 'Accommodating thermal stress during the start up of a combined cycle plant', Proc 10th Int Conf on Systems Eng, Vol II, 1151-1158, Coventry, 1994.

[12] C A Smith, K J Burnham, P A L Ham, D J G James, 'Adaptive fuzzy control of combined cycle plant', Proc IFAC Symp Control of Power Plants, Cancun, Mexico, 1995.

[13] C A Smith, K J Burnham, D J G James, P A L Ham, 'Fuzzy and model based supervisory controller for combined cycle plant start up using thermal stress constraints', Proc 3rd European Control Conf, Vol 3, 2444-2449, Rome, 1995.

Associative Readout from a Chaos Network employing Chaotic Itinerancy

Takeshi yamakawa

Department of Control Engineering and Science
Kyushu Institute of Technology
Iizuka, Fukuoka 820, Japan
Phone : +81-948-29-7712
Fax : +81-948-29-7742
E-mail : yamakawa@ces.kyutech.ac.jp

An ordinary neural network can store information in the form of distributed connective weights and thresholds, and facilitates readout of the information in the form of training output data by applying input data similar to but different from the training input data. Although storing behavior in the training mode in the network is dynamical, the readout behavior is static. Accordingly, there cannot be searching of alternative candidates to be read out. This is quite different from information retrieval in a biological brain. The brain retrieves events one by one, by chaining a train of thought or by analogy of different fields.

A group of biological neurons connected to each other often exhibits chaotic itinerancy. This is regarded as the primary behavior of chaining retrieval of stored information in the recurrent neural network.

Interesting behavior of a chaotic network is presented here, which is constructed with chaotic units connected to each other including self-feedback. The chaotic unit is constructed with the delay block and the nonlinear block which a linear combination of a sigmoidal function and an inverted-N type nonlinear function. The combinatory coefficient (one of the nonlinear parameters) can be determined to make the chaotic unit to come up to fixed point attractor, periodic attractor and strange attractor (chaos). The connective weights in the chaotic network are assigned by Hebbian learning rule. When a distorted pattern is applied to the chaotic network, the stored pattern most closed to the distorted pattern is retrieved. When an overlapped pattern is applied, separated patterns are retrieved sequentially.

When a mixed pattern or noisy pattern is applied, all the stored patterns are retrieved sequentially. This interesting behavior cannot be obtained in the ordinary mutually connected networks, e.g. Hopfield network, which is agitated by noisy signal .

This chaining retrieval of information is inherent to the chaotic network proposed here and is possibly a model of associative readout of stored information in a human brain.

Neurcomputer Control for Tracking of Moving Objects

M. Sugisaka
Department of Electrical and Electronic Engineering
Oita University
Oita, 870-11, Japan

Abstract

We developed a new neuocomputer control technique for tracking a moving object using a neurocomputer. The control is produced by the RICOH neurocomputer RN2000, which is able to learn various control laws instantly, in order to track the moving object within a certain range of errors. This paper describes the new neurocomputer control technique and presents the results obtained from the tracking experiments.

1 Introduction

We have already developed the recognition and tracking system of moving objects[1]-[4]. The system consists of one CCD video camera, two DC motors in horizontal and vertical axes with encoders, pulse width modulation(PWM) driving unit, 16 bit NEC PC-9801 microcomputer, and their interfaces. The recognition and tracking system is able to recognize the pattern and size of a moving object and is able to track the object within a certain range of errors by learning control laws.

In the meanwhile the RICOH neurocomputer RN-2000 has been recently devised for research and development[5],[6]. The RICOH neurocomputer RN-2000 consists of seven RN-200 digital neural network VLSI chips, where sixteen neurons and totally 256 synapses are integrated in a $13.73 \times 13.73\ mm^2$ VLSI chips, fabricated by RICOH 0.8 μ CMOS technology. This chip can perform 5.12 giga pulse operations per second. It corresponds to effective neural computing rate of 40M CPS or CUPS. Utilizing the neurocomputer RN-2000, we have developed a new identification technique in order to model linear and nonlinear dynamical systems to be controlled[7],[8].

Utilizing the results obtained from the identification experiments using the neurocomputer[7],[8], we devised a new recognition and tracking system of moving objects[9] called a neurocomputer control system for recognition and tracking of moving objects. The new system consists of the 32 bit microcomputer NEC PC-9801βA, the neurocomputer RN-2000, and the other remaining hardware developed in the former one stated above. The main differences from the former system are:
(1) The 16 bit microcomputer NEC PC-9801 is replaced by the 32 bit microcomputer NEC PC-9801βA in order to use the software(Nadeshiko and Yamato) for operating the neurocomputer on the Microsoft Windows.
(2) In the former one the control laws were produced from the neural network systems programmed in the 16 bit NEC PC-9801 microcomputer so that on-line learning was not possible[8].
(3)The neurocomputer RN-2000(S specification) is newly added to the system. Therefore on-line learning of control laws and patterns become possible because the learning by the neurocomputer RN-2000 is very fast.

On February 2 in 1995, we succeeded in the experiments in order to track both the desired values(sinusoidal, circular, and elliptical movements) of encoders in X and Y axes and a light from electric lamp moved by hand using the new neurocomputer control system for recognition and tracking stated above in our laboratory. We obtained the first experimental results on the control studies for the tracking using the neurocomputer control system. This paper is a brief introduction of both the neurocomputer control system and the results obtained from the control studies.

2 Neurocomputer Control

The neurocomputer control system for recognition and tracking of moving objects devised in our Laboratory is shown in Fig.1 and Fig.2. The elaborate descriptions of the system have been given in [1]-[4]. The configuration shown in Fig.1 is different from the former system[1]-[4] in the point that the neurocomputer RN-2000 is newly connected to the microcomputer NEC PC-9801βA. The general view shown in Fig.2 is the same as before.

At the first stage, in the experiments of the neurocomputer control system simple control laws are desirable for the purpose of both checking the performance of the hardware system and validity of the results. To this end, we used the neural network structure and the training data discussed below.

2.1 Neural Network Structure

The structure of the neural network used in the neurocomputer RN-2000 is shown in Fig.3. The number of neurons in the input layer is 4, the numbers of neurons

in the first and second intermediate layers are 16 and 16, respectively, and the number of neurons in the output layer is 1. The inputs to the neural network are the errors(deviations) between the center of the CCD camera and the desired values of the encoders in X and Y axes or the highest intensity of a moving light from electric lamp by hand as shown in Fig.4. The teaching signals are the duty ratios of the PWM. The reason why we use 4 neurons in the input layer is the followings.
(1)As the inputs to the neurocomputer RN-2000, the two digits 0(equals 0) and 127(equals 1) are permitted due to the hardware specification.
(2) The maximum error is 10 in the experiments and this value is transformed into the binary number of 4 bits, namely, 4 neurons. On the other hand, the digits ranged from 0 to 127 can be used as the output from the neurocomputer so that one neuron is employed in the output layer.

2.2 Data for Learning

As the learning or training data for the neural network shown in Fig.3 in order to track a moving object, we use the data produced from a proportional control indicted by the dotted line shown in Fig.5. Both the real training data and the corresponding data used for the neural network in the neurocomputer RN-2000 are shown in Table 1. In this table the input data to the neurons are deviations or errors of e_x and e_y which are equal to 1, 2, ..., 10 and the teaching data are the duty ratios of the DC motors in both horizontal and vertical axes which are equal to 0.1, 0.2, ..., 1.0. The input data are transformed into 4 bit digits by using bit transformation technique. The teaching signals are normalized such that the maximum value 1.0 and the minimum values 0 in Table 1 correspond to 127 and 0, respectively.

As one of criteria for the learning of the neural network, we introduce the average error criterion J given by

$$J = \sum_{i=1}^{N} |O_i - T_i|/N, N = 11(patterns), \quad (1)$$

where O_i is the output from the neural network and T_i is the teaching signal. We use the neural network trained until the value of this criterion becomes less than 2.0. The results, which were obtained from the forward processing by the neural network trained, are shown in Table 1 using the same input data where the iteration number per one pattern is 2378 for the learning, total learning time is 1 minute and 7 seconds, and J=1.64. It is seen from Table 1 that the data obtained from the neural network learned are different from the teaching signals due to the hardware specification, namely, the digital neural network VLSI with on-chip learning using stochastic pulse encoding. However, the neural network learned works well for the tracking problems as shown below.

It should be noted that in the experiments for the tracking the control laws indicated by the solid lines in Fig.5 are used in order to compensate the nonlinear characteristics of two DC motors in the recognition and tracking system for improving the control performance. This will be discussed in the next Section.

3 Tracking Experiments

In the experiments for tracking, we use the neural network shown in Fig. 3 in the nerocomputer RN-2000, which is constructed by the software called Nadeshiko. The configuration of the neurocontroller using the neural network is illustrated in Fig. 6. The duty ratios for two DC motors in X(horizontal) and Y(vertical) axes are produced from one neural network sequentially. In other words, at first, the error between the desired value in X axis(denoted $X_{desired}$ in Fig. 6) and the value of encoder of DC motor in X axis(denoted X motor in Fig. 6) is processed by the neural network in order to get the duty ratio of X motor. Secondly the same processing is performed in order to get the duty ratio of Y motor.

In the whole experiments, the sampling time employed is 100 ms by taking account of hardware specifications in the various parts. The procedures stated above[1]-[4] are repeated at each sampling time for the tracking experiments. The results are given below.

3.1 Tracking for Desired Values

We performed the experiments for tracking the desired values(sinusoidal, circular, and elliptical movements) of the encoders in X and Y axes using the outputs from the neural network.

At first the neural network shown in Fig. 3 is set in the neurocomputer RN-2000 and is learned or trained as stated above by using the software Nadeshiko. Thereafter the duty ratios for the X and Y motors are produced from the neural network by using the software Yamato and the control programs written by C language developed in our laboratory. The compensated proportional control laws were used in the programs to produce the duty ratios as indicated by the solid lines in Fig. 5.

For the reason of space, we show only the results for tracking the desired values of the encoders obtained from the elliptical movement in two dimensional space given by

$$X_{desired} = 40sin(5\pi t/(180 \times 100)), \quad (2)$$

$$Y_{desired} = 30cos(5\pi t/(180 \times 100)), \quad (3)$$

where the unit of time t is ms.

The locus of X and Y coordinates in the elliptical movement is illustrated from 1 second to 25 seconds by the solid line in Fig.7(a) where the dotted line is the desired value or trajectory. The values of the encoders in X and Y axes are shown in Fig.7(b), where the dotted lines are desired values.

3.2 Tracking for a Moving Light by Hand

The procedure for tracking a moving object is shown in Table 1 in [4] and the elaborate explanations are also given. Therefore, we show briefly how to detect the position with the highest intensity of the light from a lamp moved by hand and then explain how to track it. The detection procedures are as follows.
(a) Read 6 bits image data of the light with 15×12=180 pixels, which is moved by hand, from CCD video camera.
(b) Write the image data into the memory in NEC PC-9801βA.
(c) Detect the position with the highest intensity of the moving light.
The tracking procedures are as follows.
(d) Calculate the duty ratios using the neural network, which has been trained beforehand in the neurocomputer RN-2000, for both X and Y motors sequentially.
(e) Move the system in oder to coincide the center of the CCD video camera with the position of the hightest intensity using the duty ratios calculated above.

We show one of results for tracking a moving light by hand in Fig.8 where the distance between the moving light and the CCD video camera in the system was approximately 60 cm. In this figure the loci from 8 seconds to 25 seconds are illustrated. The locus of X and Y coordinates in tracking the highest intensity of the moving light is shown by the solid line in Fig.8(a) where the dotted line is the locus of the highest intensity. The corresponding values of the encoders in X and Y axes are shown in Fig.8(b).

The results obtained from tracking both the desired values and a moving light by hand are quite satisfactory. The followings are considered as the reasons.
⋆ The neural network structure constructed in the neurocomputer RN-2000 is simple as shown in Fig.3.
⋆ The control law used for training the neural network is a simple proportional law as shown in Fig.5.
⋆ The control law used for calculating the duty ratios for both X and Y motors is simple compensated as shown in Fig.5.

The results obtained using the simple proportional control law as indicated by the solid line in Fig.5 were also satisfactory although they are not shown in this paper.

4 Summary

In this paper we presented a new neuocomputer control technique for tracking a moving object using the RICOH neurocomputer RN-2000. The results obtained from the tracking experiments revealed that the neurocontrollers of simple structures developed in the neurocomputer RN-2000 have desirable features such that they are able to track both the desired values and the moving light by hand satisfactory by using a few training data. In other words, the neurocontrollers developed in the neurocomputer RN-2000 have knowledge or brain.

The other experimental data obtained by changing both the structure and the training data for the neural network could not be shown in this paper for reason of space. They are quite interesting and will be presented in a forthcoming paper. We are now developing the applications of our hardware and software obtained from the experiments for various engineering fields.

Acknowledgments

I thank graduate student Mr. Noboru Tonoya and research group in RICOH Co., Ltd. Mrs. Syuuji Motomura, Takashi Kitaguchi, Hirotoshi Eguchi, Hiroyasu Mifune, and Toshiyuki Furuta, Yutaka Ebi for their help and valuable suggestions in order to perform this research project.

References

[1] M. Sugisaka, "Recognition and Tracking System of Moving Objects", *Patent p2020181(Japan), G05D 3100(in world)*, in Japanese, Feb., 1992.

[2] M. Sugisaka, H. Kaita, M. Hara, and K. Nakao, "Recognition and Tracking System of Moving Objects Based on Artificial Neural Network and PWM Control", *Proc. of the IFToNM-jc Int. Symp. on Theory of Machines and Mechanics*, Vol.1, pp.104-107, 1992.

[3] M. Sugisaka, "Recognition and Tracking System of Moving Objects Based on Artificial Neural Network", *Proc. of the 1992 Korean Automatic Control Conf.*, pp.573-574, 1992.

[4] M. Sugisaka, M. Hara, and N. Tonoya, "Neuro-Fuzzy Control for Recognition and Tracking System of Moving Objects", *Proc. of 3rd Int. Workshop on Advanced Motion Control*, pp.1091-1098, 1994.

[5] H. Eguchi, T. Furuta, H. Horiguchi, and S. Oteki, "Neural Network Hardware with Learning Function Utilizing Pulse-Density Modulation", *Trans. of the Institute of Electronics, Information and Communication Engineers*, Vol.J74-C-II, No.5, pp.369-376, 1991.

[6] S. Oteki, A. Hashimoto, T. Furuta, S. Motomura, T. Watanabe, D. G. Stork, and H. Eguchi, "A Digital Neural Network VLSI with On-chip Learning Using Stochastic Pulse Encoding", *Proc. of 1993 Int. Joint Conf. on Neural Networks*, pp.3039-3045, 1993.

[7] M. Sugisaka, S. Motomura, T. Kitaguchi, and H. Eguchi, "Hardware Based Neural Identification-Linear Dynamical Systems", *Proc. of Int. Workshop on Intelligent Systems and Innovative Computation-The 6th Bellman Continuum-*, pp.124-131, 1994(*Computers & Mathematics with Applications, Special Issue, to be appear* 1996).

[8] M. Sugisaka, "Hardware Based Neural Identification: Nonlinear Dynamical Systems", *The 2nd Asia-Pacific Conf. on Control and Measurement(APC CM'95)*, 23-27 June, 1995, Wuhan-Chongqing, P.R.China.

[9] M. Sugisaka, "Neurocomuter Control System for Recognition and Tracking of Moving Objects", *to be submitted for a Patent*, May, 1995.

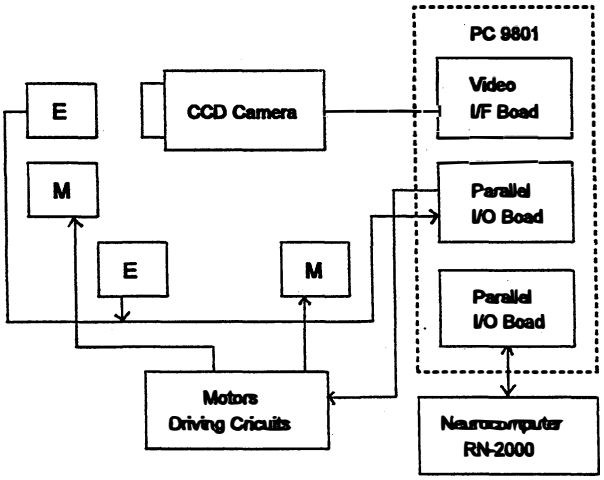


Fig.1 Neurocomputer control system

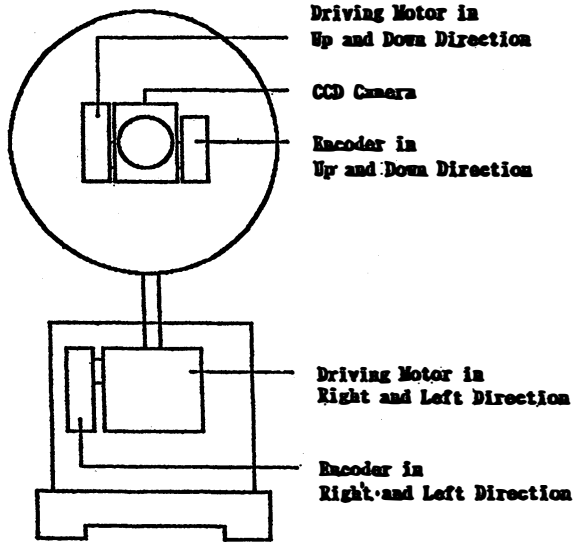


Fig.2 General view of recognition and tracking system.

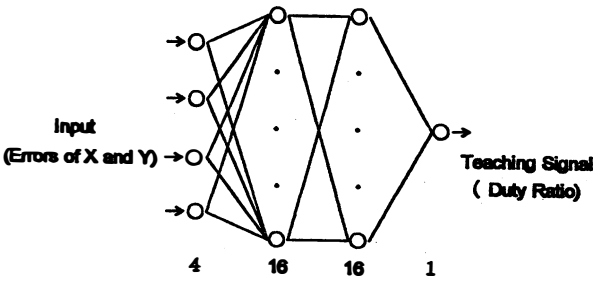


Fig.3 Structure of neural network in RN-2000.

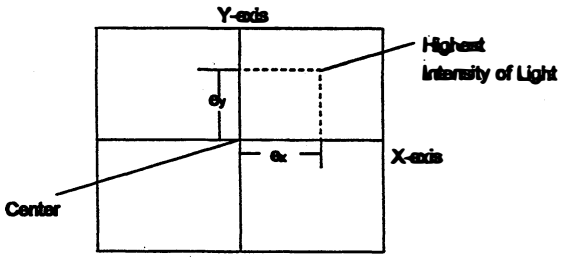


Fig.4 Errors or deviations.

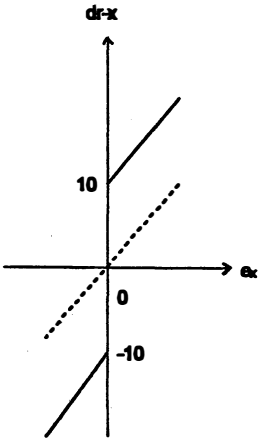


Fig.5a Nonlinear propotional control(X_{axis}).

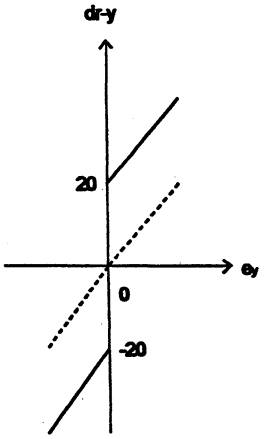


Fig.5b Nonlinear propotional control(Y_{axis}).

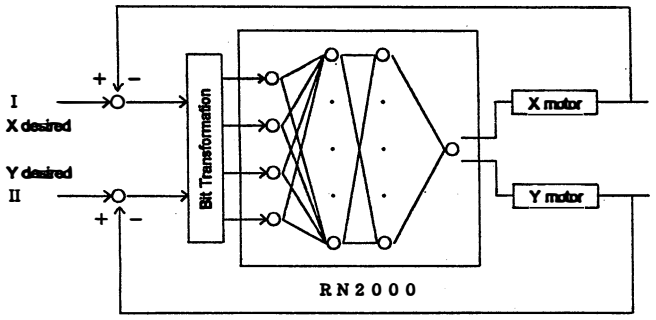
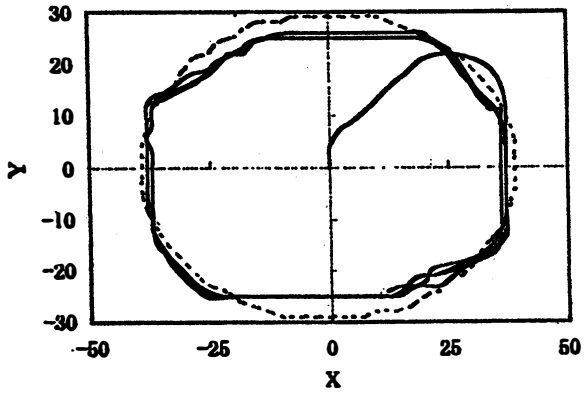


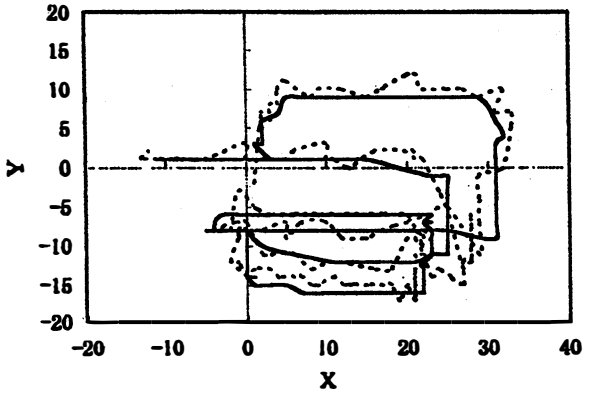
Fig.6 Configulation of neurocontroller.
(I and II show the order of processing)

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

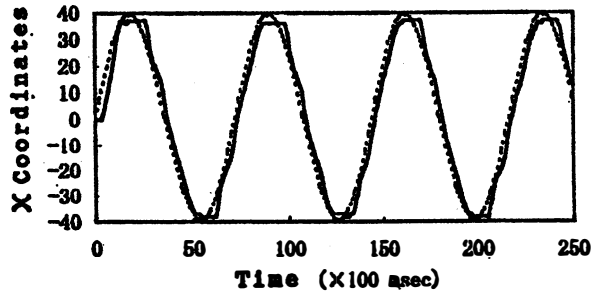
Table 1 Data for Training

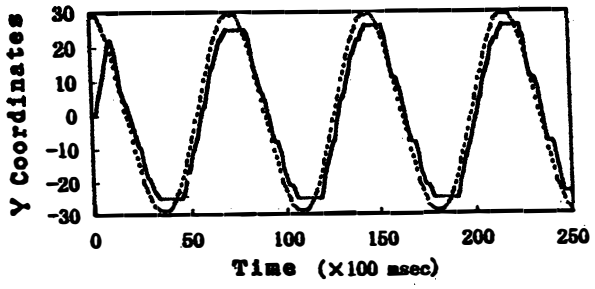


(a) Loci of desired value and X and Y coordinates.



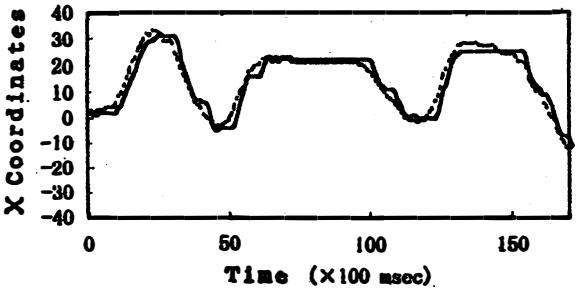
(a) Loci of light and X and Y coordinates.

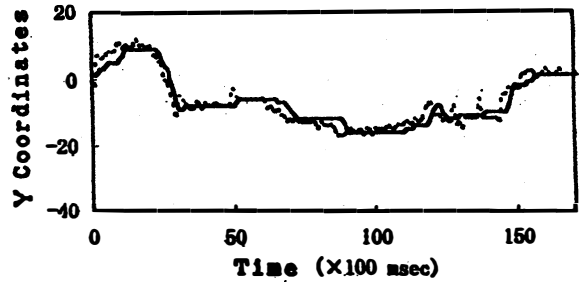




(b) Values of encoders in X and Y axes.

Fig.7 Results of tracking desired values(1-25 sec).





(b) Values of encoders in X and Y axes.

Fig.8 Results of tracking a moving light(8-25 sec).

Pattern Recognition using Neurocomputer

Masanori Sugisaka
Department of Electrical Engineering
Oita University
Oita, 870-11 Japan
e-mail msugi@cc.oita-u.ac.jp

Abstract

This paper presents a new pattern recognition system based on moment invariants using a neurocomputer. The new pattern recognition system consists of a CCD video camera, an image processing system named FDM, a monitor, two stand lights, a NEC PC-9801 microcomputer and a neurocomputer RICOH RN-2000. The experimental studies in order to recognize five dynamic patterns of Japanese chestnuts were performed. From the studies high speed of both learning and recognition has been achieved compared with the former pattern recognition system based on the software of artificial neural networks developed by us.

1. Introduction

We have already developed the fast pattern recognition method using the moment invariant computation via artificial neural networks[1]. The method was able to recognize patterns independently of rotation, translation, and size of any patterns. The pattern recognition method developed used both the moment invariants of the images obtained from CCD video camera and an artificial neural network system constructed by software in order to recognize the shapes or patterns of the images.

The system consisted of two systems, the first using the moment invariants to perform preprocessing, and the second using three-layer artificial neural network to perform dynamic pattern recognition. It was shown that the proposed method could recognize the dynamic patterns very correctly. In the system seven moment invariants of a pattern were used as the inputs in the input layer of the three layer artificial neural network with 50 units or neurons in the hidden layer and with 5 units in the output layer. The experimental studies on the learning of the dynamic patterns were performed with the image processing system.

The same system as stated above is used for the new pattern recognition system where the software of three layer artificial neural network with 50 units in the hidden layer and 5 units in the output layer is replaced by the hardware of the neurocomputer RN-2000[2]-[5]. The RICOH neurocomputer RN-2000 consists of seven RN-200 digital neural network VLSI chips, where sixteen neurons and totally 256 synapses are integrated in a $13.73 \times 13.73\ mm^2$ VLSI chips, fabricated by RICOH 0.8 μ CMOS technology. This chip can perform 5.12 giga pulse operations per second. It corresponds to effective neural computing rate of 40M CPS or CUPS.

The simulation of the proposed system is carried out in the following steps. The first step uses the image science, which is compressed in $256(16 \times 16)$ pixels by preprocessing from the original image science to the input pattern. At the second step, seven moment invariants are computed for each pattern and at the third step three moment invariants instead of seven moment invariants are used as the inputs of the artificial neural network system constructed in the neurocomputer for all samples due to the hardware specification of the neurocomputer. At the fourth step the training begins with back-propagation method by the neurocomputer. By ending the artificial neural network training in the neurocom-

puter, the outputs of the network is used to recognize the dynamic patterns. In the experiments, high speed of both learning and recognition has been achieved, compared with the former pattern recognition method based on the software, and that is the advantage of the proposed system in this paper.

2. Moment Invariants and Neural Network

The hardware of the new image processing system proposed for the patterns recognition is illustrated in Fig.1 where the difference between the new one and the former one[1] is the introduction of the neurocomputer. The moment invariants Q_1, Q_2, Q_3 are invariant under translation and rotation [6],[1]

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

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

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

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

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

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

The moment invariants above will be used as the inputs for the neural network in the neurocomputer RN-2000. In our pattern recognition system only the three moment invariants Q_1, Q_2, Q_3 are used instead of seven moment invariants due to the hardware specification in the neurocomputer. The artificial neural network system constructed in the neurocomputer is shown in Fig.2. It consists of four layers: the input layer with 15 neurons or units, the first hidden layer with 16 neurons, the second hidden layer with 16 neurons, and the output layer with 5 neurons which is equal to the number of the patterns to be recognized.

3. Simulation Study

3.1 *Data for Training*

The experimental studies on the learning of dynamic patterns are performed with the image processing system shown in Figs.1 and 2. Figure 3 shows the patterns which are used in the learning processing and gives examples of five different patterns. The three moment invariants of the five patterns are calculated for four different groups. The locations of each pattern are different in the four groups.

Table 1 shows the moment invariants calculated for the patterns in each group. The moment invariants in Table 1 are transformed into five bits digital values in order to use the neural network shown in Fig.2. Therefore, the maximum value of the normalized moment invariants becomes 31. Table 2 shows the normalized moment invariants. The teaching signals are shown in Table 3 for each pattern.

3.2 *Training Method*

It is necessary that the pattern recognition system is able to recognize the patterns exactly as possible as it can. There might exist effective methods by training the neural network in the neurocomputer. However, we used the training method as follows. The order and number of the training are:

$$G_1(\text{n times}),\ G_2(\text{n}),\ G_1(\text{n}),\ G_2(\text{n}),\ G_3(\text{n}),$$
$$G_1(\text{n}),\ G_2(\text{n}),\ G_3(\text{n}),$$

where G_i denotes group i and n is 15,000 in our experiments. The synapses weights of the neural network were accumulated in the neurocomputer according to the order stated above. The schematic configuration of the training processing is shown in Fig.4. If the number of groups is four, the same training method as shown in Fig.4 is employed. The outputs of the neural network trained by the method above are shown in Fig.5. The results obtained are considered to be satisfactory for the recognition of the five patterns.

4. Summary

In this paper we proposed a new pattern recognition system based of three moment invariants of a pattern using a neurocomputer. we obtained the first experimental results for recognizing five different dynamic patterns using the new system proposed. The experimental results confirm the effectiveness of the proposed system and show improvement of both the learning and recognition speeds, which is the main point of this paper. We are now developing the application of our hardware and software developed in this study for various engineering fields.

References

[1] M. Sugisaka and M. Teshnehlab, "Fast pattern recognition by using moment invariants computation via artificial neural networks," *Control-Theory and Advanced Technology*, **9**, 877-886, 1993.

[2] H. Eguchi, T. Furuta, H. Horiguchi and S. Oteki "Neural network hardware with learning function utilizing pulse-density modulation", *Trans. of the Institute of Electronics, Information and Communication Engineers, Vol.J74C-II*, **5**, 369-376, 1991.

[3] S. Oteki, A. Hashimoto, T. Furuta, S. Motomura, T. Watanabe, D. G. Stork and E. Eguchi, "A digital neural network VLSI with on-chip learning using stochastic pulse encoding", *Proc. of 1993 Int. Joint Con. on Neural Networks*, 3039-3045, 1993.

[4] M. Sugisaka, S. Motomura, T. Kitaguchi, and H. Eguchi, "Hardware Based Neural Identification:Linear Dynamical Systems", *Proc. of Int. Workshop on Intelligent Systems and Innovative Computation-The 6th Bellman Continuum-*, 124-131 (*Computers Math. Appli. 1996, to be appeared*), 1994

[5] M. Sugisaka, "Hardware based neural identification: nonlinear dynamical systems", *Proc. of the 2nd Asia-Pacific Conf. on Control and Measurement, Wuhan-Chongqing, P. R. China*, 255-259, 1995.

[6] B. Widrow, R. G. Winter and R. A. Baxter, "Layered neural nets for pattern recognition", *IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-36*, **7**, 1109-1118, 1998.

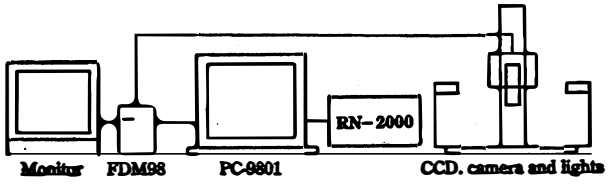


Fig.1 New pattern recognition system using neurocomputer RICOH RN-2000

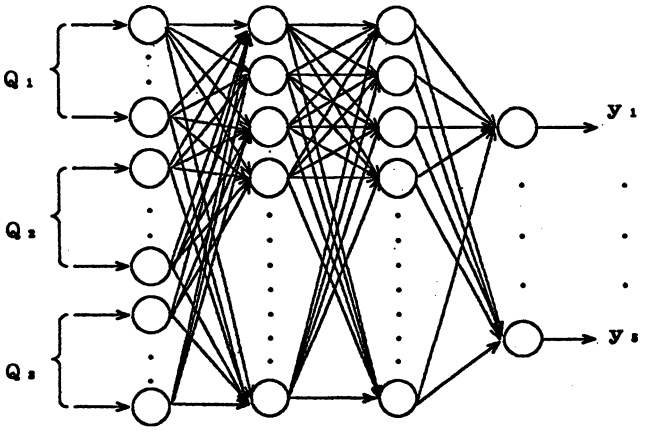


Fig.2 Neural network constructed in neurocomputer RN-2000

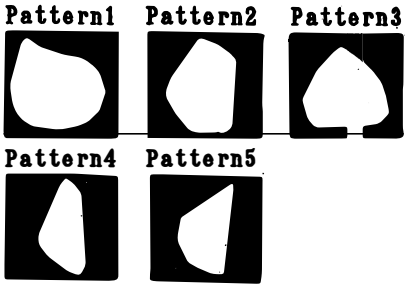
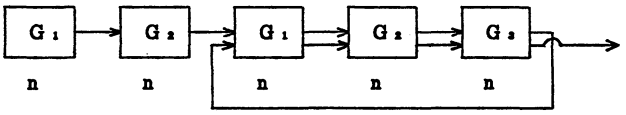


Fig.3 Examples for testing recognition of five various patterns



n=15000 times

Fig.4 Schematic configulation of training

	Pattern1	Pattern2	Pattern3	Pattern4	Pattern5	
Q_1	9.200	7.503	7.220	7.606	6.801	
Q_2	0.717	2.105	0.133	18.950	10.300	Group1
Q_3	0.470	0.154	1.403	0.562	3.025	
Q_1	8.784	7.779	7.115	6.682	6.364	
Q_2	0.204	8.418	0.090	12.380	7.590	Group2
Q_3	0.010	0.416	0.570	0.794	1.843	
Q_1	8.776	7.111	6.931	7.452	6.081	
Q_2	0.442	3.122	0.019	18.210	11.230	Group3
Q_3	0.275	0.218	0.573	1.565	1.643	
Q_1	9.011	6.913	7.484	6.819	6.969	
Q_2	0.334	2.825	0.155	9.892	9.390	Group4
Q_3	0.104	0.173	0.762	0.520	2.502	

Table 1 Moment invariants of five patterns for each group

0	0	76	0	0	0
1	127	0	0	0	0
2	0	127	0	0	0
3	2	0	126	0	0
4	0	0	0	113	0
0	0	0	0	0	0
1	126	0	0	0	0
2	0	126	0	0	0
3	2	0	126	0	0
4	0	0	0	113	0
0	0	0	0	0	0
1	127	0	0	0	0
2	0	127	0	0	0
3	2	0	126	0	0
4	0	0	0	127	0
0	0	0	0	0	0
1	126	0	0	0	0
2	0	104	0	0	0
3	2	0	126	0	0
4	0	0	0	110	0

Fig.5 Outputs of neural network trained for each group

	Pattern1	Pattern2	Pattern3	Pattern4	Pattern5	
Q_1	15	12	11	12	11	
Q_2	0	3	0	31	16	Group1
Q_3	0	0	1	0	4	
Q_1	21	19	17	16	15	
Q_2	0	21	0	31	18	Group2
Q_3	0	1	1	1	4	
Q_1	14	12	11	12	10	
Q_2	0	5	0	31	19	Group3
Q_3	0	0	0	2	2	
Q_1	28	21	23	21	21	
Q_2	0	8	0	31	19	Group4
Q_3	0	0	2	1	7	

Table 2 Normalized moment invariants of five patterns to be used for neural network

	y_1	y_2	y_3	y_4	y_5
Pattern1	0	0	0	0	0
Pattern2	127	0	0	0	0
Pattern3	0	127	0	0	0
Pattern4	0	0	127	0	0
Pattern5	0	0	0	127	0

Table 3 Teaching signals of five patterns

Rectangles, Triangles, Robots and Transients

Philip Husbands
School of Cognitive and Computing Sciences
University of Sussex
Brighton, UK

Abstract

This paper describes an experiment in real-world evolutionary robotics. A visuo-motor behaviour is evolved that allows the robot to distinguish between two different targets. The mechanisms underlying the behaviour are analysed and some surprisingly subtle features are uncovered.

1 Introduction

1.1 Evolutionary Robotics

The basic notion of Evolutionary Robotics is as follows. The evolutionary process, based on a genetic algorithm [3, (Holland 1975)], involves evaluating, over many generations, whole populations of control systems specified by artificial genotypes. These are interbred using a Darwinian scheme in which the fittest individuals are most likely to produce offspring. Fitness is measured in terms of how good a robot's behaviour is according to some evaluation criterion. The work reported here forms part of a long-term study to explore the viability of such an approach in developing interesting adaptive behaviours in visually guided autonomous robots, and, through analysis, in better understanding general mechanisms underlying the generation of such behaviours. It is one of the strands of the research program of the Evolutionary and Adaptive Systems Group, School of Cognitive and Computing Sciences, University of Sussex. For further details see e.g. ([1, Cliff et al 1993]).

1.2 Real World Evolution

A crucial decision in evolving robot control systems is whether or not to use simulation at the evaluation stage, transferring the end results to the real world. Since an evolutionary approach potentially requires the evaluation of populations of robots over many generations, a natural first thought is that simulations will speed up the process, making it more feasible. It has recently been shown that control systems evolved in carefully constructed simulations, with an appropriate treatment of noise, transfer extremely well to reality, generating almost identical behaviours in the real robot ([4, Jakobi et al 1995]. However, this example involved relatively simple robot-environment interaction dynamics. Once even low-bandwidth vision is used, simulations become altogether more problematic. They become difficult and time consuming to construct and computationally very intensive to run. Hence evolving visually guided robots in the real world becomes a more attractive option. The experiment described here uses a piece of robotic equipment specially designed to allow the real-world evolution of visually guided behaviours — the Sussex gantry-robot.

2 The Gantry-Robot

The gantry-robot is shown in Figure 1. The robot is cylindrical, some 150mm in diameter. It is suspended from the gantry-frame with stepper motors that allow translational movement in the X and Y directions, relative to a co-ordinate frame fixed to the gantry . The maximum X (and Y) speed is about 200mm/s. Such movements, together with appropriate rotation of the sensory apparatus, correspond to those which would be produced by left and right wheels. The visual sensory apparatus consists of a CCD camera pointing down at a mirror inclined at 45^o to the vertical. The mirror can be rotated about a vertical axis so that its orientation always corresponds to the direction the 'robot' is facing. The visual inputs undergo some transformations en route to the control system, described later. The hardware is designed so that these transformations are done completely externally to the processing of the control system.

The control system for the robot is run off-board on a fast personal computer, the 'Brain PC'. This computer receives any changes in visual input by inter-

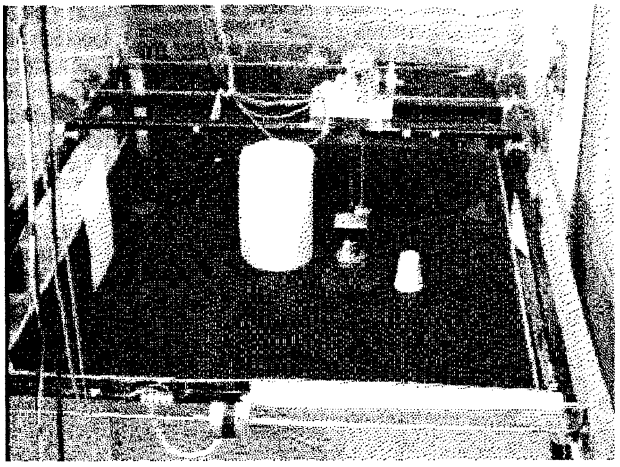


Figure 1: *The Gantry viewed from above. The horizontal girder moves along the side rails, and the robot is suspended from a platform which moves along this girder.*

rupts from a second dedicated 'Vision PC'. A third (single-board) computer, the SBC, sends interrupts to the Brain PC signalling tactile inputs resulting from the robot bumping into walls or physical obstacles. The only outputs of the control system are motor signals. These values are sent, via interrupts, to the SBC, which generates the appropriate stepper motor movements on the gantry.

The Brain PC runs the top-level genetic algorithm and during an individual evaluation, it is dedicated to running a genetically specified control system for a fixed period. At intervals during an evaluation, a signal is sent from the Brain PC to the SBC requesting the current position and orientation of the robot. These are used in keeping score according to the current fitness function. The Brain PC receives signals, to be fed into the control system, representing sensory inputs from the Vision PC and the SBC. The visual signals are derived from averaging over genetically specified circular receptive patches in the camera's field of view.

This setup, with off-board computing and avoidance of tangled umbilicals, means that the apparatus can be run continuously for long periods of time – making artificial evolution feasible. A top-level program automatically evaluates, in turn, each member of a population of control systems. A new population is produced by selective interbreeding and the cycle repeats. For full technical details of the system see ([2, Harvey et al 1994]).

3 Experimental Setup

Full details of the experimental setup for the gantry-robot can be found in ([2, Harvey et al 1994]). This paper also explains in full the genetic encodings used and the control system primitives manipulated by the GA. Experiments conducted with the gantry-robot to date have all involved relatively simple vision based navigation tasks. The experiment described below was one of a series where a converged population of robots was evolved through a series of increasingly complex behaviours.

These were based around the evolution of control architectures built from recurrent dynamic realtime networks, where the primitives were the nodes in a network, and links between them. There were no restrictions on network topologies, arbitrarily recurrent nets being allowed. When some of these nodes are connected to sensors, and some to actuators, the network acts as a control system, generating behaviours in the robot.

Rather than imposing a fixed visual sampling morphology, we believe a more powerful approach is to allow the visual morphology to evolve along with the rest of the control system. Hence we genetically specify regions of the robot's visual field to be sub-sampled, these provide the only visual inputs to the control network. It would be desirable to have many aspects of the robot's morphology under genetic control, although this is not yet technically feasible.

Starting from a converged population of robots that could move forward, but little else, the first task was to move to a large white target from random starting points and orientations. Once this was being achieved, the task was changed to approaching a small white target and evolution continued.

4 Rectangles and Triangles

The experiment then continued with a distinguish-between-two-targets task. Two white paper targets were fixed to one of the gantry walls; one was a rectangle, the other was an isosceles triangle with the same base width and height as the rectangle. The robot was started at four positions and orientations near the opposite wall such that it was not biased towards either of the two targets. The evaluation function $\mathcal{E}_3$, to be maximised, was:

$$\mathcal{E}_3 = \sum_{i=1}^{i=20} [\beta(D_{1_i} - d_{1_i}) - \sigma(D_{2_i}, d_{2_i})] \qquad (1)$$

where D_1 is the distance of target-1 (in this case the triangle) from the gantry origin; d_1 is the distance of the robot from target-1; and D_2 and d_2 are the corresponding distances for target-2 (in this case the rectangle). These are sampled at regular intervals, as before.. The value of β is $(D_1 - d_1)$ unless d_1 is less than some threshold, in which case it is $3 \times (D_1 - d_1)$. The value of σ (a penalty function) is zero unless d_2 is less than the same threshold, in which case it is $I - (D_2 - d_2)$, where I is the distance between the targets; I is more than double the threshold distance. High fitnesses are achieved for approaching the triangle but ignoring the rectangle. It was hoped that this experiment might demonstrate the efficacy of concurrently evolving the visual sampling morphology along with the control networks.

After about 15 generations of a run using as an initial population the last generation of the incremental small target experiment, fit individuals emerged capable of approaching the triangle, but not the rectangle, from each of the four widely spaced starting positions and orientations. The behaviour generated by the fittest of these control systems is shown in Figure 2. When started from many different positions and orientations near the far wall, and with the targets in different positions relative to each other, this controller repeatedly exhibited very similar behaviours to those shown.

The active part of the evolved network that generated this behaviour is shown in Figure 3. The evolved visual morphology for this control system is shown inset. Only receptive fields 1 and 2 were used by the controller.

Detailed analyses of this evolved system can be found in ([2, Harvey et al 1994]). To crudely summarise, unless there is a *difference* in the visual inputs for receptive fields 1 and 2, the robot makes rotational movements. When there is a difference it moves in a straight line. The visual sensor layout and network dynamics have evolved such that it fixates on the sloping edge of the triangle and moves towards it.

5 Transient Behaviour

Time plots of behaviour against this difference in visual inputs consistently revealed an interesting non-reactive feature to the robot's behaviour. Figure 4 shows such a plot. The behaviour axis (Z) is discretized into simple observable motor behaviours such as straight line motion, rotating on the spot, movement in the arc of a circle and so on. The final part

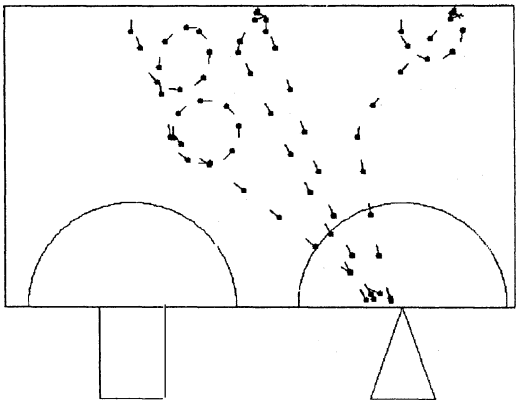


Figure 2: *Behaviour of a fit individual in the two target environment. The rectangle and triangle indicate the positions of the targets. The semi circles mark the 'penalty' (near rectangle) and 'bonus score' (near triangle) zones associated with the fitness function. In these 4 runs the robot was started directly facing each of the two target, and twice from a position midway between the two targets; once facing into the wall and once facing out.*

of the plot, a line parallel to the time axis and terminating at the point marked 'finish' at the right hand side of the cube, represents the straight line motion when the robot has fixated on the triangle edge and is moving towards it. The parallel line above this and immediately to the left represents a short lived transient behaviour which such plots revealed *always* occurred when the visual signal difference become large. Briefly, the onset of a large difference triggers a short sharp rotational movement which has very different consequences depending on whether the robot has fixated on a vertical or sloping edge. With a vertical edge, the rotation tends to move both receptors off the target, the visual signals become very different and rotational behaviours ensue. However, with a sloping edge, the rotation is not enough to move both receptive fields off the target; the visual signal difference is still there and a straight line motion follows. This is illustrated in Figure 5. This behaviour can be interpreted as a kind of 'checking' of edge orientation.

6 Conclusions

This paper has described an experiment in real-world evolutionary robotics and has shown that the

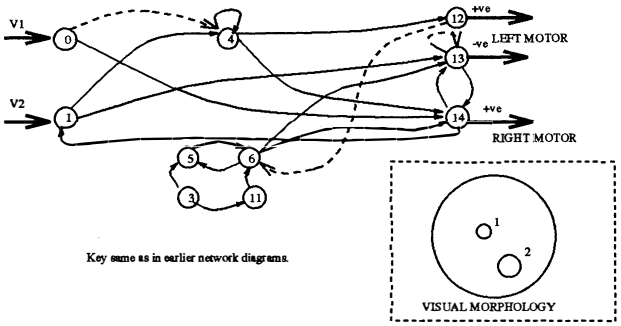


Figure 3: *Active part of the control system that generated fit behaviour for the rectangle and triangle experiment. Visual morphology shown inset.*

evolved control networks and visual morphology for a robot engaged in a simple target distinguishing behaviour display subtle dynamics resulting in non-reactive behaviour.

References

[1] Cliff D and Harvey I and Husbands P 1993 Explorations in Evolutionary Robotics *Adaptive Behavior* **2(1)** 73–110

[2] Harvey I and Husbands P and Cliff D 1994 Seeing The Light: Artificial Evolution, Real Vision *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior* eds. D Cliff and P Husbands and J-A Meyer and S Wilson (Cambridge, Mass: MIT Press/Bradford Books) pp 392–401

[3] Holland J 1975 *Adaptation in Natural and Artificial Systems* (Ann Arbor: University of Michigan Press)

[4] Jakobi N and Husbands P and Harvey I 1995 Noise and the Reality Gap: The use of simulation in evolutionary robotics *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life* eds. F Moran and A Moreno and J J Merelo and P Chacon Lecture Notes in Artificial Intelligence **929** (Berlin: Springer-Verlag) pp 704–720

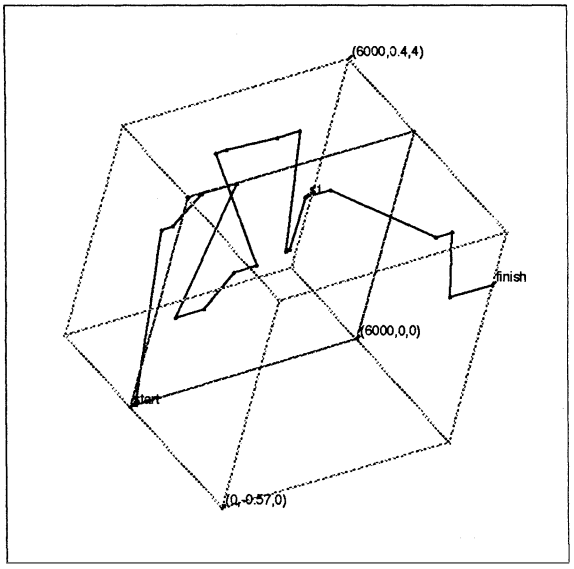


Figure 4: *Time plot of behaviour against difference in visual inputs for receptive fields 1 and 2. The time axis (X) runs left to right, the visual signal difference axis (Y) runs bottom to top on the lower face of the cube, the behaviour axis (Z) runs from lower to top face of the cube. See text for further details.*

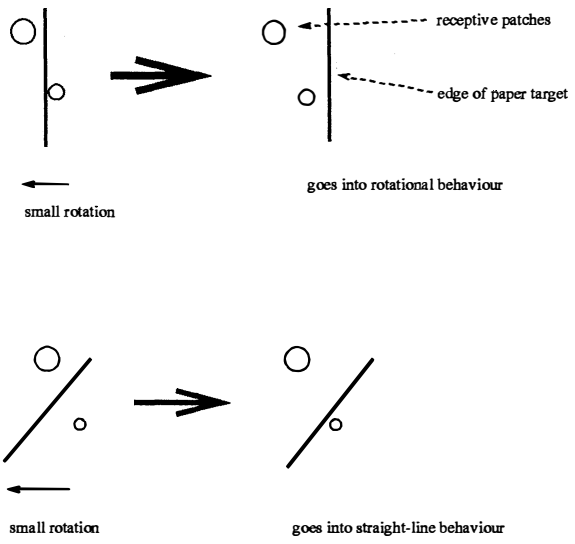


Figure 5: *The top part of the figure illustrates the outcome of the transient 'checking' behaviour when the receptive fields straddle a vertical edge, and the bottom part shows the same when they straddle a sloping edge.*

Force / Position Control of Robot Manipulator via Motion Dynamics

Kyumann Im*
Sacheul Jeoung*
* Electronic Engineering
* Chonbuk National Univ.
* Chonju, Chonbuk, Korea 560-756

Cheoljoo Ham**
Woonchul Ham*
** Electronic Engineering
** Kunjang Technical Junior College
** Kunsan, Chonbuk, Korea 573-840

Abstract

In this paper, we propose a new force/position control scheme for motion control of a rigid robot manipulator along a frictionless surface represented by a integral submanifold of some distribution. The described control scheme involves the exact cancellation of the nonlinear terms of the projected end effector dynamics in the position and force controlled directions. We introduce a new motion equation on the integral submanifold of a smooth distribution by using differential geometry. The feasibility of the proposed force control scheme is verified through a computer simulation.

1 Introduction

In this paper, we address the problem of controlling a robot manipulator which is to track a path on a frictionless surface while exerting a force (tangential and normal force) on the surface. Many manufacturing tasks such as deburring and polishing a surface require such constrained motion execution. An interesting and informative historical perspective on some past work in the area of robotic force control was presented by Whitney[1]. Considerable number of researchers have studied the problem of constrained motion control of a rigid robot manipulator along a frcitionless surface. Early researchers in this area, such as, Paul and Shimano proposed decomposing the manipulator into types of joints those which contribute to the force control and those which contribute to the motion of the end effector along the surface. Raibert and Craig[2] pursued this subdivision of tasks into position and force control frame work and developed a hybrid force/position control law, where each manipulator joint provides a hybrid torque which affects both the end effector force and position. However detailed stability analysis of the hybrid control law were not performed. Anderson and Spong[3] combined the notion of impedance control with hybrid force/ position control to allow for precise force servoing which might be required in many applications. The dynamic equations of the manipulator along an algebraic constraint surface were identified as a singular system of differential equations (or descriptor systems) by McClamroch. Khrisnan and McClamroch[4] also considered the design of a hybrid/force position control scheme for flexible joint robots by linearizing them about an operating point. Ham[5-6] proposed adaptive nonlinear control of one-link flexible arm and adaptive control based on explicit model of robot manipulator. In this papaer, we derive motion equation that can be used to make the trajectory of end-effector move from an initial point to the desired point on the constrained task plane(or hyperplane) by using the mathematical tools concerning vector fields of manifold. We also consider the robust motion equation that can be used even in the presence of uncertainties. We suggest a new force/motion control scheme of robot manipulator based on the proposed motion equation. The paper is organized as follows. In section II, we describe the concepts of submanifold and determistic and robust motion equations. In section III, we formulate the dynamics of the rigid robot manipulator with an end-effector. In section IV, we design a new force/position control law. In section V, the feasibility of the proposed a new force/position control scheme is verified through a computer simulation.

2 Mathematical Tools

In this section, we briefly discuss the concepts of submanifolds and derive motion equations based on that concepts.

Let Δ be a nonsingular and has dimension d, in a neighborhood U^o of x^o. Then the following lemma is satisfied.

Lemma 2.1 *Let Δ be a smooth distribution and x^o a regular point of Δ. Suppose $dim(\Delta(x^o)) = d$, then*

there exist an open neighborhood U^o of x^o and a set $\{f_1, ..., f_d\}$ of smooth vector fields defined on U^o with the property that

i) the vectors $f_1(x), ..., f_d(x)$ are linearly independent at each x in U^o.
ii) $\Delta(x) = span\{f_1(x), ..., f_d(x)\}$ at each x in U^o.

Remark 2.1 *Moreover, every smooth vector field τ belonging to Δ can be expressed, on U^o, as*

$$\tau(x) = \sum_{i=1}^{d} c_i(x) f_i(x)$$

where $c_1(x), ..., c_d(x)$ are smooth real-valued function of x, defined on U^o.

Based on the above mathematical tool, we suggest two theorems concerning motion equation under the following assumption 2.1. One is for a deterministic motion equation which has no uncertainty and the other one is for a motion equation which has some uncertainties.

Let x_o, x_1 be two points on a maximal submanifold S of nonsingular d–dimensional smooth distribution $\Delta(x)$ and define a set B_δ contained in S as follows

$$B_\delta = \{x | \parallel x - x_o \parallel < \delta, x \in S\}.$$

Assumption 2.1 *The absolute angle between T_xS and T_yS is less than $\pi/2$ for any points $x, y \in B_\delta$ where $\delta = \parallel x_1 - x_o \parallel$.*

Theorem 2.1 (WITHOUT UNCERTAINTY)
Let x_o and x_1 be the points on a same maximal integral submanifold S of the distribution Δ. Then there exist some time functions $k_1(t), \cdots, k_d(t)$ such that the following motion equation move the state trajectory from x_o to x_1 without deviating from the integral submanifold S.

$$\dot{x}(t) = k_1(t) f_1(x) + ... + k_d(t) f_d(x) \qquad (2.1)$$

where $f_1(x), ..., f_d(x)$ are independent and selected such that

$$\Delta(x) = span\{f_1(x), ..., f_d(x)\}. \qquad (2.2)$$

Proof: The proof is based on the Lyapunov-like function as follow

$$V(x) = \frac{(x_1 - x)^T (x_1 - x)}{2}. \qquad (2.3)$$

Taking the time derivative of V along the trajectory of (2.1) yields

$$\begin{aligned} \dot{V}(x) &= -(x_1 - x)^T \dot{x} \qquad (2.4) \\ &= -(x_1 - x)^T (k_1(t) f_1(x) + ... + k_d(t) f_d(x)) \end{aligned}$$

Let us define

$$k_i(t) \cong (x_1 - x)^T f_i(x) \beta_i(t)$$

where $\beta_i(t) > 0$, $i = 1, 2, ..., d$. Then, we can write as follow

$$\begin{aligned} \dot{V}(x) &= -[((x_1 - x)^T f_1(x))^2 + ((x_1 - x)^T f_2(x))^2 \\ &+ ... + ((x_1 - x)^T f_d(x))^2] \beta_i(t) \leq 0. \qquad (2.5) \end{aligned}$$

Let Ω_1 be the largest invariant set of contained in set Ω_2 where $\Omega_2 = \{x \in B_\delta : \dot{V}(x) = 0\}$, then we can find $\Omega_1 = \{x_1\}$.. By using LaSalle's theorem, we can find $x(t)$ approaches x_1 as time goes to infinity. ∎

Lemma 2.2 *If $\dot{x}(t) = f(x, t)$ is a stable, then*

$$\ddot{x}(t) = F(x, \dot{x}, t),$$

where

$$F(x, \dot{x}, t) = \frac{\partial f(x,t)}{\partial x} \dot{x}(t) + \frac{\partial f(x,t)}{\partial t} - \gamma(\dot{x}(t) - f(x, t)),$$

is also stable for any constant $\gamma > 0$.

Proof: Let $z(t) = \dot{x}(t) - f(x, t)$, then from the above equation we obtain

$$\dot{z}(t) = -\gamma z(t).$$

Because $\gamma > 0$, the above system is stable. So for any $z(0)$

$$\lim_{t \to \infty} z(t) = 0$$

and therefore we can see that

$$\lim_{t \to \infty} x(t) = 0. \qquad \blacksquare$$

Example 2.1 Consider the following distribution, defined on R^3

$$\Delta(\mathbf{x}) = span\{(0, -x_3, x_2)^T, (x_2, -x_1, 0)^T\}.$$

Let S be integral submanifold of the above distribution Δ. Then

$$S = \{(x_1, x_2, x_3) | \; x_1^2 + x_2^2 + x_3^2 = c \;\; for \; some \; c\}.$$

Consider the following points $x_o = (1,0,0)^T, x_1 = (0,1,0)^T$ and we choose $k_1(t), k_2(t)$ as follows

$$\begin{aligned} k_1(t) &= (x_1 - x)^T f_1(x)(1 - e^{-3t}) \\ k_2(t) &= (x_1 - x)^T f_2(x)(1 - e^{-3t}) \end{aligned} \quad (2.6)$$

then, from theorem 2.1, we obtain the motion equation that move the state trajectory from x_o to x_1 as follows

$$\begin{aligned} \dot{x}(t) &= f_1(x)(x_1 - x)^T f_1(x)(1 - e^{-3t}) \\ &+ f_2(x)(x_1 - x)^T f_2(x)(1 - e^{-3t}). \end{aligned} \quad (2.7)$$

The state trajectory of above system is shown in Fig.1. From Fig.1, we can check the validness of theorem 2.1.

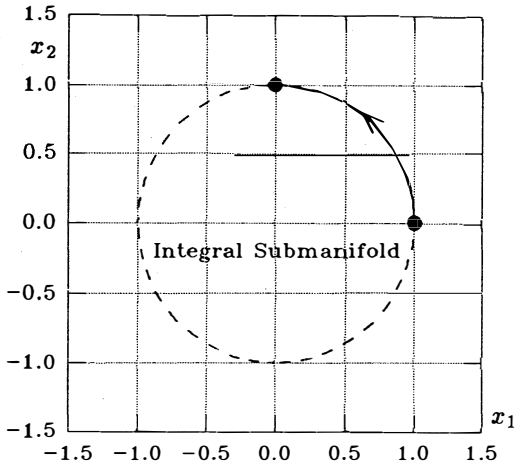


Fig.1 The state trajectory of motion equation ($x_3 = 0$).

Theorem 2.2 (WITH UNCERTAINTY)
Let x_o and x_1 be the points on a same maximal integral submanifold S of the distribution Δ. Then there exist some time functions $k_1(t), \cdots, k_d(t)$ such that the following motion equation move the state trajectory from x_o to x_1 without deviating from the integral submanifold S.

$$\begin{aligned} \dot{x}(t) &= k_1(t)f_1(x) + k_2(t)f_2(x) + ... + k_d(t)f_d(x) \\ &+\eta_1(t)f_1(x) + \eta_2(t)f_2(x) + ... + \eta_d(t)f_d(x) \end{aligned} \quad (2.8)$$

where $|\eta_i(t)| < \delta_i$, $i = 1,2,...,d$ and $f_1(x), ..., f_d(x)$ are independent and selected such that

$$\Delta(x) = span\{f_1(x), ..., f_d(x)\}. \quad (2.9)$$

Proof: The proof is based on the Lyapunov-like function as follows

$$V(x) = \frac{(x_d - x)^T(x_d - x)}{2}. \quad (2.10)$$

Taking the time derivative of V along the trajectory of (2.8) yields

$$\begin{aligned} \dot{V}(x) &= -(x_d - x)^T \dot{x} \\ &= -(x_d - x)^T \sum_{i=1}^{d}(k_i(t) + \eta_i(t))f_i(x). \end{aligned} \quad (2.11)$$

Let us choose

$$k_i(t) = (x_d - x)^T f_i(x) + K_i tanh(s_i) \quad (2.12)$$

where $s_i = (x_d - x)^T f_i$ and $K_i > 0$, $i = 1,2,...d$.

Let us define $\dot{V}_i(x)$ as follows

$$\dot{V}_i(x) = -(x_d - x)^T(k_i(t) + \eta_i(t))f_i(x)$$

then, by substituting equation (2.12) into equation (2.11), we obtain

$$\begin{aligned} \dot{V}(x) &= \sum_{i=1}^{d}(-s_i^2 - s_i\eta_i(t) - s_i K_i tanh(s_i)) \\ &\leq 0, \end{aligned} \quad (2.13)$$

if $|s_i| < \frac{\delta_i}{1+K_i}$.

If the absolute value of s_i is greater than $\frac{\delta_i}{1+K_i}$, then the value of $\dot{V}_i$ is less than zero. Therefore we can guarantee that the following inequality holds as time goes to infinity

$$\begin{aligned} |s_i| &< \frac{\delta_i}{1 + K_i} \\ |s_i| &= \| (x_d - x)^T f_i(x) \| \\ &\cong \| (x_d - x)^T f_i(x_d) \| < \frac{\delta_i}{1 + K_i}. \end{aligned} \quad (2.14)$$

Let the angle between $x_d - x$ and $f_i(x_d)$ be θ_i. Then we can obtain the following inequalities

$$\| x_d - x \| \| f_i(x_d) \| cos\theta_i < \frac{\delta_i}{1 + K_i}$$

$$\| x_d - x \| < \min_{i \in I}\{\frac{\delta_i}{\| f_i(x_d) \| cos\theta_i(1 + K_i)}\} \quad (2.15)$$

where $I = \{1, 2...d\}$.
Therefore if K_i is selected to be large, then $\| x_d - x \|$ becomes small. ∎

3 Dynamics of Two-Link Robot Manipulator

In this section, we derive the dynamics of two-link rigid robot manipulator with an end-effector. In

derivation of dynamics, we consider end-effector as point-mass. .

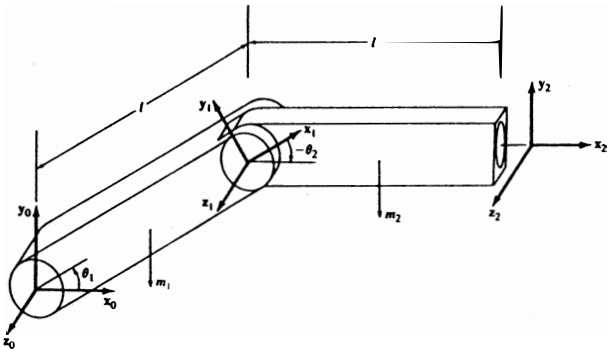


Fig.2 The two-link robot manipulator.

When we apply Euler-Lagrangian equation to the robot manipulator shown in Fig.3, we obtain the inertia matrix term as follows

$$\begin{aligned} D_{11} &= l^2[\frac{1}{3}m_1 + \frac{4}{3}m_2 + 2m_l + m_2c_2 + 2m_lc_2] \\ D_{12} &= l^2[\frac{1}{3}m_2 + \frac{1}{2}m_2c_2 + m_l + m_lc_2] \\ D_{21} &= D_{12} \\ D_{22} &= l^2[\frac{1}{3}m_2 + m_l] \end{aligned} \tag{3.1}$$

and the Coriolis and centrifugal matrix terms as follows

$$\begin{aligned} C_{11} &= -\frac{1}{2}m_2l^2s_2\dot{q}_2 - m_ll^2s_2\dot{q}_2 \\ C_{12} &= -\frac{1}{2}m_2l^2s_2(\dot{q}_1 + \dot{q}_2) - m_ll^2s_2(\dot{q}_1 + \dot{q}_2) \\ C_{21} &= \frac{1}{2}m_2l^2s_2\dot{q}_1 + m_ll^2s_2\dot{q}_1 \\ C_{22} &= 0 \end{aligned} \tag{3.2}$$

and the gravity vector terms as follows

$$\begin{aligned} G_1(q) &= \frac{1}{2}m_1glc_1 + \frac{1}{2}m_2glc_{12} + m_2glc_1 \\ &+ m_llc_1g + m_llc_{12}g \\ G_2(q) &= \frac{1}{2}m_2glc_{12} + m_llc_{12}g. \end{aligned} \tag{3.3}$$

where $c_1 = cosq_1, c_2 = cosq_2, s_2 = sinq_2, c_{12} = cos(q_1 + q_2), m_l$ denotes point mass of an end-effector.

Therefore the resulting dynamic equation of two-link robot manipulator considering the normal force of the end-effector can be expressed as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + J^T(q)F_n(q) = u \tag{3.4}$$

where $F_n(q)$ is normal force(or contact force) of the end-effector on frictionless surface.

4 Controller Design

In this section, we propose new force/position control law for the robot manipulator based on the motion equation discussed in section II. The new force/position control law guarantees that closed-loop system has the uniformly bounded stability. The mathematics that relate the world coordinate to the joint angle coordinate is inherently nonlinear and can be expressed by a nonlinear vector valued function as

$$x = H(q) \tag{4.1}$$

where $x \in R^6$ and $q \in R^n$. The velocity of the end-effector is related to the joint velocity $\dot{q}$ as follows,

$$\dot{x} = J(q)\dot{q}. \tag{4.2}$$

where $J(q)$ is the manipulator's Jacobian. The end-effector and joint accelerations are related by the following equation

$$\ddot{x} = \dot{J}(q)\dot{q} + J(q)\ddot{q}. \tag{4.3}$$

Now, we design the force/position control law such that above equation is the same as motion equation suggested in Lemma 2.2, i.e.,

$$\begin{aligned} \ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q} &= F(H(q), J(q)\dot{q}) \\ &= F(x, \dot{x}, t) \end{aligned} \tag{4.4}$$

From equation (3.4), we obtain

$$\begin{aligned} \ddot{q} + D^{-1}(q)C(q,\dot{q})\dot{q} &+ D^{-1}(q)G(q) \\ &+ D^{-1}(q)J^T(q)F_n(q) \\ &= D^{-1}(q)u. \end{aligned} \tag{4.5}$$

If we premultiply $J(q)$, then equation (4.5) becomes

$$\begin{aligned} J(q)\ddot{q} &+ J(q)D^{-1}(q)C(q,\dot{q})\dot{q} + J(q)D^{-1}(q)G(q) \\ &+ J(q)D^{-1}(q)J^T(q)F_n(q) \\ &= J(q)D^{-1}(q)u. \end{aligned} \tag{4.6}$$

From (4.3) and (4.6), we get

$$\begin{aligned} \ddot{x} &- \dot{J}(q)\dot{q} + J(q)D^{-1}(q)C(q,\dot{q})\dot{q} \\ &+ J(q)D^{-1}(q)G(q) + J(q)D^{-1}(q)J^T(q)F_n(q) \\ &= J(q)D^{-1}(q)u. \end{aligned} \tag{4.7}$$

If we apply equation (4.4) to above equation, we get

$$\begin{aligned} F(x,\dot{x},t) &- \dot{J}(q)\dot{q} + J(q)D^{-1}(q)C(q,\dot{q})\dot{q} \\ &+ J(q)D^{-1}(q)G(q) \\ &+ J(q)D^{-1}(q)J^T(q)F_n(q) \\ &= J(q)D^{-1}(q)u. \end{aligned} \tag{4.8}$$

Let us define vector v as follow

$$D^{-1}(q)u = J^T(q)v. \tag{4.9}$$

Then, we can find the vector v as follow by using pseudo inverse of $J^T(q)$

$$\begin{aligned} v &= (J(q)J^T(q))^{-1}[F(x,\dot{x},t) - \dot{J}(q)\dot{q} \\ &+ J(q)D^{-1}(q)C(q,\dot{q})\dot{q} + J(q)D^{-1}(q)G(q) \\ &+ J(q)D^{-1}(q)J^T(q)F_n(q)]. \end{aligned} \tag{4.10}$$

Therefore a new force/position controller of robot manipulator as follows

$$\begin{aligned} u &= D(q)J^T(q)(J(q)J^T(q))^{-1}[F(x,\dot{x},t) - \dot{J}(q)\dot{q} \\ &+ J(q)D^{-1}(q)C(q,\dot{q})\dot{q} + J(q)D^{-1}(q)G(q) \\ &+ J(q)D^{-1}(q)J^T(q)F_n(q)]. \end{aligned} \tag{4.11}$$

5 Computer Simulation

Computer simulations are conducted to verify the validity, effectiveness and performance of the proposed a new force/position control scheme. We simulated the proposed a new force/position controller to control the joint angle θ_1, θ_2 of the two-link rigid robot manipulator. Table.1 show the physical parameters of two-link robot manipulator. We set the task plane of end-effector to elliptic function $(x^2 + 4y^2 = 2)$ and set the initial angles and velocities of the link 1 and link 2 to $\theta_1 = \frac{\pi}{4}, \theta_1 = 0, \theta_2 = -\frac{\pi}{2}, \dot{\theta}_2 = 0$. Hence the initial location and velocity of end-effector is $(\sqrt{2}, 0)$ and 0 respectively. We also set the desired location and velocities of end-effector to $(\sqrt{\frac{2}{5}}, \sqrt{\frac{2}{5}})$ and 0.

parameters	values	units
m_1	2	kg
m_2	1.2	kg
m_l	0.2	kg
l	1	m
g	9.8	N/kg

Table 1. Physical parameters of two-link robot manipulator

6 Conclusions

In this paper, we propose a new force/position control scheme to perform force and position control of the robot manipulator when the end-effector is moving along a frictionless surface represented by algebraic equation. The presented control scheme is based on the motion equation that can be obtained by using differential geometry. The magnitude of vector fields that charaterize the motion equation is designed by using Lyapunov-like function. The uniformly ultimate boundness of the control scheme is guaranteed and has been demonstrated by a simulation. In near future, we will study and develop the adaptive version of the proposed control law which can be applied in real situation when a part of parameters of robot dynamics are unknown.

References

[1] Whitney, D.E., "Historical Perspectives and State of the Art in Robot Force Control," *Proceedings of the 1985 IEEE Conference on Robotics and Automation*, pp.262-268,St.Louis,MO,March 25-28,1985.

[2] Raibert, M.H., J.J., "Hybrid Position/Force Control of Manipulators ," *ASME Journal of Dynamic Systems Measurement and Control*, 102, pp.126-133 June 1981.

[3] Spong, M.W., "On the Force Control Problem for Flexible Joint Manipulators," *IEEE Trans. Automat. Contr.*, Vol.34,No.1, pp.107-111,January 1989.

[4] McClamroch, N. H., "Singular Systems of Differential Equations as Dynamical Models for Constrained Robot Systems," *Proceedings of the 1986 IEEE Robotics and Automation Conference* , April 1986 San Francisco, CA.

[5] W.Ham, "Adaptive control based on explicit model of robot manipulator ,"*IEEE Trans. Automat. Contr.*, vol.38, Apr,pp.654-658, 1993.

[6] W.Ham, J.J.Lee "Adaptive nonlinear control of one-link flexible arm," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.1, pp.3-7, Yokohama, Japan, 1993.

[7] W.Ham, J.J.Lee "Adaptive Control Based on Speed gradient Algorithm for Robot Manipulators," *IECON '94 Proceedings of the 20th International Conference on Industrial Electronics Control and Instrument.*, vol.2, pp.776-781, Bologna,Italy,September 1994.

[8] Isidori, "Nonlinear Control Systems : An Introduction ," Springer-Verlag, Berlin, New York, 1985.

Autonomous Micro-Robot 'Kity' for Maze Contest

Jong-Hwan Kim, Myung-Jin Jung, Hyun-Sik Shim and Seon-Woo Lee
Department of Electrical Engineering
KAIST
Taejon, 305-701, Korea
johkim, mjjung, hsshim, senu@vivaldi.kaist.ac.kr

Abstract

In this paper, we propose a table-look-up immune network which is extracted from an immune network to solve problems that arise from autonomous micro-robots to achieve a given goal with limited memory and calculation capacity. This method is implemented and tested with the micro-robot 'Kity', the size of which is less than 1 inch^3. It is possible to generate enough rules to make the robot achieve the goal to navigate freely in a maze with a very small number of sensors. Experimental results show that the immune network can be used to control a robot in a restricted environment. Kity demonstrated its efficient algorithm by taking the 1st prize at the 4th International Micro Robot Maze Contest held in Nagoya, Japan on October 6, 1995.

1 Introduction

Since the first mechanical devices were invented centuries ago, the size of the parts was significantly reduced and the precision increased tremendously. Miniaturization of real life objects has always been a fascination for the researchers. Small models exist for everything. Powering these small objects or giving them some autonomy or even *intelligence* is still a challenge left for future generations. Small mechanisms are machined and assembled by different techniques called precision engineering, micro-mechanics or micro-mechatronics, with this last word being used when some control electronics is embedded into the system.

In this paper, a table-look-up immune network is proposd for the proper behaviors of the autonomous micro-robot in a maze. The table consists of antibodies (behaviors) extracted from an immune network[1] to solve a problem of finding a way from start point to goal point via five control points in the maze. The key technology is to implement an autonomous micro-robot sized less than 1 $inch^3$ for the achievement of a given objective with limited memory and calculation capacity.

The algorithm is implemented and tested with the micro-robot Kity, the size of which is less than 1 $inch^3$. With a very small number of sensors it is possible to generate enough rules to make the robot achieve the objective to move freely in a maze that consists of channels and nodes.

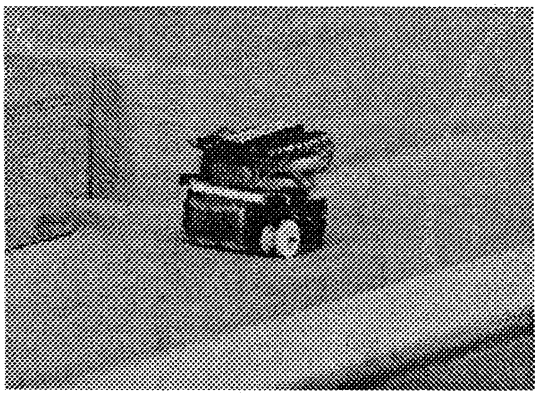


Fig. 1. Kity in the maze

The immune system, in particular, has various interesting features such as immunological memory, immunological tolerance, micro-pattern recognition, non-hierarchical distributed structure, and so on from the standpoint of engineering. In addition to this, recent studies on immunology have clarified that the immune system does not only detect and eliminate the non-self materials called *antigens* such as virus, cancer cells and so on which come from inside or outside of the living system, but play important roles to maintain its own system against dynamically changing environments. Therefore, the immune system would be expected to provide a new paradigm suitable for dy-

namic problem dealing with unknown environments rather than static problem. So far, several methods have been proposed to realize phenomena of the immune system from standpoint of mathematical analysis. However, the immune system has little been applied to engineering fields in spite of its productive characteristics. Based on the above facts, Ishiguro et. al. [2] studied the immune system for the engineering purpose, and applied it to the mobile robot. They expected that there would be an interesting AI technique for dynamically changing environment by imitating the immune system in living organisms.

Experiment results show that immune network can be used to control a robot in a restricted environment. Kity demonstrated its efficient algorithm by winning the 1st prize at the 4th International Micro Robot Maze Contest held in Nagoya, Japan on October 6, 1995.

2 Description of Kity

Figure 2 shows the structure of Kity, an autonomous micro-robot of less than 1 $inch^3$ in size.

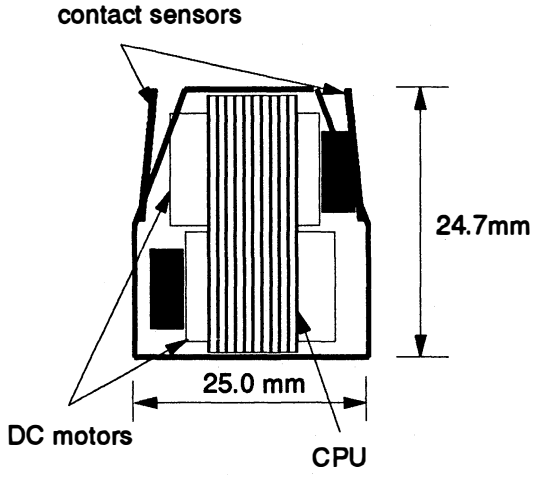


Fig. 2. The structure of Kity.

It consists of two contact sensors, two DC motors, two 3-Volt lithium batteries, and an *i*8051 compatible micro-processor [3]. An *i*8051 processor is one of widely used Intel's 8 bit micro-processors for controlling small machineries. The compatible chip has two kilo bytes of flash memory that makes it easy to install the code. To control Kity only two digital control signal outputs to the DC motors and two digital input channels to receive signals of contact sensors are used. Light weight of Kity makes it critical to choose adequate contact sensors that operate under very weak force. The wheels are geared to the DC motors with the gear ratio 9 : 100. Because only 'on-off' control is possible in the system, Kity can go forward, turn left, and right, but can take no backward motion.

3 Immune Networks

Given rich chemical environment present in a highly evolved organism, it is inevitable that foreign organisms will attempt to invade in an effort to make use of these resources. To counteract this, in vertebrates the immune system has evolved to identify and dispose of foreign materials. This is done in part by antibody molecules that tag foreign materials and mark them for eventual removal by lymphocytes, phagocytic cells, and the complement system. The specialized portion of the antibody molecule used for identifying other molecules is called the *antibody combining region* or *paratope*. The sequence of amino acids making up the paratope determines its shape and hence the set of other molecules that it can react with. If the shape of an foreign antigen molecule matches that of the paratope, the antibody can attach itself to the antigen, leading to its eventual demise. The regions on any molecule (antigen or antibody) that the paratopes can attach to are called *epitopes*. The set of all paratopes is like a very large collection of keys, and the set of all possible epitopes a very large collection of locks. Recent studies on immunology have clarified that each type of antibody has also its specific antigen determinant called *idiotope*

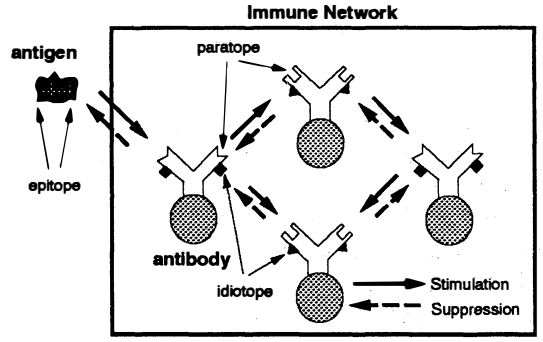


Fig. 3. Basic concept of immune network.

Based on this face, Jerene who is an immunologist proposed a remarkable hypothesis *idiotypic network hypothesis* [4]. This hypothesis explains that antibodies do not exist independently in living organism, but

communicate with each other through idiotope and paratope.

This idea is schematically shown in Figure 3. These mutual interactions (stimulation and suppression) chains between different species of antibodies form the large-scale closed chain loop which works as a self and non-self recognizer. The important point of Jerne's idea is that the immune system does not act as a unit-level recognition system but as a system-level recognition system.

4 Application of Immune Network to Kity Control

A a table-look-up immune network is employed to solve problems that arise to achieve a given goal of an autonomous micro-robot. The algorithm is implemented and tested with the micro-robot Kity. With a very small number of sensors it is possible to generate enough rules to make the robot achieve the goal to navigate freely in a maze that consists of channels and nodes.

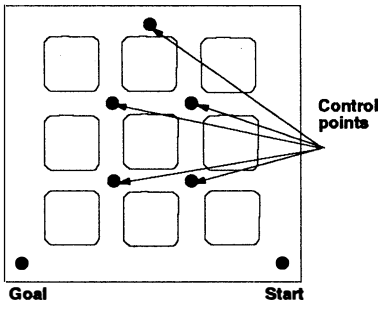


Fig. 4. Maze for 1 $inch^3$ robot.

Figure 4 shows the maze in which Kity should run. Control points are where Kity should pass by at least once. Because of limitation of the structure due to the size, we mainly focused on solving the following problems for Kity to navigate any path in the maze with only the 2 contact sensors.

- Turning a corner; left or right at a node.
- Traversing a node.
- Percepting when it is stuck by the wall because of the rough terrain on the playground.

The first two problems are graphically described in Figure 5 and 6. Figure 5 (a) shows a cornering motion. In this motion, Kity should check if it has

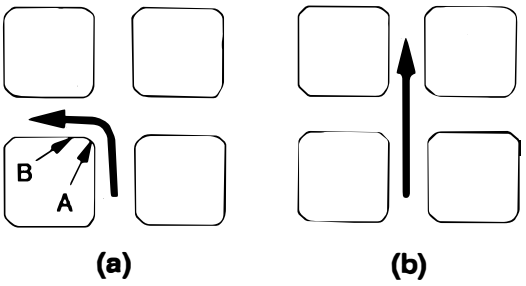


Fig. 5. Trajectories for turning a corner and traversing in a node.

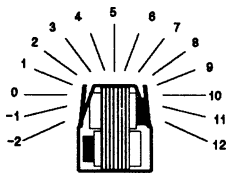


Fig. 6. Numbered orientation for antibody.

turned a corner or has not yet. Figure 5 (b) shows a going straight motion in a node. If Kity has a good command of these two motions, it is able to navigate any path in the maze by just inputting a sequence of traversing a node, turning left at a node, or turning right at a node. Here, let us define Mode 1 as a motion of traversing a node and Mode 2 as a motion of turning at a node. For these motions, antibodies are defined as in Figure 7 and Figure 8, respectively. Orientation of Kity was defined and denoted by numbers as in Figure 6. The number of antibodies used for Mode 1 is seven, and that for Mode 2 is nineteen. The longer the antigen, the smoother Kity's motion at a node in Mode 2.

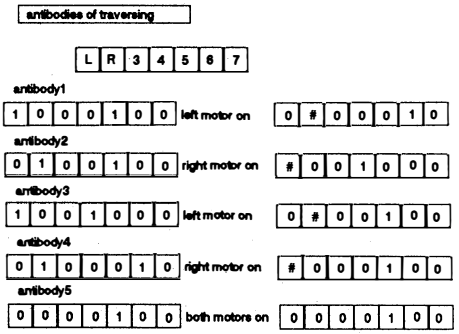


Fig. 7. Antibodies for traversing a node.

The last problem is an additional one. Because of the discrepancy of the center of wheels, it is liable to occur when Kity runs on a channel with a rough

terrain. The problem should be considered in mind when attending a contest.

Figure 7 shows some examples of antibodies for traversing straight and Figure 8 for turning left. Similarly, antibodies for turning right can be defined. Two sensor information and the orientation of Kity are used for each antigen. L indicates the left-hand side sensor detects a wall, R indicates the right-hand side one does, and each element of the rest parts is a corresponding orientation. The symbol # denotes 'don't care' condition. The behavior of antibody of maximum concentration is chosen as a proper one.

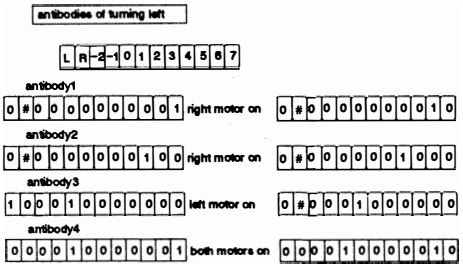


Fig. 8. Antibodies for turning left.

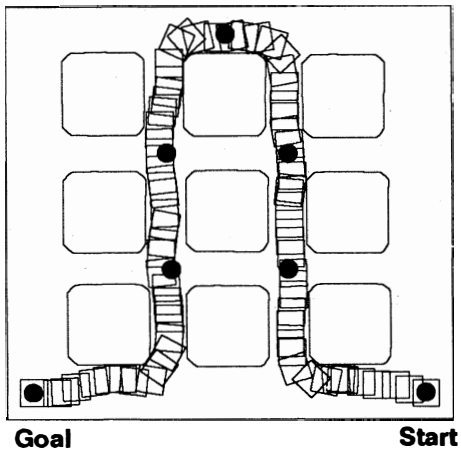


Fig. 9. Kity's Trajectory.

When Kity navigated in Mode 1, the concentration of antibodies for running straight was increased such that it could keep Kity run straight. After traversing a node and when reaching another one, it changes the mode according to the input sequence of Mode 1 and Mode 2. It was observed that when Kity ran in Mode 2 at a node, the concentration of antibodies for rotation was increased (at point A in Figure 5 (a)) and after entering a channel (at point B in Figure 5 (b)) the concentration of antibodies for going straight was increased. This makes Kity notice that it has turned a corner completely. This confirmation of having turned a corner is very important because Kity doesn't have a precise encoder or odometer. These points at which the confirmation has been done are used as internal landmarks; Kity compensates its accumulated error at these points. It was possible to generate a look-up-table that describes the sensor input and output to actuators. With this table, the calculation time could be reduced.

Figure 9 shows an experimental result of Kity in a maze. Kity's trajectory might be almost perfect one that a micro-robot can have.

5 Conclusion

Limitation in size of micro-robots makes it difficult to realize them as autonomous ones. In this paper, to achieve an objective under a restrictive environment the immune network has been used for an autonomous micro-robot. By making a look-up-table of the network results, the calculation time could be reduced. This table-look-up method makes it easy to implement in a small size micro-robot like Kity. Kity ran the shortest path in the maze with this rule table by 52 seconds at the contest, which gave the first winner place to Kity.

References

[1] J. D. Farmer and N. H. Packard, "The immune system, adaptation, and machine learning," *Physica* 22D, pp. 187-204, 1986.

[2] A. Ishiguro, T. Kondo, Y. Watanabe and Y. Uchikawa, "An Immnunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots," *IEEE Conf. IROS*, pp. 495-500, 1995.

[3] S. K. Park, *i8051 programming in C*, Ohm publishing company, 1995.

[4] N. K. Jerne, "Idiotypic networks and other preconceived ideas," *Immunological Rev.*, Vol.79, pp.5-24, 1984.

Towards Designing Artificial Neural Networks by Evolution*

Xin Yao and Yong Liu
Computational Intelligence Group, Department of Computer Science
University College, The University of New South Wales
Australian Defence Force Academy, Canberra, ACT, Australia 2600
Email: xin@csadfa.cs.adfa.oz.au

Abstract

Designing artificial neural networks (ANNs) for different applications has been a key issue in the ANN field. Although there are many training algorithms available for learning ANN's connection weights, algorithms which can learn ANN's architectures are relatively few. At present, ANN design still relies heavily on human experts who have sufficient knowledge about ANNs and the problem to be solved. As ANN's complexity increases, designing ANNs manually becomes more difficult and unmanageable. Simulated evolution offers a promising approach to tackle this problem. This paper describes an evolutionary approach to design ANNs. We call these ANNs designed by the evolutionary process as evolutionary ANNs (EANNs). In other words, EANNs refer to a special class of ANNs in which evolution is another fundamental form of adaptation in addition to learning (refers to weight training here). We use an evolutionary algorithm similar to evolutionary programming (EP) to evolve both architectures and connection weights (including biases) of ANNs. Five mutation operators have been proposed for our evolutionary algorithm. In order to improve the generalisation ability of evolved ANNs, these five operators are applied sequentially and selectively. We have also used validation sets in our studies to further improve generalisation. The evolutionary algorithm allows ANNs to grow as well as shrink during the evolutionary process. The evolutionary algorithm also incorporates the weight learning process as part of its mutation process. In a sense, our EANN system is a hybrid evolution and learning system. Extensive experimental studies have been carried out to test our EANN system. This paper will give some of the experimental results which show the effectiveness of the system.

*This work is supported by the Australian Research Council through its small grant scheme.

1 Introduction

Designing artificial neural networks (ANNs) through simulated evolution has been investigated for many years [1, 2, 3, 4]. It offers a very promising and automatic alternative to designing ANNs manually. The advantage of the automatic design over the manual design becomes clearer as the complexity of ANNs increases. Manual design of ANNs requires the designer to have very good knowledge in both ANNs and the problem to be solved by the ANN. However, such knowledge is often unavailable for a non-ANN-expert facing a real-world problem.

Evolutionary artificial neural networks (EANNs) [3, 4] refer to a special class of artificial neural networks (ANNs) in which evolution is another fundamental form of adaptation in addition to learning. They provide a general framework for investigating various aspects of simulated evolution and learning, including the automatic design of ANNs. The general framework make it clear where the automatic design of ANNs fits into the whole picture of a general adaptive system which can change its behaviours through changing its "hardware", i.e., weights and architectures, and its "software", i.e., learning rule. We will review such a general framework [5, 6] in Section 2. Then we discuss the issue of designing ANNs through simulated evolution in Section 3. Section 4 presents our latest evolutionary system which can evolve ANN's weights and architectures at the same time. Finally, Section 5 gives our conclusions and indicates some future research topics.

2 A General Framework for EANNs

Evolution can be introduced into ANNs at various levels. At the lowest level, evolution can be intro-

duced into weight[1] training, where ANN's weights are evolved. This evolutionary process is similar to the learning process in the connectionist paradigm where weights are adjusted in order to learn certain functions. At the next higher level, evolution can be introduced into ANN's architecture adaptation, where the architecture is evolved, rather than designed by human beings. At the highest level, evolution can be introduced into ANN's learning rule, i.e., the rule which specifies how to adjust weights in weight training. Since the weight training has traditionally been regarded as a learning process, the evolution of learning rules can be regarded as a process of learning to learn (weights). A general framework of EANNs which includes the above three levels of evolution is given in Figure 1 [3].

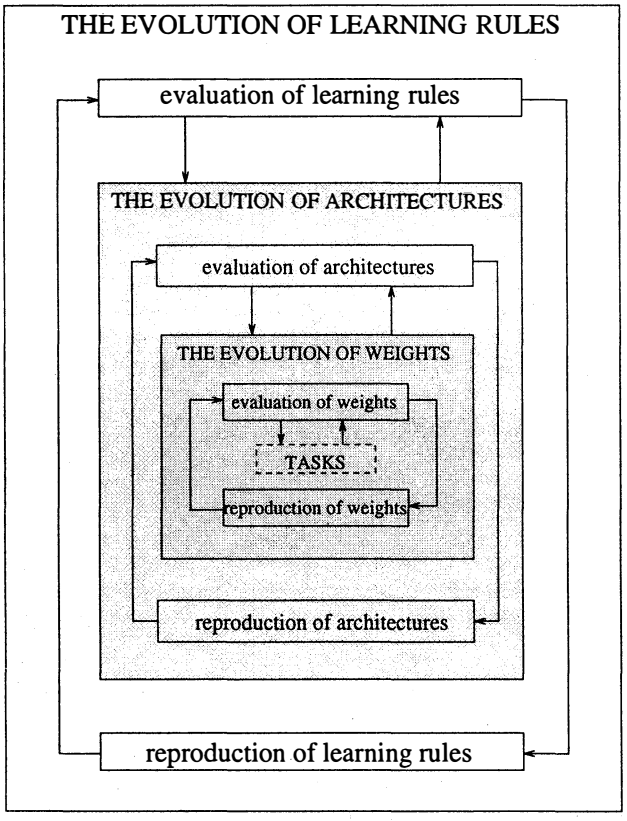


Figure 1: A general framework for EANNs.

There have been some discussions on whether the evolution of learning rules is at the highest level among the three in Figure 1 [3, 4]. From the point of view of engineering, the decision on the level of evolution depends on what kind of prior knowledge is available. If there is more prior knowledge about EANN's architectures than that about their learning rules or a particular class of architectures is pursued, it is better to put the evolution of architectures at the highest level because such knowledge can be used to reduce the (architecture) search space and the lower level evolution of learning rules can be more biased towards this kind of architectures. On the other hand, the evolution of learning rules should be at the highest level if there is more prior knowledge about them available or there is a special interest in certain type of learning rules.

Figure 1 provides us with a common framework for discussing various EANN models if we interpret the simulated evolution in a broader sense, that is, if we interpret simulated annealing (SA), gradient descent search, exhaustive search, etc., as special cases of evolutionary algorithms. For example, the traditional back-propagation (BP) network can be considered as a special case of our general framework with one-shot (only-one-candidate) search used in the evolution of architectures and learning rules and the BP algorithm used in the evolution of connection weights. In fact, our general framework defines a three-dimensional space where 0 represents one-shot search and +1 represents exhaustive search along each axis. Each EANN model corresponds to a point in this space, where the three coordinates represent the three algorithms used by the EANN model in searching for the weights, architecture and learning rules. In this paper, we will describe an EANN model which uses one-shot algorithm to search for the learning rule, a variant of evolutionary programming (EP) to search for the architecture, and a hybrid BP-SA algorithm to search for the weights.

3 Automatic Design of ANNs Through Evolution

The evolution of architectures provides an automatic way to design ANNs. It is obvious from the general framework in the previous section that any algorithm can be used to implement such simulated evolution. There are constructive and pruning algorithms which learn ANN's architectures and weights, but they are "susceptible to becoming trapped at structural local optima" [7]. In addition, they "only investigate restricted topological subsets rather than the complete class of network architectures" [7]. Evolutionary algorithms, like genetic algorithms, EP, and evolution strategies, are better suited to the task of evolving ANN's architectures [8].

Evolving ANN's architectures by evolutionary algo-

[1] Weights include connection weights and biases.

rithms is not an easy task. The many to many mapping between genes and phenotypes has caused two major problems in evolving ANNs. The first problem is the noisy fitness evaluation problem. That is, we normally use one phenotype's fitness to approximate its genotype's fitness. For example, the fitness of a trained ANN (from a random set of initial weights) is often used to represent the fitness of the ANN's architecture. Such evaluation of the architecture is noisy because it depends on the random initialisation and the training algorithm used. One genotype may have many different phenotypes.

The second problem is the well-known permutation problem (or competing conventions problem). That is, one phenotype may have many different genotypes. For example, two strictly-layered feedforward ANNs which order their hidden nodes differently are functionally equivalent, but may have different genotypic representations.

In order to alleviate the above two problems in the evolutionary design of ANNs, we have decided to evolve the architecture and the weights at the same time using an EP algorithm. The simultaneous evolution of the architecture and weights means that each individual in the population is an ANN with the architecture and weights. Its evaluation is more accurate because we are evaluating an ANN with the architecture and weights. We do not use the fitness of an ANN with the architecture and weights to represent that of the architecture. The use of an EP algorithm avoids crossover operators which will not be effective due to the permutation problem.

4 A New System for Designing ANNs — EPNet

We have developed a new evolutionary system for designing ANNs automatically. The main structure of the system, EPNet, is given in Figure 2.

The EPNet system is built upon an EP algorithm which adopts a rank-based selection scheme [9] and five mutations; hybrid training, node deletion, connection deletion, connection addition and node addition [10, 11, 12]. Hybrid training is the only mutation in EPNet which modifies ANN's weights. It is based on a modified BP (MBP) algorithm with an adaptive learning rate and an SA algorithm. The other four mutations are used to grow and prune hidden nodes and connections. Only feedforward ANNs are considered in EPNet at present.

The number of epochs used by MBP to train each

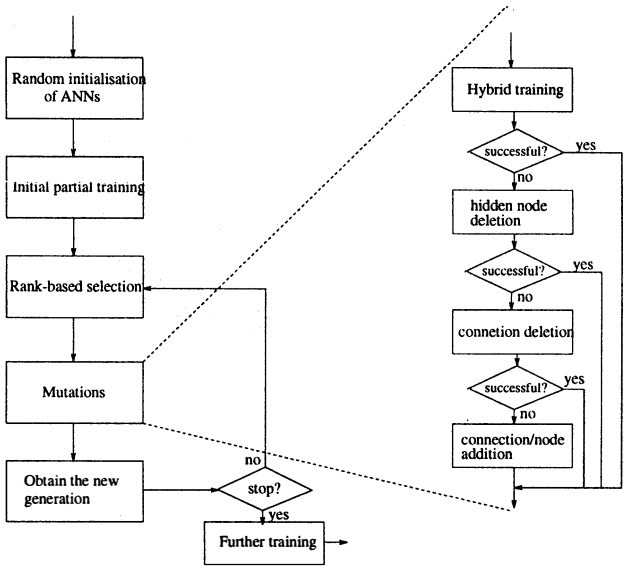


Figure 2: The main structure of EPNet.

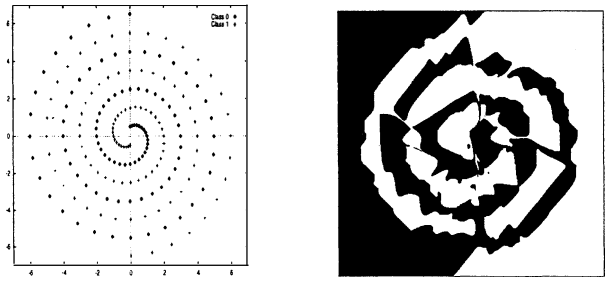
ANNs in a population is defined by two user-specified parameters. There is no guarantee that an ANN will converge to even a local optimum in a generation. Hence this training process is called partial training.

The five mutations are attempted sequentially. If one mutation leads to a better offspring, it is regarded as successful. No further mutation will be applied. In other words, only one of the five mutations will be applied each time. The order of deletion first and addition later encourages the evolution of compact ANNs. EPNet puts a lot of emphasis on the behavioural links between parents and offsprings. It deletes and adds connections probabilistically according to their importance in the ANN. Node deletion is done at random, but node addition is achieved through splitting an existing node.

In order to improve the generalisation ability of evolved ANNs, we have used two validation sets in EPNet. The first one is used to compute the fitness of each ANN. After the evolutionary process, all the ANNs in the final population will be trained further by MBP on the combined training and the first validation set. The ANN which performs the best on the second validation set will be the final output of EPNet.

We have tested our EPNet on a number of artificial and real-world data sets, including the parity problem [10, 12] of various sizes, four medical diagnosis problems [11], and the Australian credit card assessment problem [12]. We also tested our EPNet on another very difficult artificial problem — the two-spiral problem. The result of a typical run by EPNet on the

problem is shown in Figure 3.



(a) (b)

Figure 3: (a) The training data set of the two spiral problem with 194 examples. (b) The decision region formed by the ANN evolved by EPNet. There are 14 hidden nodes and 131 connections in the ANN.

5 Conclusion

This paper addresses the issue of automatic design of ANNs. It is argued that the evolutionary approach offers a very promising and competitive alternative to designing ANNs manually or by a constructive/pruning algorithm. The evolutionary design of ANNs is discussed in a general framework of EANNs, which provides a common basis for comparing and investigating various EANN models. A new evolutionary system for designing ANNs, EPNet, is described. Experimental studies were carried out for a number of problems. This paper reports the result on the well-known two-spiral problem. Our future work includes parallelisation of the EPNet system and improvement of the hybrid training since it is the mutation which consumes most of the computation time.

References

[1] J. D. Schaffer, D. Whitley, and L. J. Eshelman, "Combinations of genetic algorithms and neural networks: a survey of the state of the art," In D. Whitley and J. D. Schaffer, editors, *Proc. of the Int'l Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, pages 1–37. IEEE Computer Society Press, Los Alamitos, CA, 1992.

[2] X. Yao, "A review of evolutionary artificial neural networks," *International Journal of Intelligent Systems*, 8(4):539–567, 1993.

[3] X. Yao, "Evolutionary artificial neural networks," *International Journal of Neural Systems*, 4(3):203–222, 1993.

[4] X. Yao, "Evolutionary artificial neural networks," In A. Kent, J. G. Williams, and C. M. Hall, editors, *Encyclopedia of Computer Science and Technology*, volume 33, pages 137–170. Marcel Dekker Inc., New York, NY 10016, 1995.

[5] X. Yao, "Evolution of connectionist networks," In T. Dartnall, editor, *Preprints of the Int'l Symp. on AI, Reasoning & Creativity*, pages 49–52, Queensland, Australia, 1991. Griffith Univ.

[6] X. Yao, "The evolution of connectionist networks," In T. Dartnall, editor, *Artificial Intelligence and Creativity*, pages 233–243. Kluwer Academic Publishers, Dordrecht, 1994.

[7] P. J. Angeline, G. M. Sauders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Trans. on Neural Networks*, 5(1):54–65, 1994.

[8] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms," In J. D. Schaffer, editor, *Proc. of the Third Int'l Conf. on Genetic Algorithms and Their Applications*, pages 379–384. Morgan Kaufmann, San Mateo, CA, 1989.

[9] X. Yao, "An empirical study of genetic operators in genetic algorithms," *Microprocessing and Microprogramming*, 38:707–714, 1993.

[10] Y. Liu and X. Yao, "A population-based learning algorithm which learns both architectures and weights of neural networks," In *Proc. of ICYCS'95 Workshop on Soft Computing*, pages 29–38, Beijing 100088, P. R. China, 1995. National Research Center for Intelligent Computing Systems. To appear in *Chinese Journal of Advanced Software Research* (Allerton Press, Inc., New York, NY 10011), Vol. 3, No. 1, 1996.

[11] X. Yao and Y. Liu, "Evolving artificial neural networks for medical applications," In *Proc. of 1995 Australia-Korea Joint Workshop on Evolutionary Computation*, pages 1–16. KAIST, Taejon, Korea, September 1995.

[12] X. Yao and Y. Liu, "Evolutionary artificial neural networks that learn and generalise well," Submitted to *the 1996 IEEE International Conference on Neural Networks, Washington, DC, USA*. 3-6 June 1996.

Deadlock Resolution Using Hand-to-Hand Motion in Group Robots

Hidenori ISHIHARA, Toshio FUKUDA, Fumihito ARAI and Naohiro HIRAOKA

Department of Micro System Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-01, JAPAN
E-mail: ishihara@mein.nagoya-u.ac.jp

Abstract: This paper deals with an approach for resolving deadlock conditions in the case of carrying loads by Distributed Autonomous Robotic System (DARS). Deadlock condition sometimes occurs when a large number of robots move and carry loads autonomously. Therefore, we propose an algorithm to resolve deadlock conditions by cooperative Hand-to-Hand Motion based on only partial local information. Through the numerical simulation, we show the effectiveness of Hand-to-Hand Motion and consider influences of environment.

1. Introduction

Recently, robots become playing important roles in various fields such as industrial robots. As evolving the robotic systems, they become to be given complex tasks, and multiple robots are applied to the several situations. However, present robots have specific functions for some specific aims and lack extendibility or flexibility to different situation from the initial purposes at designing. The reasons are considered to be why robots are connected by the fixed order, and why the specific manager controls the whole robots centrally. Centralized control is the good control method in the minority group, but it is not suitable for the system which consists of large number of the robots because the manager must handle so much information of whole system that it is impossible for the manager to control the whole system. Recently, the distributed autonomous robotic system (DARS) has been proposed as a system to solve those problems[1]-[7]. The DARS aims at developing the intelligent system which consists of multiple autonomous robot existing independently each other, and which improves its efficiency according to the cooperation of constructing robots.

We have proposed the Cellular Robotic System (CEBOT), which consists of the single-functional units, namely cell, and can dynamically reconfigure its structure by the connection and detachment of cells to adapt to the given circumstances or the tasks[8]-[13]. The schematic drawing of CEBOT is shown in Fig. 1.

We have built and tested the CEBOT Mark I-V experimentally, and then we have evaluated the several system architecture such as the optimal reconstruction of the structure and the optimal distribution of knowledge allocation[10]-[13]. Through the previous studies, those results clarify that the moving cells are important for the reconstruction and other functions. One of those reasons is that the moving cell can play several roles such as to carry other non mobile cells, and to expand the working area according to the movement by its sensing capabilities. In the industrial fields, unmanned transfer vehicle or Automated Guided Vehicle was introduced as the mobile robots. In future, the increasing number of such mobile robots from now on can be considered to be used densely in a system. In such a system with mobile robot, the deadlock avoidance and resolution are important problems to control them smoothly. In the previous studies on the deadlock avoidance and resolution, the several methods of the single mobile robot have been reported, but only a few methods for multiple mobile robots have been proposed[14]-[16]. In the DARS with the mobile robots, when the mobile robots carry the loads as their task, the deadlock condition of carrying the loads is easily considered to happen often. Especially, when two robots with loads come across on the narrow path, it is difficult to solve such a deadlock condition. At that situation, it is considered to be effective in solving such a situation that each robot changes its direction to move according to handing over its load each other. Therefore, to resolve deadlock condition of carrying loads, we take notice of the motion to hand over the loads from one robot to the other, namely Hand-to-Hand Motion, and propose the Hand-to-Hand Motion strategy the as a method of the deadlock resolution.

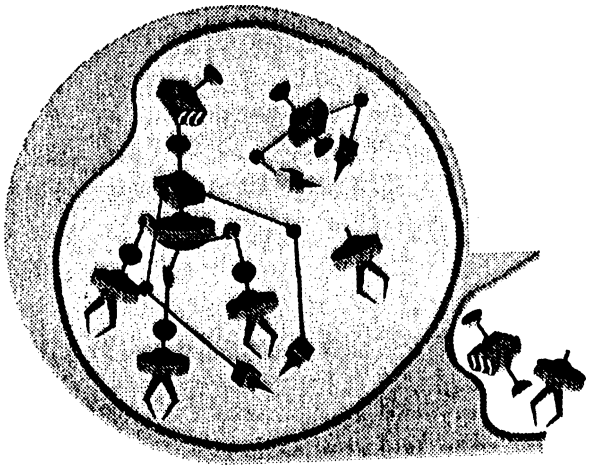


Fig. 1 Concept of Cellular Robotic System (CEBOT)

2. Deadlock Problem and Deadlock Resolution by Hand-to-Hand Motion

In the DARS including the CEBOT, since each robot does not have overall global information and decides its behavior according to the partially local information, there appear various problems of communication, cooperation, competition, and collision avoidance, etc. When we consider the task to carry loads, for example, several deadlock conditions such as carrying of load and moving of the robots occur. If each robot does not accept to perform cooperatively, robots can not clearly solve such deadlock situations. Because of that, the deadlock resolution requires for the robots to work the tasks cooperatively.

At the first, we define the environment and the tasks for the group robots. Several situations which the deadlock condition happens are considered, but since it is difficult to make the algorithm to solve all the deadlock condition, we propose a method to solve them in the limited situation. As the workspace, we assume the following environment shown in Fig. 2:

(1) workspace consists of narrows paths where robots can take over,

(2) there are two goals to be carried loads,

(3) there are two types of loads, and different types of loads are carried to the different goal.

In the defined environment shown in Fig. 2, several deadlock conditions are considered to happen.

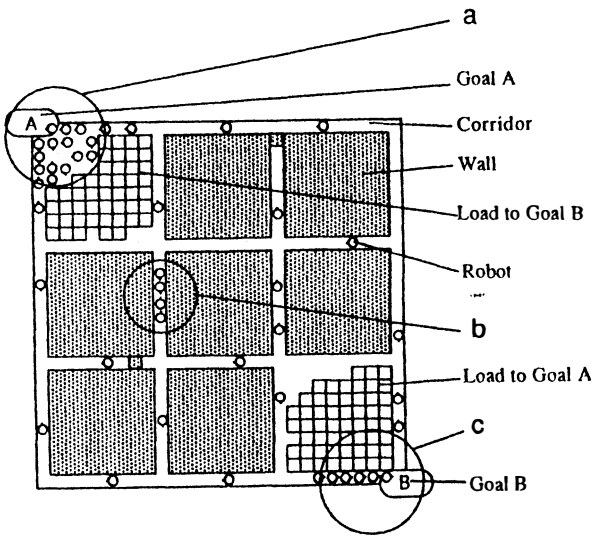


Fig. 2 Environment for the robots

First, take notice of the cases, 'a' or 'c' in Fig. 2 Under the condition 'a,' many robots move towards one goal, and robots around the goal get deadlocked by other coming robots even after they completed their tasks, because the goal A can be approached from more than two directions. When it takes much time to put the load off, a queue as the case 'c' is formed. In both cases, these queues include some robots not only with a load but also without a load. Within such a situation, the pileup load problem can be solved by handing the load over from one robot with load to the other robot without. Then, at the end of the queue, robots which finished to hand its load over to the other generally become able to move, and the deadlock of movement can be also solved.

Next; the case 'b' is to be considered. This case presents that facing two robots moving in the different direction on the narrow path are the obstacles each other on each way ahead. The reason is that each robot tends to go toward the opposite direction of their goals, and when they exchange their loads, the problem will be solved.

In those cases, handing over the loads is considered to be a way effective to solve the deadlock condition happening in such environment as in Fig. 2. Therefore, this study proposes the Hand-to-Hand Motion, which means the motion of handing over the loads each other, as a method of deadlock resolution. Figure 3 is the schematic drawing of hand-to-hand motion under the situation in Fig. 2.

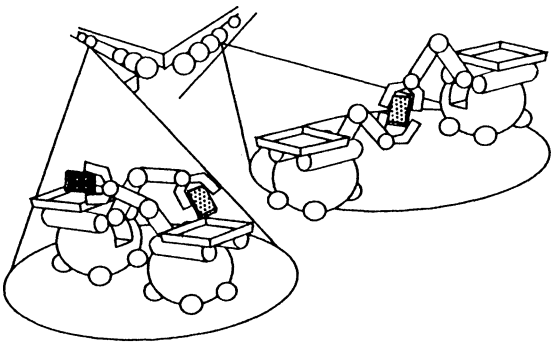


Fig. 3 Concept of Hand-to-Hand Motion

Except above three cases, the deadlock situation that two robots try to hand over the load to a robot at the same time is easily considered to occur. In such a situation, it is considered to be effective to form a one-to-one relation between robots to hand and to be handed abiding by some agreements (i.e. protocol) using communication[12]. Because of that, the algorithm of Hand-to-Hand Motion using communication is defined as followings, and is shown in Fig. 4:

[Step 1] a robot, namely robot A, detects the deadlock condition based on only the partial local information,

[Step 2] robot A look for the other robot, namely robot B, which can be transferred the loads,

[Step 3] robot A informs robot B of "I would like to transfer you the load,"

[Step 4] robot B receives these information,

[Step 5] robot B informs robot A of "I am ready,"

[Step 6] robot A receives these information,

[Step 7] robot A hands over his load to robot B.

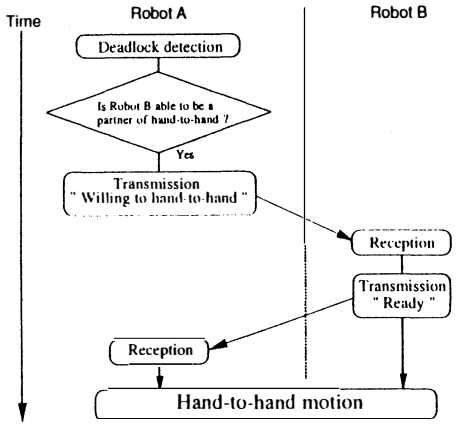


Fig. 4 Flowchart of Hand-to-Hand Motion

3. Numerical Simulation

Numerical simulation is carried out for three cases; the case using robots with hand-to-hand motion, the case without hand-to-hand motion but on the same basic rules, and the case based on heuristics. Detail is given as follows. Workspace is set up as in Fig. 2, and more complex situations such as in Fig. 6 are used for the simulation. Robots can not pass each other as we see from the figure. The task is to carry the lower right load to the upper left goal A and the upper left load to the lower right goal B.

3.1. Assumptions of Simulation

The following conditions are assumed to the simulations:
(1) each robot can moves toward 4 directions (x, -x, y and -y),
(2) each robot know the absolute position in the environment,
(3) each robot has the identification number (ID),
(4) each robot does not know the map of a whole environment, but know the position of goals,
(5) it takes no time to communicate between robots,
(6) each robot can recognize the intersection, and passes it by the rule.

3.2. Principle of Hand-to-Hand Motion

We define the rules of basic behavior as follows:
•A robot with load in deadlock conditions first looks for another robot to hand over the load to and then hands over to it.
•A robot without load performs random search.
•A robot reaching at the dead end moves along by the left wall.

Each robot sees whether any other robots are coming into the intersection when it comes just before it. If there are other robots, they all communicate each other the position, the IDs, the holding of load, and the direction to go. Then each of them calculates its order of priority to cross the intersection[13]. The order of priority is decided as follows:
(1) Robot crossing the intersection is of the highest priority.
(2) Robot which come across no robots in his way comes next.
(3) Robot with load does prior to that of without.
(4) Then the priority comes on robot of younger ID.

The idea to arrange this order of priority of crossing the interaction is the same as semaphore, but is deferent from it at the point that there's no guarantee of crossing because the flow is not a single direction.

The following equations are used as the detection of deadlock.

$$f_{dead} = \sum_{k=1}^{n} \left(d_{k-1} - d_k \right) \quad (1),$$

$$d_k = \left((x_k - x_{goal})^2 + (y_k - y_{goal})^2 \right)^{\frac{1}{2}} \quad (2),$$

where, d_k provides the distance between a robot and the goal, when (x_k, y_k) is the robot's position at that time, k steps, and (x_{goal}, y_{goal}) is the position of the goal. In the Eq. (1), $(d_{k-1} - d_k)$ represents the approached distance to the goal during the time from k steps to $(k-1)$ steps, and becomes minus as coming close to the goal. Under the deadlock condition, since robot does not move at all, the distance of movement within n steps is zero, that is $f = 0$, and so the deadlock can be detected. Since n is the memorable number of steps by each robot, we define it as n=5. This number is defined after repeated trial and error.

3.3. Heuristics

Considering from the environment of the simulation and the task, there appears a heuristic rule as following:
A robot in the area y-x<0 divided by the diagonal as shown in Fig. 5 moves toward x or y direction, and a robot in the area x-y<0 moves toward --x or -y direction. This rule makes a path just as an elliptical orbit through the two goals on the workspace.

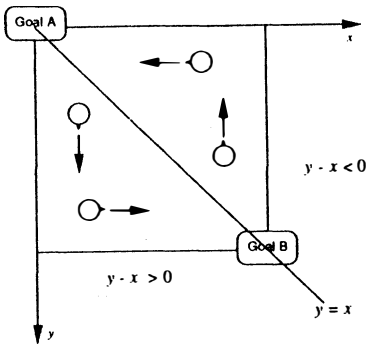
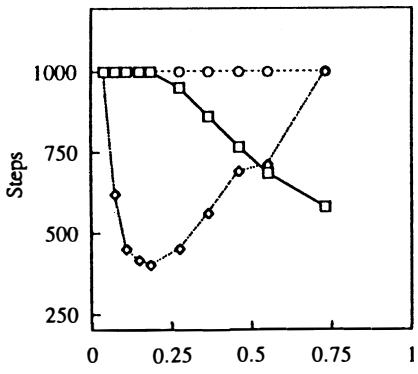
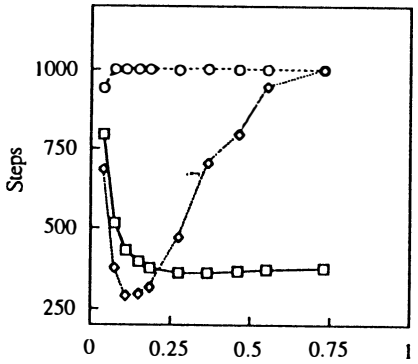


Fig. 5 Rules of Heuristics

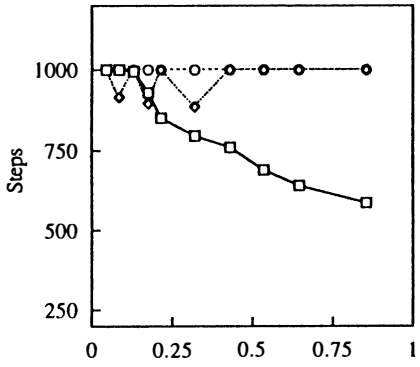


(a) Environment 1-1 (a kind of load)

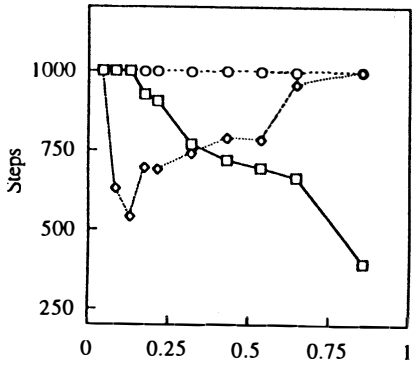


(b) Environment 1-2 (two kind of loads)

Fig. 7 Simulation results on Environment 1

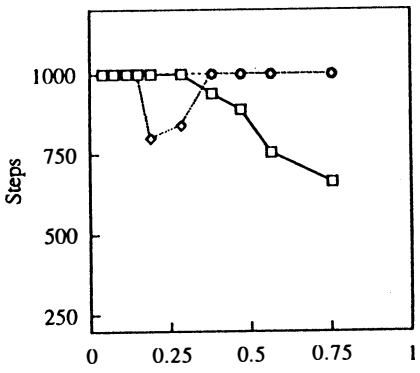


(a) Environment 2-1 (a kind of load)

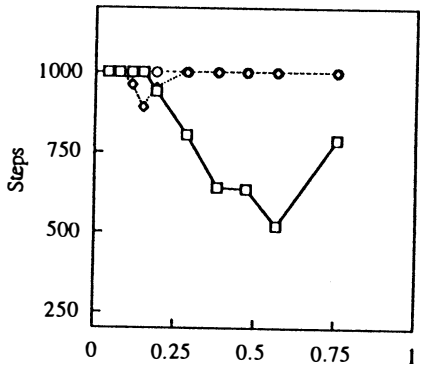


(b) Environment 2-2 (two kind of loads)

Fig. 8 Simulation results on Environment 2



(a) Environment 3-1 (a kind of load)



(b) Environment 3-2 (two kind of loads)

Fig. 9 Simulation results on Environment 3

Table 2 shows the number of steps of finishing the task at each environment when the number of robots is 150 (the highest occupation ratio of robot population). It can be said that the Hand-to-Hand motion is effective with the large number of robots. Without Hand-to-Hand motion, there appear the cases that the task cannot be completed even after 1000 steps passed as the number of robots increasing.

Table 2 steps completing the task

	Environment 1		Environment 2		Environment 3	
Kinds of Load	1	2	1	2	1	2
Without Hand-to-hand motion	1000	1000	1000	1000	1000	1000
With Heuristics	712	946	1000	963	1000	1000
With Hand-to-hand motion	688	371	643	666	758	522

In order to estimate the effect of each environment, three parameters of the environment, the number of corridor, average of corridor length, and average length of path to goal A, which are shown in Table 3, are defined as followings:

The corridor is the path form one corner of a block in the environment to the other corner of it along a

This rule makes the path of the robots one-way that makes the robots pass near the loads and goals, and then dissolves the deadlock condition such as the case 'a' in Fig. 2. For this reason, it is used as the comparison to the Hand-to-Hand motion we propose.

3.4. Simulation Environment

The environments of the simulations are three types of 26 x 26-square fields as in Fig. 6. Figure 6-(a) shows a simple environment distributing the path in a square, while Fig. 6-(b) is arranged to have some longer path, and Fig. 6-(c) some dead ends. The small blocks forming the obstruction making the path are a basic unit of representing the scale of the environment.

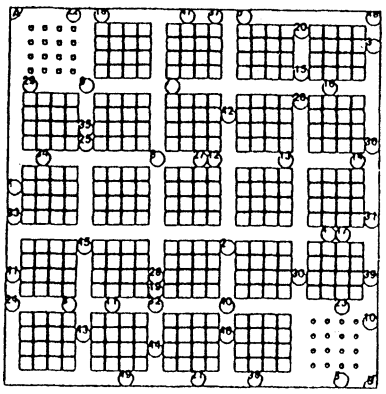
The evaluation of simulation results is given by comparison of the working efficiencies as changing the number of robots in each workspace. The efficiency is estimated by the number of steps to complete the task of carrying loads to the goal. The parameters are shown in Table 1. In Table 1, six kinds of environments are defined. Environment 1-1 and Environment 1-2 use the same map, Fig. 6-(a), but there is a kind of loads in the environment 1-1 and 2 kinds of loads in the environment 1-2. As same way, two cases are set in both environments 2 and 3. In the environment including two kinds of loads, the deadlock condition is considered to happen more than the other including a kind of loads.

Figures 7, 8 and 9 show the results of simulation. The horizontal axis in Figs. 7, 8 and 9, the Occupation ratio of Robot population to Space, is obtained by (number of robots) / (number of squares of the path). At the simulation results as shown in Figs. 7, 8 and 9, it can be stated that the Hand-to-Hand motion strategy solves the deadlock problem by keeping the ability of carrying the loads even when it becomes harder to move as the robot population increases. While with the small number of robots, heuristic leads the better efficiency.

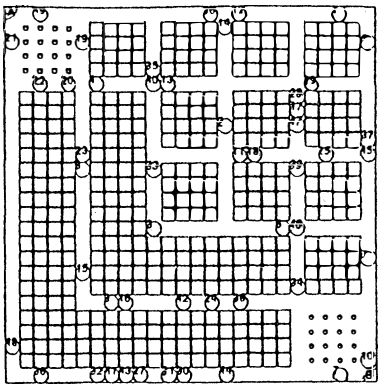
The most remarkable change is obtained by the environment 1 with 2 kinds of loads as shown in Fig. 7. This is the most general workspace though it's rather simple, and heuristics is effective especially under this environment and the best efficient value with this environment is obtained by the case of heuristics. This result has some relations to the rule that the Hand-to-Hand motion is performed after the detection of deadlock. More over, even under the same environment 1, when it is the case with a kind of load, random search costs wastefully. With the environment 3, extra motion of advance along by the left wall at the dead end, the characteristic of its statistic data may be of slightly different kind, event though it can be concluded that the noticeable difference of efficiencies indicates the effectiveness of the Hand-to-Hand motion.

Table 1 Parameter for the simulations

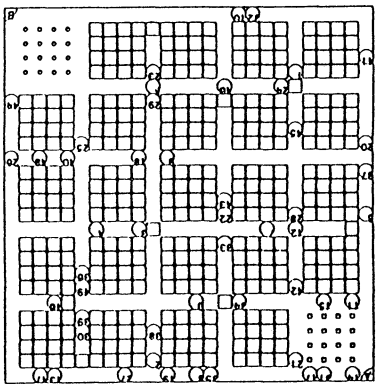
Environment	1_1, 2_1, 3_1	1_2, 2_2, 3_2
Work space	26×26	26×26
Kinds of Load	1	2
Number of Load	100	50×2
Robot population	10, 20, 30, 40, 50, 75, 100, 125, 150, 200	10, 20, 30, 40, 50, 75, 100, 125, 150



(a) Environment 1 for the simulation:
26 x 26-square without long paths and dead ends



(b) Environment 2 for the simulation:
26 x 26-square with long paths



(c) Environment 3 for the simulation:
26 x 26-square with dead ends

Fig, 6 Environment for the simulation

side, while the average length of path to goal A is a value taking the average of the length of the path from any positions to the goal A. The average of corridor length and the average length of path to goal A are calculated based on the basic units in Fig. 6.

Table 3 Environmental Parameters

	Environment 1	Environment 2	Environment 3
Number of Corridor	60	40	49 44
Average of Corridor length	4.0	5.25	4.27 4.20
Average length of Path to Goal	25 50	25 44	26 53

In Table 3, the difference between environment.1 and 2 is remarkable. It may be stated that there is a direct proportional interrelation between the average of corridor length and the working efficiency. The number of corridor is closely related with this average of corridor length, and the number of intersections inevitably becomes small, when, as environment 2, the average of corridor length is long and the number of corridor is little. Thus it becomes difficult to solve the deadlock problems due to the slow movement of line. This causes the inefficiency in the case by heuristics which is without Hand-to-Hand motion. On the contrary, as environment 1 with many short corridors and intersections, less deadlock problems are caused and the resolution can be gotten easier.

From the comparison of environment 1 and 3, it is found that environment 3 has smaller number of valid corridors as it has some dead ends. According to this, the efficiency also becomes lower. From this comparison, there also seems to be direct proportional interrelation between the length of path to goal and the efficiency. If we consider this together with the previous comparison, at the first sight, it seems to be less influence on the efficiency. However, the difference between the case of heuristics and the case with Hand-to-Hand motion is considerable, the value of heuristics is double of the value of the Hand-to-Hand motion. Then it can be read from the Table 2 that the average of the length of the path influences to the heuristics.

4. Conclusions

We have proposed the new method of deadlock solution using the Hand-to-Hand Motion and have done numerical simulation under several environments. Through the consideration of the simulation results, it can be concluded that the Hand-to-Hand Motion to the carrying loads by autonomous robots is effective for both the deadlock resolution and the earlier completion for the task with multiple robots.

References

[1]J.Wang and S.Premvuti, "Fully Distributed Traffic Regulation and Control for Multiple Autonomous Mobile Robots Operating in Discrete Space," Proceedings of International Symposium on Distributed Autonomous Robotic Systems (DARS'94), pp.134-161, (1994)

[2]R.A.Brooks, "A Robust Layered Control System For A Mobile Robot," IEEE Journal of Robotics and Automation, Vol.RA-2, No.1, pp.14-23, (1986)

[3]L.E.Parker, "Designing Control Laws for Cooperative Agent Teams," Proc. of International Conference on Robotics and Automation (ICRA'93), pp.582-587, (1993)

[4]A.Agah and G.A.Bekey, "A Novel Cognate Architecture for Simulated Robots in an Artificial World," Proc. of ICRA'94, pp.2309-2314, (1994)

[5]F.Hara and S.Ichikawa, "Effects of Population Size in Multi-Robots Cooperative Behaviors," Proc. of DARS'92, Wako, Saitama, Japan, Sep.21-22, pp.3-9, (1992)

[6] Y.Ohmori, Y.Nakauchi and Y.Anzai, "Task Allocation Algorithm for Cooperation of Distributed Autonomous Mobile Robots," Proc. of DARS'92, Wako, Saitama, Japan, Sep.21-22, pp.159-165, (1992)

[7] J. Wang, "On Sign-board Based Inter-Robot Communication in Distributed Robotic Systems," Proc. of ICRA'94, pp.1045-1050, (1994)

[8] T.Fukuda and T.Ueyama, "Cellular Robotics and Micro Robotic Systems," World Scientific Series in Robotic and Automated Systems Vol.10, World Scientific, pp.1-47, (1994)

[9] T. Fukuda, " Structure Decision Method for Self Organizing Robots based on Cell Structure-CEBOT," Proc. ICRA'89, pp. 695-700, (1989)

[10] T. Fukuda, et. al, "Communication Method of Cellular Robotics CEBOT as a Selforganizing Robotic System," Proc. of International Workshop on Intelligent Robots and Systems '89 (IROS'89), Tsukuba, Japan, Sept.4-6, pp.291-296, (1989)

[11] T. Fukuda, et. al, "Concept of Cellular Robotic System (CEBOT) and Basic Strategies for its Realization," Computers Elect. Eng. Vol. 18, No. 2, Pergamon Press, pp. 11-39, (1992)

[12] T. Fukuda et. al, "Optimization of Group Behavior on Cellular Robotic System in Dynamic Environment," Proc. of ICRA'94, pp.1027-1032, (1994)

[13] A.-H Cai, T.Fukuda, et al., "Hierarchical Control Architecture for Cellular Robotic System -Simulations and Experiments-," Proc. of ICRA'95, Nagoya, Japan, pp.1191-1196, (1995)

[14] A. Matsumoto, H. Asama, et. al, "Communication in the Autonomous and Decentralized Robot System ACTRESS," Proc. of IROS'90, July 3-6, Tsuchiura, Japan, pp.835-840, (1990)

[15] S.Yuta and S.Premvuti, "Coordinating Autonomous and Centralized Decision Making to Achieve Cooperative Behaviors Between Multiple Mobile Robots," Proc. of DARS'92, Wako, Saitama, Japan, Sep.21-22, pp.173-181, (1992)

[16] J. Wang, "Deadlock Detection and Resolution in Distributed Robotic System," Proc. of DARS'92, Sept.21-22, Wako, Japan, pp.93-101, (1992)