PROCEEDINGS OF THE EIGHTH INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB 8th '03)

Vol.2

Jan. 24-Jan. 26, 2003
B-Com Plaza, Beppu, Oita, JAPAN

Editors : Masanori Sugisaka and Hiroshi Tanaka
ISBN4-9900462-3-4

Proceedings of The Eighth International Symposium on

ARTIFICIAL LIFE AND ROBOTICS

(AROB 8th '03)

for Cognitive and Behavioral Intelligent Artificial Liferobot-2

January 24-26, 2003
B-Con Plaza, Beppu, Oita, Japan

Editors: Masanori Sugisaka and Hiroshi Tanaka

THE EIGHTH INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB 8th '03)

ORGANIZED BY

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

CO-SPONSORED BY

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

COOPERATED BY

ISICE, IEICE, IEEE Japan Council, JARA

SUPPORTED BY

Ministry of Education, Culture, Sports, Science and Technology, Japanese Government
Kyushu Bureau of Economy, Trade and Industry
Oita Prefecture
OITA CITY
Beppu City
Kyodo News
Jiji Press Ltd.
OITA GODO SHINBUNSYA
The Asahi shimbun
The Mainichi Newspapers
THE YOMIURI SHIMBUN
The Nishinippon Newspaper Co.
THE NIKKAN KOGYO SHINBUN. LTD
Nihon Keizai Shimbun, INC.
NHK (Japan Broadcasting Corporation)
OITA BROADCASTING SYSTEM, INC.
Television Oita System
Oita Asahi Broadcasting Co. Ltd.
Oita Prefectural Industrial Organization
Oita System Control Research Society

ADVISORY COMMITTE CHAIRMAN

F. Harashima (Tokyo Denki University, Japan)

GENERAL CHAIRMAN

M. Sugisaka (Oita University, Japan)

CO-GENERAL CHAIRMAN (PROGRAM)

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

CO-CHAIRMAN

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

ADVISORY COMMITTEE

F. Harashima (Tokyo Denki University, Japan)
H.Kimura (The University of Tokyo, Japan)
T.Fukuda (Nagoya University, Japan)
S.Ueno (Kyoto University, Japan)

INTERNATIONAL ORGANIZING COMMITTEE

K.Aihara (The University of Tokyo, Japan)
W.B.Arthur (Santa Fe Institute, USA)
W.Banzhaf (University of Dortmund, Germany)
C.Barrett (Los Alamos National Laboratory, USA)
M.Bedau (Reed College, USA)
Z.Bubnicki (Wroclaw University of Technology, Poland)
J.L.Casti (Santa Fe Institute, USA)
H.S.Cho (KAIST, Korea)
J.M.Epstein (The Brookings Institution, USA)
T.Fukuda (Nagoya University, Japan)
H.Hashimoto (The University of Tokyo, Japan)
D.J.G.James (Coventry University, UK)
S.Kauffman (Santa Fe Institute, USA)
K.Kosuge (Tohoku University, Japan)
C.G.Langton (Santa Fe Institute, USA)
M.H.Lee (Pusan National University, Korea)
J.J.Lee (KAIST, Korea)
M.Nakamura (Saga University, Japan)
G.Obinata (Nagoya University, Japan)
S.Rasmussen (Santa Fe Institute, USA)
T.S.Ray (University of Okulahoma , USA)
M.Sugisaka (Oita University, Japan) (Chairman)
H.Tanaka (Tokyo Medical & Dental University, Japan)
C.Taylor (University of California-Los Angeles, USA)
K.Tsuchiya (Kyoto University, Japan)
G.Wang (Tsinghua University, China)
W.R.Wells (University of Nevada-Las Vegas, USA)

Y.G.Zhang (Academia Sinica, China)

INTERNATIONAL PROGRAM COMMITTEE

K.Abe (Tohoku University, Japan)
K.Aihara (The University of Tokyo, Japan) (Co-chairman)
T.Arita (Nagoya University, Japan)
H.Asama (RIKEN, Japan)
M.Bedau (Reed College, USA)
R.Belew (University of California-San Diego, USA)
Z.Bubnicki (Wroclaw University of Technology, Poland)
T.Christaller (Fraunhofer Institute for Autonomous intelligent Systems-AiS)
C.S.Han (Hanyang University, Korea)
I.Harvey (University of Sussex, UK)
H.Hashimoto (The University of Tokyo, Japan)(Co-chairman)
K.Hirasawa (Kyusyu University, Japan)
H.Hirayama (Asahikawa Medical College, Japan)
N.Honma (Tohoku University, Japan)
T.Ikegami (The University of Tokyo, Japan)
H.Inooka (Tohoku University, Japan)
K.Ito (Tokyo Institute of Technology, Japan)
J.Johnson (The Open University, UK)
Y.Kakazu (Hokkaido University, Japan)
O.Katai (Kyoto University, Japan)
S.Kawaji (Kumamoto University, Japan)
S.Kawata (Tokyo Metropolitan University, Japan)
S.Kitamura (Kobe University, Japan)
T.Kitazoe (Miyazaki University, Japan)
S.Kumagai (Osaka University, Japan)
J.M.Lee (Pusan National University, Korea)
C.G.Looney (University of Nevada-Reno, USA)
H.H.Lund (University of Southern Denmark, Denmark)
H.Miyagi (Ryukyu University, Japan)
M.Nakamura (Saga University, Japan)
R.Nakatsu (KWANSEI GAKUIN University, Japan)
S.Omatsu (University of Osaka Prefecture, Japan)
M.Okamoto (Kyusyu University, Japan)
R.Pfeifer (University of Zurich-Irchel, Switzerland)
T.S.Ray (University of Oklahoma, USA) (Co- chairman)
T.Sawaragi (Kyoto University, Japan)
T.Shibata (MITI, MEL, Japan)
K.Shimohara (ATR, Japan)
M.Sugisaka (Oita University, Japan)
H.Tanaka (Tokyo Medical & Dental University, Japan) (Chairman)
M.Tanaka-Yamawaki (Miyazaki University, Japan)
N.Tosa (ATR, Japan)
K.Ueda (The University of Tokyo, Japan)
K.Uosaki (Tottori University, Japan)
H.Wakamatsu (Tokyo Medical & Dental University, Japan)

K.Watanabe (Saga University, Japan)
M.Yano (Tohoku University, Japan)

LOCAL ARRANGEMENT COMMITTEE

T.Kubik (Oita University, Japan)
K.B.Kubik (Oita University, Japan)
A.Loukianov (Oita University, Japan)
K.Shibata (Oita University, Japan)
M.Sugisaka (Oita University, Japan) (Chairman)
Y.Suzuki (Tokyo Medical & Dental University, Japan)
X.Wang (Oita Institute of Technology, Japan)
F.Xiongfeng (Oita University, Japan)

HISTORY

This symposium was founded in 1996 by the support of Science and International Affairs Bureau, Ministry of Education, Science, Sports and Culture, Japanese Government. Since then, this symposium was held every year at B-Con Plaza, Beppu, Oita, Japan except Tokyo, Japan (AROB 6th '02). The Eighth symposium will be held on 24-26 January 2003, at B-Con Plaza, Beppu, Oita, Japan. This symposium invites you all to discuss development of new technologies concerning Artificial Life and Robotics based on simulation and hardware in twenty first century.

OBJECTIVE

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

TOPICS

Artificial brain research
Artificial intelligence
Artificial life
Artificial living
Artificial mind research
Bioinformatics chaos
Brain science
Cognitive science evolutionary computations
Complexity
Computer graphics
DNA computing
Fuzzy control
Genetic algorithms
Human-machine cooperative systems

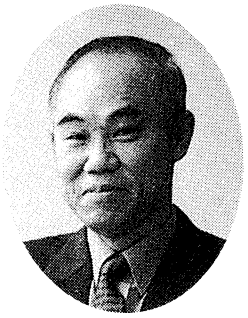
Human-welfare robotics
Innovative computations
Intelligent control and modeling
Micromachines
Micro-robot world cup soccer tournament
Mobile vehicles
Molecular biology
Multi-agent systems
Nano-biology
Nano-robotics
Neural networks
Neurocomputers
Neurocomputing technologies and its application for hardware
Pattern recognition
Robotics
Robust virtual engineering
Virtual reality

COPYRIGHTS

Accepted papers will be published in the proceeding of AROB and some of high quality papers in the proceeding will be requested to re-submit fir the consideration o publication in an international journal ARTIFICIAL LIFE AND ROBOTICS (Springer) and APPLIED MATHEMATICS ANDCOMPUTATION (North-Holland). All correspondence related to the symposium should be addressed to AROB Secretariat

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

MESSAGE



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

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

This symposium is supported by Monkasho and other institutions. The symposium invites you to discuss the development of new technologies in the 21st century, concerning Artificial Life and Robotics, based on simulation and hardware.

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

Masanori Sugisaka
M. Sugisaka

December 20, 2002

MESSAGE

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

On behalf of the program committee, it is truly my great honor to invite you all to the Eighth International Symposium on Artificial Life and Robotics (AROB 8th '03). This symposium is made possible owing to the cooperation of Oita University and Santa Fe Institute. We are also debt to Japanese academic associations such as SICE, RSJ, and several private companies. I would like to express my sincere thanks to all of those who make this symposium possible.

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

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

I am looking forward to meeting you in Beppu, Oita.

Hiroshi Tanaka

H. Tanaka

December 20, 2002

TIME TABLE

	RoomA	RoomB	RoomC	RoomD
1/23(Thu.) 8:00				
13:00	Registration (Registration Desk)			
17:00	Welcome Party(in Hotel Arthur 10th Floor)			
1/24(Fri.) 8:00	Registration (Registration Desk)			
9:00	Chair A. Loukianov GS1(6)	Chair T. Kitazoe IS1(6)	Chair W. Wells IPS1(3)	
10:30	Coffee Break			
10:40				Opening Ceremony
11:00				Plenary Talk Chair J. Casti PT1 M.Sugisaka
12:00	Lunch			
13:00	Chair T. Arita GS2(5)	Chair J. Johnson GS3(5)	Chair H. Kang GS4(5)	
14:00 14:15	Chair X. Feng GS5(4)	Chair A. Ohuchi GS6(3)	Chair H. Hirayama GS19(4)	
15:15	Coffee Break			
15:30	Chair J. Casti GS7(4)	Chair T. Hoya GS8(4)	Chair T. Kubik GS9(4)	
16:30	Chair K. Shibata GS10(4)	Chair H. Suzuki IS3(4)	Chair H.Tanaka GS11(3)	
17:30				

GS: General Session IS: Invited Session IPS: Invited Professor's Session

GS1 Robotics-I
GS2 Multi-agent systems
GS3 Bioinformatics, Molecular Biology & Brain Science
GS4 Intelligent Control and Modeling-I
GS5 Genetic Algorithms & Evolutionary Computations-I
GS6 Artificial Intelligence
GS7 Virtual Reality & Economic Data Mining
GS8 Computer and Robot Vision-I
GS9 Fuzzy Control
GS10 Reinforcement Learning-I
GS11 Image Processing & Pattern Recognition-I
GS12 Reinforcement Learning-II
GS13 Image Processing & Pattern Recognition-II
GS14 Micro-Robot World Cup Soccer Tournament & Micromachines
GS15 Robotics-II
GS16 Mobile Vehicle-I
GS17 Neural Networks
GS18 Intelligent Control and Modeling-II
GS19 Genetic Algorithms & Evolutionary Computations-II
GS20 Computer and Robot Vision-II
GS21 Mobile Vehicle-II
GS22 Artificial Life
GS23 Computer & Internet Security
IS1 Robot Control and Image Processing
IS2 Neural Network Applications
IS3 Evolution of codes, Behaviors, and networks
IS4 Artificial Chemistry-I
IS5 Artificial Chemistry-II
IS6 Welfare and Medical Engineering
IS7 Nonlinear Modeling and its Applications
IS8 Machine Intelligence and Robotic Control
IS9 Neural Network
IS10 Genetic Algorithms for Engineering Optimization
IS11 Interaction and Intelligence
IS12 Artificial Brain
IS13 Artificial Mind
IS14 Genetic Algorithms for Production and Distribution
IS15 Artificial Life and Application
IS16 Soft Robotics and Information
IS17 Applications for Intelligent Control
IPS1 Invited Professor's Session (I)
IPS2 Invited Professor's Session (II)
IPS3 Invited Professor's Session (III)

	RoomA	RoomB	RoomC	RoomD
1/25(Sat.) 8:00	Registration (Registration Desk)			
9:00	Chair H. Suzuki IS4(4)	Chair J.M. Lee GS12(4)	Chair K. Ida IS10(4)	
10:00	Chair H. Suzuki IS5(3)	Chair M. Nakamura GS14(3)	Chair T. Ishimatsu IS6(3)	
10:45	Coffee Break			
11:00				Plenary Talk Chair Y. Zhang PT2 T. Fukuda
12:00	Lunch			
13:00	Chair S. Omatsu GS15(6)	Chair C. Zhang GS17(5)	Chair Z. Bubnicki IPS2(2)	
14:30	Coffee Break			
14:45	Chair M. Gen IS14(5)	Chair H. Sayama GS16(6)	Chair I. Yoshihara IS9(5)	
16:00		Chair H. Hashimoto IS11(5)	Chair A. Buller IS12(5)	
16:15	Chair X. Wang GS13(3)			
17:00				
17:15				
18:00	AROB Award Ceremony (Chair: K. Watanabe)			
18:20	Banquet (Hotel Shiragiku)			
20:30				

GS: General Session IS: Invited Session IPS: Invited Professor's Session

GS1 Robotics-I
GS2 Multi-agent systems
GS3 Bioinformatics, Molecular Biology & Brain Science
GS4 Intelligent Control and Modeling-I
GS5 Genetic Algorithms & Evolutionary Computations-I
GS6 Artificial Intelligence
GS7 Virtual Reality & Economic Data Mining
GS8 Computer and Robot Vision-I
GS9 Fuzzy Control
GS10 Reinforcement Learning-I
GS11 Image Processing & Pattern Recognition-I
GS12 Reinforcement Learning-II
GS13 Image Processing & Pattern Recognition-II
GS14 Micro-Robot World Cup Soccer Tournament & Micromachines
GS15 Robotics-II
GS16 Mobile Vehicle-I
GS17 Neural Networks
GS18 Intelligent Control and Modeling-II
GS19 Genetic Algorithms & Evolutionary Computations-II
GS20 Computer and Robot Vision-II
GS21 Mobile Vehicle-II
GS22 Artificial Life
GS23 Computer & Internet Security
IS1 Robot Control and Image Processing
IS2 Neural Network Applications
IS3 Evolution of codes, Behaviors, and networks
IS4 Artificial Chemistry-I
IS5 Artificial Chemistry-II
IS6 Welfare and Medical Engineering
IS7 Nonlinear Modeling and its Applications
IS8 Machine Intelligence and Robotic Control
IS9 Neural Network
IS10 Genetic Algorithms for Engineering Optimization
IS11 Interaction and Intelligence
IS12 Artificial Brain
IS13 Artificial Mind
IS14 Genetic Algorithms for Production and Distribution
IS15 Artificial Life and Application
IS16 Soft Robotics and Information
IS17 Applications for Intelligent Control
IPS1 Invited Professor's Session (I)
IPS2 Invited Professor's Session (II)
IPS3 Invited Professor's Session (III)

	RoomA	RoomB	RoomC	RoomD
1/26(Sun.) 8:00	Registration (Registration Desk)			
9:00	Chair K. Shimohara IS13(5)	Chair K. Aihara IS7(5)	Chair Y. Zhang IPS3(2)	
10:30				
10:45	Chair B. Price GS18(4)	Chair S. Omatsu IS2(4)	Chair J. Wang GS20(5)	
12:00				
13:00	Chair H. H. Lund GS21(5)	Chair K. Watanabe IS8(5)	Chair T.X. Yan GS22(3)	
13:45			Chair P. Sapaty GS23(3)	
14:15		Chair T. Yamamoto IS16(4)		
14:30	Chair K-B. Sim IS15(4)		Chair J.J. Lee IS17(4)	
15:30	Farewell Party (Room D)			
16:30				

GS: General Session IS: Invited Session IPS: Invited Professor's Session

GS1 Robotics-I
GS2 Multi-agent systems
GS3 Bioinformatics, Molecular Biology & Brain Science
GS4 Intelligent Control and Modeling-I
GS5 Genetic Algorithms & Evolutionary Computations-I
GS6 Artificial Intelligence
GS7 Virtual Reality & Economic Data Mining
GS8 Computer and Robot Vision-I
GS9 Fuzzy Control
GS10 Reinforcement Learning-I
GS11 Image Processing & Pattern Recognition-I
GS12 Reinforcement Learning-II
GS13 Image Processing & Pattern Recognition-II
GS14 Micro-Robot World Cup Soccer Tournament & Micromachines
GS15 Robotics-II
GS16 Mobile Vehicle-I
GS17 Neural Networks
GS18 Intelligent Control and Modeling-II
GS19 Genetic Algorithms & Evolutionary Computations-II
GS20 Computer and Robot Vision-II
GS21 Mobile Vehicle-II
GS22 Artificial Life
GS23 Computer & Internet Security
IS1 Robot Control and Image Processing
IS2 Neural Network Applications
IS3 Evolution of codes, Behaviors, and networks
IS4 Artificial Chemistry-I
IS5 Artificial Chemistry-II
IS6 Welfare and Medical Engineering
IS7 Nonlinear Modeling and its Applications
IS8 Machine Intelligence and Robotic Control
IS9 Neural Network
IS10 Genetic Algorithms for Engineering Optimization
IS11 Interaction and Intelligence
IS12 Artificial Brain
IS13 Artificial Mind
IS14 Genetic Algorithms for Production and Distribution
IS15 Artificial Life and Application
IS16 Soft Robotics and Information
IS17 Applications for Intelligent Control
IPS1 Invited Professor's Session (I)
IPS2 Invited Professor's Session (II)
IPS3 Invited Professor's Session (III)

TECHNICAL PAPER INDEX

January 24 (Friday)

Room D

11:00~12:00 PT-1 Plenary Talk1
Chair J.Casti

January 25 (Saturday)

11:00~12:00 PT-2 Plenary Talk2
Chair Y. Zhang

January 24 (Friday)

Room A

9:00~10:30 GS1 Robotics-1
Chair: A. Loukianov (Oita University, Japan)

13:00~14:15 GS2 Multi-agent systems
Chair: T. Arita (Nagoya University, Japan)

14:15~15:15 GS5 Genetic Algorithms & Evolutionary Computation- I
Chair: X. Feng (Oita University, Japan)

15:30~16:30 GS7 Virtual Reality & Economic Data Mining
Chair: J. Casti (SFI, USA)

16:30~17:10 GS10 Reinforcement Learning- I
Chair: K. Shibata (Oita University, Japan)

Room B

9:00~10:30 IS1 Robot Control and Image Processing
Chair: T. Kitazoe (Miyazaki University, Japan)
Co-Chair: M. Tabuse (Miyazaki University, Japan)

13:00~14:15 GS3 Bioinformatics, Molecular Biology & Brain Science
Chair: J. Johnson (Open University, UK)

14:15~15:00 GS6 Artificial Intelligence
Chair: A. Ohuchi (Hokkaido University, Japan)

15:30~16:30 GS8 Computer and Robot Vision- I
Chair: T. Hoya (BSI RIKEN, Japan)

16:30~17:30 IS3 Evolution of codes, Behaviors, and network
Chair: H. Suzuki (ATR Human Information Science Labs., Japan)
Co-Chair: K. Shimohara (Kyoto University, ATR, Japan)

Room C

9:00~10:30 ISP1 Invited Professor's Session(I)
Chair: W. Wells (University of Nevada-Las Vegas, USA)

13:00~14:15 GS4 Intelligent Control and Modeling- I
Chair: H. Kang (Chung- Ang University, Korea)

14:15~15:15 GS19 Genetic Algorithms & Evolutionary Computations- II
Chair: H. Hirayama (Asahikawa Medical College, Japan)

15:30~16:30 GS9 Fuzzy Control
Chair: T. Kubik (Oita University, Japan)

16:30~17:30 GS11 Image Processing & Pattern Recognition- I
Chair: H. Tanaka (Tokyo Medical and Dental University, Japan)

January 25 (Saturday)

Room A

9:00~10:00 IS4 Artificial Chemistry- I

Chair: H. Suzuki (ATR Human Information Science Labs, Japan)
Co-Chair: J-Q. Liu (ATR Human Information Science Labs, Japan)

10:00~10:45 IS5 Artificial Chemistry-Ⅱ
Chair: H. Suzuki (ATR Human Information Science Labs, Japan)
Co-Chair: J-Q. Liu (ATR Human Information Science Labs, Japan)

13:00~14:30 GS15 Robotics-Ⅱ
Chair: S. Omatsu (Osaka Prefecture University, Japan)

14:45~16:00 IS14 Genetic Algorithms for Production and Distribution
Chair: M. Gen (Ashikaga Institute of Technology, Japan)
Co-Chair: M. Sasaki (Ashikaga Institute of Technology, Japan)

16:00~17:15 GS13 Image Processing & Pattern Recognition-II
Chair: X. Wang (Oita Institute of Technology, Japan)

Room B

9:00~10:00 GS12 Reinforcement Learning-Ⅱ
Chair: J.M. Lee (Pusan National University, Korea)

10:00~10:45 GS14 Micro-Robot World Cup Soccer Tournament & Micromachines
Chair: M. Nakamura (Saga University, Japan)

13:00~14:30 GS17 Neural Networks
Chair: C. Zhang (Tsinghua University, P.R.China)

14:45~16:00 GS16 Mobile Vehicle- I
Chair: H. Sayama (University of Electro-Communication, Japan)

16:00~17:15 IS11 Interaction and Intelligence

Chair: H. Hashimoto (The University of Tokyo, Japan)

Room C

9:00~10:00 IS10 Genetic Algorithms for Engineering Optimization
Chair: K. Ida (Maebashi Institute of Technology, Japan)
Co-Chair: T. Yokota (Ashikaga Institute of Technology, Japan)

10:00~10:45 IS6 Welfare and Medical Engineering
Chair: T. Ishimatsu (Nagasaki University, Japan)

13:00~14:30 IPS2 Invited Professor's Session(Ⅱ)
Chair: Z. Bubnicki (Wroclaw University of Technology, Poland)

14:45~16:00 IS9 Neural Network
Chair: I. Yoshihara (Miyazaki University, Japan)
Co-Chair: M. Yasunaga (University of Tsukuba)

16:00~17:15 IS12 Artificial Brain

Chair: A. Buller (ATR Human Information Science Labs, Japan)
Co-Chair: K. Shimohara (ATR Human Information Science Labs, Japan)

January 26 (Sunday)

Room A

9:00~10:30 IS13 Artificial Mind
Chair: K. Shimohara (ATR Human Information Science Labs, Japan)
Co-Chair: H. Kozima (Communications Research Laboratory, Japan)

14:30~15:30 IS15 Artificial Life and Application
Chair: K.B. Sim (Chung-Ang University, Korea)

Room B

9:00~10:30 IS7 Nonlinear Modeling and its Applications
Chair: K. Aihara (The University of Tokyo, Japan)
Co-Chair: H. Suzuki (The University of Tokyo, Japan)

10:45~12:00 IS2 Neural Network Applications

Chair: S. Omatsu (Osaka Prefecture University, Japan)
Co-Chair: A. Selamat (Osaka Prefecture University, Japan)

13:00~14:15 IS8 Machine Intelligence and Robotic Control
Chair: K. Watanabe (Saga University, Japan)
Co-Chair: K. Izumi (Saga University, Japan)

14:15~15:30 IS16 Soft Robotics and Information
Chair: T. Yamamoto (University of the Ryukyus, Japan)
Co-Chair: H. Kinjo (University of the Ryukyus, Japan)

Room C

9:00~10:30 IPS3 Invited Professor's Session(Ⅲ)
Chair: Y. Zhang

10:45~12:00 GS20 Computer and Robot Vision-Ⅱ
Chair: J. Wang (Oita University, Japan)

13:00~13:45 GS22 Artificial Life
Chair: T.X. Yan (University of Science & Technology Beijing, P.R.China)

13:45~14:30 GS23 Computer & Internet Security
Chair: P. Sapaty (Oita University, Japan)

14:30~15:30 IS17 Applications for Intelligent Control
Chair: J.J. Lee (Korea Advanced Institute of Science and Technology, Korea)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Task-Oriented Multiagent Reinforcement Learning Control for a Real Time High-Dimensional Problem

M.A.S. Kamal
Graduate School of Information Science and Electrical Eng.
Kyushu University, Japan
kamal@cig.ees.kyushu-u.ac.jp

J. Murata
Graduate School of Information Science and Electrical Eng.
Kyushu University, Japan

K. Hirasawa
Graduate School of Information, Production and Systems,
Waseda University, Japan

Abstract

The task-oriented approach of reinforcement learning reduces complexity of a high-dimensional problem in a more realistic and human-like way of thinking. In this paper we investigate the performance of task-oriented learning for elevator group control considering two main tasks. The first task treats, from the viewpoint of passengers, how the most suitable elevator can be called, and the second one is related to, from the viewpoint of the elevators, how they can provide best service by selecting the suitable floor to wait for passengers. The simulation results show significantly improved performance in controlling this very large-scale stochastic dynamic problem.

1 Introduction

Most of the real world problems have their own typical characteristics and there is rarely a known optimal policy. In such cases reinforcement learning agent can successively improve its control strategy through experience and reinforcement from the system. Instead of its conceptual simplicity, the reinforcement-learning algorithm often cannot scale well to non-uniform problems with large or infinite state and action spaces.

This paper presents task-oriented reinforcement learning scheme [1,2] to solve a high dimensional problem proposing some logical task-oriented agents. It has been proved that the task-oriented learning has faster convergence characteristic for an episodic task [1]. For the continuous and dynamic system, where the agent does not have any opportunity to repeat its trial with same initial conditions, the task-oriented learning can be applied successfully [2]. In both cases, an agent uses more than one Q-table for different tasks. Many researchers have proposed multiple agents in lieu of a single agent to make a complex learning task easier and to achieve better performance, through combining outcomes of multiple agents. Here we propose two kinds of agents. According to the types of tasks the goal and policy of each group of agents are different, but by their combined efforts the ultimate goal of the system is attained. The whole task is decomposed into some logical sub-problems according to the types of actions needed, and the learning is carried out from the viewpoint of the task considering its expectation. The task-decomposition limits the size of state space hence ensures less memory and less computation requirement with faster convergence, and thinking from the viewpoint of the task leads the agent to choose the action more precisely.

2 Task-oriented Reinforcement Learning

Reinforcement learning [3] is a process of trial-and-error whereby an agent seeks to find the combination of actions that maximizes the rewards as its performance feed back. One of the most commonly used reinforcement learning method is Q-learning. This algorithm does not need a model of the environment and directly computes the approximate function of optimal action-value independent of the policy followed. The updating rule of Q-learning is as follows:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)] \quad (1)$$

where, α is the learning rate, γ is the discount factor, r_t is the achieved reward at time t, and $Q(s_t, a_t)$ is the value of action a_t in state s_t.

The usual approach of Q-learning is to take into account all kinds of information of the environment to constitute state and only one Q-table is used for the whole task. The agent receives different information of the environment to constitute its state and tries to choose better action maintaining a balance between exploration and exploitation according to the certain policy. The convergence of the Q-learning is proven under the assumption that each state-action pair is visited infinitely often. Unfortunately, most of the complex problems have high dimensional state space and it is hardly possible to visit all states in short interval to reach convergence and almost impossible to converge if different types of actions are needed in different situations.

The task-oriented approach of RL reduces the complexity of the problem by decomposing the whole task into some logical subtasks according to the types of actions (for example, searching the environment, moving to a particular location, conducting a job, etc). For each subtask a separate Q-table is used in which the state signal represents the job condition with respect to the corresponding agent, and the action space is the indications to the agent how the task

should be carried out. The goal of each subtask may be different apparently, but it helps attaining the global goal of the system. This method provides one Q-table for each subtask, therefore, the same agent may deal all Q-tables [2] or separate agent can be used for each subtask. The main objective of this method is to simplify the learning process considering less information related to corresponding task only, which reduces the state-space size, hence requires less memory and faster convergence can be achieved.

3 Elevator Group Control

3.1 Complexity in Elevator System

The elevator system poses significant difficulties as it operates in high-dimensional continuous state spaces and in continuous time as discrete event dynamic system [4]. The elevator systems are driven by passenger arrival, which varies during the course of the day. The stochastic passenger arrival pattern has different peak level, and traffic density also varies with time. In typical residential building, the morning rush brings a peak level of down traffic; while a peak in up traffic occurs during evening, and moderate or light profile in up or down traffic exist in other times. The performance criteria of an elevator system are to minimize the average (or average squared) wait time (the time between the arrival of a passenger and his entry into a car) for any traffic pattern and density.

Beside the standard service provided by the elevator during operated by the inside passengers, it needs to make some critical decision such as which elevator should respond when a hall call button is pushed by an incoming passenger. Whether light or heavy traffic is, if the down traffic is much larger than the up traffic the elevator should wait in upper floors, and when up traffic is much larger than the down traffic the elevator should stay near lobby to keep the waiting time minimum. The other operations such as opening and closing the door, acceleration and deceleration for starting and stopping the car, are fixed for all operations.

3.2 Implementation

Real time control of elevator group for a twelve-storied residential building with two elevator cars serves as our test bed. First thing before applying reinforcement learning to controlling an elevator group is to model it as a discrete event system representing its dynamics in terms of small time units. The usual approach of reinforcement learning control for elevator group provides one autonomous agent to control each elevator car [4]. The agent considers signal of internal car buttons, current floor and moving direction of all elevators, and calling signal from all floors as its current state to decide the next action. So the size of the state space increases exponentially with both the number of floors and the number of elevators and for this test bed there would be approximately 10^{16} states. In this system, sometimes more than one elevator may rush to the called floor, or all of them may not respond mistakenly presuming other elevator is going there since all elevator-agents respond independently after receiving any call.

In this test bed we found three kinds of tasks, which should be performed for its successful operation. The first task is operating the elevator according to some priority rules considering commands of inside passengers and waiting passengers. An elevator must stop in next floor if the hall call button is on or if the inside passenger want to get off. The second task is to consider how to keep the wait time minimum for forthcoming passengers. The elevator should stay at the floor where the possibility of getting a passenger is higher. The third task is to select the appropriate elevator when any hall call button is pressed. Only the elevator, which may take minimum time to reach the calling floor, should respond.

To solve the above three tasks we introduce two kinds of task-oriented agents: elevator-agent and calling-agent.

The elevator-agent operates the elevator-car considering the commands of inside-passengers and the calls from calling-agents, according to some priority rules and restrictions without learning (the first task). Beside this, the agent carries out a separate task which demands learning: it learns in which floor the agent should wait for the passenger (the second task). This is done only when the elevator has no signal of any inside or incoming passenger for a certain interval. In this learning the elevator agent compete with the other to be qualified in getting a passenger earlier. There are two elevator agents one for each elevator car and they share a single Q_E-table of 108 possible states. The elevator-agent considers the current location of all elevators, status (idle, moving-up, or moving-down) of the other elevator, and the direction of traffic flow as its state and chooses the suitable floor as the action where it may get a passenger earlier. The relative traffic flow is estimated by eq.(2),

$$T = \frac{T_{UP} - T_{DOWN}}{T_{UP} + T_{DOWN}} \times 100, \qquad (2)$$

where the value of T_{UP} is calculated by eq.(3) at each call by an up-going passenger at lobby and discounted by eq.(4) at every minute,

$$T_{UP} \leftarrow T_{UP} + 1, \qquad (3)$$

$$T_{UP} \leftarrow \lambda T_{UP}, \qquad (4)$$

where constant λ is the discount factor.
Similarly T_{DOWN} is calculated by the call of down-going passengers. The elevator-agent estimates the relative traffic flow direction as: Peak-up when T>20%, Peak-down when T<–20% and Normal otherwise.

Figure 1 shows a brief explanation of selecting action of the elevator-agent. The elevator-agent chooses an action (going to 4th-floor) after 200sec

elapsed time, goes to that floor and stays there for 200sec and finds no call from any floor (i.e. reward 0). So it updates the Q-value and again chooses the action (going to 4th-floor). Staying there for 90 sec it receives a call from 10th-floor, which gives a reward in terms of its travel time from 4th floor to 10th floor and updates the Q-value. The reward function is defined in terms of d_{floors}, distance of calling and constants A and B of suitable values chosen heuristically as in eq.(5),

$$r = \frac{A - d_{floors}}{B + d_{floors}}. \quad (5)$$

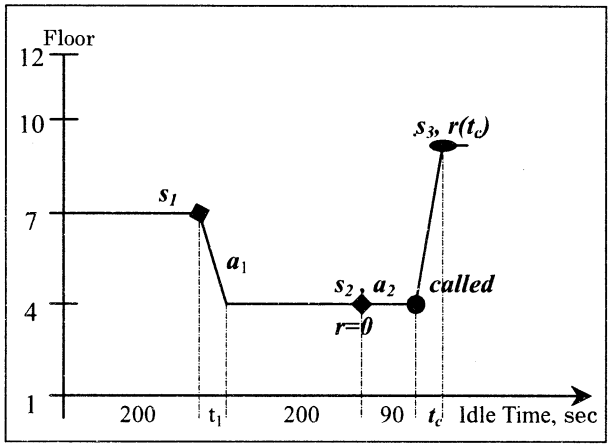


Figure 1: Typical way of action selection, reward obtaining and updating Q-table by the idle elevator agent.

There are 22 calling-agents for hall call buttons (two for each floor, except 1st and 12th floor, which have only one button) and they learn to choose the most suitable elevator that takes minimum time to reach after being called by a passenger (the third task). They consider the current location and status of elevators, and direction of the incoming passenger as the state information and call the suitable elevator after selecting it as its action. They share a single Q_C-table of 7744 possible states. The reward function is defined by eq.(6) in term of T_{wait}, the waiting time of the first passenger who pushed the hall call button, and constants C and D of suitable value chosen heuristically,

$$r = \frac{C - T_{wait}}{D + T_{wait}}. \quad (6)$$

The waiting time of other passengers in the queue are ignored since it is unknown to the system.

The task-oriented implementation of the control system greatly reduces the state space size, as it requires only 7852 states in two Q-tables, whereas the usual implementation would require 10^{16} states. The calling agent chooses the most suitable elevator estimating minimum time requirement to reach the calling floor, so in this competitive learning, both the elevator-agents try to be fit to receive a call from the calling-agent when a passenger pushes the hall call button.

5 Simulation Results

To investigate the performance of the proposed task-oriented system simulations are carried out approximating the elevator system dynamics by the following parameters:
Floor Time (the time to move one floor at maximum speed): 3 sec;
Stop time (the time needed to decelerate, open and close the doors, and accelerate again): 10 sec;
Turn time (the time needed for a stopped car to change direction): 1 sec;
Load time (the time for one passenger to enter or exit a car): 1 sec;
Car capacity: 15 passengers;

Q-learning algorithm and ε-greedy policy with decreasing value of ε is used for the simulation. The discount factor for the calling agent γ_C is set to 0.5 and for elevator agent γ_E is set to 0.95, and initial learning rate parameter α is set to 0.5, which decreases at a rate of 0.1% up to the value of 0.02. Since an elevator may reach the calling floor unexpectedly, we consider a comparatively low value of γ_E to reduce the noise effect. The value of constants λ of eq.(4) is set at 0.9. The constants A, B, C and D of the reward functions are set at 8, 9, 35 and 30, respectively.

The passenger-arrival data are generated using a typical probability distribution for 24 hours considering approximately 2000 incoming passengers per day for the residential building. Figure 2 shows the probability distribution for both up-going and down-going incoming passengers per second. 90% of up-going passengers get on at the 1st floor and 90% of down-going passengers get off at the 1st floor. Only 10% of inter floor passengers exist in the system.

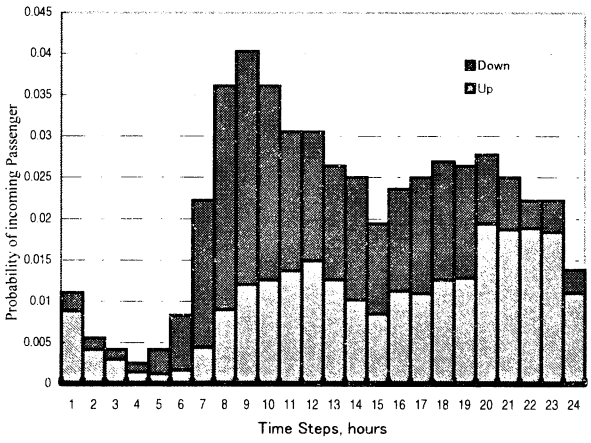


Figure 2: The probability distribution of an incoming passenger in each second for 24 hours a day in both up and down direction.

To compare the performance of the proposed task-oriented system, first we have tested the system using a single elevator, then using both elevators with random selection of elevators, and then we have tested the system using only calling-agent and finally the proposed system considering both calling-agents and elevator-agents.

Figure 3 shows the learning performance of three systems all for two-elevators, where the random selection shows a constant waiting time approximately 34 sec, whereas the system with only calling-agents reduces the waiting time from initial 34 sec to 26.38 sec. The system with both calling and elevator agents shows the best performance, and it reduces the waiting time to the lowest value of 26.02sec. In this simulation we also investigated the performance of the single elevator system to handle the same traffic profile and found an average waiting time of 49 sec, which is not shown in the figure.

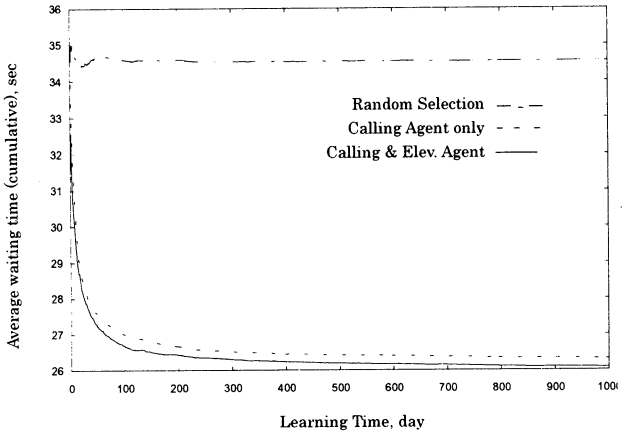


Figure 3: The cumulative average waiting time of incoming passengers for three systems as the systems in learning.

After a course of 1000 days learning we investigated the performance of these systems using the same traffic rate for each day. Figure 4 and Table 1 show the comparative statistics of these systems for a day taking an average value over 30 days. It is found that the waiting time increases for the hours with high traffic density for all systems as Fig.4.

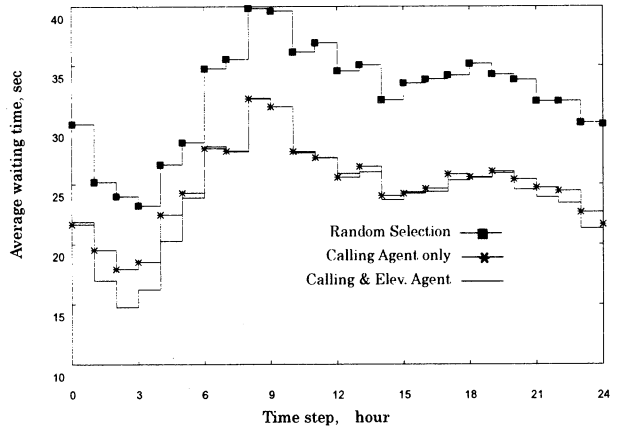


Figure 4: The variation of average waiting time of incoming passengers for three systems after learning for each hour of a day.

The calling-agent system and the calling and elevator-agent system both show almost the same average waiting time for the high traffic hours. The system with both calling and elevator agents shows significant improvement in waiting time for the hours with low traffic rate. The only way to keep the waiting time within reasonable value for the high traffic hours is to use additional elevators as we found comparing the performance with a single elevator system. Table 1 shows the percentage of the passengers waited more than 60sec. The both systems show approximately the same results since the longer waiting time occurs during high traffic hours only. If any passenger pushes the call button when both elevators are in working there is high possibility to have a longer waiting time. The other systems with random elevator selection and with single elevator show relatively worse performance.

Table 1: Statistics of all systems after learning in terms of waiting time and percentage of passengers who have waited more than 60sec.

	Observation time	Single elevator	Random elevator selection	Using Calling Agents	Calling & Elevator Agents
Av. Waiting time, sec	Max Passengers hour, 9.00-10.00	63.38	39.62	31.54	31.53
	Min Passengers hour, 3.00-4.00	30.10	26.65	22.45	20.27
	A whole day	49.54	34.46	26.30	25.98
Passengers, W-time>60 sec (%)	Max Passengers hour, 9.00-10.00	48.62	21.13	10.37	10.74
	Min. Passengers hour, 3.00-4.00	5.54	2.06	0.32	0.47
	A whole day	32.82	13.87	5.75	5.79

Conclusion

The task-oriented scheme simplified the problem reducing the size of state-spaces and fast convergence is achieved. To gradual or sudden change in passenger arrival pattern the system adapts well within a short time on its continuous operation. These results show the implied significance of task-oriented system for real time control system.

References

[1] M.A.S. Kamal, J. Murata and K. Hirasawa "Task-Oriented Reinforcement Learning in Cooperative Multi-agent System", Procedings of the 20th SICE Kyushu Branch Annual Conference, pp. 477-480, Dec 2001.

[2] M.A.S. Kamal, J. Murata and K. Hirasawa "Task-Oriented Reinforcement Learning for Continuous Tasks in Dynamic Environment", Proceeding of the SICE annual conference, pp.932-935, Aug 2002.

[3] Sutton, R.S. and Barto, A.G, "Reinforcement Learning: An Introduction," MIT press, 1998.

[4] Robert H. Crites and Andrew G. Barto, "Elevator Group Control Using Multiple Reinforcement Learning Agents", Machine Learning, 33(2-3), pp.235--262, 1998.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Action Selection by Voting with Learning Capability for a Behavior-based Control Approach

†S. M. Jeong †S. R. Oh †D. Y. Yoon *W. K. Chung **I. H. Suh **C. C. Chung
†Intelligent System Control Research Center, KIST
* Robotics Laboratory, POSTECH
**Division of Electrical and Computer Engineering, HanYang University
P.O.BOX 131, CHEONGRYANG, SEOUL, KOREA
sroh@kist.re.kr

Abstract

In this paper, we suggest a new action selection algorithm on behavior-based control approach. It uses a voting algorithm for the action selecting process to be self-improved by reinforcement learning algorithm in the dynamic environment. Proposed voting algorithm improves the navigation performance by adapting the eligibility of the behaviors and the command set to the faced environment. It introduces the command set generator to makean adapted command set according to the situation, and a set of behaviors votes for the command set. The voting performance is improved online by means of Q-learning. The robot's final action is determined by another action selection part of "action selector." It coordinates the selected command for navigation with other kinds of behaviors and the action selector's performance is also improved by Q-learning. Simulation results show the good performance for the action selection.

Key words: Behavior-Based Robotics, Voting, Q-Learning, Navigation, and Action Selection Problem.

1 Introduction

The study of robotics has trended into the artificial intelligence (AI). The trend of this research is that human-like robots have been developed rapidly, but mission to control robot in dynamic and complex environment is still challengeable. In the field of controlling a robot, the Action Selection Problem (ASP) is a main subject. ASP has early introduced by [1][2][3]. The approaches for ASP can be classified into two categories of hierarchical paradigm and reactive one. The reactive paradigm has a representative control scheme of behavior-based robotics. Behavior-based scheme is based on coordination among behaviors for running so that it is closely related to the ASP. This research proposed an architecture for ASP based on the behavior-based frame. The employed behavior-based strategy takes advantages of bottom-up intelligence and addressing real-world environment, but the simplicity of the pure behavior-based strategy has defect in missing experiences. We employed the learning capability to the coordination of the behaviors, e.g., ASP in order to make use of experiences. It may be referred to as a reactive-deliberative hybrid paradigm [4] since it deliberates the action selection process with past experiences rather than reactive response. The suggested architecture consists of three part of "action selector", "command set generator (CSG)" and "navigator". The ASP is executed by following manner: First, the CSG makes a command set based on velocity vector. Second, the navigator can selects a command set with voting values. Finally, action selector compares the chosen navigation command with other heterogeneous behaviors. In these process, the voting and the final action selection processes have the self-improving capability by Q-learning.

2 Definition of the ASP

The objective of the action selection problem in this research is to select the navigation commands. As shown in figure 1, the objective is navigating for the goal while avoiding obstacles. The environment proposes conflictive behaviors of obstacle avoidance and movement to the target. The dotted arrows indicate the directions that will satisfy the "obstacle avoidance" behavior. The solid arrow indicates the direction to the target. This conflict are fused into possible multiple velocity commands of various direction, and definition of the ASP is a choice among the fused commands. This is a homogeneous ASP for navigation.

Another ASP is in the choice of actions among heterogeneous [5] goals. Final, action command is selected in competition among the chosen command above and other kind of goals by Q-learning.

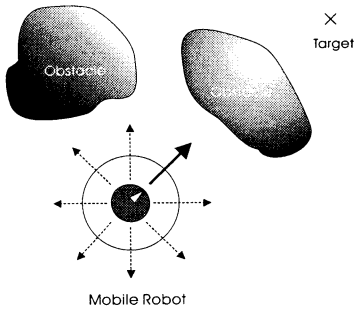


Figure 1: Conflicted behaviors : between the Move to goal and the Obstacle Avoidance

3 Proposed Control Architecture

The proposed control architecture is shown in figure 2. It has a overall SENSE-ACT structure. The behaviors are divided into two kinds. One is for navigation to the goal, and the other is for other missing such as battery charging. For ASP, the voting of the homogeneous behaviors which choose a best command set performed firstly. The best navigation command then compete with other heterogeneous behaviors for robot's final action. This final stage is for the selection among actions that cannot be made from the main CSG. The voting process and the final action process use the Q-learning strategy. The reward values are taken from the two reward functions. The features of the reward functions are explained in section 3.4

3.1 Command Set Generator

The CSG has the role of making the adapted command set according to the situation, e.g., velocity. The CSG makes the linear and the angular velocity as follows:

$$V_{basis} = V_{avg} = \frac{V_1 + V_2 + \cdots + V_n}{n} \quad (1)$$

$$V_{CSG} = \{-2 * V_{gap}, -V_{gap}, V_{basis}, V_{gap}, 2 * V_{gap}\} \quad (2)$$

Where, V_{gap} makes a decision by the difference between the maximum velocity and the minimum velocity.

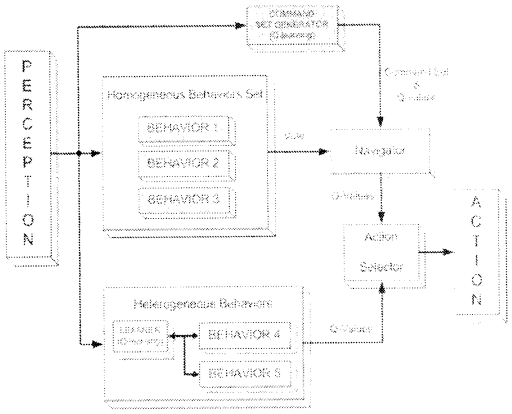


Figure 2: Control Architecture for the ASP

$$\omega_{basis} = 0.0 \quad (3)$$

$$\omega_{max} = \max\{\omega_1, \omega_2, \cdots, \omega_n\} \quad (4)$$

$$\omega_{gap} = \frac{\omega_{max} - \omega_{basis}}{2} \quad (5)$$

$$\omega_{CSG} = \{-\omega_{max}, -\omega_{gap}, \omega_{basis}, \omega_{gap}, \omega_{max}\} \quad (6)$$

Using above equations, we could obtain the final output of the CSG as follows.

Table 1: Output of the CSG, state = 614

		Step	CSG_v	CSG_ω	$Diff_{behavior}$
State	614	0	0	2	0.75
v_1	1.5	1	0.25	1	2
v_2	0	2	0.75	0	
ω_1	-2	3	1.25	-1	
ω_2	2	4	1.75	-2	

3.2 Navigator

In order to increase further the ability of ASP to deal with a dynamic environment, the navigator part is proposed. The navigator selects the best command set that produced by the CSG. The selection is performed by means of voting strategy, and the voting performance is improved by Q-learning.

3.3 The employment of Q-learning

Reinforcement learning has one of the methods used to adapt robotics control systems to changing environments. A mobile robot has to operate in unknown environments and must learn to predict the

consequences of their own actions. A representative method of reinforcement learning, the Q-learning [6] was employed. Q-learning is a model-free reinforcemente arning based on stochastic dynamic programming. It provides the robot with the capability of learning to act optimally in Markovian domains by experiencing the consequences of actions without map-building. It is desirable characteristics to the our behavior-based frame. We assume that the robot is a computational agentmo ving around some discrete, finite world, choosing one from a finite collection of actions at every time step. The learning process is summarized as follows:

- For a policy π, define Q value as:

$$Q^{\pi}(x,a) = R_x(a) + \gamma \sum_y P_{xy}[\pi(x)]V^{\pi}(y). \quad (7)$$

- Update Q-values.
  - If the state $x = x_n$ and action $a = a_n$

$$Q_n(x,a) = (1-\alpha_n)Q_{n-1}(x,a) + \alpha_n[r_n + \gamma V_{n-1}(y_n)] \quad (8)$$

  - otherwise

$$Q_n(x,a) = Q_{n-1}(x,a). \quad (9)$$

3.4 The reward function

The implementation of the Q-learning strategy has focus on the implementation of the reward function. Due to the difference of reward values, a certain controller could not control to adapt action selection to the situation. The reward function should bemad e from sensor readings. In general, reward function can be classified into dense rewards and sparse rewards[7].

3.4.1 Sparse Reward Function

Sparse reward function corresponds to physical events in the environment. This function is easy to be subjected to simple reward function for many tasks. For example, for a navigation task, the robot might get a reward of -1 for conflicting an obstacles and 1 for reaching the target. However a sparse reward function has a lot of problems that are zero elsewhere.

3.4.2 Dense Reward Function

On the other hand, the dense reward function gives non-zero rewards most of the time. A dense reward function for obstacle avoidance might be the sum of distances to the obstacles divided by the distance to the target. But it is much more difficult to be implemented than sparse reward function.

$$\mathbf{Reward_{Total}} = \frac{\sum \mathbf{Dist_{Obstacle}}}{\mathbf{Dist_{Target}}} \quad (10)$$

3.4.3 Proposed Reward Function

In order to complementth e defect of above two kind of reward function, we suggest a new reward function that guarantees robustness at everywhere. The proposed new reward function is designed to provide both sparse and dense reward capabilities. As the result , a new reward function is more flexible than the existing reward functions.

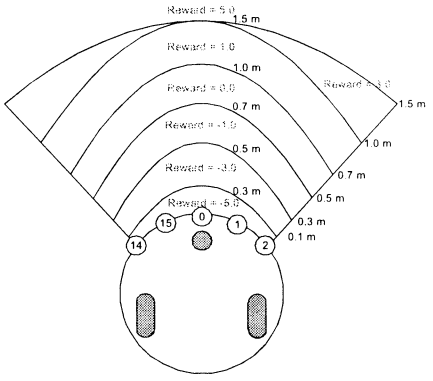


Figure 3: Reward Function

3.5 Voting algorithm

We have chosen weighted voting. Weighted voting is not harder to implement than simple voting. The used weighted voting is a majority voting method which $\omega > 1/2V$. The navigator is properly taken the velocity vector by voting. In the process, voting algorithm is implemented by solving the following equations:

$$\arg\max_x [R_1(x), R_2(x), \cdots, R_n(x)] \quad (11)$$

$$\textbf{subject to} x \in X, \textbf{where} x = (x_1, x_2, \cdots, x_n) \in R'' \quad (12)$$

3.6 Action Selector

As mentioned before, this section introduces an action selector which is contrived to solvean other action selection problem. This module compares the Q-values of the navigator-generating command and the other heterogeneous behaviors. These Q-values are updated by iterative learning process.

4 Simulation

The architecture has been implemented in the simulator in order to test the correct performance of the model and its ability to adapt different characteristics in the environment. The mobile robot in the simulator learns about Q-value to select one of actions. The needed learning parameters are as follows.

Table 2: Learning Parameters

learning rate	$\alpha = 0.5$
discount rate	$\gamma = 0.6$
initial Q-values	zero
number of state	1040EA ($20m \times 13m$)
number of target	5EA
resolution	$0.25m^2$
sensor(Ultrasonic)	16EA
the state-action table is updated at each step	

The first experiment compares voting algorithm with potential fields for navigation. These algorithms have a lot of problems for action selection problem. Especially, they can not find a target correctly. Both voting algorithm and potential fields method have been showed that a mobile robot incorrectly fulfills the mission. On the other side, the robot is capable of stabilizing its weights of the voting algorithm with Q-learning. As figure 4, the suggested architecture implemented to prove a reliability in a complex environment. As the result, the performance of the robot improved as the episode increase.

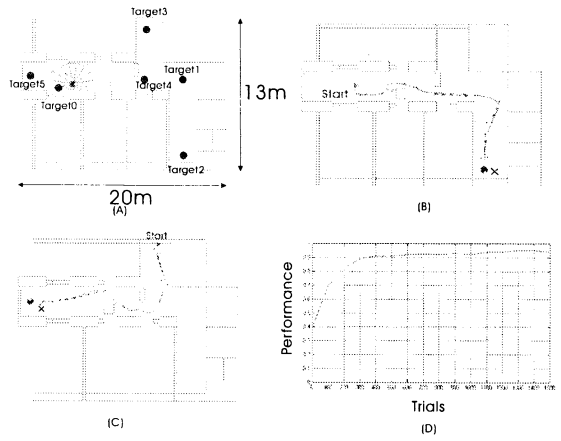


Figure 4: Scope of several successive destinations: (A) simulation environment, (B) task 1 : Target 0 - 2, (C) task 2 : Target 3 -5 ,(D) results of Q-learning

5 Conclusion

There are a variety of control architectures for action selection techniques that work effectively on a lot of small problems. This work was partly inspired by DAMN[3] that is a planning and control architecture in which a collection of independently operating modules collectively determine a robot's actions. But the structure[3] has a difficult problem to determine the weighting value of behaviors. In this paper, new control architecture was considered for this problem using Q-learning. Future work will be focused on the localization and real-world experiment.

6 Acknowledgments

This work was partially supported by Biomimetic Control National Research Lab program and Tangible Space Initiative KIST program.

References

[1] R. Arkin, "Motor Schema Based Navigation for a Mobile Robot.", *IEEE International Conference on Robotics and Automation* 1987.

[2] R. Brooks, "A Roboust Layered Control System for a Mobile Robot.", *IEEE Journal of Robotics and Automation*,Vol.2 RA-2, no. 1, pp.14-23, 1986.

[3] Julio K. Rosenblatt, "DAMN: A Distributed Architecture for Mobile Navigation.", *Ph.D. dissertation, Carnegie Mellon University* , 1997.

[4] Robin R. Murphy. *Introduction to AI Robotics* The MIT Press, 2000.

[5] Mark Humphrys, "Action Selection methods using Reinforcement Learning.",*Ph.D. dissertation, University of Cambridge*, 1997.

[6] C. Watkins, P. Dayan, "Technical Note: Q-learning.", *Machine Learning*, Vol.8, pp.279-292, 1992.

[7] William D. Smart and Leslie Pack Kaelbling, "Effective Reinforcemnt Learning for Mobile Robots.", *IEEE International Conference on Robotics and Automation*, Vol.4, pp.3404-3410, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Development of A Novel 4-dof Mobile Microrobot with Nanometer Resolution

Zhu Tao, Tan Dalong, Zhang Jiangbo, Wang Zheng
Robotics lab, Shenyang Institute of Automation,
Chinese Academy of Sciences
Shenyang, P.R.China, 110016.
Email:tzhu@sia.ac.cn

Abstract

A novel 4 dof mobile microrobot with nanometer resolution is designed in the paper . It is based on deformations of piezoelectric tubes. In contrast to existing mobile micro mechanisms, the microrobot is actuated by rolling friction force instead of sliding friction force. The robot can obtain translation in x- and y- directions and a rotation around its center in the plane. The movable range is infinite in principle. In addition, it is easy to get a motion in z direction by extending or contracting all the tubes simultaneously. The microrobot has advantages in compact volume, quick response, heavy load, long displacement, and high resolution. The paper mainly deals with problems in its design, driving principles and modeling. Some experimental results are also investigated to identify the effects of friction coefficients, load and other related factors. The mobile microrobot has promise in microassembly, cell manipulation, injection of DNA, and the repair of large scale circuit.

Key words: piezoelectric, mobile microrobot, frictional force

1 Introduction

With the development of MEMS, the demands for precious micro components are increasing dramatically. However, for the production of hybrid microsystems that consist of several microcomponents made of different materials and manufactured by different techniques, one or more steps assembly is needed. Besides, it is often necessary to combine conventional and microcomponents within a system. Therefore, automatically controlled smart microrobots are desired to free humans from the tedious tasks of having to manipulate very small objects directly.

Many researches have been focused on the actuation and motion control of mobile microrobots, most of which are based on inchworm or impulse driven principles. These two types are based on dry friction resulting from reaction force to gravity, as shown in T.Higuchi [1]. Recently, new principles are proposed by researchers. Mark Versteyhe [2]et al built a rigid and accurate piezo-stepper based on hybrid force-position controlled clamping. They use a series of piezo driven units to realize accurate linear motion. Sergej Fatikow[3] et al developed an automated desktop station, in which "slip-stick" actuation principle has been implemented. Sylvain Martel[4] et al proposed a three-legged wireless miniature robot, using the bending of piezo tubes to actuate the robot. We have developed a novel 4-dof mobile microrobot based on deformation of piezoelectric tubes. In contrast to existing mobile micro mechanisms, the microrobot is actuated by rolling friction force instead of sliding friction force. The robot has a resolution about 20 nanometer and theoretically unlimited displacement, as well as good flexibility.

The paper is arranged as follows: in section 2, the driving principle and design of the robot are proposed. Then, from physical and geometrical point of views, the paper establishes mathematical model for the microrobot. In section 4, some experimental results are given to identify the effects of related parameters to the motion. The last section concludes the paper.

2 Drive Principle and design

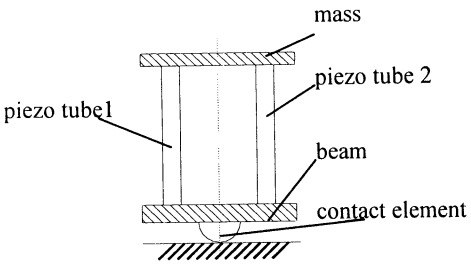


Fig. 1. Piezo-driven unit

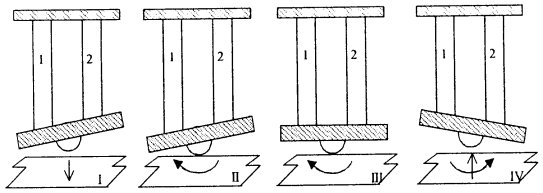


Fig. 2. Stepping principle

Due to its characteristics of high efficiency. compact volume, quick response, high load capacity and high resolution., piezoelectric element is often preferable for precious positioning. The microrobot we designed is based on three piezo driven units acting as the legs of the robot, which can transform the vertical deformation of the piezo tubes to the horizontal motion of the robot. Each unit consists of two piezo tubes, a beam, and a contact element (fig.1).

As shown in fig 2, while being applied inverse voltage, the two piezoelectric tubes generate different deformations, one extends, while the other contracts. The combination of deformations drives the semi-sphere bonded to the beam to roll. Because of rolling friction force between the sphere and the base surface, the mechanism moves on the surface.

Four phases are involved in motion of each unit: descending, rolling, ascending, and recovering. By repeating such phases, continuous stepping movement can be obtained (fig.2). When three units are arranged at equilateral triangular locations, through controlling the applied voltages on the piezo tubes, the robot can

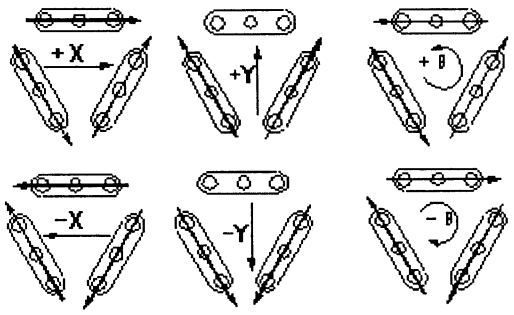


Fig. 3. Concept of 3-axial movement

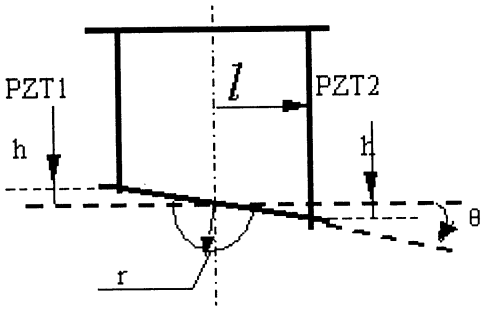


Fig.4. Displacement of a unit

achieve translation in x- and y- directions and a rotation around its center on the plane(fig.3) .The movable range is infinite in principle. In addition, it is easy to get a motion in z direction by extending or contracting all the tubes simultaneously.

3 Modeling

There are several ways to obtain a mathematical description of a physical system by the modeling of dynamic systems. Mostly, we use geometrical and physical views to analyze the motion of the microrobot. The motion of the robot is decided by the displacement of the sphere, so we should calculate the rotation angle of the sphere first. According to figure 4, we can get

$$\theta = \frac{h}{l} \tag{1}$$

where h is the deformation of the piezo-tube, l is the distance between the tube and the center of the sphere. θ is the angle of the rotation of the beam, and r is the radius of the sphere. Because the sphere is bonded to the beam, it follows the rotation of the beam, therefore we can get the displacement in the planar direction.

$$S = \theta \times r = \frac{hr}{l} \tag{2}$$

According to the equation (2), the step size is determined by transformation h, as well as the ratio of r to l.

Based on energy equivalent equation, we can get the angular acceleration of the sphere. The unit's moment comes from the elastic energy generated by the transformation of the piezo tubes. Hence, we can get the following equations for each unit

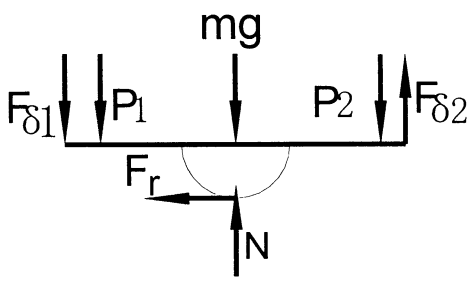


Fig. 5. Force analysis of the sphere

$$ma_{cx} = \sum X \tag{3}$$

$$ma_{cy} = \sum Y \tag{4}$$

$$J\dot{\omega} = \sum M \tag{5}$$

$$\sum X = F_r \tag{6}$$

$$\sum Y = -F_{\delta 1} - P_1 - P_2 - mg + N + F_{\delta 2} \tag{7}$$

$$\sum M = (F_{\delta 1} + P_1 + F_{\delta 2})L - (P_2 + F_r)r \tag{8}$$

where m is the mass of the sphere, M is the mass of the robot on each unit, N is the support force of the surface, Fr is friction force between the sphere and the base surface, $F_{\delta 1}$ is the reduction of piezo tube 1, $F_{\delta 2}$ is the elongation of the piezo tube 2, and P_1 and P_2 are the load of the mass on each piezo tube, $\dot{\omega}$ is angular acceleration of the sphere.

where $a_{cy} = 0$

$$P_1 = P_2 = \frac{M_1 g}{6}, \quad F_{\delta 1} = F_{\delta 2} = K\delta ,$$

$$F_r = f_r N, \quad J = \frac{2}{5} mr^2$$

where K is the stiffness of the piezo tube.

From equation (4) to (8), angular acceleration $\dot{\omega}$ can be estimated as

$$\dot{\omega} = \frac{5K\delta L}{mr^2} - \frac{5(m + M_1)gf_r}{2mr} \tag{9}$$

If there is no sliding, the linear acceleration of the unit can be calculated as

$$a_{cx} = a_c = r\dot{\omega} \tag{10}$$

From the equation (9) we can get that the angular acceleration $\dot{\omega}$ is decided by a series of parameters such as frictional coefficient, load, frequency, driven voltage, etc. because the driving force is rolling frictional force, the surface of the substrate is decisive to the motion.

4 Experimental results

Based on above mentioned driving principle, we have built up an experimental system and implemented some experiments. In order to reduce the mass of the robot, organic glass is used as platform and beams. The contact elements are glass semi-spheres. To simplify the structure and minimize the volume, only three drive units are involved in the robot.

Since the transformations of the piezo-tubes are determined by applied voltages, we can control the step size and speed of the mechanism by implementing different voltages and different frequency. In order to obtain proper motion, the voltages on the two tubes must be applied according to certain sequence (fig.6).

To identify the characteristics of the microrobot, we implement some experiments. Figure 7 shows the comparison of displacements under different voltages (65V and 100V respectively). Figure 8 shows the comparison of displacement under different payload, namely empty load , 50g and 91g. The results show : the increasing of the drive voltage and the frequency can accelerate the motion, while the effect of load is opposite. We also investigate the effect of different work surfaces

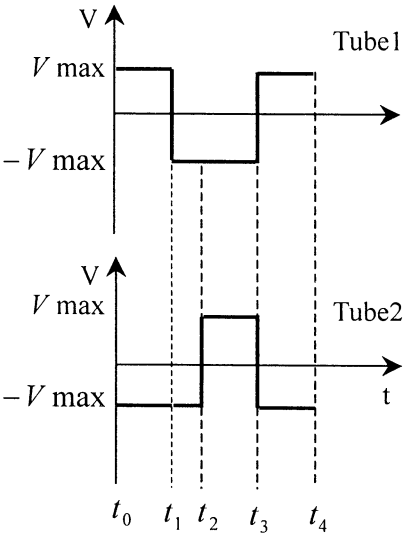


Fig.6. Drive voltages on one unit

to the robot, including glass, organic glass, and wood. Experiments indicate that the microrobot moves faster on glass surface than on the organic glass surface, but can't move on wood surface. However, the frictional coefficient must be within a proper range, too slippy or too rough is not good for the motion. For precious movement, the noise and vibration from the environment decline the accuracy of the robot. Besides, the forces generated by the wires also affect on the motion greatly.

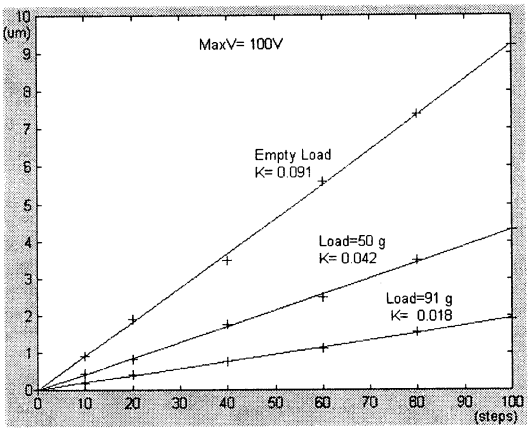


Fig.8. Displacement vs different payload

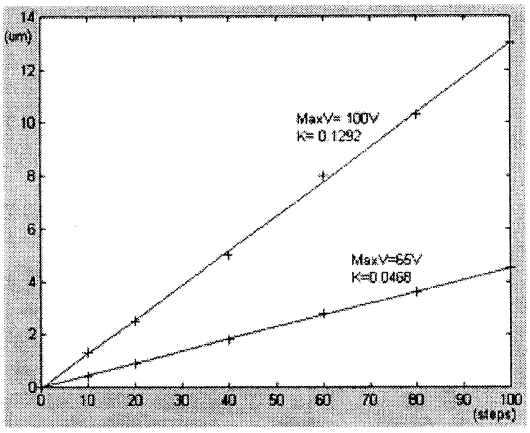


Fig.7. Displacement vs different maximum drive voltages

5 Conclusion and future work

In the paper, a novel 4-dof mobile microrobot is developed. It has the advantages of small size, high resolution, high stiffness and theoretically unlimited displacement. However, it has some defects yet: first, the base surface must be smooth enough, or the robot will not move. Second, because of the tension of the wires, the linearity of the motion is not very good. In order to get rid of the tension of the wires, our next goal is to build a teleoperated robotic system. Meanwhile, stack piezo elements will be used instead of tube-shaped ones, so that the payload capacity will be significantly improved. With attachment of different micro manipulators, the microrobot will be more useful for cell manipulations and micro assembly, and so on.

Acknowledgement

This research work has been performed at Shenyang Institute of Automation, It is supported by Innovative Grant of SIA F020110.

References

[1] T.Higuchi, Y.Yamagata, Kfurutani, et.al, "Precise Positioning Mechanism Utilizing Rapid Deformations of Piezoelectric Elements", Proc. of IEEE Micro Electro Mechanical Systems, February 1990, pp.220-226

[2] Mark Versteyhe, Dominiek Reynaerts, and Hendrik Van Brussel, "A rigid and accurate piezo-stepper based on smooth learning hybrid force-position control clamping", Proc. of the 1998 IEEE International Conference on Robotics and Automation. Leuven, Belgium, May 1998 ,pp.3059-3064

[3] Sergej Fatikow, Ulrich Rembold, and Heinz Wörn, "Design and control of flexible microrobots for an automated microassembly desktop station", Proc. of SPIE on Microrobotics and Microsystem Fabrication, Pittsburgh Pennsylvania, USA, November 1997, pp.66-77

[4] Sylvain Martel, Mark Sherwood, Chad Helm, et al, "Three-legged wireless minature robots for mass-scale operation operations at the sub-atomic scale", Proc. of the 2001 IEEE International Conference on Robotics and Automation. Seoul, Korea, May 2001,pp.3423-3428

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A Simulator for Strategy developing and realization in Robot Soccer Game

Jing-Sin Liu, Tzu-Chen Liang and Yi-An Lin
Institute of Information Science 20
Academia Sinica
Nankang, Taipei 115, Taiwan, R.O.C.
liu@iis.sinica.edu.tw

Abstract

We build a control command driven mobile robot motion simulator with controller and dynamics of mobile robots included. Kick motion follows physical law, and a simplified collision check and response model is utilized for efficiently detecting the hitting of robot with the ball or other robots. Dynamics of robots, which is very important to describe the high speed motion, are used together with the nonholonomic kinematic constraints for accurate simulation of the real robot dynamical behavior for development of strategies and control methods and evaluation of the overall planning and control algorithms. A demonstration of three robots to pass a ball is shown by the visualization of the simulator.

I. Introduction

In recent years, robot soccer game has inspired very fruitful research issues such as multi-agent systems, multi-robot cooperative teams, autonomous navigation, sensor fusion, fast pattern recognition and vision-based real time control [1],[10]. It also has been proposed as a benchmark problem for developing and comparing new methods in the fields of artificial intelligence and multiple robotic systems.

In robot soccer games, there are two teams of wheeled mobile robots embedded with local on board intelligence. A video camera captures the stadium image and the host computer extracts the locations of the home soccer robots, the opponent soccer robots and the ball. The communication between soccer robot and host computer is via wireless communication. On the other hand, both major world robot soccer game leagues, Robocup and FIRA, hold simulation competition, which assume the availability of locations of ball and robots. The objective of simulation league is for joiners who are interested in the software design to concentrate on the study of artificial intelligence or strategy development. Because of its comparably low cost in participation, the number of teams joining this league is much more than other physical robot leagues. The simulator used in Robocup Simulation League, named Soccer Server [7], mimics some sensory and motion abilities of a human-like soccer player. It is very different from the one provided by FIRA, called Simurosot. Teams joining Robocup simulation game should develop robot soccer skills, strategies and robot intelligence like human playing soccer, while teams joining FIRA simulation game need to design controller, trajectory planning method and others "machine" soccer skills.

Robot simulation is a useful tool for verifying the performance of overall system in a controllable, repeatable software environment. It affords greater flexibility to adapt to new situations and serves as valuable resources for the development of real robot systems at low cost, though the simulations may miss significant features to exhibit the necessary faithfulness in prediction. To port simulation results on real robots with acceptable performance is to a large extent founded in the accuracy of the simulations. An accurate simulation of the robot dynamical behavior is thus essential to describe the high-speed motion and for use in learning control program, as small differences between real and model robots are amplified through the robot learning program. The model of robot behavior may be learned from the interactions with environment by a simulated/physical approach using recorded data from real robot runs. In this paper, we build a dynamic mobile robot simulator where the Lagrangian of robot motion is built into where a ball-passing strategy for three robots is implemented for demonstration. Example of simulators for multiple robots are reported in [2-4] developed for different objectives of investigation such as multirobot coordination and control, behavior-based control.

This paper introduces a simulator for mobile robot motion planning and control. The simulator is aimed to serve as a platform for developing and verifying strategies, and trajectory planning and control methods for robot soccer games. We choose the unicycle mobile robot as robot soccer player. We choose the unicycle mobile robot as soccer player because of its superior mobility and abundant research information. Our robot simulator is an imitation of Simurosot software of FIRA [1], except that the robot motion is simulated by dynamical model and collision response is included. Ball motion is simplified as a pure slipping motion with friction. When the ball collides with the robot, a non-elastic collision model simulates the ball reaction after collision.

The paper is organized as follows. In the next section we will describe the details of mobile robot motion simulator and the models of the motion of ball, collision and kick. In Section 3,

a demonstration of the ball passing strategy and its realization is shown. Conclusion is made in section 4.

II. The Simulator

2.1 Dynamic Model of Mobile Robot

The shape of robot is modeled as rectangle with center of mass at the center of polygon. The vehicle position is described by the coordinate (x, y) of the midpoint between the two driving wheels, and by the orientation angle θ with respect to a fixed frame. Under the hypothesis of "pure rolling" and "non slipping", of wheels motions, unicycle model of mobile robot satisfies the nonholonomic constraint $\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$. The motion of the robot can be described by the following kinematical model,

$$\begin{aligned} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= w \end{aligned} \qquad (1)$$

where v is the linear velocity and w is the angular velocity of robot. (1) is the kinematic model of wheeled mobile robot useful for the design of path planning algorithms. However, an accurate simulation of the robot dynamical behavior is very important to describe the high speed motion, such as abrupt changes of velocity often occurred in robot soccer game. The dynamical model of vehicle is described by the following equations [5],

$$\begin{aligned} \ddot{x} &= -\sin(\theta)\left[\dot{x}\cos(\theta) + \dot{y}\sin(\theta)\right]\dot{\theta} + \frac{\cos(\theta)}{mr}(\tau_R + \tau_L) \\ \ddot{y} &= \cos(\theta)\left[\dot{x}\cos(\theta) + \dot{y}\sin(\theta)\right]\dot{\theta} + \frac{\sin(\theta)}{mr}(\tau_R + \tau_L) \\ \ddot{\theta} &= \frac{l}{Ir}(\tau_R - \tau_L) \end{aligned} \qquad (2)$$

where τ_L and τ_R are driving torques of left and right wheels; m , I are the robot mass, moment of inertia, respectively; r is the wheel radius.

Wheel torque-acceleration relation

Though a discrete version of equation (2) can simulate the robot motion successfully, we still need the kinematical model of equation (1). This is because most mobile robots are controlled by the velocities of their wheels, which are related to the radius of curvature of the car-like vehicles. Therefore, the velocity command is more natural than torque command for mobile robot control. There is no wheel velocity terms appeared in equation (2) and the constraint equation is implicit. Therefore, a relationship between wheel torque and wheel velocity should be developed and the kinematics would be incorporated into the relationship to assure satisfaction of the kinematic constraint.

Let l be the distance between the ground contact points of two driving wheels and R be the turning radius of the midpoint point of robot, then

$$\frac{1}{R} = \frac{2(v_R - v_L)}{l(v_R + v_L)}$$

Let v_L and v_R denote the velocities of the left driving wheel and the right driving wheel, respectively. Kinematically, the velocity of mobile robot can be computed by the average of two wheel velocities,

$$v = (v_R + v_L)/2$$

Differentiating the above equation yields the acceleration of the robot

$$\dot{v} = (\dot{v}_R + \dot{v}_L)/2 \qquad (3)$$

On the other hand, the robot velocity and wheel torques are always at the same direction, we also have,

$$\dot{v} = \frac{\tau_R + \tau_L}{mr} \qquad (4)$$

From equations (3) and (4), we have,

$$\dot{v}_R + \dot{v}_L = 2(\tau_R + \tau_L)/mr \qquad (5)$$

Furthermore, without loss of generality assume that $v_R > v_L$. From figure 1, we can build the relationship of differential wheel velocity and the angular velocity of robot $w = \dot{\theta} = v/R = (v_R - v_L)/l$ .Differentiating this equation and from the third equation of (2), we have

$$\dot{v}_R - \dot{v}_L = \frac{l^2}{Ir}(\tau_R - \tau_L) \qquad (6)$$

Solving equations (5) and (6), the relationship between wheel acceleration and wheel torque can be derived as,

$$\begin{aligned} \dot{v}_R &= \frac{1}{mr}(\tau_R + \tau_L) + \frac{l^2}{2Ir}(\tau_R - \tau_L) \\ \dot{v}_L &= \frac{1}{mr}(\tau_R + \tau_L) - \frac{l^2}{2Ir}(\tau_R - \tau_L) \end{aligned} \qquad (8)$$

This set of equations shows a coupled relation that one side wheel velocity is not solely dependent on its wheel torque.

Torque-acceleration Decouple Method

Figure 1 is the block diagram of the torque-acceleration decouple method. The robot model represents the equation (8), in which the relationship of torques and accelerations of two wheels are coupled. The "Nominal Robot Model" is the series connection of the robot model and a decoupler. As a result, the relationship between input torques (τ_R^+, τ_L^+) and output accelerations is decoupled as,

$$\dot{v}_R = \frac{2}{m\,r}\tau_R^+ , \dot{v}_L = \frac{2}{m\,r}\tau_L^+ \qquad (9)$$

suitable for designing control for each wheel. The "Actual Controller" is the designed controller cascading with the decoupler.

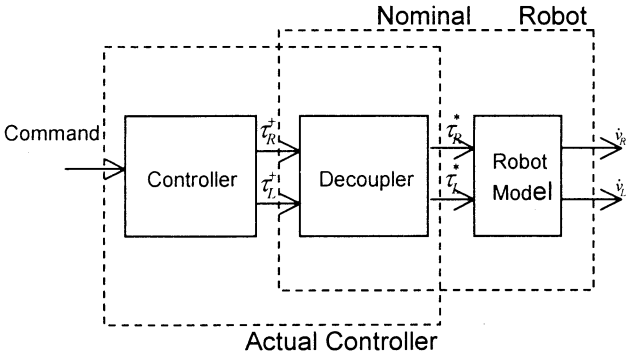


Fig.1. The connection of decoupler, controller and robot model.

2.2 Collision and Collision Response

(1) Robot-robot Collision Check. In physical robot soccer games, collision response between two mobile robots is a complex phenomenon involving impact effects in dynamics and cannot be simulated to perfect reality by a simple mathematical model. Since our simulator is developed for the purpose of strategy development, path planning and controller design, a complex yet accurate collision model does not quite fit the use for efficiency concern in simulation of dynamically changing environment. Here, we employ a simplified collision response model which can avoid the overlap of mobile robots when the robots collide. The robot is modeled as a square in the simulator. The robot-robot collision checks are activated after every position update of the robots. For ease of collision check, the collision is checked for its enclosing circle of radius *r*. Two robots are colliding if the distance between two centers of enclosing circles is less then *2r*. When a collision happens, to avoid overlapping, two robots can only move along the tangential direction of the collision surface, i.e., the normal component of velocity with respect to the collision plane is set to be zero.

(2) Kick Model: responding motion after kick. A kick means a mobile robot collides with the ball. A point contact between robot and ball is assumed. In the simulator, the robot soccer player kicks the ball by its front surface, which is orthogonal to the velocity direction. When a collision of ball and robot occurs, the kick model is utilized to simulate the ball reaction. We assume that comparing with the ball mass, the robot is heavy and would not change its velocity after kicking the ball. But since the simulator is discrete in time, we shall first solve the real kick time and position for reflection computation needed for describing motion after kicking. Referring to Fig.2, suppose at $T_k = kT$ the ball and robot are collision free while at $T_{k+1} = (k+1)T$ the ball and the robot are overlapping, where $T = \Delta T_1 + \Delta T_2$, ΔT_1 is time-to-collision. This means that the time step is too large, and a smaller time step must be found. A kick must happen at a certain time instant between *t=kT* and *t=(k+1)T*. An estimate time of ΔT_2 is provided by the penetration distance at T_{k+1} divided by the velocity of robot at T_{k+1}. The real kick position and time are then solved by simple geometric computations. By the principle of particle mechanics, the ball velocity vector after kicked by the robot can be computed by adding robot velocity vector $\vec{v}_m$ right before collision and the zero-speed-collision reflection velocity vector $\vec{v}_{bo}$. The ball velocity after kick is the vector $\vec{v}_m + \vec{v}_{bo}$.

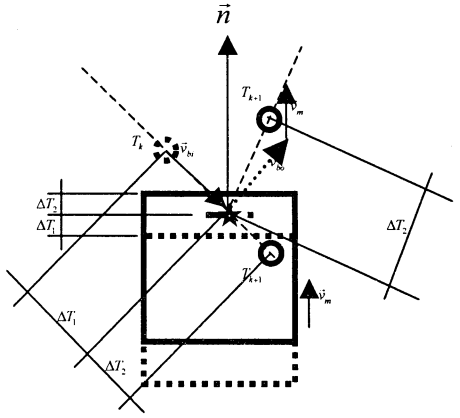


Fig.2. kick model

Finally, the ball position at *(k+1)T* is the vector of kick position plus ball velocity vector multiplying the travel time ΔT_2.

(3) Ball Motion. In the simulator, the ball locus is a straight line and the motion is pure slipping subject to a friction force proportional to its linear velocity.

2.3 Simulation Process

1. Discrete Time Simulation: update equations at successive time instants. Now we can update the location of robot from the input torque (9) by incorporating the nonholonomic constraints (1). From equation (9), for wheel input torque (τ_R, τ_L) at time instant *t=kT*, the wheel acceleration pair $(\dot{v}_R, \dot{v}_L)$ can be computed. Let the wheel acceleration pair at *t=kT* be denoted as $(\dot{v}_{Rk}, \dot{v}_{Lk})$, then wheel velocities at *t=(k+1)T* can be calculated from wheel accelerations. Let (x_k, y_k, θ_k) denote position and orientation of the robot at *t=kT*. Its position and orientation at *t=(k+1)T*, is updated by discretizing nonholonomic constraints (1):

$$x_{k+1} = x_k + v_k \mathrm{T}\cos(\theta_k + \dot{\theta}_k T/2)$$
$$y_{k+1} = y_k + v_k \mathrm{T}\cos(\theta_k + \dot{\theta}_k T/2)$$

$$\theta_{k+1} = \theta_k + \dot{\theta}_k T$$

where $v_k=(v_{Rk}+v_{Lk})/2$, $\dot{\theta}_k=(v_{Rk}-v_{Lk})/l$.

3. Update cycle. The whole simulation process of the simulator in one sampling interval is shown in the following recursive process

```
t=kT
    Receive Control Commands
    Update Robots Motion
    Compute Ball Motion
    Check Collision and Kick
    Compute Collision and Kick Reaction
    Let k=k+1
Loop
```

III. A demonstration: ball passing

A demonstration of the simulator is shown by visualizing the ball passing strategy [9]. There are three identical robots to pass the ball alternatively. For a passing movement among three soccer robots, we assume that relative locations of three robots are initially in a ready formation for passing. As the passing cycle starts, one robot goes to a position behind the ball to kick the ball toward a designed direction and the other two robots move to suitable locations to anticipate a possible pass, following a collision-free trajectory generated by a path planning system. For this simulation, the simulation tick time is set as T=0.02 sec and is equal to the sampling time. All computations about strategy, including ball motion prediction and robot trajectory planning, should be completed in a time duration less than T. This imposes a demand of the computational speed in practical realization of the ball passing strategy among multiple mobile robots. The control command at time $(k+1)T$ can be computed by all motion data accessed at time $t=kT$. The simulator could be used to verify the path planning, velocity planning, tracking control and tune the parameters for performance visualization. The simulator is written in VB.A snapshot of trajectories of three mobile robots for the ball passing is shown in Figure3.

IV. Conclusion

In this paper, a simulator for dynamic motion of multiple mobile robots and ball is built for soccer strategy development and realization by path planning, and tracking controller design, thus could serve as a platform whose simulation results can be transferred to real soccer robots for performance validation. A demonstration is shown for ball passing strategy among three mobile robots with changing formation. Our future work is to embed sensory capabilities into the simulator and to implement more coordination strategies of multiple robots on the simulator.

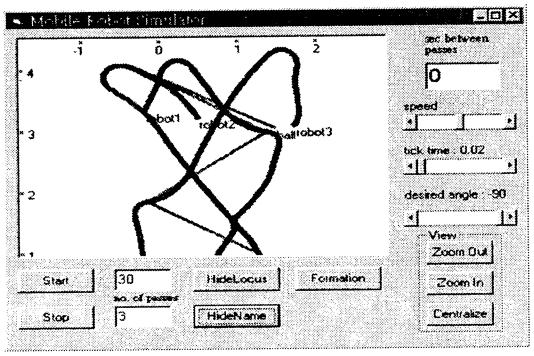


Fig.3. Snapshot of the simulator.

Acknowledgment. This research was supported by National Science Council of R.O.C. under contract NSC 91-2212-E-001-001.

References

[1]Hong B, Gao O, et al. (2000), Robot Soccer Simulation Competition Platform Base on Multi-agent, FIRA-KAIST Cup Workshop.

[2]Ye W, Vaughan R, Sukhatme G, Heidemann J, Estrin J, Mataric M (2001), "Evaluating control strategies for wireless-networked robots using an integrated robot and network simulator," 2001 IEEE Int. Conf. on Robotics and Automation, pp.2941-2947.

[3]Balch T, Hybinette M (2000), "Social potentials for scalable multirobot formations," 2000 IEEE Int. Conf. On Robotics and Automation, pp.73-80.

[4]Gerkey B, Vaughan R, Stoy K, Howard A, Sukhatme G, Mataric M (2001), "Most valuable player: A robot device server for distributed control, IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp.1226-1232.

[5]Corradini ML and Orlando G (2001), Robust Tracking Control of Mobile Robots in the Presence of Uncertainties in the Dynamical Model, Journal of Robotic Systems, 18(6):317-323.

[6]Buck S., Beetz M. and T. Schmitt T. (2002), "M-ROSE: a multi robot simulation environment for learning cooperative behavior," Distributed Autonomous Robotic Systems 5, pp.197-214.

[7]Noda I, Matsubara H, Hiraki K, and Frank I (1998), "Soccer server: a tool for research on multiagent systems," Applied Artificial Intelligence, 12(2-3):233-250.

[8]Lee T, Nehmzow U, and Hubbold R (1999), "Computer simulation of learning experiments with autonomous mobile robots," Proc. TIMR'99 "Towards Intelligent robots".

[9]Liang TC and Liu JS (2002), "Coordinated trajectory planning and formation control of soccer robots to pass a ball cyclically," Distributed Autonomous Robotic Systems 2002, Springer-Verlag.

[10]Messom C (1998), "Robot soccer: sensing, planning, strategy and control, a distributed real time intelligent system approach,"3rd Int. Conf. Artificial Life and Robotics.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Embodied AI in Humanoids

Henrik Hautop Lund Luigi Pagliarini Leonid Paramonov Morten Winkler Jørgensen

Maersk Mc-Kinney Moller Institute for Production Technology
University of Southern Denmark, Campusvej 55, 5230 Odense M., Denmark

hhl@mip.sdu.dk luigi@mip.sdu.dk leonid@mip.sdu.dk mwj@mip.sdu.dk
www.adaptronics.dk

Abstract

In contrast to the top-down approach of equipping a humanoid with as many sensors, motors, power, etc. as possible, we developed a bottom-up approach to the construction of humanoids – an approach that attempts to minimize the robot complexity. The approach is shown with the development of the Viki humanoid that won the RoboCup Humanoids Free Style World Championship 2002. For the development of the bottom-up approach we find inspiration from recent work in embodied artificial intelligence that puts emphasis on the correspondence and interrelatedness between material, electronic hardware, energy use, and control. By finding the right balance and relationship between these components of the system, it becomes possible to develop biped walking and other humanoid behaviors with much simpler hardware and control than is traditionally envisioned for humanoids. Indeed, the Viki humanoid robots were able to win the world championship though they include much less sensors, motors and energy use than their competitors.

Introduction

In order to explore embodied artificial intelligence, we developed the humanoid robot Viki. It is our working hypothesis that morphology plays a crucial role in intelligence and intelligent system. Unfortunately, in the past, many researchers have neglected the investigation of the role of morphology. In artificial intelligence robotics, many researchers looked at optimization and adaptation of control on a fixed hardware platform, and therefore optimized to a specific hardware platform only, and not to the overall problem solving behavior. With the Viki humanoid work, we would like to emphasize that optimization towards the best behavior on a global task should happen by finding the right balance between hardware, material, energy use, and software. Indeed, for the first prototype, our software is fairly simple, and becomes a primitive form of a behavior-based system, inspired by the work on behavior-based robotics [1], and our own work on using behavior-based systems for edutainment robotics purposes [2,3]. By purpose, we chose a simple form of control in order to show that it is the right bottom-up mentality in the design process that leads to the result rather than the control in isolation. Indeed, we find that only few motors and inexpensive sensors are necessary, and that a simple control is sufficient, if they are used in a bottom-up approach where all components (hardware, software, mechanics, energy use, control, etc.) are designed for the integration to achieve the overall behavior.

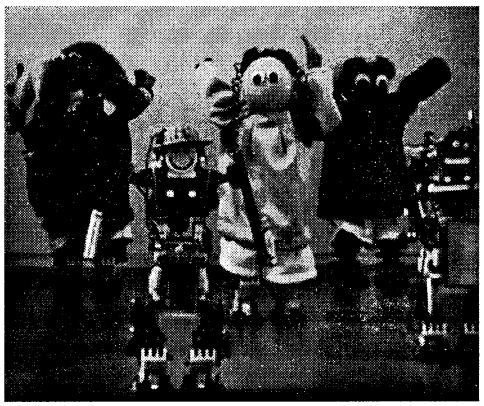


Figure 1. Viki humanoid robots making dancing performance.

Mechanical Structure of the Robot

The mechanical structure of the robot is based on several simple parallel or non-parallel prismatic structures which allow us to simplify the

mechanical design so that very few actuators are needed for mobility.
In the robot 5 DC 6 Volt motors are applied for the upper body, the hips, the legs rotation, the arms shift. We designed specific plastic-made units (12 units in total) to increase stability of the robot and to allow motors installation (see fig.1). The units were made so that they fit with the standard LEGO sizes, which allowed us to use a number of LEGO units in the final robot's structure where it was possible from durability point of view.

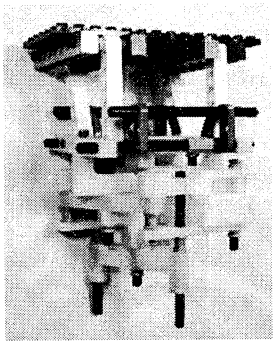


Figure 2. Plastic structure compiled

All the motions of the robot are assumed quasi-static on the design stage but it is possible in the later prototypes apply some dynamic motions using principles, which we introduced in our recent publication [4].
The upper body structure uses one motor for both the main weight shift and legs extraction-contraction. One non-parallel prismatic structure is used. The operation is based on combination of two pulling strings (at the left side and at the right side of the upper body) and legs structures loaded by springs (rubber bandages in the simplest case). Strings actuated by the same motor in a way that while one string is pulled in by a round pulley the other one is released with the same speed from another pulley. The properly adjusted structure works so that the main weight shifts in between two extreme positions. While the main weight is shifting it performs a very small effect on the motion of the legs. But when the extreme position is reached and the main weight is shifted, the leg extraction from one side and contraction from the other side is started (high load phase figure 3).
The extraction of one leg together with contraction of the other one leads to balancing of the robot on extracted leg. The maximum possible extraction of one leg allows taking the other leg off the ground for about 2-3cm. So, it is possible to perform a step. The step action is performed by hips structure that uses one parallel prismatic structure actuated by one motor in the middle. So, depending on which leg is on the ground, the left step forward or the right step back (for instance) will be performed by the same hips action.

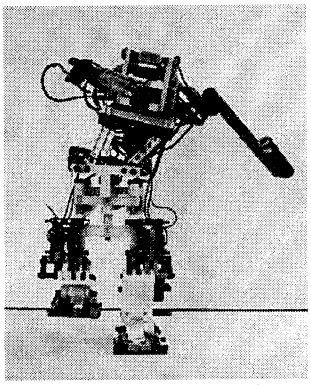


Figure 3. Height load phase.

Two motors for rotation of the legs motor are installed in left and right hips units so that it is possible to rotate one leg for more than a half of a revolution. As a result not only the straight step could be done but also steps combined with rotation around the foot place, which turn the robot around the standing leg and also allow the "swing" leg to prepare for the next action.
Finally, one single motor is used to shift both arms to the same side.

Electronic hardware

Viki's control system is built to be minimalistic in nature, modular and highly reconfigurable. The control system is centered on a rather powerful CPU with the peripherals connected on an I2C bus with a single motor controller being interfaced directly to the CPU as the exception. This allows for quick reconfiguration and can be expanded or shrunk as desired for the particular purpose. Since Viki is controlled by five motors, four motor drivers with local computational power was attached to the bus along with eight analog to digital converters for feedback from the angular sensors. Two 3.6-volt lithium-ion polymer batteries[1] connected in series to offer 7.2 volt power to the entire system.
The CPU in Viki is a AMD186ES micro controller which essentially is an Intel 186 clone

[1] Battery prototypes sponsored by Danionics

wrapped in a micro controller layer that, amongst others, offer two UARTS, timers and several bi-directional I/O pins. The micro controller is supported by 512Kb of working RAM and approximately 0.7Mb of FLASH disk for program storage and file-creation. The system runs an embedded DOS compatible with the IBM DOS allowing for program-development on a PC with any DOS compiler.
The I2C interface to the CPU was implemented in software on top of the DOS and got therefore limited to a bandwidth of 0.1 kHz.
To control the motors used for actuating Viki's legs, hips and upper body a distributed motor control was implemented outsourcing the workload of maintaining speed and direction of the motors. A PIC16F876 micro controller was programmed to act as an I2C slave that could receive information on and maintain the speed and direction of four motors. If the CPU needs the motor for left leg to rotate counter clockwise, it would send a command to the PIC16F876 that makes the motor do so. This is a very low cost and flexible solution that allows for any number of "outsourced motor control" to be attached to the bus. For Viki only a single was needed.
Both solutions, the outsourced and the internal, were using the L293D, a full H-bridge motor driver with internal protection diodes, to drive the motors allowing applying ±7.2 volt to the motors.
For feedback on the angular position of hips, the rotation of the legs and the displacements of the arms four commercial linear potentiometers costing 0.70 US$ each was built into Viki at appropriate places to offer a continuous signal for A/D conversion.
Flexing of the legs is done by displacing the upper body thus pulling a string, and extraction is done by rubber bandages. Two extremities sensors (switches) were implemented to detect the end of these motions.
All six feedback values was converted to an eight-bit value using the PCF8591 analog-to-digital converter from Philips interfaced to the CPU via the I2C bus.
The design of the control architecture and the actuation of the body allow Viki to operate for approximately half an hour before a recharge is needed. Minimizing the CPU further to an even more simple architecture as the current one consumes about 2/3 of the total power dissipation may increase this significantly. In contrast to that an even simpler solution with a PIC micro controller would consume only 7% of the total power dissipation.

The software

We decided to work out some strictly sequential control routines. Therefore, the Behavior-Based Algorithm that we implemented, at a primitive behaviors level, had to control one single motor at the time. Despite of such limitations, results were quite satisfying.
With the sensors, we could measure the arms position (one rotation sensor); the hip position (one rotation sensor); the legs rotation (two rotation sensors, one for each leg); the robot's upper body oscillation (two contact sensors, one for the left and the other for the right side upper body movements).
We also implemented two so-called "virtual" sensors, a time measure we used to let the robot keep on bending the upper body to one side in order to lift up on the two legs. In general, inputs were quite "noisy" and sensors, as usually happen with cheap ones, differed one from the other quite a lot, both in performance and measurements (i.e. in accuracy and fidelity). Also the motors performance did not show accuracy and coherence. This was not due to batteries power consumption dynamics, only. In practice, they all differed in speed and, consequentially, in power.
Due to the high level of turbulence in both the input and the output flow, we needed to elaborate a fairly accurate and robust calibration routine to be run on each robot. Such a process was somehow necessary to control the robots behavior at the best, and it would decide parameters for inputs (i.e. both the real and the virtual ones) and output flow (i.e. motors speed, timing and etc.). Such parameters were then recorded on a log file and downloaded within the robots hard disk to be used with our behavior control algorithm.

The simple algorithm

To control Viki, we decided to develop a behavior-based algorithm. To do so, we first implemented (and carefully tested) a whole set of *primitive* behaviors. Both because of theoretical issues and because of the partial unreliability of the input-output system, we had to shape out such primitive behaviors in a very 'molecular' way. Basically, each single movement or fraction of it was coded.
After the primitive layer was drawn, we started up by building a second layer trying to combine, as many as possible, significant and compatible couple of primitives, the *mates*. Two primitive

behaviors do not need to belong to two different input-output systems, but they can also be a proper combination, of the same one. For example, (A) move arms to the center, plus, (B) move arms to the left.
The third layer is the *patterns*. Patterns, assemble together both single primitives and single mates. Also in this case, we applied the same criteria of significance and compatibility used to build the mates. For example, in such robots, a pattern for the upper-body control is built in such a way that balancing the body is at the principal level of priority. Therefore, moving the upper-body from left to the center can be done by: (A) Move upper-body to the left to center, first, and then (B) move arms to the center, plus, (C) move arms to the left. On the contrary, a sequence, or pattern, like (B and C) and (A) would bring the robot in a very unstable state. (This is when robot the body is still on the left side and the arms move to the center, therefore loosing their function of counterweight).
After this second step, a further behavior level, the *movements* one, was realized. A movement can be any possible sequence of *primitive* + *mates* + *patterns*. For example, in the "lift the left leg" routine, one of the most simple and reliable ones we built, we probably had many more routines than one would expect. They were (we here suppose the upper-body is known to be on the left): Move upper-body to the left touch sensor off; Timer Start; Move upper-body T milliseconds. (i.e. to the center); Timer and Motor Stop; Move arms to the center; Move arms to the right extreme; Move upper-body to the right touch sensor on; Timer Start; Move upper-body right for T milliseconds. (i.e. lift the left leg); Timer and Motor Stop.
Once the movements layer was ready we moved on the *actions* level. Again, an action can be any possible sequence of *primitive* + *mates* + *patterns* + *movement*. For example, an action can be the (A) lift the left leg described above, plus, (B) move hip left.
By combining movements (and, of course, their sub-categories) we, finally, reached the *basic behaviors* level. A basic behavior, for example, could be the one just described, plus, a "move the left leg down" mate (i.e. it simply counts time before turning off one motor). The resulting basic behavior is "one step forward left".
By assembling basic behaviors (and their sub-categories) we could then obtain *intermediate behaviors*. For example, "one step forward left" plus "one step forward right" produced a "walk forward" intermediate behavior.
At this point the game is done since we only needed to link intermediate levels (and their sub-categories) into chains of *high-level behaviors* to obtain beautiful, efficient and coordinate acts such as the twelve humanoid robots dance, rewarded as the best free style show in RoboCup 2002. With this control architecture, it is very easy also to design numerous other behaviors.

Discussion and conclusion

For a future development of Viki, we envision different issues to be touched upon. First of all, we will include a third battery, in order to be able to control all motors simultaneously, ensuring much faster movements.
We developed the Viki humanoid in order to show the importance of a bottom-up approach to the design of humanoid robots. In such an approach, one should look at finding the right balance between hardware, software, mechanics, material, and energy use. Our experiments show that we were able to win the RoboCup Humanoids Free Style World Championship 2002 with such an approach, even though / because it resulted in a much more inexpensive design than the competitors, and because it resulted in the focus on achieving the overall behavior by an interplay between the necessary components. In general, we hope that this and other similar experiments/results can open a discussion in the scientific community regarding the importance of morphology in intelligence and intelligent systems.

Acknowledgement
The Viki Humanoid robot development is sponsored by Entertainment Robotics.

References

[1] R.A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1):14--23, 1986.
[2] H. H. Lund. Adaptive Robotics in Entertainment. Applied Soft Computing, 1:1, pp. 3-20, 2001
[3] H. H. Lund and L. Pagliarini. Robot Soccer with LEGO Mindstorms. In Asada (ed.) Proceedings of RoboCup'98, LNAI 1604, Springer-Verlag, Heidelberg, 1999.
[4] L. Paramonov, and H. H. Lund "A minimalistic Approach to Humanoids". In Proc. of Humanoids 2001, IEEE Press, 2001.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A Kernel Based Neural Memory Concept and Representation of Procedural Memory and Emotion

Tetsuya Hoya
Laboratory for Advanced Brain Signal Processing,
BSI RIKEN, 2-1, Hirosawa, Wakoh-City, Saitama 351-0198, Japan
e-mail: hoya@bsp.brain.riken.go.jp

Abstract

This paper explores a general concept of two-stage dynamic memory by means of Gaussian kernels and associated processing mechanisms. The memory system can be viewed as the extension to the previous work of hierarchically arranged generalised regression neural network (HA-GRNN) and its evolutionary process, in which two psychological functions, attention and intuition, are interpreted. Within the proposed mechanism, both the functionality of procedural memory and emotion are newly modeled together with the aforementioned two psychological functions, in terms of the associated interactive processes between the short-term (STM) and long term memory (LTM).

1 Introduction

To elucidate the cognitive processes of humans and the relevant psychological functions is a challenging but inevitable subject for the development of artificial intelligence (AI). In psychology, the study of human learning and memory has long been established [1]. In the study, a great variety of models have been proposed, which suggests that the human memory system is a central part of the cognitive process.

In the artificial neural network field, multilayered perceptron neural networks (MLP-NNs) have played a significant role especially in the study of pattern recognition tasks [2]. However, it is now well-known that in practice the learning of the MLP-NN parameters by a backpropagation (BP) [3] type algorithm quite often suffers from becoming stuck in local minima and requiring long period of learning, both of which are good reason for detracting their utility in on-line processing. Such networks also need for training from scratch when new training data is used. MLP-NNs therefore have appeared as unsuitable candidates for elucidating the learning mechanism of the brain [4].

In the early 1990's, Specht rediscovered the effectiveness of kernel discriminant analysis [5] within the context of artificial neural networks. This led him to define the notion of a probabilistic neural network (PNN) [6]. Subsequently, Nadaraya-Watson kernel regression [7, 8] was reformulated as a generalised regression neural network (GRNN) [9]. In the neural network context, both PNNs and GRNNs are categorised into a family of radial basis function neural networks (RBF-NNs) [10] in which the hidden neurons are represented by Gaussian response functions (or, Gaussian kernels). It was reported that the RBF-NNs are also biologically appealing [11].

The advantage of PNNs and GRNNs is that they are essentially free from the 'baby-sitting' required for the MLP-NNs, i.e., the necessity to tune a number of network parameters to obtain good convergence rate or worry about any aforementioned numerical problems. By exploiting the property of PNNs and GRNNs, simple and quick incremental learning is possible due to their inherent memory-based architecture, whereby the network growing/shrinking is straightforwardly performed [12]. Moreover, in [13], it is reported that a PNN even exhibits a capability to accommodate new classes whilst maintaining a reasonable generalisation performance. These then give a substrate for modeling psychological functions [12], which leads to a framework for the development of brain-like computers, or, in a more true sense of, 'artificial intelligence'. On the basis of the remarks in [14], the aforementioned features of PNNs are considered to be crucial for the development of brain-like computers.

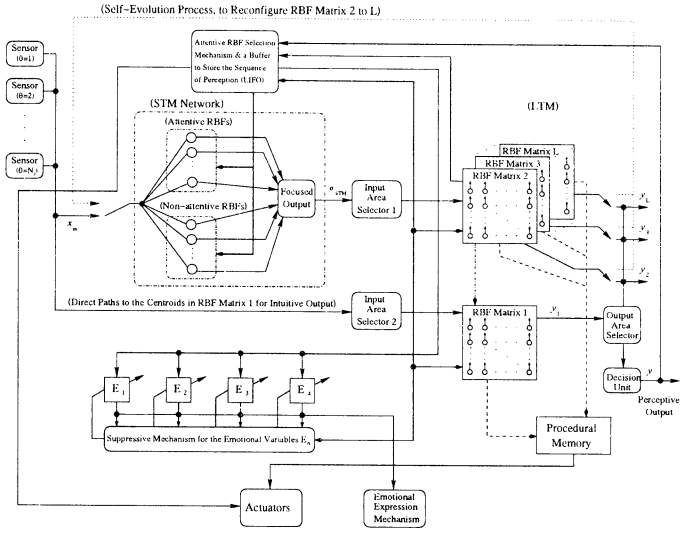


Figure 1: The new memory system.

2 The Two-Stage Dynamic Memory Concept

Fig. 1 shows the two-stage dynamic memory system based upon the RBF (kernel) networks. As in the figure, the STM consists of 1) a single modified RBF-NN with an attentive selection mechanism, 2) a buffer to store the sequence of perception given as the interactive processing within the Gaussian kernel matrices in the LTM, 3) four emotional variables connected to the emotional expression mechanism of AI, and 4) a suppressive mechanism for the four emotional variables. In contrast, the LTM consists of 1) input/output area selectors, 2) the Gaussian kernel (RBF) matrices, and

3) memory arrays representing the procedural memory.

2.1 The STM Network

The STM network of the proposed memory system has a simple two-layered structure in itself as that of the HA-GRNN [12]. As in Fig. 1, within the STM network, the RBFs can be divided into two parts; 1) the attentive RBFs which tend to generate relatively higher activations for a particular set of incoming vectors and 2) the rest. The main role is thus to temporarily buffer the incoming sensory vectors and eventually transfer (some of) them to the LTM. The STM network output $\boldsymbol{o}_{STM}$ is given as a vector rather than a scalar:

$$\boldsymbol{o}_{STM} = [\boldsymbol{o}'_{STM}; \theta]^T \tag{1}$$

where $\boldsymbol{o}'_{STM} = [o_1, o_2, \ldots, o_L]^T$ is the feature vector transferred from the STM network and θ denotes the sensory input number (i.e. $\theta = 1, 2, \ldots, N_s$) for which the feature vector was obtained (e.g., $\theta = 1$: microphone, 2: CCD camera, ..., with varying L). The learning of the STM network is then summarised as follows:

Step 1: (At an initial stage) if the number of the centroids is less than M_{STM}, add an RBF with its centroid vector $\boldsymbol{c}_i = \boldsymbol{x}$ in the STM. Then, set $\boldsymbol{o}'_{STM} = \boldsymbol{x}$.

Step 2: Otherwise,

1) If the activation of the least activated centroid (h_j, say) $h_j < th_{STM}$ (where th_{STM} is a given threshold), replace it with a new one with $\boldsymbol{c}_j = \boldsymbol{x}$ and set $\boldsymbol{o}'_{STM} = \boldsymbol{x}$.

2) Otherwise,

$$\boldsymbol{o}'_{STM} = \lambda \boldsymbol{c}_k + (1 - \lambda)\boldsymbol{x} \tag{2}$$

where $\boldsymbol{c}_k$ is the centroid vector of the most activated centroid (k-th, say) h_k and λ is a *smoothing* factor ($0 \leq \lambda \leq 1$).

In Step 2 above, a smoothing factor λ is introduced in order to regulate how fast the attentive focus of the STM network is changed by the newly incoming sensory vector which has not appeared before. This can then be regarded as 'selective attention' to a particular event/object. For instance, if the factor is small, $\boldsymbol{o}'_{STM}$ becomes more like $\boldsymbol{x}$, then this shows a sign of 'carelessness'. In contrast, if the factor is large, the STM network becomes 'sticky' to particular patterns. The above learning scheme can then be considered as a process similar to last in first out (LIFO) stack.

2.2 The Input/Output Area Selectors

In the LTM, the incoming STM network output $\boldsymbol{o}_{STM}$ is firstly transferred to the input area selector as in Fig. 1. The role of the input area selector is to collect the input nodes of all the RBF units in the RBF matrices that belong to the particular area corresponding to the sensory input specified by the number θ and then forward the STM network output $\boldsymbol{o}_{STM}$. In contrast, the output area selector will gather all the output values of the RBF units, within the area for a particular pattern classification task, and form a decision unit (by following the 'winner-takes-all' strategy) that will tell us the final classification result. The right part of Fig. 2 illustrates an example of this; there are two modal columns depending upon the types of stimuli; one for auditory and the other for visual. Then, the respective columns are subdivided further into several areas corresponding to the specific

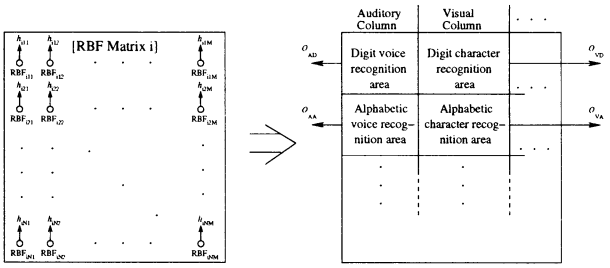


Figure 2: The i-th RBF matrix and its modality-dependent areas.

cognitions/perceptions. In the right part of Fig. 2, it is considered that there are four distinct areas, i.e., digit character/voice recognition (visual/auditory) and alphabetic character/voice recognition areas (visual/auditory), with the respective classification results, o_{VD}, o_{AD}, o_{VA}, and o_{AA}. In each area, a PNN/GRNN (with the assembly of the RBF units and a decision unit) so formed thus represents the dynamic memory for the corresponding pattern classification. In Fig. 1, the final classification result y from the RBF matrices is then given as the largest value among all the weighted outputs of the particular area within the RBF Matrix (1 to L):

$$y = \max(y_1, y_2, \ldots, y_L), \tag{3}$$

where $y_i = v_i \cdot o_i$, o_i is the output from the particular area in RBF Matrix i, and the weights $v_1 >> v_2 > v_3 > \ldots > v_L$. Note that the weight value v_1 for RBF Matrix 1 is much larger than the others. This discrimination indicates the generation of the 'intuitive output'. In practice, the intuitive outputs can be exploited to obtain e.g., faster classification results [12].

2.3 A Layer of RBF Matrices

As in Fig. 1, a layer of RBF matrices represents the LTM of the proposed new memory. Each RBF matrix (from 1 to L) consists of the RBF units, and multiple neural networks can coexist within the RBF matrices by way of the addressing pointers to other RBF units. As in the LTM networks of HA-GRNN, the hierarchical structure of the RBF matrices is constructed based upon the 'significance' or 'attractiveness' of information represented by the activation of centroids. In Fig. 1, RBF Matrix 1 indicates a collection of RBF units which can generate intuitive outputs as LTM Net 1 within the HA-GRNN [12], as there is no buffering process by the STM. In contrast, RBF matrices (2-L) represent the normal LTM. (Note that in practical implementation RBF Matrix 1 is not necessarily apart from other matrices and thus it does not mean there is a special agency for 'intuition'.) Although this layered LTM principle is inherited from that of HA-GRNN, the formation of each LTM layer (i.e., RBF matrix) here is based simply upon an assembly of RBF units, whilst the formation within the HA-GRNN is limited by a single GRNN. Hence, the neural representation is quite versatile and not restricted to the conventional fixed representations, such as layered networks or two-dimensional neuronal map (as in Kohonen's self-organising maps), which leads to a more brain-like memory representation.

2.4 The RBF Unit

In the kernel matrices, as shown in Fig. 3, each RBF unit can be seen as a memory element and is composed of

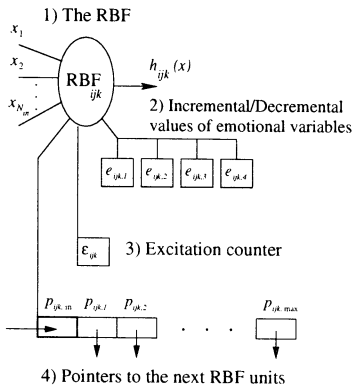


Figure 3: A Gaussian kernel (RBF) unit.

1) Gaussian kernel (or RBF) itself, 2) an excitation counter ϵ_{ijk}, 3) incremental/decremental values e_{ijk} to update the emotional variables E_n $(n = 1, 2, 3, 4)$ within the STM, and 4) addressing pointers $p_{ijk,m}$ $(m = 1, 2, ..., \max)$ which enable the RBF to *link* with other RBF units specified by the addresses.

The activation (output) of the kernel unit at the j-th row and k-th column of the i-th RBF matrix is given as

$$h_{ijk}(\boldsymbol{x}) = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_{ijk}\|_2^2}{\sigma^2}) \quad (4)$$

where $\|\ldots\|_2^2$ denotes the L_2 norm, $\boldsymbol{c}_{ijk}$ is the centroid vector, and σ is the radius. In another point of view, the Gaussian response function in (4) can be also translated as a measurement of similarity, since as the input vector $\boldsymbol{x}$ becomes closer to the centroid vector $\boldsymbol{c}_{ijk}$, the output $h_{ijk}(\boldsymbol{x})$ is increased.

Then, by exploiting the addressing pointers, a single RBF unit can be also seen as a multiple-input-multiple-output (MIMO) system and thereby the 'memory-chain' concept is developed. To illustrate the memory-chain concept, an example using four RBF units is given in Fig. 4 (note that in Fig. 4 only the RBF and the addressing pointers are depicted for convenience). In the figure, suppose that for a certain

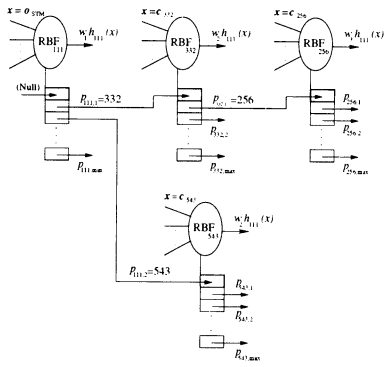


Figure 4: An example of a memory-chain represented by the RBF units.

modality RBF_{111} was most strongly activated by the STM output vector $\boldsymbol{o}_{STM}$ among all the RBF units within a particular area of RBF Matrix (1 to L) and that the strong activation of RBF_{111} is transferred to the remaining three RBF units in the chain, i.e., RBF_{332}, RBF_{256}, and RBF_{543}, via the addressing pointers. Then, the four associated RBF units become active for the subsequent perceptive processes. This memory-chain concept can be extended to, for instance, form a network for digit voice classification task, i.e., a star-shaped network consisting of ten distinct clusters of RBFs (i.e., each consists of single-layered sub-networks) describing the pattern space from /ZERO/ to /NINE/), each having a set of RBF units for the corresponding digit represents a network for the digit voice classification task. Then, the actual pattern classification can be performed as: 1) forward the incoming feature vector to all the RBF units within the neural network represented by the star-shaped network for each cluster (or, sub-network), obtain the sum of output values, (in this case, we will have ten different values) and 3) applying the decision unit to the star-shaped graph (and followed by the 'winner-takes-all' strategy), find the maximum amongst the ten sum. values and the class number which corresponds to the digit.

Similar to the memory-chain concept above, Minsky developed a concept called as 'Knowledge-line' (K-line) [15] in which each node in a semantic network is linked to other networks (agencies) via the K-lines. However, unlike K-lines, the proposed memory-chain concept also takes account for the generalisation capability in each node (RBF), which is advantageous in practical sense, i.e., the memory space required can be small compared to that of K-lines.

2.5 Representation of Procedural Memory

If the AI has skilled a learned sequence of particular actions/movements, then, in the context of memory representation, some information about the perceptive sequence contained in the memory-chain of RBF units will become solid and be transformed in a different form of representation. For clarity, this transformation is represented by the area for procedural memory shown in the right corner in Fig. 1. In such representation, all the RBF units except the first will have much smaller variance , e.g., in (4), the variance asymptotically approaches a certain small value. The procedural memory then contains only the centroid vectors transferred from the RBF units in the chain plus a special link connected to the first RBF unit within the normal LTM (i.e., the RBF matrices) for receiving the STM output $\boldsymbol{o}_{STM}$ (or, filtered sensory input vector) and executing the sequence of real actions/movements. This indicates that the experienced actions can be performed without being conscious [1]. In other words, during the execution, the sequence described by the procedural memory-chain is not monitored by the STM at all. For instance, let us turn back to the example of the memory-chain in Fig. 4. Provided that here the memory-chain with the three RBF units, RBF_{111}, RBF_{332}, and RBF_{256}, reside in a part of the motor area and the sequence describes a kinetic action, it can be formulated that the values of their activations directly controls the actuators of robots. For instance, the values stored within the centroid vectors could be used as, e.g., the target positions for PID controllers.

3 Interpretation of Emotion

In Hobson's argument [16], the psychological function 'emotion' is considered as one of the fundamental components describing consciousness. In the philosophical context, it is considered/implied that short-term memory plays a key role for describing the functionality of consciousness (e.g., [17]). On the other, an ethological model for entertainment robots (such as SDR-3X/AIBO) with emotional expressions has appeared in the literature [18]. The model-

ing, however, seems to resort to rather static symbolic mechanisms and it is thus considered that its extension to more reconfigurable and flexible brain-like representation is questionable.

Inspired/motivated by these studies, in this paper, the STM with four emotional states is considered as in Fig. 1. In the figure, the four emotional variables E_n $(n = 1, 2, 3, 4)$ representing i) pleasure, ii) anger, iii) sadness, and iv) amenity, respectively. Accordingly, in Fig. 3, every RBF unit RBF_{ijk} comes also with four auxiliary variables $e_{ijk,n}$ $(n = 1, 2, 3, 4)$ used for updating the corresponding emotional variables within the STM.

As in Fig. 1, a buffer is used to temporarily store the locations/activations of subsequently excited RBF units during the successively occurred perceptions. For instance, the following scenario is considered in the context of pattern classification: if the AI performs a series of perceptions, 1) to look at a picture of his father (face image recognition) passed by a few years ago, 2) to remember its father's voice (voice identification), and 3) to remember his pen (object image recognition), then it is naturally considered that the three pattern classification tasks, 1) face image recognition, 2) voice identification, and 3) object image recognition, are subsequently involved. Now, let us assume that within the RBF matrices there is already formed the memory representation (or, a neural network) for each pattern classification task and suppose that RBF_{111} was firstly activated when the feature vector extracted from the image of the father's face was received from the STM. Then, suppose that the two RBF units RBF_{332} and RBF_{256} in the memory-chain shown in Fig. 4 were successively activated, which respectively match the feature vector of his utterance and the image of his pen. Then, the three perceptive processes involve the respective updates of the emotional variables E_n $(n = 1, 2, 3, 4)$, e.g., by a simple summation operation:

$$E_n = E_n + e_{111,n} + e_{332,n} + e_{256,n}$$

Note that here the interpretation of the memory-chain in Fig. 4 is totally different from the example of procedural memory in Section 2.5; since, except the first (i.e., RBF_{111}), the RBFs so activated correspond to the consequent of 'recalling' the previously stored events/objects. Namely, for the recalling, it is postulated that the activation of $\mathrm{RBF}_{332}/\mathrm{RBF}_{256}$ is led by not the actual sensory input (i.e., the image of its father) but the linkage due to the RBF pointers. In such a case, the input to those two RBFs is given simply as their respective centroid vectors (as illustrated in Fig. 4). Then, since the activation was caused by the RBF linkage, the actual output values h'_{332} and h'_{256} so obtained may be set to the decaying value of $h_{111}(\boldsymbol{x})$:

$$\begin{aligned} h'_{332}(\boldsymbol{x}) &= w_2\, h_{332}(\boldsymbol{c}_{332}) h_{111}(\boldsymbol{x}) = w_2\, h_{111}(\boldsymbol{x}) \quad (5) \\ h'_{256}(\boldsymbol{x}) &= w_3\, h_{111}(\boldsymbol{x}) \end{aligned}$$

where $w_1 (= 1, \text{say}) > w_2 > w_3$. This may represent 'memory-fading' within the association of memory.

4 Conclusion

In this paper, a novel two-stage dynamic memory concept has been proposed. The memory system exploits the properties of kernel regression neural networks. Based upon the memory concept, the three psychological functions, attention, intuition, and emotion, have been modeled in terms of the interactive processes between the STM and LTM. In the LTM, a layer of RBF matrices and the associated mechanisms have newly been proposed as an extension of the LTM networks in the HA-GRNN [12]. Then, the representation of procedural memory and interpretation of emotion by an artificial model have been attempted. This paper has focused only upon presenting a framework for modeling psychology-oriented brain functions. Due to the limit of the space, many other aspects and features (e.g., the actual learning procedures) are left unexplained and thus a complete picture has not been given in this paper. The whole picture will be presented elsewhere. Future work is directed towards the development of the AI motivated by the proposed memory concept and justification of the effectiveness by observing and analysing the behaviour.

References

[1] J. A. Anderson, *Learning and Memory: an Integrated Approach*, 2nd Ed., New York: John Wiley, 1995.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford, U,K.: Oxford Univ. Press, 1996.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, chapter 8, Cambridge, MA: MIT Press, 1986.

[4] A. Roy, "Artificial neural networks - a science in trouble," *SIGKDD Explorations*, 1(2), pp. 33-38, 2000.

[5] D. J. Hand, *Kernel Discriminant Analysis*, Research Studies Press, 1984.

[6] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, 3, pp. 109-118, 1990.

[7] E. A. Nadaraya, "On estimating regression," *Theory Probab. Applic.*, 10, pp. 186-190, 1964.

[8] G. S. Watson, "Smooth regression analysis," *Sankhy, Series A*, 26, pp. 359-372, 1964.

[9] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, 2-6, pp. 568-576, 1991.

[10] P. D. Wasserman, *Advanced Methods in Neural Computing*, in chapter 8, "Radial basis-function networks" pp.147-176, New York: Van Nostrand Reinhold, 1993.

[11] T. Vetter, A. Hurlbert, and T. Poggio, "View-based models of 3D object recognition: invariance to imaging transformations," *Cerebral Cortex*, vol. 3, pp. 261-269, 1995.

[12] T. Hoya, "Notion of intuition and attention modeled by hierarchically arranged generalized regression neural networks," subject to revision for the *IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics.*

[13] T. Hoya, "On the capability of accommodating new classes within probabilistic neural networks," subject to revision for the *IEEE Trans. Neural Networks.*

[14] G. Matsumoto, Y. Shigematsu, and M. Ichikawa, "The brain as a computer," in *Proc. Int. Conf. Brain Processes, Theories, and Models*, Cambridge, MA: MIT Press, 1995.

[15] M. Minsky, *The Society of Mind*, Simon & Schuster: New York, 1985.

[16] J. A. Hobson, *Consciousness*, W. H. Freeman and Company: New York, Japanese translation; Tuttle-Mori Agency Inc.: Tokyo, 1999.

[17] D. C. Dennett, (1991). *Consciousness Explained*, Little Brown: Boston, 1991.

[18] R. C. Arkin, M. Fujita, T. Takagi, and R. Hasegawa, "Ethological modeling and architecture for an entertainment robot," *Proc. Int. Conf. Robotics and Automation*, pp.453-458, Seoul Korea, 2001.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Flexible Neural Network with PD-type Learning

Myeonghee Kim, Nobutoma Matsunaga and Shigeyasu Kawaji
Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami,Kumamoto 860-8555,JAPAN
mhkim@st.cs.kumamoto-u.ac.jp, matunaga, kawaji@cs.kumamoto-u.ac.jp

Abstract

Neural networks have been successfully applied to control field, especially for identification and control on nonlinear systems. But many problems on determining structure and parameters of neural network have been unsolved and these are usually determined on trial and error. To cope with this, a new scheme will be required for improving the learning ability. In this paper, a new neural network with PD-type learning algorithm is proposed, in which a new activation function is introduced so as to have a proportinal and differential functions. Thus it is expected that our new network is more flexible in the aspects of high learning ability than conventional P-type neural network.

1 Introduction

Neural networks have been successfully applied to control field, especially for identification and control of nonlinear systems [1][2]. It is well known [3] that the multilayered neural network has an ability to form a nonlinear mapping in order to match the given data. However, mechanism of creating the function and properties to possess after learning have not been proved enoughly, so it is not certain whether the necessary function can be obtained or not. Further, the problem of determining structure and parameters of neural network has been unsolved, and there are usually determined on trial and error. To cope with this, it will be required to find a new scheme for improving the learning ability.

In this paper, firstly by revealing input/output relationship of multilayered neural network, the learning mechanism of network and the role of hidden layer units are re-examined, and secondly we propose a new flexible neural network with PD-Type learning. From the analysis, it is found that each unit of hidden layers traces learning process independently. This suggests that, in order to improve learning speed, it is indispensable to train the slope of sigmoid function in each unit, which is fixed in the conventional neural networks. Increment of the slope can be decided by considering error change according to the change of the slope, so the learning algorithm, similar to backpropagation method, can be simply obtained. We refer to this algorithm as D-type learning, meaning to train the slope which is derivatives of sigmoid function. Meanwhile, the conventional algorithm to fix the slope of sigmoid function is called P-type. The advantages and disadvantages of P-type learning will be clarified and PD-type learning, the fusion of P-type and D-type, will be proposed. A simple test example is used to show the effectiveness of the proposed neural network.

2 Conventional Learning method

Input-output relational expression of unit in basic neural network can be represented as,

$$y = f(n) \tag{1}$$

$$n = \sum w \cdot x + \theta \tag{2}$$

where x is the input into the unit, y is the output from the unit, w is the connection weights between the units, θ is a bias. The sigmoid function

$$f(n) = \frac{1}{1 + exp(-an)} \tag{3}$$

is usually used, where a is the slope of the function.

Backpropagation, which is the learning algorithm of multilayered neural network, minimizes the squared error expressed by $E = \frac{1}{2}(t - y)^2$, using the steepest descent method, where t is teaching signal. Updated connection weight Δw_{jk} in hidden-output layer is given by

$$\Delta w_{jk} = -\eta_\omega \frac{\partial E}{\partial w_{jk}} = \eta_\omega \delta_k y_j \tag{4}$$

where

$$\delta_k = e \cdot f'(n_k) \tag{5}$$

And Δw_{ij} in input-hidden layer is given by

$$\Delta w_{ij} = -\eta_\omega \frac{\partial E}{\partial w_{ij}} = \eta_\omega \delta_j y_i \tag{6}$$

where

$$\delta_j = f'(n_j) \sum_k \delta_k w_{jk} \tag{7}$$

In these equations, the index i represent the neuron in input layer, j in hidden layer, and k in output layer. $f'(n)$ is the derivatives with respect to n, and η_ω is learning coefficient.

To re-examine the ability to form a function of multi-layered neural network, a neural network of three layers in Fig.1 is used, which includes one input layer, one output layer and 5 units of hidden layer. The neural network is trained with 5 arbitrary x-coordinate input values and the outputs of y-coordinate values given in table 1.

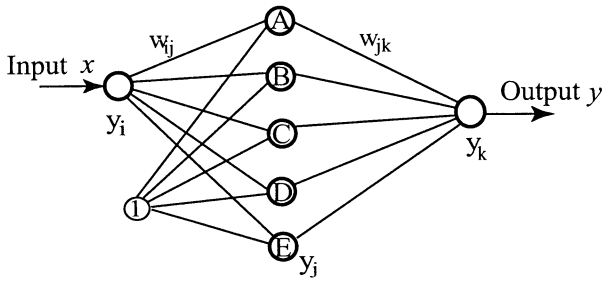


Figure 1: Neural Network Structure

Table 1: Teaching Signal

x	0.0	0.2	0.5	0.6	1.0
y	0.5	0.9	0.1	0.6	0.4

The connection weight is initialized randomly, the learning is performed by using moment method with the following equation, and the learning parameters are given as $\eta_\omega = 0.6$, $\alpha = 0.9$ [4].

$$\Delta w(t) = -\eta_\omega \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \tag{8}$$

During the learning, when x-values at the interval [0, 1] are inputted, the network sends out its output after training. Fig.2 shows the network output and the behavior of 5 units in hidden layer during the learning, where round points show the input/output data. The training number is repeated until the summation of squars error of 5 points becomes under 0.001. The internal structure of neural network, which is not understood from the value of connection weight, can be visible by diagramming input/output relationship.

Paying attention to the change of output state of each unit of Fig.2, the whole unit is not changed at a time, the unit C largely changes at first 1000 times training, after the change gets slow, then units A, E start to change. Then the unit D is changed. The learning is performed with an aspect that change of a unit is saturated, then the other unit is changed by causing low learning rate. If the whole unit can be changed at a time, it may improve the learning speed.

3 Learning of slope: D-type learning

In order to speed up the learning rate, new method to change the slope of sigmoid function will be proposed. In the conventional neural network, the slope is decided usually in trial and error, and fixed with the constant value. It is expected to improve learning effect by controlling the slope at each unit.

Self-control method of the slope of sigmoid function, using the steepest descent method similar to backpropagation method, decides the updated Δa by following equational algorithm.

$$\Delta a = -\eta_a \frac{\partial E}{\partial a} \tag{9}$$

where η_a is a learning coeffient. Δa is obtained in the output layer as follows,

$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial a_k} \tag{10}$$

In hidden layer,the following holds.

$$\frac{\partial E}{\partial a_j} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial n_k} \frac{\partial n_k}{\partial y_j} \frac{\partial y_j}{\partial a_j} \tag{11}$$

Specifically mentioned for this, we have

$$\Delta a = \eta_a \delta_a \tag{12}$$

$$\delta_a^k = e \cdot f_a'(a_k n_k) \tag{13}$$

$$\delta_a^j = f_a'(a_j n_j) \sum \delta_k w_{jk} \tag{14}$$

If $f(t)$ is sigmoid function, the derivatives are given as

$$f'(t) = f(t)(1 - f(t)) \tag{15}$$

from which $f_n'(\cdot)$, $f_a'(\cdot)$ are obtained by

$$f_n'(an) = af(an)(1 - f(an)) \tag{16}$$

$$f_a'(an) = nf(an)(1 - f(an)) \tag{17}$$

As seen in (12) $\sim$ (14), the increment of a can be calculated in similar way to resemble closely to backpropagation and don't need a long calculation time . We call this learning method with the new method to use derivatives of the slope of sigmoid function as "D-type learning".

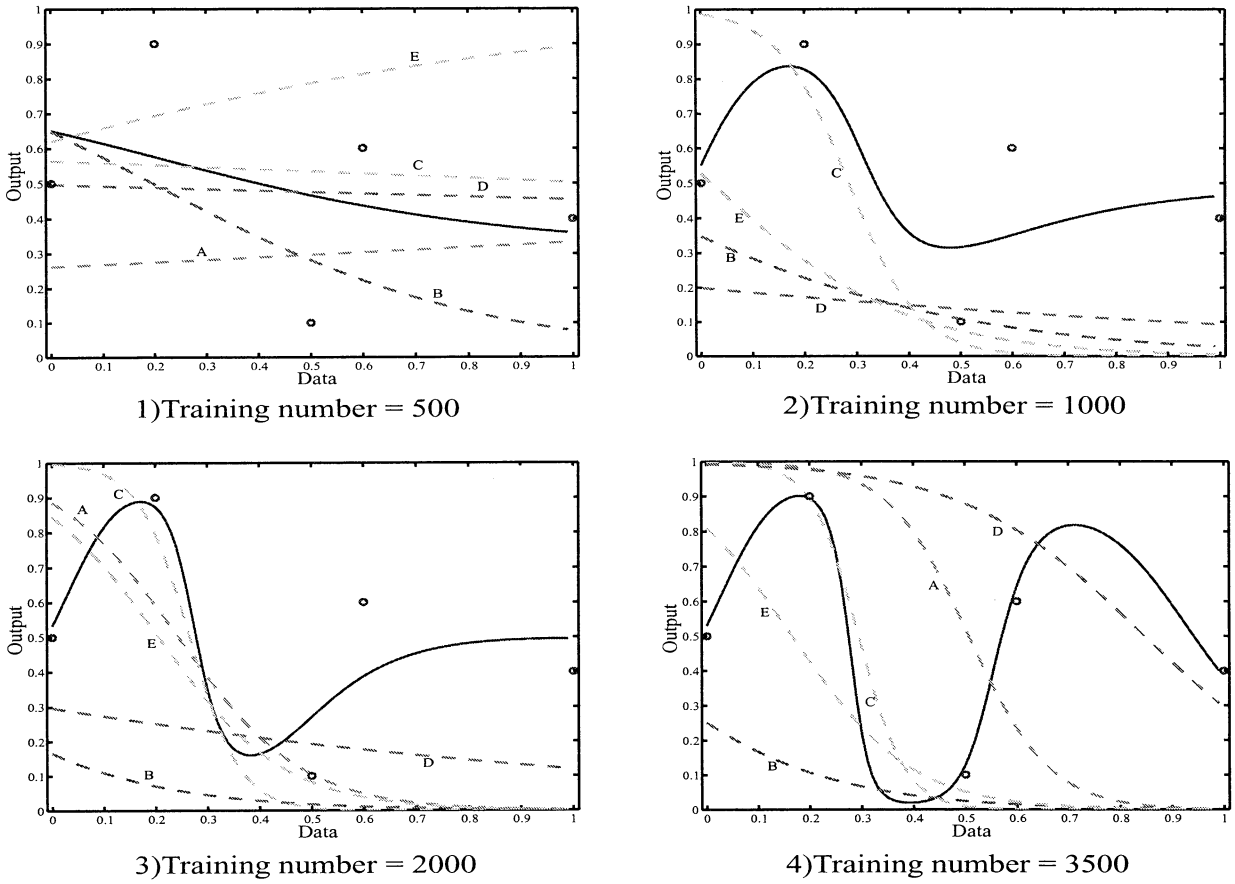


Figure 2: P-type learning

4 PD-type learning

In P-type learning, the unit changes fast in the first steps (a rough shape of the function is rapidly formed), but slow in the later steps. Whereas, in D-type learning the unit changes slow at the first steps, but fast in the later steps. This suggests that change of slope results in better effect to updates of connection weights and threshold.

In this section, we develope PD-type learning, the fusion of P-type and D-type, which combines back-propagation method with the new method to use derivatives of the slope of sigmoid function.

Fig.3 shows comparison of error rates between P-type and PD-type learning. The convergence speed of P-type learning is a little faster than PD-type at the first steps. However passing over 1000 times of training number, PD-type is converged rapidly. Fig.4 shows the network output and the behavior of units in hidden layer during the learning, when learning algorithm to train slope is used in combination with back-propagation method. We can see that learning time of PD-type learning requires only half as much as time for P-type learning.

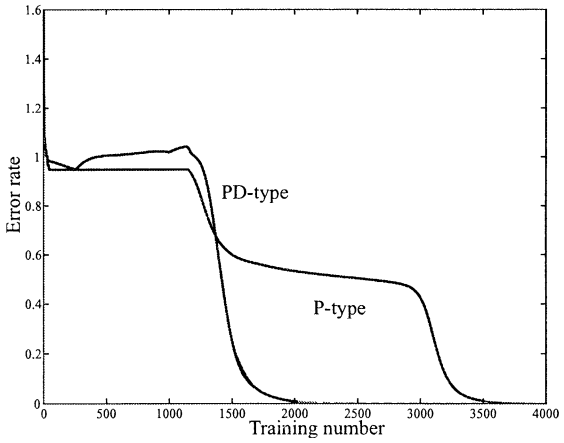


Figure 3: Comparison of learning rates

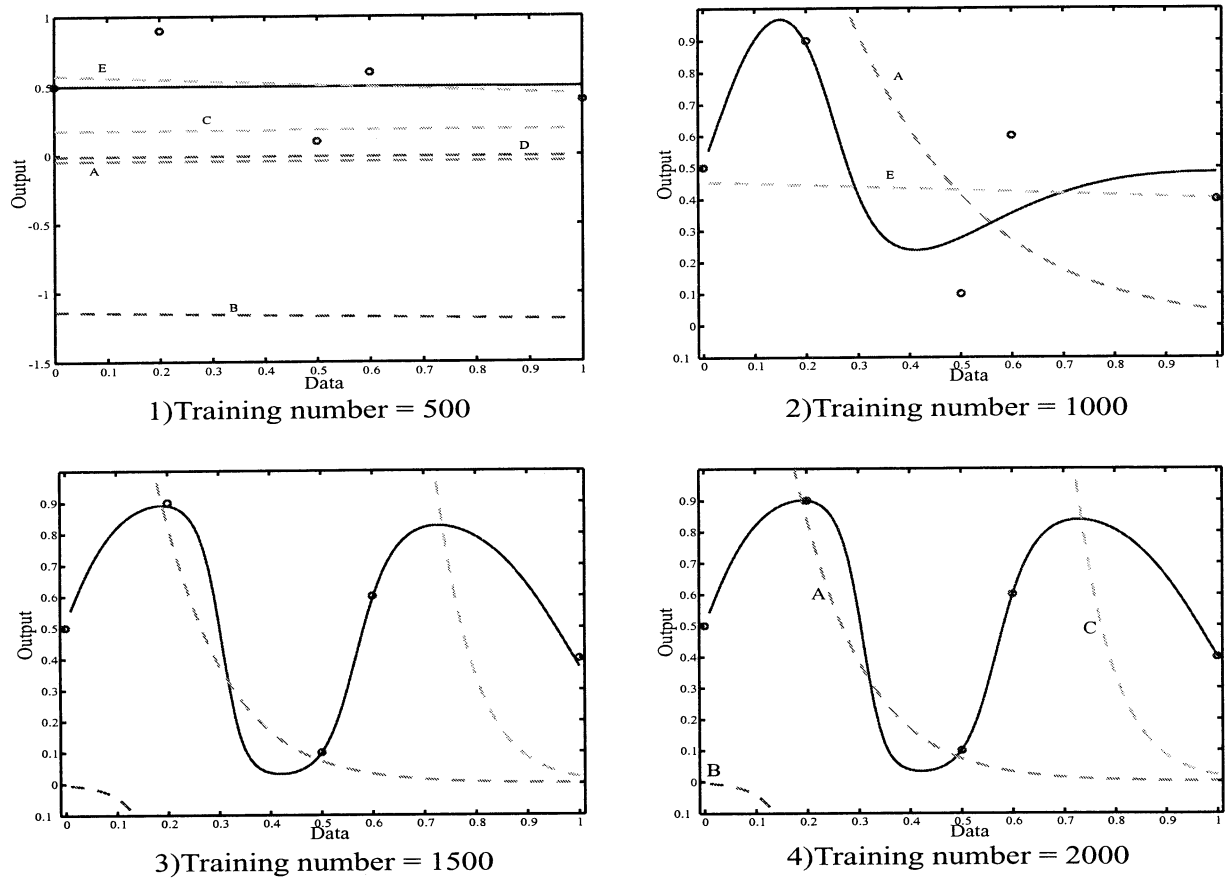


Figure 4: PD-type learning

5 Disscusion

In this paper, a new flexible neural network with PD-type learning algorithm has been proposed, in which a new activation function is introduced to have proportinal and differential functions.

The effectiveness of PD-type learning has been proved by comparing with conventional neural network with fixed sigmoid function, called P-type learning. Learning time of PD-type is required only half as much as time for P-type learning and realizes the high speed and flexible learning. As a result, a new neural network proposed in this paper is more flexible in the aspects of high learning ability than conventional one, which leads to less units of hidden layer.

In this paper we have discussed with the sigmoid function (3), but other type of sigmoid functions can be chosen depending on function and application of neural network. In [5], we proposed a flexible neural network with

$$f(x,a,b) = \frac{b}{1+exp(-ax)}$$

and the effectiveness was proved by applying to estimation of cutting torque in complicated drilling systems.

References

[1] S. Omatsu, M. Khalid and R. Yusof, *Neuro-control and its applications*, Springer-Verlag, 1996.

[2] M. Worgaard, O. Raun, N.K. Paulsen and L.H. Hasen, *Neural Networks for Modelling and Control of Dynamic Systems*, Springer, 2000

[3] K. Furuhashi,"On the approximation realization of continuous mappings by neural networks" , *Neural Networks*, vol.2,pp.183-192, 1989

[4] K. Asakawa, N. Watanabe, A. Kawamura, R. Masuoka, and J. Tanahashi, "Functions of Multi-Layered Neural Networks and their Fast Learning Algorithm", *T.IEE Japan*, Vol. 110-C, No. 3, pp.141-147, 1990.

[5] M. Kim and S. Kawaji,"Estimation of Cutting Torque based on Flexible Neural Network", *Proc. of Technical Meeting on Industrial Instrumentation and Control of IEE Japan*, Vol. IIC-02-82, pp.65-70, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A Realization of Optimum-Time Firing Squad Synchronization Algorithm on 1-Bit Cellular Automaton

Jun Nishimura
MegaChips Co., LTD.,
Yodogawa-ku, Miyahara, 4-1-6
Osaka, 532-0003, Japan

Takashi Sogabe
Internet Initiative Japan Inc.,
Chiyoda-ku, Kanda Nishiki-cho, 3-13
Tokyo, 101-0054, Japan

Hiroshi Umeo
Osaka Electro-Communication Univ.
Neyagawa-shi, Hatsu-cho, 18-8
Osaka, 572-8530, Japan

Abstract

In the long history of the study of cellular automata, the amounts of bit-information exchanged at one step between neighboring cells have been assumed to be O(1)-bit. In the present paper, we introduce a new class of cellular automata, CA1-bit, whose inter-cell communication is restricted to 1-bit and propose an optimum-time $(2n-2)$-step firing squad synchronization algorithm for n cells on CA1-bit. The number of internal states in each cell implemented is 78 and the total number of transition rules is 208. The algorithm we propose is based on Waksman's optimum-time algorithm which has been shown valid for any n.

1 Introduction

Cellular automata (CA) are considered to be a nice model of complex systems in which an infinite one-dimensional array of finite state machines (cells) updates itself in a synchronous manner according to a uniform local rule. In the long history of the study of the CA, generally speaking, the number of internal states of each cell is finite and the local state transition rule is defined in a such way that the state of each cell depends on the previous states of itself and its neighboring cells. Thus, in the finite state description of the CA, the amount of communication bits exchanged at one step between neighboring cells is assumed to be O(1)-bit, however, such bit-information exchanged between inter-cell has been hidden behind the definition of *conventional* automata-theoretic finite state description.

In the present paper, we focus our attention to the communication bits exchanged between inter-cell and introduce a new class of cellular automata, CA1-bit, whose inter-cell communication at one step is restricted to 1-bit. We refer the model as 1-bit CA. The number of the internal states of CA1-bit is assumed to be finite in a usual sense. The next state of each cell is determined by the present state of itself and two binary 1-bit inputs from its left and right neighbor cells. Thus the 1-bit CA can be thought of as one of the most powerless and simplest models in a variety of CAs.

We study a synchronization problem that gives a finite-state protocol for synchronizing a large scale of cellular automata. The synchronization in cellular automata has been known as the firing squad synchronization problem since its development, in which it was originally proposed by J. Myhill to synchronize all parts of self-reproducing cellular automata [5]. The firing squad synchronization problem has been studied extensively for more than 40 years [1-10]. First, we introduce a cellular automaton with 1-bit inter-cell communication and define the firing squad synchronization problem on CA1-bit. Then, we give an optimum-time firing squad synchronization algorithm on CA1-bit. The algorithm is based on the classical synchronization scheme developed by Waksman [10] and it will be implemented on a CA1-bit with 78 internal states and 208 transition rules.

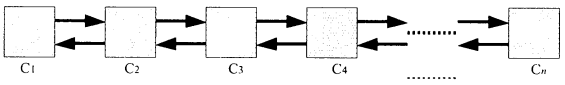


Figure 1: A one-dimensional cellular automaton with 1-bit inter-cell communication.

A one-dimensional 1-bit inter-cell communication cellular automaton consists of an infinite array of identical finite state automata, each located at positive integer point. (See Figure 1.) Each automaton is referred to as a cell. A cell at point i is denoted by C_i, where $1 \leq i \leq n$. Each C_i, except C_1 and C_n, is connected with its left and right neighbor cells via a left or right one-way communication link, in which those communication links are indicated by right- and left-going arrows, as is shown in Figure 1, respectively.

Each one-way communication link can transmit only one bit at each step in each direction. A cellular automaton with 1-bit inter-cell communication (abbreviated by CA1-bit) consists of an infinite array of finite state automaton $A = (Q, \delta, F)$, where

1. Q is a finite set of internal states.
2. δ is a function, defining the next state of any cell and its binary outputs to its left and right neighbor cells, such that δ: $Q \times \{0,1\} \times \{0,1\} \rightarrow Q \times \{0,1\} \times \{0,1\}$, where $\delta(p,x,y) = (q,x',y')$, p, $q \in Q$, $x, x', y, y' \in \{0,1\}$, has the following meaning: We assume that at step t the cell C_i is in state p and is receiving binary inputs x and y from its left and right communication links, respectively. Then, at the next step $t+1$, C_i assumes state q and outputs x' and y' to its left and right communication links, respectively. Note that binary inputs to C_i at step t are also outputs of C_{i-1} and C_{i+1} at step t. A quiescent state $q \in Q$ has a property such that $\delta(q,0,0) = (q,0,0)$.

Thus the CA1-bit is a special subclass of *normal* (i.e., *conventional*) cellular automata studied so far.

2 Waksman's Optimum-Time Algorithm

2.1 The Outline

Waksman's algorithm is constructed on the conventional CA, and it can synchronize any cellular array that consists of n cells at exactly $2n-2$ steps. Figure 2 shows its time-space diagram. In this figure, the horizontal axis means the cellular space, and the vertical axis means the step. The top of the horizontal line means the cellular array at $t = 0$, and it's called the initial configuration. The bottom of the horizontal line means the cellular array at $t = 2n-2$, and it's called the firing configuration.

Each cell is denoted C_1, C_2, ..., C_n. At $t = 0$, a general is at C_1, and soldiers fill up the other cells. The general is indicated G_0, and the cellular array G_0 manages is referred as S_0. $|S_0|$ is equal to n.

The general G_0 generates signal-a, wave-b_2, wave-b_3, ..., and wave-b_k, where $k = \lceil \log_2 n \rceil - 1$, at $t = 0$. The signal-a propagates the right at the sloop of $\frac{1}{1}$ (one cell per step). The wave-b_k propagates the right at the sloop of $\frac{1}{2^k-1}$ (one cell per $2^k - 1$ step). It reaches C_n at $t = n-1$, and generates G_1 there. After it reflects to the left, crosses wave-b_1, wave-b_2, ..., wave-b_k, and generates G_2, G_3, ..., G_k. These new generals divide S_0 to k parts of cellular array that are S_1, S_2, ..., S_k. The new generals G_i, $1 \leq i \leq k$, manages S_i the same as the management of G_0.

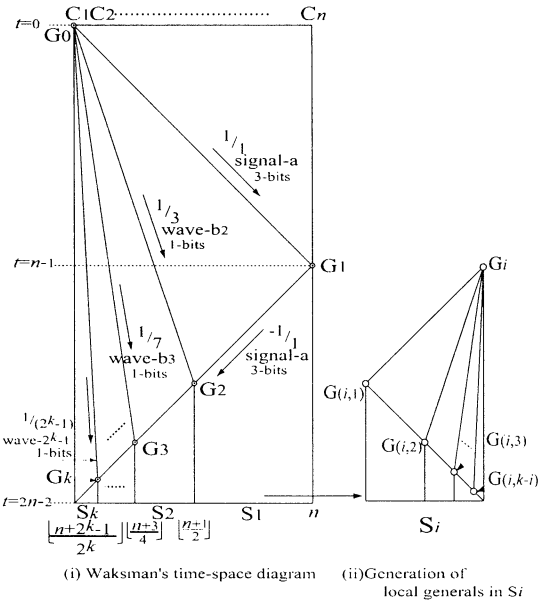


Figure 2: Time-Space Diagram of Waksman's Algorithm

2.2 Generation of Generals

G_i is generated at the position which divides exactly $|S_{i-2}|$. Therefore the parity of $|S_{i-2}|$ is an important factor for generating G_i. If the parity of $|S_{i-2}|$ is odd, G_i includes a cell. And if the parity of $|S_{i-2}|$ is even, G_i includes two cells. This parity information is determined by the cell on G_{i-1}, and is transmitted to the cell of G_i by signal-a. It determines a elements that is inclues G_i.

Figure 3 is a time-space diagram for generating G_2 in case of that n is even. G_2 is on C_m, and includes two cells. One of the cells includes the left part of the cellular array that is denoted C_1C_m, and another includes the right part of the cellular array that is denoted $C_{m+1}C_n$. In this case, G_2 of the right part is generated at $t = \frac{3n-4}{2}$, and G_2 of the left part is generated at $t = \frac{3n-2}{2}$. Therefore the right part has been synchronized earlier that the left part.
In this case, Waksman' solution is that the right part

starts to synchronize with delayed 1 step. This technique is referred to as delaying. In the case that the parity of $|S_{i-1}|$ is even, G_i uses the delaying. When the parity of $|S_{i-1}|$ is odd, G_i doesn't use it.

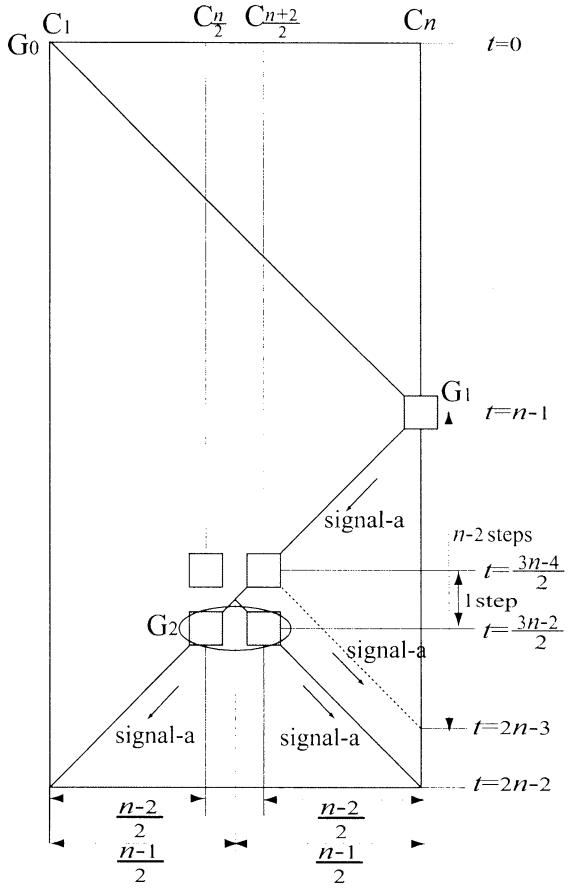


Figure 3: Generation of G_2 (n =*even*)

3 Firing squad synchronization problem on CA1-bit

We design a firing squad synchronization algorithm on CA1-bit based on Waksman's Algorithm. We need two 1-bit information that are the parity of $|S_{i-2}|$ and the parity of $|S_{i-1}|$ to generate any G_i. But on CA1-bit, we can't include them in a 1-bit signal-a. Therefore we must find a new technique instead of Waksman's ones. In this section, we propose these new techniques.

3.1 Parity of $|S_{i-2}|$

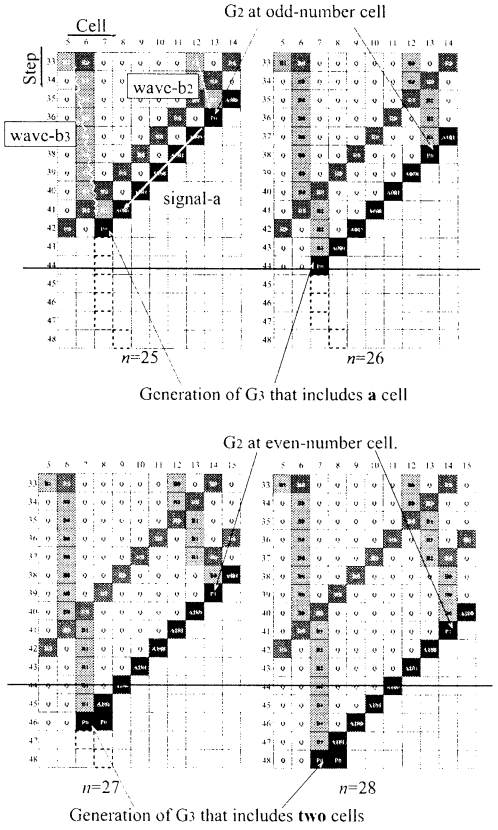


Figure 4: Generation of G_3 on C_7

Figure 4 shows the cases in which G_3 is generated on C_7. In this figure, n means $|S_0|$. G_3 is generated by the crossing wave-b_3 and signal-a. If G_2 positions at odd-number cell, G_3 always includes a cell. And if G_2 positions at even-number cell, G_3 always includes two cells. These are obvious. And the position information, which G_2 stays on, is same as $|S_{i-2}|$ size one. Then, we investigate how to get the parity of $|S_{i-2}|$ on C_7. It is noted that we decide the parity of $|S_{i-2}|$ by the crossing wave-b_3. At first, wave-b_3 stays on C_7 for 7 steps. In this figure, a horizontal line, which across 4 snapshots, is a half time of wave-b_3 full staying time. If the crossing happens before the half time, G_2 always positioned at odd-number cell. And if the crossing happens after the half time, G_2 always positioned at even-number cell.

The wave-b_i stays on C_{m_i} from $t = (2^i - 1)m_i - 2^i$

through $t = (2^i - 1)m_i - 2$. We define α_i, where $1 \leq \alpha_i \leq 2^i - 1$ as a variable that means the offset time. We express the time that b_i staying time range by using this equation.

$$t = (2^i - 1)m_i - 2^i - 1 + \alpha_i \quad (1)$$

And, signal-a, which crosses wave-b_i on C_{m_i}, arrives at C_{m_i+1} at time=t, where t is expressed by the following equation.

$$t = -m_i + 2n - 2 \quad (2)$$

We get the next equation from Equations (1) and (2).

$$2^i m_i = 2n + 2^i - 1 - \alpha_i, \quad where 1 \leq \alpha_i \leq 2^i - 1 \quad (3)$$

m_i is always an integer that leads to Equation(3), since α_i must be odd. When α_i is even, wave-b_i doesn't cross to signal-a on C_{m_i}.

In Addition, we investigate the crossing by wave-b_{i-1} and signal-a on $C_{m_{i-1}}$. We can also express m_{i-1} by this equation such that:

$$2^{i-1} m_{i-1} = 2n + 2^{i-1} - 1 - \alpha_{i-1}, \quad where 1 \leq \alpha_{i-1} \leq 2^{i-1} - 1 \quad (4)$$

Then we have Equation(5).

$$2^i \cdot m_i - 2^{i-1} \cdot m_{i-1} = 2^i - 2^{i-1} - \alpha_i + \alpha_{i-1} \quad (5)$$

In this case, α_i and α_{i-1} are expressed as follows.
(i)If $C_{m_{i-1}}$ positions on odd-number cell,

$$\alpha_i = \alpha_{i-1} \quad (6)$$

(ii)If $C_{m_{i-1}}$ positions on even-number cell,

$$\alpha_i = \alpha_{i-1} + 2^{i-1} \quad (7)$$

Therefore we get Equations (8) and (9).
If m_{i-1} is odd,

$$m_i = \frac{m_{i-1} + 1}{2} \quad (8)$$

And if m_{i-1} is even,

$$m_i = \frac{m_{i-1}}{2} \quad (9)$$

But, wave-b_i on $CA_{1\text{-bit}}$ needs trigger signal that is known to wave-b_i half time passed. Thus we have:
[Lemma 1] There exist 1-bit signals that can determine the parity of $|S_{i-2}|$.

3.2 Parity of $|S_{i-1}|$

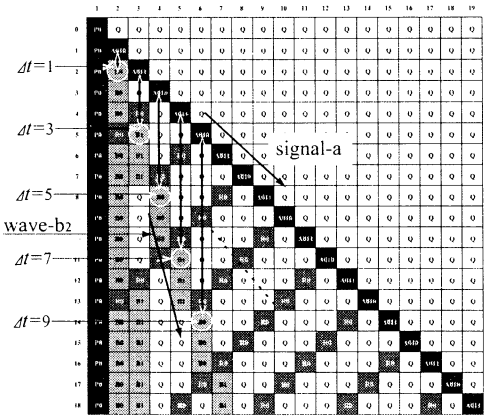


Figure 5: Relation of signal-a and wave-b_2

G_i positions just center of S_{i-2}, then, $C_1C_{G_i}$, which the left part of cellular array, is same size as S_{i-1}. Therefore the parity of $|S_{i-1}|$ is same as the position information , which G_i stays on. Figure 4 propagates signal-a and wave b_i in Waksman's algorithm. In this figure, we notice offset time that from signal-a passing through reached wave-b_2. On any C_m, where m is positive integer and it is bigger than 1, the offset time can lead to this equation.

$$\Delta t = 2p - 3 \quad (10)$$

In this case, the remainder that is Δt divides by 4, If $p = 2x$,

$$\Delta t mod 4 = 4x - 3 = 4(x - 1) + 1 \quad (11)$$

If , $p = 2x + 1$,

$$\Delta t mod 4 = 4x - 1 = 4(x - 1) + 3 \quad (12)$$

[Lemma 2] There exist 1-bit signals that can determine the parity of $|S_{i-1}|$.

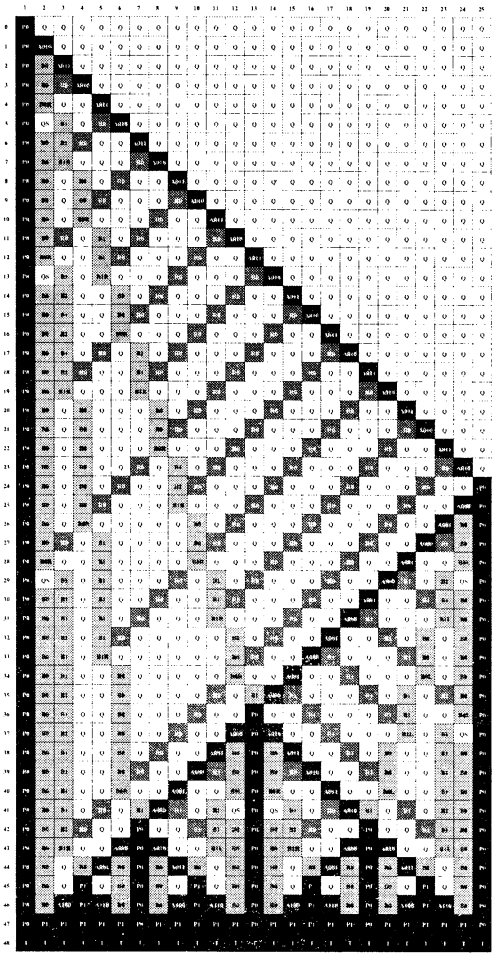


Figure 6: Snapshots of modified Waksman's synchronization algorithm

The key idea is the construction of an infinite set of 1-bit waves which propagate at $\frac{1}{2^{k+1}-1}$-speed in one-way direction on a CA_{1-bit}.

[Lemma 3] There exists a CA_{1-bit} that can generate an infinite signals which are described in Waksman' algorithms. Precisely, For any $n \geq 2$, the initial left-end *General* G generates k signals w_1, w_2, ... , w_k propagating at speed $1/(2^{k+1}-1)$ on n cells, where $k = \lfloor \log_2(2n-2) \rfloor - 1$.

Based on Lemmas, our main theorem is stated as follows:

[Main Theorem] There exists a CA_{1-bit} which can synchronize n cells in $2n-2$ steps. The CA_{1-bit} constructed has 78 internal states and 208 transition rules.

In Fig. 7 we show snapshots of the synchronization processes. Small right and left black triangles ▶ and ◀, shown in the figure, indicate a 1-bit signal transfer in the right or left direction between neighbor cells. A symbol in a cell shows its internal state.

4 Conclusion

We designed an optimum-time firing squad synchronization algorithm on $CA_{1\text{-bit}}$. Each cell has 78 states and 208 transition rules, and we checked its validity from n =2 through n=10000 by computer simulation.

References

[1] R. Balzer: An 8-state minimal time solution to the firing squad synchronization problem. *Information and Control*. vol. 10 (1967), pp. 22-42.

[2] J. Mazoyer: A minimal time solution to the firing squad synchronization problem with only one bit of information exchanged. *Technical report of Ecole Normale Superieure de Lyon*. no. 89-03. April. (1989). p.51.

[3] J. Mazoyer: On optimal solutions to the firing squad synchronization problem. *Theoretical Computer Science*. vol. 168 (1996), pp. 367-404.

[4] K. Michisaka, H. Yahara, N. Kamikawa and H. Umeo: A generalization of 1-bit-communication firing squad synchronization algorithm. *Proc. of The 15th Annual Conference of Japanese Society for Artificial Intelligence*, 2C3-06. (2001). pp.1-4.

[5] E. F. Moore: The firing squad synchronization problem. in *Sequential Machines, Selected Papers* (E. F. Moore, ed.). Addison-Wesley, Reading MA., (1964), pp. 213-214.

[6] J. Nishimura, T. Sogabe and H. Umeo: A design of optimum-time firing squad synchronization algorithm on 1-bit cellular automaton. *Technical Report of IPSJ*, vol. 32-12 (2000), pp. 41-44.

[7] J. Nishimura, T. Sogabe and H. Umeo: A Realization of Optimum-Time Firing Squad Synchronization Algorithm on 1-Bit Cellular Automaton. *Technical Report of IPSJ*. vol. 87-8 (2002), pp. 59-66.

[8] H. Umeo, J. Nishimura and T. Sogabe: 1-bit inter-cell communication cellular algorithms (*invited lecture*). *Proc. of the Tenth Intern. Colloquium on Differential Equations*, held in Plovdiv in 1999, *International Journal of Differential Equations and Applications*, vol. 1A, no. 4 (2000), pp. 433-446.

[9] H. Umeo: Cellular Algorithms with 1-Bit Inter-Cell Communications. *Proc. of MFCS'98 Satellite Workshop on Cellular Automata*(Eds. T. Worsch and R. Vollmar), Interner Bericht 19/98, University of Karlsruhe, (1998), pp.93-104.

[10] H. Umeo, T. Sogabe and Y. Nomura: Correction, optimization and verification of transition rule set for Waksman's firing squad synchronization algorithm. *Proc. of the Fourth Intern. Conference on Cellular Automata for Research and Industry*, Springer, (2000), pp. 152-160.

[11] A. Waksman: An optimum solution to the firing squad synchronization problem. *Information and Control*. vol. 9 (1966), pp. 66-78.

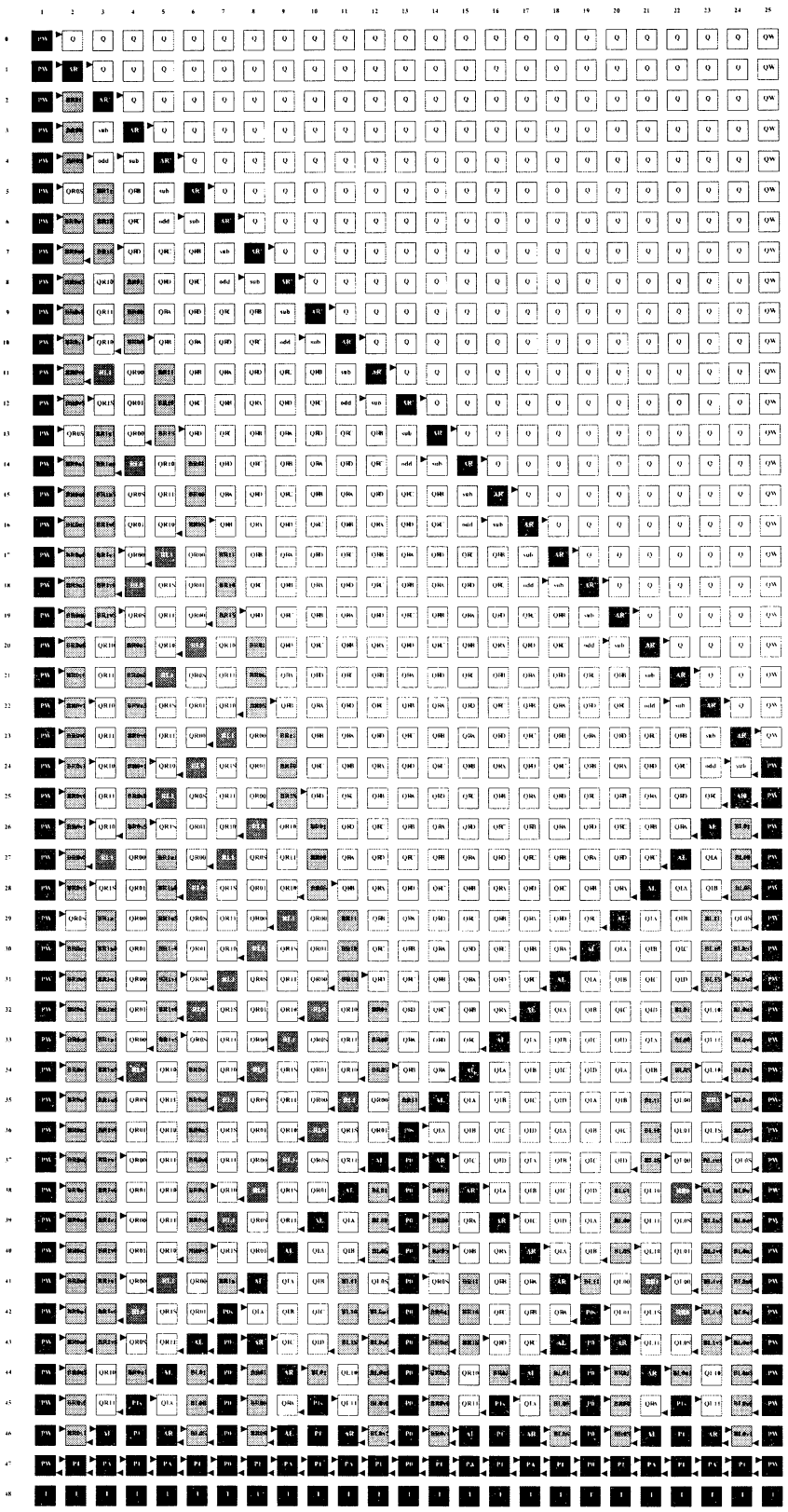


Figure 7: Snapshots of our algorithm on $CA_{1\text{-bit}}$

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

An Improved SMO Algorithm[1]

Xiang Wu, Li Tan, Wenkai Lu, Xuegong Zhang
State key Laboratory of Intelligent Technology and Systems
Department of Automation, Tsinghua University, Beijing(100084), CHINA

ABSTRACT

Support Vector Machine is a recently developed effective tool for solving pattern recognition problems. The mathematical model of its training process can be reduced to a constrained quadratic programming (QP) problem. A key problem in the application of SVM is how to solve this QP problem quickly and precisely. Among various SVM fast algorithms, SMO (Sequential Minimal Optimization algorithm) is an efficient one. We improve this algorithm in the light of the idea of Chunking. The improved method is called Chunking SMO algorithm (CSMO). The new algorithm has great advantage in terms of both speediness and memory saving. It is especially suitable for problems with small number of support vectors. Two-dimensional artificial data are used to test the algorithm.
key words: SVM; SMO; Chuning; quadratic programing

1. Introduction

From 1995 to now, SVM, which is a main product of Statistical Learning Theory, has become a promising field in machine learning study. Due to its immense size, the quadratic programming (QP) problem that arises from SVM's training cannot be easily solved via standard QP techniques. First, Traditional QP algorithms contain iteration approach, which requires kernel matrix, a matrix that has a number of elements equal to the square of the number of training examples, being held in memory. This matrix cannot be fit into 128 Megabytes if there are more than 4000 training examples [1]. Second, QP is NP-hard and its time consumption also increases rapidly when training examples increases. Third, because error is accumulated in every step of iteration, iterative algorithm's precision drops rapidly with the increasing of training examples.

Unfortunately, large sample problem is frequently encountered in practical use of SVM. For example, in human face recognition usually 50,000 training examples are needed. Therefore fast SVM training algorithm has received more and more study. Most fast algorithms' basic idea is to break down the large QP problem into a series of smaller QP sub-problems. In each step the algorithm process a training subset and then adjust examples in the subset by specific adding-discarding mechanic. For example, at every step Chunking algorithm (Vapnik, 1982) solves a QP problem that consists of the following examples: every non-zero Lagrange multiplier from the last step, and the M worst examples that violate the KKT conditions. Osuna, et al. (1997) suggests keeping a constant size matrix for every QP sub-problem, which implies adding and deleting the same number of examples at every step. Sequential Minimal Optimization (J.C. Platt, 1998) can be deemed as an extreme instance of Osuna's algorithm. Only two examples are analytically optimized at every step, so that each step is very fast [1]. SMO adopts choice heuristics for choosing which multipliers to Optimize. Finally all the multipliers will satisfy Kuhn-Tucker condition and the global minimum is found.

The advantage of SMO lies in the fact that solving for two Lagrange multipliers can be done analytically. Thus, numerical QP optimization, which is notoriously tricky to get right and slow, is avoided entirely. However, when more examples are added, the times of two Lagrange multipliers' optimization will increase sharply, and time consumption will still be too huge. Aiming at the cases in which large training set contains a few support vectors, we improve SMO according to the idea of Chunking. That is, divide the training set into many subsets, at each step the SVs in a subset is found and preserved while non-SVs being discarded. Since SVs are minority, the algorithm always processes a small subset, so computing speed is greatly increased. In our experiment, the improved algorithm turned out to be 30 times faster than original SMO algorithm while examples are more than 5000. The more examples processed, the bigger portion of time is saved.

2. Mathematical Model of SVM

SVM is a new machine learning method based on Structural Risk Minimization. The fundamental principle of SVM is to correctly divide two classes of examples by a

[1] This paper is sponsored by State's natural science foundation, project No. 69885044.

super plane and locate the plane so that margin (the minimum distance from any example to the super plane) is maximized. Consequently, function set with low VC dimension is constructed in high dimensional space, and good generalization ability is guaranteed.

Thus, the original problem is a constrained quadratic programming problem:

$$\text{Minimize} \qquad \Phi(\mathbf{w},b)=\frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t. \qquad y_i(\mathbf{x}_i^{\mathrm{T}}\mathbf{w}+b)-1\geq 0 \qquad i=1,2,...,l$$

where $\mathbf{x}_i$ is training example and y_i is its class label (+1 or −1). Its solution $\mathbf{w}$ is the normal vector of the optimal classification super plane. According to the theory of quadratic programming, the problem is transformed to its Wolfe dual:

$$\text{Maxmize} \qquad W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$$

$$\text{s.t.} \qquad \sum_{i=1}^{l}\alpha_i y_i=0$$

$$\alpha_i\geq 0 \qquad i=1,...,l$$

where α_i is example $\mathbf{x}_i$'s Lagrange multiplier. According to Kuhn-Tucker theorem, the Lagrange multiplier of non-support-vectors must be zero. Thus, the classification function is independent of non-support-vectors and is determined only by support vectors, the examples on the boundary of the margin.

In linear inseparable case, soft margin classification surface is introduced. The problem's Wolfe dual is similar to the dual in linear separable case, except that an upper bound is imposed on α. Finally, if we convert maximization to negating and then minimizing, the final expression of SVM's mathematical model turns out to be:

$$\text{Minimize} \qquad W(\alpha)=\frac{1}{2}\alpha^{\mathrm{T}}\mathbf{H}\alpha-\mathbf{u}_l^{\mathrm{T}}\alpha$$

$$\text{s.t.} \qquad \sum_{i=1}^{l}\alpha_i y_i=0$$

$$0\leq\alpha_i\leq C \qquad i=1,...,l$$

where $\mathbf{H}$ is a semi-definitive symmetric matrix $[y_i y_j(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)]_{i,j=1}^{l}$. $\alpha=[\alpha_1, \alpha_2, ..., \alpha_l]^{\mathrm{T}}$, $\mathbf{u}_l=[1,1,...,1]^{\mathrm{T}}$. To construct a nonlinear classifier, we only need to replace the inner product in the quadratic form by kernel $K(\mathbf{x}_i,\mathbf{x}_j)$. Generally, studies on SVM algorithm start from Wolfe dual, not the original QP problem.

3. Chunking SMO Algorithm

3.1 basic principle

As has been mentioned in introduction, though SMO is relatively efficient algorithm for large training set, its computation complexity still increases much faster than the size of training set, which makes it inapplicable in huge sample problems. Since SVM's classification only depends on SVs in training set, non-SVs are redundant information and can be discarded without affecting training result. It is processing this redundant information that account for most time consumption. Therefore, we consider discarding non-SVs in the process of training. As long as the cut training set preserves all the SVs, the resulted classifying super plane will remain the same. To do this, we divide the whole training set into chunks, and process only a chunk with SMO in a step. After each step, non-SVs are discarded while SVs are preserved and merged into the next chunk. If the SVs in training set increase much slower than the size of training set (as in many circumstances larger training set doesn't contain more SVs than smaller training set), an SVM whose training time is linearly dependent on training set size can be gotten. This improved algorithm is called Chunking SMO (CSMO).

3.2 protection against support vector lost

The above method merely helps us to introduce the basic idea of CSMO. In fact it risks losing SVs. Because SVs only determine classifying surface but can't determine the training set, when we delete non-SVs of a training subset, true SVs of the whole training set may be deleted. To avoid SV lost, we don't copy Chunking's method entirely. On the one hand, our algorithm preserves not only SVs of a chunk, but also examples that are likely to be SVs of the whole training set. That is, it preserves examples close to the classification surface of the chunk. On the other hand, each chunk is consist of preserved examples of the previous chunk and all the examples in the next subset, instead of examples in the next subset that violate Kuhn-Tucker condition. In addition, other techniques can be useful. For example, after processing all the chunks, the training set is redivided in a different measure and the preserved examples are merged into the first chunk. The training set is gone through again. Because after the first traverse the preserved examples contain most SVs of the whole training set, fewer SV will be lost in the second traverse. SV lost can be reduced greatly by multiple traverses.

3.3 adjusting chunk size dynamically to achieve optimum computing speed

The number of examples in each subset will affect computing speed. Consider processing a subset with n examples in it at one time and processing k subsets with n/k examples in each chunk. Denote the number of

previously preserved examples as n_0, and we hypothesize that after training by the n_0+n examples still approximately n_0 examples are to be preserved (as has been hypothesized previously that SVs always account for a tiny portion of training examples). The average computation time of SMO algorithm, denoted as $t(n_x)$, is a function of training set size n_x. Then the two method's computation time are $t(n_0+n)$ and $kt(n_0+n/k)$, respectively. If we set n much smaller than n_0 and t increases slowly with the increasing of its variable, $t(n_0+n)$ will be not much greater than $t(n_0+n/k)$. So $kt(n_0+n/k)$ is approximately k times of $t(n_0+n)$. However, if $n>>n_0$ and t increase sharply, which caused $t(n_0+n)>> t(n_0+n/k)$, the latter method is more likely to get faster speed.

This comparison demonstrates that neither a very large subset size nor a very small one is in favor of time saving. An optimum size locates between them. Furthermore, this optimum size ought to be adjusted dynamically according to n_0, the number of preserved examples from the previous chunk. We model this optimization as

$$\min_{n} \quad \eta = t(n_0 + n)/n,$$

where η is the average time consumption per example.

For convenience, n is regarded as continuous variable, let

$$d\eta / dn = 0$$

We get $nt'(n)=t(n)$. If t is an exponential function $t(n)=e^{an}$-1, n is a constant $1/a$, else if t is a power function $t(n)=n^a$, n is n_0 multiplied by a constant $1/(a-1)$.

Unfortunately, the computation complexity of SMO algorithm can't be precisely estimated. According to Platt, the average computation time of SMO is approximately linearly dependent on $n\sim n^{2.2}$ [3]. Therefore, in our algorithm we take Pn_0 new examples together with the n_0 preserved examples to form the next chunk. P is a parameter adjustable.

4. Experiment Results and Analysis

We implement CSMO algorithm with Matlab program. Two-dimensional artificial data are used to test the algorithm. As a demonstration, the result of 300 two-dimensional data is shown in Figure 1. The same example generator is used to generate 300 test examples. Only one test example is misclassified. Misclassification rate on test set is 0.3%.

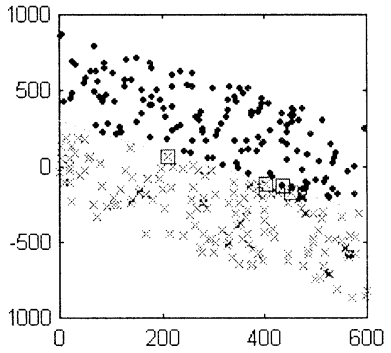


Figure 1. CSMO's classification surface for 300 examples The points denote positive examples while the crosses denote negative examples. SVs are marked by squares. The straight line is classifying surface.

CSMO's computation complexity is linearly dependent on the number of training examples (provided that the training set doesn't have many SVs). Hence, it is especially suitable for problems with huge training set and tiny SV set. In our experiments, SMO can deal with at most 10,000 examples, while CSMO can process 500,000 examples in 3 hours (11,265 seconds).

The following table lists the computation time comparison between SMO and CSMO while processing the same training set:

Training set size	300	1000	6000	500,000
SMO time consumption (in seconds)	32	226	3337	Can't get result
CSMO time consumption (in seconds)	5	24	116	11,265

Table 1. Time consumption comparison between SMO and CSMO

When tested by inseparable training sets, CSMO is also faster than SMO, but not as significant as tested by linear separable training set. In our experiment each class of examples is generated by a Gaussian distribution. When 1000 training examples are used, CSMO's training time is as much as approximately 60% of SMO's. This is because inseparable training sets usually contain relatively more SVs, which is not favorable for CSMO.

5. Summary

This paper puts forward CSMO algorithm, which is used to solve the QP problem that arises from SVM's training. CSMO is an improvement on SMO algorithm in the light of the idea of chunking. CSMO processes each

chunk with SMO algorithm in turn and pick out examples far away from classification surface, and then merges retained examples into the next chunk. CSMO is based on SMO and inherits the main merits of SMO. Analytic method is used to circumvent complicated iteration procedure encountered in numeric solution of quadratic programming problem. Not only the computation efficiency is improved, but also the accumulated error introduced by iteration is avoided. (In many algorithms accumulated error brings much trouble.) CSMO are also suitable for situations that require as small memory occupation as possible. Because training examples are processed separately, memory occupation of the algorithm will increase linearly, not squarely. CSMO is especially suitable for situation of huge training set with small support vector set. Its computation time is linearly dependent on the number of training examples and can solve classification problem including millions of examples.

6. References

[1] Vladimir N. Vapnik(2000), The Nature of Statistical Learning Theory, Springer-Verlag New York,Inc.

[2] John C.Platt(1999), Using Analytic QP and Sparseness to Speed Training of Support Vector Mections, MIT Press.

[3] John C.Platt(1998), Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Technical Report MSR-TR-98-14

[4] Vapnik, V.(1982), Estimation of Dependences Based on Empirical Data, Springer-Verlag

[5] Osuna, E., Freund, R., Girosi, F.(1997), Improved Training Algorithm for Support Vector Machines, Proc. IEEE NNSP '97

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Acquisition of 2-layer structure in a growing neural network

Ryusuke KURINO Masanori SUGISAKA and Katsunari SHIBATA
Dept. of Electronic Engineering, Oita University.
700 Dannoharu Oita 870-1192 Japan. Email: shibata@cc.oita-u.ac.jp

Abstract

Neural network are broadly used to approximate non-linear functions. However, it is difficult to decide an appropriate structure for a given problem. In this paper, "growing neural network" is proposed as an extension of Back Propagation (BP) learning. The propagated error signal is diffused from a target neuron as a substance. The axon of a growing neuron grows according to the concentration gradient of the substance. In a simulation, it was examined that the most simple problem such as "AND" and "OR" could be solved by the neural network and 2-layer structure was properly obtained.

1. Introduction

Artificial neural networks suggested by the natural nerve system of living things are broadly used to approximate non-linear functions because of its advantage of the leaning and generalization ability.

It is said that there are billions of neurons in the brain of human, and they are mutually connecting complicatedly. As opposed to it, when the artificial neural network is used, a simple 3-layer structure is employed in most cases. That is because the way to decide an appropriate structure for a giving problem has not been established yet.

There are well-know several methods to decide the structure of the neural network at present: those are the method using AIC information criterion[1], the method to cut out unnecessary connections[2], the method to append hidden neurons successively[3], and the method using genetic algorithm (GA)[4]. In these methods, since rough structure is assumed, the role of the designer is large and the degree of freedom to decide the structure is small. Especially, in the field of intelligent robot hereafter, high order functions will be required. Accordingly autonomous acquisition of an appropriate structure of the neural network including recurrent structure must be required for the robot's brain.

It has been considered that in the brain of living things, the number of neurons decreases after its birth, while the number of connections between neurons rapidly increases[5]. This suggests a strategy in the brain. More neurons than necessity are prepared at first, then they grow their axons and learn their synapse weights, and an appropriate network structure is formed flexibly. After that, unnecessary neurons are removed. It has been also reported that chemical substances such as NGF (Nerve Growth Factor) make a role of promoting the growth of axons and maintaining the connections[6].

In this paper, "growing neural network" is proposed in which the concept of "growth" is introduced in the conventional artificial neural networks. Then, the growth is formulized as an extension of Back Propagation (BP) learning that is a popular supervised learning. Getting a hint from NGF mentioned above, axons grows according to the concentration gradient of chemical substance which is diffused according to the error signal propagated in BP leaning. Therefore, the degree of freedom to decide the structure is larger than the conventional methods. Furthermore, since the growth is performed as an extension of learning, it is expected that purposive structure can be obtained. The final goal of this research is to build growing neural network that is able to obtain various structures including recurrent structure. In this paper, as the first step of this research, considering the acquisition of two-layer structure, the fundamental algorithm is introduced at first, and then the simulation result when the algorithm is applied to the simplest logical function such as "AND", "OR" is reported.

2. Growing neural network (NN)

2.1 Fundamental algorithm

In the growing NN, the growth is formulized as an extension of BP learning as mentioned above. Fig.1 shows the basic idea of the growing NN. Fig.2 shows the flowchart of the algorithm. At first, the output and error is calculated, and the error signal is propagated backward. The neuron that receives the signal diffuses it as a chemical substance. Then, the concentration gradient is formed by the diffusion around the neuron. The growing neuron

that does not have enough connections extends its axon according to the concentration gradient. After making the connection (synapse), the learning is started from 0 connection weight according to the regular BP algorithm. In the growing NN, the position of neurons also influences the structure. That is different from the conventional methods. By this property, it is expected that the symmetry between neurons are broken, and strcturization of network is promoted.

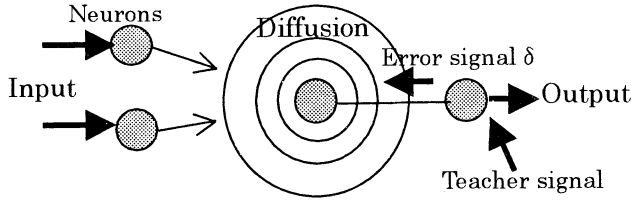


Fig.1 Basic idea of the growing neural network

①Decision of each neuron's position
②Decision of whether each connection is made or not.
③Setting of input and training signals
④Calculation of output and error
⑤Back propagation of the error signal
⑥Diffusion of the error signal
⑦Calculation of the axon growth
⑧Judge of whether each connection has been made or not
⑨Update of connection weights
(only for connected neurons)
⑩Return to ③

Fig.2 Flow chart of the growing neural network

2.2 Diffusion of the error signal

Neurons of living things diffuse chemical substance like NGF, and grow its axon according to the concentration gradient that is formed by diffused substance. To realize such functions in the growing NN, the error signal is diffused as the chemical substance around the output neuron. The error can be positive or negative, but the diffusion of negative is hard to imagine. For this reason, here, it is assumed that there exists the substance for each of negative and positive error, and they diffuse their substance independently. In this paper, since simplest two-layer structure is assumed, the output neuron diffuses the error signal. The error signal is calculated as

$$\delta_j = -\frac{\partial E}{\partial net_j} = (d_j - o_j) f'(net_j) \quad , \qquad (1)$$

where net_j : the internal state of the neuron j , o_j : the output, d_j : the training signal, $f'(net_j)$: the derivative of the output function, j =0,.., $NODE$, $NODE$: the number of output neurons, E : the error function. The diffusion is calculated for each of the positive and negative error signal respectively as

$$\kappa \frac{\partial u^p_{x,y}}{\partial t} = \rho \delta_i + \frac{\partial^2 u^p_{x,y}}{\partial x^2} + \frac{\partial^2 u^p_{x,y}}{\partial y^2} \quad , \qquad (2)$$

$$\kappa \frac{\partial u^n_{x,y}}{\partial t} = -\rho \delta_i + \frac{\partial^2 u^n_{x,y}}{\partial x^2} + \frac{\partial^2 u^n_{x,y}}{\partial y^2} \quad , \qquad (3)$$

where $u^{n,p}_{x,y}$: the concentration, κ : a diffusion constant, p indicates positive, n indicates negative, ρ : a divergence constant. The quantity of the diffused substance is proportional to the error signal. At the place where no neuron exists, Eq (2) and (3) is calculated with setting δ to be 0.0. By dealing with the error signal as the diffused substance, the growth of neuron according to the necessity can be realized.

2.3 Extension of the axon

The axon extends according to the state of growing neuron and the concentration gradient at the tip of the axon. Two types of neurons are prepared. One of them makes a positive connection, while the other makes a negative connection. The former grows its axon to the direction of the gradient of the positive error signal, while the latter grows its axon to that of the negative error signal. The reason is as follows.①If there are only one type of neurons, the neuron makes only one of negative or positive connection.②In the neuron of living things, it is known that which of the positive or negative connection the neuron makes is decided by the neurotransmitter that is generated at the synapse. The extension of the axon is calculated as,

$$\Delta x_i = \xi \frac{\partial (u^p - u^n)}{\partial x} S_i flag_i \quad , \qquad (4)$$

$$\Delta y_i = \xi \frac{\partial (u^p - u^n)}{\partial y} S_i flag_i \quad , \qquad (5)$$

where ξ : a growing constant, i =0,..., $NODE$, $NODE$: the number of input neurons, and the state S of the growing neuron is defined in the section 2.5. If the growing neuron is for the positive connection, $flag$ =1, while for the negative connection, $flag$ = -1.

2.4 Update of the connection weight

The connection weight is always 0 before the connection is formed. When the axon grows according to the concentration gradient and the connection is formed, the synapse starts learning from 0 connection weight. The update of connection weight is calculated as well as the conventional BP leaning,

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = \eta \delta_j o_i \qquad (6)$$

2.5 Adjustment for diffusion delay

It takes sometime that the diffused error signal reaches a growing neuron. If the input patterns change frequently, the input has changed when the error signal arrives at the growing neuron. Then, the first-order delay is introduced to the output of the growing neuron to adjust the gap of the timing. The state of neuron S is defined as the first-order delay of the output of the input neuron as

$$\tau \frac{dS_i}{dt} = -S_i + o_i \qquad (7).$$

If a time constant τ is too large, the state of neuron does not change so much. While, if the time constant is too small, the state of the neuron is not different from the original output value. Thus, the appropriate time constant is required.

3. Simulation

3.1 Set up

A simple two-layer structure is assumed to verify the fundamental functions of the growing NN, and two simple logical function "AND","OR" were learned. The simulation is done in 50×50 of area. Fig.3 shows the position of each neuron. The left output neuron learns to output "AND" of the input 'a' and 'c', and right one learned to output "OR" of the input 'b' and 'd'. The input 'b' and 'c' have to connect to the father output neuron. The parameters used in the simulation as follows. Leaning constant η =0.2, growing constant ξ =0.2, diffusion constant κ =0.2, divergence constant ρ =2.0, each of four input signals is chosen randomly from 0 and 1, and is fixed for one cycle. The one cycle is defined as 1000 steps, and 300 cycles are performed in one simulation. 0.1 or 0.9 was used as the training signal on behalf of 0 or 1. The output function is sigmoid with the range from 0 to 1. The inertia term was not used.

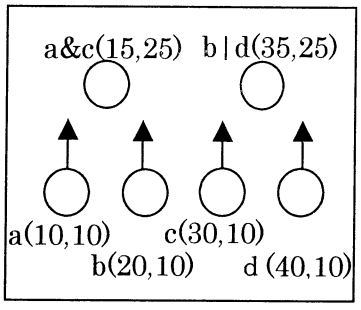


Fig.3 Position of each neuron

3.2 Result

Fig.4 shows how the axon of each input neuron grows. Although, there are two types of neurons, for positive connection and for negative connection, only the neurons for positive are shown in the figure. The neuron grows with repetition of extension and degeneration and the input 'a' forms the connection at 28000 steps and the input 'd' forms at 35000 steps. Since the input 'b' and 'c' are located closer to the output neuron that should not be connected, the loci prowl by the influence of the closer output neuron. However, there is no correlation between the input and the output. Finally, the both loci crossed and arrived at the correct at 87000 step and 89000 step respectively.

Fig5, 6 show the changes of the error for each output. In these figure, even when the error becomes large, it decreases quickly. That is because the bias of each output neuron is soon adjusted even if the neuron does not have any connections. The arrows in these figures indicate the timing when each connection was made. After making the second connection, the error decreases quickly. Moreover, it is seen that a small error remains after convergence. That is because the output for AND is stuck at 0.0 when the input is (0,0), and the output for OR is stuck at 1.0 when the input is (1,1) even though the training signal is 0.1 and 0.9 respectively.

Fig.7, 8 show the changes of the connection weights for the "AND" and "OR" output respectively. Since the

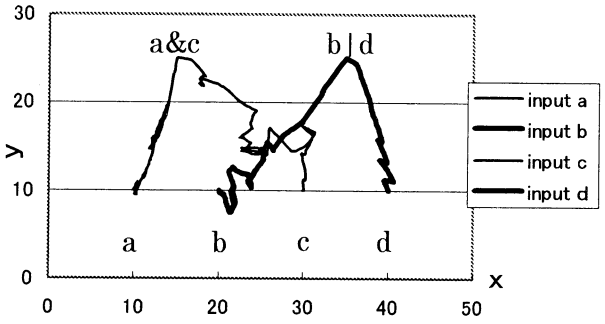


Fig.4 Growth of each input neuron

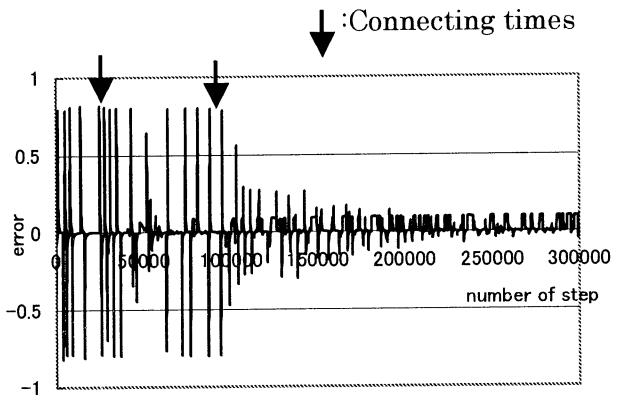


Fig.5 Change of the error for the AND output

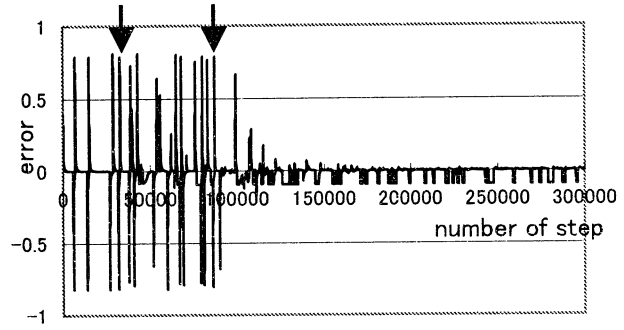


Fig.6 Change of the error for the OR output

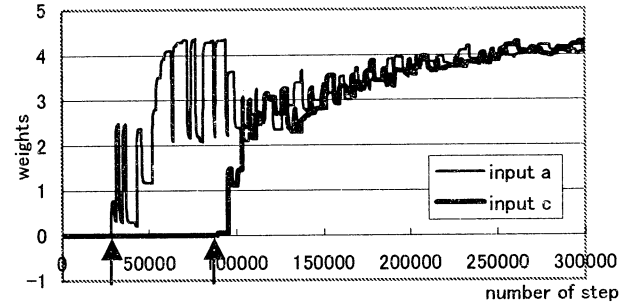


Fig.7 Change of the connection weights for the AND output

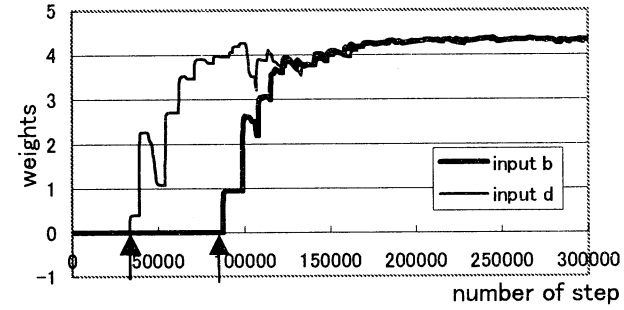


Fig.8 Change of the connection weights for the OR output

neuron 'a' and 'd' make their connection, the learning of the connection weight for the neuron 'a', 'd' starts at first. The connection weight of the neuron 'a' fluctuates more than that of the neuron 'd'. That is because when the input signal 'a' is 1, the training signal is 0.1or 0.9, but when the input signal 'd' is 1, the training signal is always 0.9.

4. Conclusion

In this paper, the growing NN was proposed that is formulized as an extension of the conventional BP learning. In the simulation with two output neurons, it was confirmed that the input neurons grow their axons to the appropriate output neuron even if the output neuron is farther than the other output neuron.

5. Future work

The growing NN is extended so as to solve the problems those needs hidden neurons, such as EXOR. There are two difficult points to be solves. One is that there are sometimes no correlation between each input and each output. The other is that the hidden neuron cannot receive either of input signals and error signals at first.

Acknowledgement

A part of this research was supported by the Sci. Res. Foundation of the Ministry of Edu., Sci., Sports and Culture of Japan (#13780295).

Bibliography

[1]D.B. Fogel, An information criterion for optimal neural network, vol.9,no.3, pp.509-521(1996)

[2]Masumi. Ishikawa, A Structural Connectionist Learning Algorithm with Forgetting Artificial Intelligence, vol.5, No.5, pp.595-603(1990)

[3]S. E. Fahlman and C. Lebiere, The cascade– correlation learning architecture, Advances in Neural information Processing System, vol.2, pp.524-532 (1990)

[4]Naoki SHIBA, Mnabu KOTANI and Kenzo AKAZAWA, designing Multi-layered Neural Networks Using Genetic Algorithm, SICE, vol.34,No.8,1080-1087(1998)

[5]Tadaharu TUMOTO, Brain and Development, Asakura Publishing Co., Ltd. (1986)

[6]Akimiti Kaneko, Koki Kawamura, Keiiti Uemura, Brain and neurons A guide to molecule nerve biochemistry, Kyoritsu Shuppan Co., Ltd. , pp58-73, 101-110(1999)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Absolute Position Estimation for Mobile Robot Navigation in an Indoor Environment

Park Soomin, Lee Bongki, Jin Taeseok and Lee JangMyung

Intelligent Robot Lab. Dept. of Electronics Eng., Pusan Nat. Univ.
30, Changjeon-dong, Kumjeong-Ku, Pusan 609-735, Korea
Tel : +82-51-510-1696, Fax : +82-51-510-5190, http://robotics.ee.pusan.ac.kr/
E-mail : psmim@pusan.ac.kr, jmlee@pusan.ac.kr

Abstract

Position estimation is one of the most important functions for the mobile robot navigating in the unstructured environment. Most of previous localization schemes estimate current position and pose of mobile robot by applying various localization algorithms with the information obtained from sensors which are set on the mobile robot, or by recognizing an artificial landmark attached on the wall, or objects of the environment as natural landmark in the indoor environment. Several drawbacks about them have been brought up. To compensate the drawbacks, a new localization method that estimates the absolute position of the mobile robot by using a fixed camera on the ceiling in the corridor is proposed. And also, it can improve the success rate for position estimation using the proposed method, which calculates the real size of an object. This scheme is not a relative localization, which decreases the position error through algorithms with noisy sensor data, but a kind of absolute localization. The effectiveness of the proposed localization scheme is demonstrated through the experiments.

1. Introduction

By the end of the 21st century, robots may not be strangers to us anymore. Compared with in recent years, the useful range for robots has gradually spread to a wide variety of areas. Mobile robots are especially being used as a substitute for humans in inhospitable environments or to do simple work that is either in or outside. In addition, they are used for investigating planets in space[1]. In such a mobile robot system, getting exact information on its current position is very important.

The mobile robot mainly calculates its position with the data acquired from a rotary encoder which is connected to the wheel, and from a gyroscope sensor, but it couldn't perceive the correct position because of slippage, a rough surface, and sensor error such as gyroscope drift. Many solutions have been proposed to overcome these unavoidable errors. For example some researchers presented a method that estimates the current position by applying information obtained by a rotary encoder and an ultrasonic sensor by applying an EKF(extended Kalman filter)[2-3]. And a researcher updated the positioning of mobile robots by fusing data from multi-sensors such as magnetic compasses, gyroscopes, rotary encoders with the EKF[4]. These methods need much calculation for a mobile robot to perform a task, which results in a sharp drop in the total system efficiency. Another disadvantage is a great localization uncertainty which is the result of the statistical error accumulated from sensors and control over long distances.

Contrary to the methods mentioned above, which intended to reduce the position error with relative positioning sensors, the following method provides an absolute position regardless of the distance moved and working time of a mobile robot. And some researchers presented a method that estimates the position of a robot through geometric calculation, after it recognizes a landmark[5-6]. Even though a CCD camera set on a robot is used for avoiding obstacles and tracking objects and so on, in these methods, the camera system was consumed unnecessarily for a robot to search and recognize the exact landmark. Another way that is presented for a robot equipped with a CCD camera, estimates its position by recognizing a characteristic topography or an object, and compares it with the model image saved in advance[8]. In general, we have utilized some feature points such as a wall or a corner as landmark in the workspace. However it has low confidence in recognition and requires much calculation. Therefore, its disadvantage is noticed that the processing speed of the system is low.

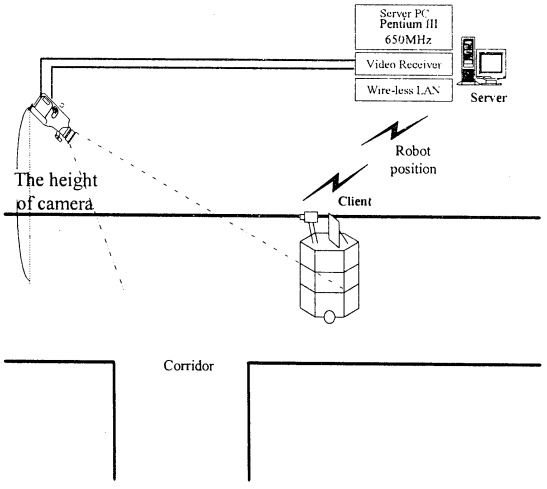


Fig. 1. Basic model.

In this paper, to overcome these problems we propose a new localization method as illustrated in Fig. 1. A camera installed on the ceiling of the corridor is utilized for the localization of the mobile robot. The system compensates for robot positioning by means of the following sequence.

First, the system recognizes whether it is a moving robot or not, with a CCD camera. Secondly, if the object is a moving robot, the system obtains the position of the robot. Finally, the system transmits the position data to the robot for the localization.

2. Object segmentation through image process

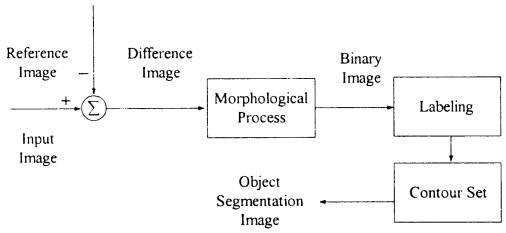


Fig. 2. Object segmentation model.

We extract moving objects using the difference image which is obtained as the difference between an input image, which is being inputted consecutively, and a reference image, which is captured and stored in advance.

2.1 Image pre-processing

In this paper, we applied a Gaussian mask with the nine pixels for removing illumination dependent image noises, and selected a modular image to shorten the processing time. In this paper, we have used modular four images which have 160×120 pixels for an image.

2.2 Filtering and labeling

In this paper, a filtering method that has been used widely, a morphological filtering method is adopted. Through Labeling, we can separate objects and search for their features using labels [9].

2.3 Reference image modification

In order to extract a moving object in a dynamic environment correctly, the reference image needs to be updated dynamically instead of keeping the initial reference image.

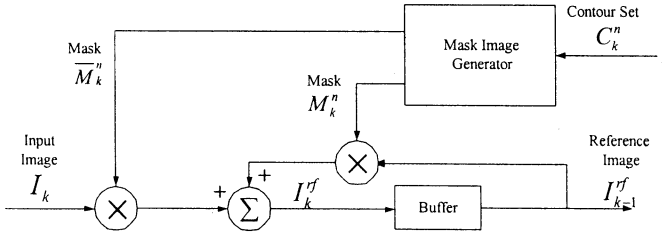


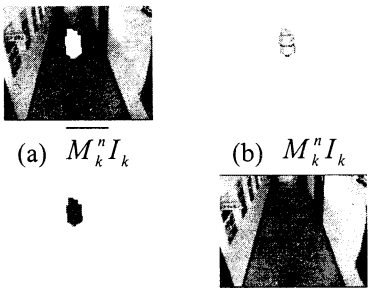
Fig. 3. Updating the reference image.

In Fig. 3, I_k^{rf} is an updated reference image that will be used for the next frame. Also, a mask image M_k^n is represented as follows:

$$M_k^n = \begin{cases} 1 & if\,(x,y) \in C_k^n \\ 0 & otherwise \end{cases} \quad (1)$$

This new reference image can be represented as,

$$I_k^{rf} = \overline{M_k^n} I_k + M_k^n I_{k-1}^{rf} \quad (2)$$



(a) $\overline{M_k^n} I_k$ (b) $M_k^n I_k$

(c) $M_k^n I_{k-1}^{rf}$ (d) I_k^{rf}

Fig. 4. Update process of background image.

3. Transformation from image coordinates to real coordinates

We obtain the distance between camera and object using a single camera so that, such distance can be represented as real coordinates[8].

As shown in Fig. 5, the solid square border in the center has a screen image for a mobile robot. This image is projection of the mobile robot on the corridor, which is in real three dimensions. Here, we can transform from the image coordinates to real coordinates to obtain the location of the robot.

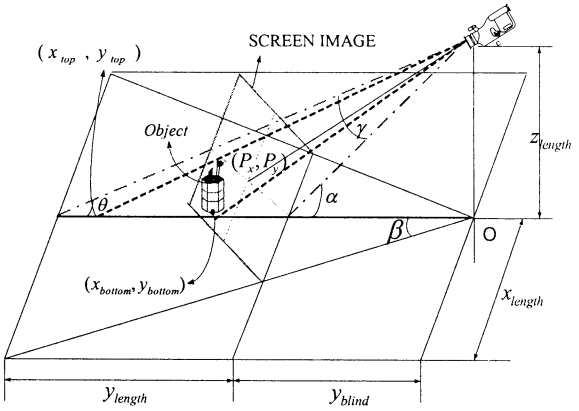


Fig. 5. Modeling for the correspondence between 2D image and 3D coordinates.

$$\alpha = \tan^{-1}\left(\frac{z_{Length}}{y_{blind}}\right) \quad (3)$$

$$\beta = \tan^{-1}\left(\frac{x_{length}}{y_{blind} + y_{length}}\right) \quad (4)$$

$$\gamma = \alpha - \theta \quad (5)$$

where $\theta = \tan^{-1}(\frac{z_{length}}{y_{blind} + y_{length}})$.

The screen image is described in detail as Fig. 6.

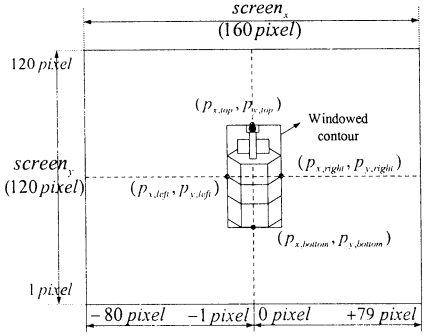


Fig. 6. Screen image.

The real coordinates of the robot center on the floor, $(x_{robot\_position}, y_{robot\_position})$, can be calculated as follows:

$$y_{bottom} = z_{length} \times \tan[(90° - \alpha) + \gamma \times (\frac{P_{y,bottom}}{screen_y})] \quad (6)$$

The y-axis center of the robot can be obtained as,

$$y_{robot\_position} = y_{bottom} + (L/2) \quad (7)$$

where L is width of the robot. And,

$$x_{robot\_position} = y_{robot\_position} \times \tan\beta(\frac{2p_{x,bottom}}{screen_x}) \quad (8)$$

4. Feature extraction

4.1 Height and width of an object

we can obtain the height and width of the robot using geometric analysis.

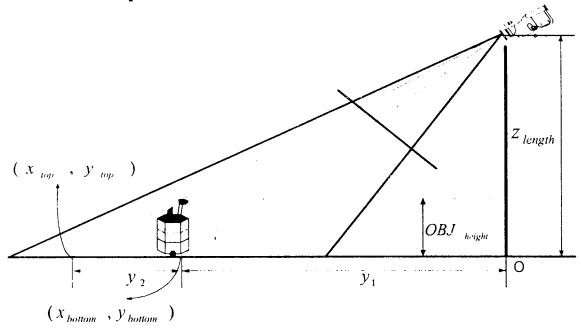


Fig. 7. Height measurement using a camera.

As shown in Fig. 7, the distance y_1 from the lowest coordinates of the object to the origin is calculated using y_{bottom} in Eq.(6) as,

$$y_1 = y_{bottom} - O \quad (9)$$

where O represents the origin.

In the same manner, y_{top} can be calculated from Eq.(6) by replacing y_{bottom} as y_{top} and $P_{y,bottom}$ as $P_{y,top}$. Therefore, the distance y_2 from the highest coordinates of the object to y_{bottom} is calculated as,

$$y_2 = y_{top} - y_{bottom}. \quad (10)$$

When the coordinates, y_1 and y_2 are obtained, the height OBJ_{height} of the robot can be calculated as,

$$OBJ_{height} = \frac{z_{length} \times y_2}{(y_1 + y_2)} \quad (11)$$

from the similarity properties of triangles.

Following the same procedure, the width of the mobile robot can be obtained as follows:

The real length $length_{pixel}$ per pixel is calculated as follow:

$$length_{pixel} = OBJ_{height}/(P_{y,top} - P_{y,bottom}). \quad (12)$$

Then, the width, OBJ_{width}, of the object is calculated as

$$OBJ_{width} = length_{pixel} \times (p_{x,right} - p_{x,left}). \quad (13)$$

4.2 Extraction of color information

To recognize the mobile robot, the height, width and color information have been used for a neural network. Since most color cameras used for acquiring digital images utilize the RGB format, we get RGB values for the object image. Each R, G, B values are represented as 8 bit data and the biggest value is 255.

5. Experiment and discussion

5.1 Mobile robot for experiment

Fig. 8. Ziro3.

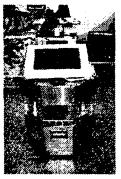
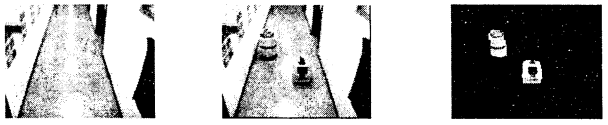


Fig. 9. IRL-2002.

Two mobile robots shown in Fig. 8 and 9 were used for experiment.

5.2 Object segmentation



(a) Ref. Image (b) Input Image (c) Ext. Image

Fig. 10. Extraction of mobile robot image.

We extracted the images of robots, which were navigating in corridor through the image processing. The experimental results are shown in Fig. 10.

5.3 Recognition of a robot through neural network

First of all, we need to exactly recognize an object to estimate the exact position of the robot. To do this, we have to decide whether an extracted object in the image

is a robot or not. In this paper, for this purpose, a neural network is utilized.

Table 1. Success rate of recognition

	recognition by using only color information		recognition by using color information and size information	
Classification / Object	number of trials	number of success	number of trials	number of success
IRL-2002	40	30	40	35
Ziro3	40	32	40	38
People	20	12	20	18

As shown in Table 1, with the size information, the success rate was improved a lot.

5.4 Acquisition of a Robot position and results of experiments

when a mobile robot was driven 10m forward, experimental results are shown in Fig. 11. Using only an encoder sensor and the kinematics of the mobile robot[7], there exists an approximately 40cm deflection along the x axis. However using our scheme, the robot trajectory was very similar to the center course of a driven corridor as seen in Fig. 11.

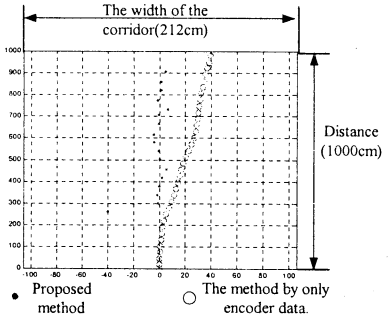


Fig. 11. Motion trajectory of a robot.

The error by the proposed method is shown in Table 2.

Table 2. Position error

Distance from camera	Axis	Maximum error (cm)	Minimum error (cm)	Average error (cm)
3m	X axis error	0.5729	0.2006	0.4195
	Y axis error	2.5828	0.1250	1.7061
4m	X axis error	2.3858	0.3178	1.1751
	Y axis error	4.833	0.3639	3.0734

As shown in Table 2, the further the robot moved from the camera, the greater the error became in real coordinates. The cause of error in the x axis is proportional to the distance, which is influenced by the β angle and the real distance which gradually increased. Consequently, in the limited camera view area, the exact robot position is recognized accurately without missing the robot.

6. Conclusion

In this paper, a new localization method with a fixed camera is proposed, which utilizes the external monitoring camera information under the indoor environment. When a mobile robot is moving the corridor, it helps the localization of robot by estimating the current position through the geometric analysis of the mobile robot image. The exact position of the mobile robot was obtained and demonstrated to be correct by the real experiments. And through the experiments, the advantages and efficiency of the proposed method are demonstrated illustratively.

For a future research topic, an efficient image processing scheme is necessary to improve and reduce the absolute error.

Acknowledgement

The author would like to acknowledge financial support from Center for Intelligent & Integrated Port Management Systems[CIIPMS] at Dong-A university.

References

[1] Clark F. Olson, "Probabilistic Self-Localization for Mobile Robots", *IEEE Trans. on Robotics and Automation*, vol. 16, no. 1, pp. 55 –66, Feb. 2000.

[2] Leopoldo Jetto, Sauro Longhi and Giuseppe Venturini, "Development and Experimental Validation of an Adaptive Extended Kalman Filter for the Localizaiton of Mobile Robots", *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 219 –229, Apr. 1999.

[3] A. Curran and K.J. Kyriakopoulos, "Sensor-Based Self-Localization for Wheeled Mobile Robots", *Proc. of ICRA*, vol. 1, pp. 8 –13, May 1993.

[4] Ching-Chih Tsai, "A localization system of a mobile robot by fusing dead-reckoning and ultrasonic measurements", *IEEE Trans. on Instrumentation and Measurement*, Vol. 47, no. 5, pp.1399 –1404, Oct. 1998.

[5] Hognbo Wang, Cheolung Kang, Shin-ichirou Tanaka and Takakazu Ishimatsu, "Computer Control of Wheel Chair by Using Landmarks", *Proc. of KACC*, Oct. 1995.

[6] M. Mata, J.M. Armingol, A. de la Escalera and M.A. Salichs, "A visual landmark recognition system for topological navigation of mobile robots", *Proc. of ICRA*, Vol. 2, pp. 1124 –1129, May 2001.

[7] Il-Myung Kim, Wan-Cheol Kim, Kyung-Sik Yun and Jang-Myung Lee, "Navigation of a Mobile Robot Using Hand Gesture Recognition", *Trans. on Control, Automation and Systems Engineering*, vol. 8, no. 7, pp. 599-606, Jul. 2002.

[8] Myong Ho Kim, Sang Cheol Lee and Kwae Hi Lee, "Self-Localization of Mobile Robot with Single Camera in Corridor Environment", *Proc. Of ISIE*, vol. 3, pp. 1619 –1623, Jun. 2001.

[9] Sung-Yug Choi and Jang-Myung Lee, Chung Kun Song and Hyek Hwan Choi, "The Detection of Lanes and Obstacles in Real Time Using Optimal Moving Window", *JSME International Journal*, vol. 44, no. 2, Jun. 2001.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Message passing implementation for the distributed robot control system

Tomasz Kubik[†,*] Hidenori Kimura[‡,*] Masanori Sugisaka[*,‡]

[†] Institute of Cybernetics Engineering, Wroclaw University of Technology, Janiszewskiego 11/17, Wroclaw 50-372, Poland
[*] Department of Electrical and Electronic Engineering, Oita University, Oita 870-1192, Japan
[‡]The Institute of Physical and Chemical Research (RIKEN), Bio-Mimetric Control Research Center, Shimoshidami, Moriyama-ku Nagoya, 463-0003, Japan
[*] Complex Systems Department of Complexity Science and Engineering Graduate School of Frontier Science, The University of Tokyo, Tokyo 113-8656, Japan
(Tel : 81-97-554-7831; Fax : 81-97-554-7818; E-mail:{tkubik|msugi}@cc.oita-u.ac.jp, kimura@crux.t.u-tokyo.ac.jp)

Keywords: mobile robot, distributed processing

Abstract

In the following paper we present an implementation of distributed control system based on message passing. Such a system was applied in our laboratory to control an indoor mobile robot.

The system has a distributed architecture: the task of intelligent robot control is partitioned into several subtasks exchanging data and synchronizing themselves through messages and message queues. This architecture allows all control subtasks to be run simultaneously on different, but connected via LAN, computers.

Message passing parts of the system are based on omniORB CORBA implementation,[1], for which some C++ wrapper classes were created. C++ wrapper classes simplify a process of system design, hiding all network related details inside their source code. Thus, all networking in the system is done on the omniORB level, but it is visible as a use of standard C++ classes. For the message passing a simple client-server model is proposed. This model requires an instance of CORBA naming service to be active. In our case we use `omniNames` application as a naming service.

1 Introduction

The intelligent robot control require several difficult and computationally intensive tasks to be solved in a short time. Most of these tasks, like visual data processing, speech processing, navigation, planning, etc., require large number of resources. More over, some of the tasks require different hardware devices (cameras, sensors, and sound devices in example) to communicate with at the same time. So, it is very difficult to stay within resources limits when implementing robot control system based only on one computer. It is also difficult to deal with different hardware devices simultaneously in this case. To assure enough resources and computing power the intelligent robot control system must have a distributed architecture.

In this paper we concern some issues connected with a distributed system implementation. In [2] mobile robot control system has been seen as a system composed of several tasks achieving certain behaviors. Some remarks about parallel processing and distributed architecture were discussed in [3] and [4].

The crucial part in our distributed control system design is networking (i.e. data exchange and process synchronization). There are some standard approaches to tackle this subject, [5]. In our case we decided to develop a system, which is based on message passing mechanism. The main concept of the system is described in the section 2.1. Sections 2.2 and 2.3 gives some details about system implementation. An example of a distributed control system is discussed in the section 3. Conclusions are given in the last section of this paper.

2 Distributed robot control system

2.1 Main concept

The main concept of our implementation is a concept of a message queue. The message queue is a place, where messages can be stored, and where messages can be retrieved from. So, in our implementation each module of the control system must have at least one instance of `MsgQueue` class. The instance of this class keeps messages inside a priority queue and provides methods to operate on them locally (in fact, the priority queue is an array of four queues of different priorities). The class is thread safe and may notify application (through events) about new messages in a queue (see Fig. 1). The `PutMsg()` method can be called remotely from the other applications, but this must be done through C++ wrapped CORBA classes. The `MsgQueue` class can be described shortly as follows:

To store a message in a priority queue `PutMsg()` method is used. The messages can be retrieved from a priority queue with the use of one of three "get" methods: `GetMessage()`, `GetMessage(timeout)`, and `GetIdleMsg()`. `GetMessage()` simply retrieves the highest priority message from the queue. The timeout parameter in GetMessage() tells, how long the

method should wait, if the priority queue is empty. `GetIdleMsg()` retrieves other than highest priority message from the queue. It should be used by the thread operating on instance of `MsgQueue` in order to avoid message starvation. All "get" methods retrieve messages by returning pointers to them (messages are stored in the queues). After use messages must be removed from a memory by the call to its destructor (application must take care about memory freeing). For events handling `GetReceiveEvent()` method is used.

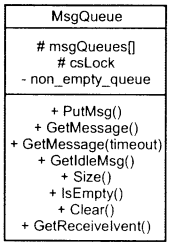


Figure 1: `MsgQueue` class

2.2 Idea of asynchronous message passing

The asynchronous message passing was invented according to the following idea:

There are several application running simultaneously on different computers (but not necessary). Each application can receive, or send, or receive and send data. The data are transfered between applications in form of messages. The message consist of: a header and a raw data field.The header specify the category of the message and the message's type. The raw data bits include encoded data (optionally).

In a given application all incoming messages are stored in a priority queue. Each message stored must be "processed" by a message reading thread. This thread reads messages from a queue and removes them when processed. The message reading thread runs in a passive waiting mode. It means that the thread is asleep if there is no messages in the queue. When a message arrive, the thread wakes up and do processing.

The priority queue is assign to the queue server which works as a receiving part of one-way communication channel. To create a communication channel between two applications at least one queue server and one queue client must exist (not necessary in two application). But in general one queue server can communicate with several queue clients which work as sending parts of one-way communication channels.

If a message broadcasting between different applications is expected, one of the applications should implement a router server. Router server provides methods for applications registering, which are accessible trough the calls of corresponding methods in router clients. It is allowed to create several router clients in the messaging system, but the system must have only one router server.

Router server is assigned to the one of queue servers. This special queue server can not be distinguish from the other queue servers by the queue clients. Each message sent to the queue server will be broadcast by the router server assigned. This is a user duty to design a messaging system, in which a special queue server will play its role correctly. In general, the messaging system can have many router servers (but usually it has one).

Router server is visible to the applications only through the methods of corresponding router clients. These methods allows to register any given queue server (message receiver) for receiving messages of a given type.

2.3 Distributed queues in CORBA

In the presented idea of a system the following methods were design to be available remotely: `PutMsg()` (which should put the message in a queue), and `Register()`, `Unregister()`, `UnregisterAll()` (which should inform a router about routing scheme).

In the standard implementation of distributed application with CORBA, [6] all starts with interfaces definitions in IDL. We declared two interfaces: `MsgQueueInterface` with `PutMsg()` method and `RouterInterface` with registering methods. The IDL specification of these interfaces was following:

```
#ifndef __MESSAGING_IDL__
#define __MESSAGING_IDL__
typedef sequence<octet> SeqData;

interface MsgQueueInterface {
 oneway void PutMsg(in long uID, in long flags,
                    in long param, in SeqData data);};

interface RouterInterface {
 oneway void Register(in string cname,
                in octet category, in octet ioFlag);
 oneway void UnRegisterAll(in string cname);
 oneway void UnRegister(in string cname,
                in octet category);};
#endif
```

After IDL to C++ compilation, we got set of CORBA classes with skeletons and stubs. These classes were bases for our C++ classes that wrapped all CORBA stuff.

`MsgQueueServer` is a C++ wrapper of CORBA class working as a queue server (implementing skeleton for `MsgQueueInterface`). This class has a unique name and instance of the `MsgQueue` assigned with it. The assignment of a `MsgQueue` causes that remote `PutMsg()` call (issued by the corresponding `MsgQueueClient`) results in a call of `PutMsg()` of the `MsgQueue`.

`MsgQueueServer`'s `PutMsg()` method is available remotely for the other applications trough the corresponding clients methods calls after registering to the `NameService` with server's name. The registering is

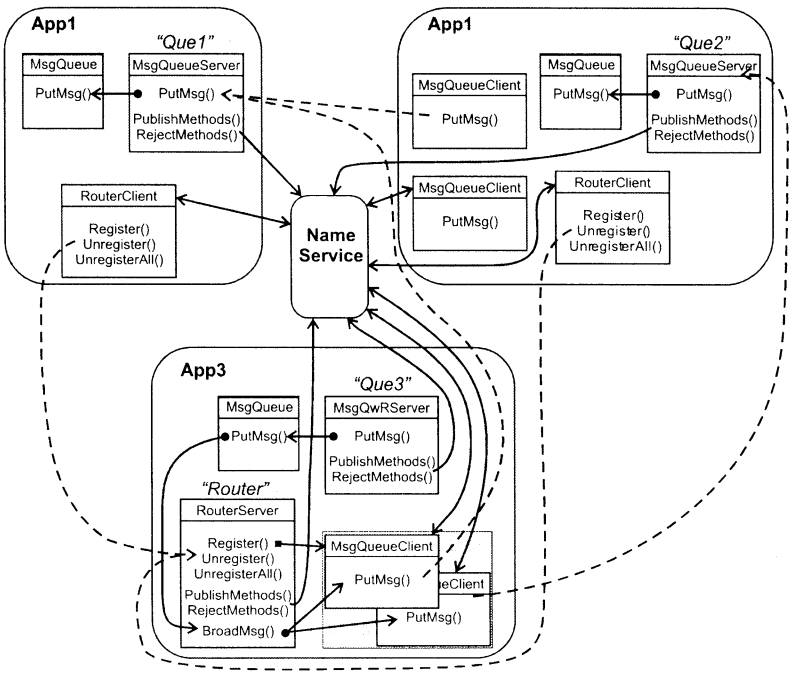


Figure 2: Example

done in the `PublishMethods()`. `RejectMethods()` unregisters a queue server and blocks all remote calls of the server's `PutMsg()`. For implementation completeness `MsgQueueServer` is equipped with a local version of `PutMsg()` method (which is a direct call to the `PutMsg()` of the `MsgQueue` assigned.)

`MsgQwRouterServer` is a subclass of the `MsgQueueServer` class which, apart from a name and a reference to the `MsgQueue`, it keeps also a reference to the `RouterServer`. `MsgQwRouterServer` works similarly to `MsgQueueServer`. The only difference is that a call of `PutMsg()` of the `MsgQueue` is followed by the call of `BroadMsg()` of the associated `RouterServer`.

`MsgQueueClient` is a C++ wrapper of CORBA class working as a queue client (implementing stub for `MsgQueueInterface`). It provides `PutMsg()` method which is a remote call of the corresponding `MsgQueueServer`'s method. Both `MsgQueueClient` and corresponding `MsgQueueServer` must have the same name assigned. If there is no `MsgQueueServer` registered in the `NameService` with name that match the name of the `MsgQueueClient`, the `PutMsg()` can not succeed. In such a case the return value of that call is 0 (1 otherwise).

`RouterServer` and `RouterClient` are C++ wrappers of CORBA classes working, respectively, as a router server (implementing skeleton for `RouterInterface`) and as a router client (implementing stub for `RouterInterface`).

`RouterServer` can be registered and unregistered to the `NameService`, similarly as `MsgQueueServer` (server and its clients must have the same name). `PublishMethods()` makes `RouterServer`'s methods `Register()`, `Unregister()`, `UnregisterAll()` available for `RouterClients`, `RejectMethods()` blocks them. For implementation completeness `RouterServer` is equipped with local versions of these three methods. Additionally it has `BroadMsg()` method. This method is used to broad directly a given message to all `MsgQueueServers` already registered in `RouterServer` by the `RouterClients`. For a normal work `RouterServer` should be assigned to the distinguished queue server (`MsgQwRServer`).

`RouterClient` is used for registering `MsgQueueServers` to the `RouterServer`. The registering is done trough the use of `RouterClient` methods `Register()` and `Unregister()` (which are remote calls of corresponding router server methods). These two methods require some parameters. One of them is a name of existing `MsgQueueServer` (not necessary in the same application) that is registered to the `NameService`. After `Register()` call, the `RouterServer` will receive the notification describing the type of messages that a `MsgQueueServer` with a given name is "interested in". This causes the creation of a `MsgQueueClient` with a given name by the `RouterServer` (if such a client does not exist yet) and message broadcasting according to the specification in parameters (if a new message will be received by the `MsgQwRServer`).

`NameService` is an application that implements CORBA naming service, [7]. This application registers objects with names in a tree-like structure, and,

on request, provides references to these objects. The references are required for accessing objects remotely in order to call their methods. We used `omniNames` application (from omniORB) as a naming service. In our case all names of registered objects follows the scheme: `test.queue_context/''name''.Object`, where `''name''` stands for a `MsgQueueServer`'s, or `MsgQwRouterServer`'s, or `RouterServer`'s name.

3 An example of distributed system implementation

In the figure 2 an example of distributed system is presented. This system consists of three distributed applications: Appl, App2, and App3. All three can exchange data, but:

- Appl can work as a listener only. It has `MsgQueueServer` that serves `PutMsg()` method to the other applications. This server is registered in `NameService` as *"Que1"*. Appl has also `RouterClient`. This client connects to `RouterServer` (which is registered in `NameService` with a name "Router") and registers *"Que1"* as a listener of messages of specified type.
- App2 can work as a listener and sender as well. It has `MsgQueueServer` serving `PutMsg()`, registered in `NameService` with a name "Que2" (listener). It has two `MsgQueueClients` (senders) - one for sending messages to the `MsgQwRServer` registered in the `NameService` as *"Que3"* (which is a queue server with a `RouterServer`); - second for sending messages to the `MsgQueueServer` registered in the `NameService` as "Que1". The `RouterClient` in App2 plays a similar role as a `RouterClient` in Appl. It informs `RouterServer` about the types of messages *"Que2"* is interested to receive.
- App3 has `MsgQwRServer` - this server works as a message listener and distributor. It is registered in the `NameService` with a name *"Que3"*. `MsgQueueServer`, apart from working as a queue server (i.e. storing incoming messages in a message queue), keeps the reference to the `RouterServer`. Each time the `PutMsg()` of `MsgQwRServer` is called, the `BroadMsg()` of `RouterServer` is also called (this is done inside `PutMsg()`). Messages are broadcast then to all registered listeners in the router. It is done by calling `PutMsg()` method of `MsgQueueClients`. In the example there are two such clients. One was constructed for *"Que1"*, second for *"Que2"*.

4 Conclusion

We have shown that with the aid of CORBA and C++ wrapper classes process of distributed system design can be a matter of its decomposition. Independent system's modules can be spread over the LAN, and a system designer does not need to know, how the networking is done. The system designer duty is to create an algorithm that will control whole system based on messages.

The main concept of presented implementation is a concept of a message queue. The queue stores incoming messages and provides methods for retrieving them. Each message can be interpreted as a command, which can carry some arguments. C++ classes described provides sources for clients and servers that are used for communication (message passing and message routing).

We have tested a message passing based distributed control system on our mobile robot (see [8] for robot details). We have seen that presented idea suits well for distributed robot control task. The limitation of the system are connected with the capacity of the network. Thus, the message passing based system does not work well, if messages are used to carry a large amount of data (like images). In one-way message implementation it is possible to loose some messages, if their number exceeds network capacity. This is an analogy to the human perception system. If the number of messages is too large, they are simply lost. Because messages are sent only in one way, there is no confirmation about message receiving. But the confirmation can be always created, if needed, as a new one-way message from a receiver.

References

[1] Lo Sai-lai, Riddoch D., Grisby D. (2000), *The omniORB version 3.0. User's Guide.* AT&T Laboratories Cambridge

[2] Brooks R. A., Connell J. H. (1986), Asynchronous Distributed Control System for a Mobile Robot. In *Proceedings of the SPIE*, volume 727, pp. 77–84

[3] Hamilton D.L., Walker I.D., Bennett J.K. (1998), Parallel robot control using speculative computation. *Journal of Robotics and Automation*, 13(4):101–112

[4] Rosenblatt J (1997), The Distributed Architecture for Mobile Navigation. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2/3):339–360

[5] Chow Mo-Yuen, Tipsuwan Y. (2001), Network-based control adaptation for network QoS variation. In *Communications for Network-Centric Operations: Creating the Information Force, MILCOM 2001, Communications for Network-Centric Operations: Creating the Information Force*, volume 1, pp. 257–261 IEEE

[6] (1999), *formal/98-12-01: CORBA 2.3 full specification* Object Management Group, revision 2.3 edition

[7] (2001), *formal/01-02-65: Naming Service specification, revised Specification* Object Management Group, revised edition edition

[8] Kubik T., Sugisaka (2001), Intelligent navigation and control system for a mobile robot based on different programming paradigms. In *Proceedings of the International Conference on Control, Automation and Systems (ICCAS2001)*, pp. 288–291, Jeju Island, Korea

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Implementing Distributed Control System for Intelligent Mobile Robot

Andrey A. Loukianov[1]
aloukian@cc.oita-u.ac.jp

Hidenori Kimura[2,3]
kimura@crux.t.u-tokyo.ac.jp

Masanori Sugisaka[1,2]
msugi@cc.oita-u.ac.jp

[1)]Department of Electrical and Electronic Engineering, Faculty of Engineering, Oita University, Dannoharu 700, Oita 870-1192, Japan.
[2)]The Institute of Physical and Chemical Research (RIKEN), Bio-Mimetic Control Research Center, Shimoshidami, Moriyama-ku, Nagoya 463-0003, Japan.
[3)]Department of Complexity Science and Engineering, Graduate School of Frontier Science, University of Tokyo, Tokyo 113-8656, Japan.

Abstract

The control system of a mobile robot has a number of real-time issues to deal with in order to operate: motion control, mapping, localization, path planning, sensor processing, etc. Intelligent reasoning, task-oriented behaviors, human-robot interfaces and communications add more tasks to be solved. This naturally leads to a complex hierarchical control system where various tasks have to be handled concurrently. Many low-level tasks can be handled by robot's onboard (host) computer. But other tasks, such as speech recognition or robot vision processing, are too computationally intensive for one computer to process. In this case it is better to consider distribute design for the control system in networked environments. In order to achieve maximum use of the the distributed environment it is important to design the distributed system and its communication mechanisms in effective and flexible way.

The paper describes our approach to designing and implementing the distributed control system for intelligent mobile robot. We present our implementation of the distributed control system for prototype mobile robot. We focus our discussion on the system architecture, distributed communication mechanisms and distributed robot control.

Keywords: mobile robot, control, software agents, distributed computing

1 Introduction and discussion

Mobile robot controllers are rather complex systems that have to deal in real-time with a number of tasks in order to allow the robot to operate autonomously. These tasks include motion control, sensing, planning, navigation, etc. The robot controllers are usually designed as modular and hierarchical systems. This makes it easier to realize complex system functions using a composition of more simple task-oriented modules. There is a vast literature on designing mobile robot control architectures [1, 3, 4, 5]. A number of mobile robot controllers were successfully implemented using these architectures [2, 5].

If the mobile robots are to perform useful tasks in open environments their controllers should be provided with more intelligent features like natural human-robot interfaces: speech recognition, face and sign recognition, and other ways of human-robot interaction. The problem here is that algorithms which may provide these features are usually very computationally intensive. If these high-level programs run on the same computer which also handles low-level control and sensing then they will worsen the response time of critical system components and will run slowly themselves. To avoid this bottleneck the control system can be implemented as a distributed one using a number of computers connected by the network. Then the computationally intensive tasks may run on separate computers. Other beneficial features of distributed architecture are openness, dynamical extensibility and mobility [6]. For mobile robots the computational intensive tasks can run on stationary off-board computers thus extending robot's battery life or even providing their services to a group of robots. Distributed approach to robot control was used, for example, in [7] to achieve minimum response in critical situations.

This paper reports our approach to designing and implementing distributed networked control system for intelligent mobile robot. We present our implementation of distributed control system for our prototype mobile robot. The control system of our robot consists of three computers connected to the high-

speed 100MBps network. Distributed design of the control system allows us to share computational load evenly between available computers to achieve maximum system performance. We discuss the system architecture, distributed communication mechanisms and robot control approaches in the following sections.

2 Control system architecture

From the general point of view our system architecture follows well established structure [2, 5]. There are two levels of control build one on the top of the other: hardware control level and robot control level (Fig 1). The hardware control level includes a set of procedures for controlling robot hardware: motors, encoders, sensors, pan-tilt-zoom cameras, etc. On this level all hardware-specific control issues are resolved and presented to the higher level in form of abstract robot state. Motor input voltages are represented by desired velocities, encoder pulses are processed to reflect current robot velocity and odometry statuses, ultrasonic sensor readings are filtered and so on.

The robot control level includes procedures for dealing with higher level control issues from motion control and sensor data interpretation to navigation, planning and intelligent interfaces. To reduce the complexity, these procedures are realized as a set of task-specific separate modules (components) that share the information and services with others. On this level the hardware control level is represented as a server component that shares with other modules the information about current robot state and set of commands to control this state. In conventional control system implementations the components of the robot control level are closely integrated between each other to establish coordinated basis for robot control.

In case of distributed networked system there are several issues that have to be addressed. The tight coupling between robot control modules is no longer possible. The components of robot control system may be executing on different computers and will not have access to the internal information of the others. So, the distributed implementation should allow the lose coupling between system components where the information is exchanged only through communication mechanisms. These communication mechanisms should be flexible enough to enable different communication strategies between system components such as broadcasting, one-to-one or one-to-many (server-client) connections. In our system we achieve loose

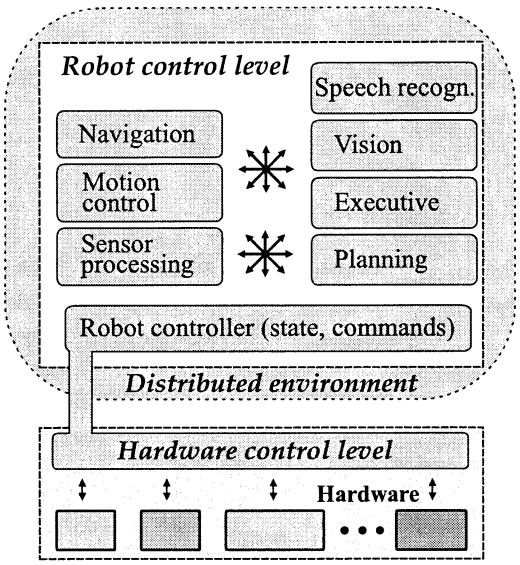


Figure 1: Distributed system architecture.

coupling and flexible communication mechanisms by using CORBA open distributed object architecture.

The distributed components and communication protocols also need to support resource sharing and fast system response in case of emergencies. Resource sharing is needed to handle situations when several modules compete for a single resource, for example to control robot motions or to move rotating camera. In our system we use queues, locking and priorities to deal with these problems.

3 Communication mechanisms

In this section we discuss the communication mechanisms which we used to implement loosely coupled distributed system with different communication strategies. In our system we use Common Object Request Broker Architecture (CORBA) architecture and infrastructure to link distributed modules of the control system together. This architecture is vendor and platform independent and can be used in a variety of areas including real-time applications. We used omniORB free implementation of CORBA.

In our system we have two levels of communications between our modules: object level and messaging level.

The object level allows a component to export its procedures and data to the the networked environment. Other system modules access exported procedures and data through module's CORBA object. This makes system architecture open and easy to extend. For example, the robot controller component

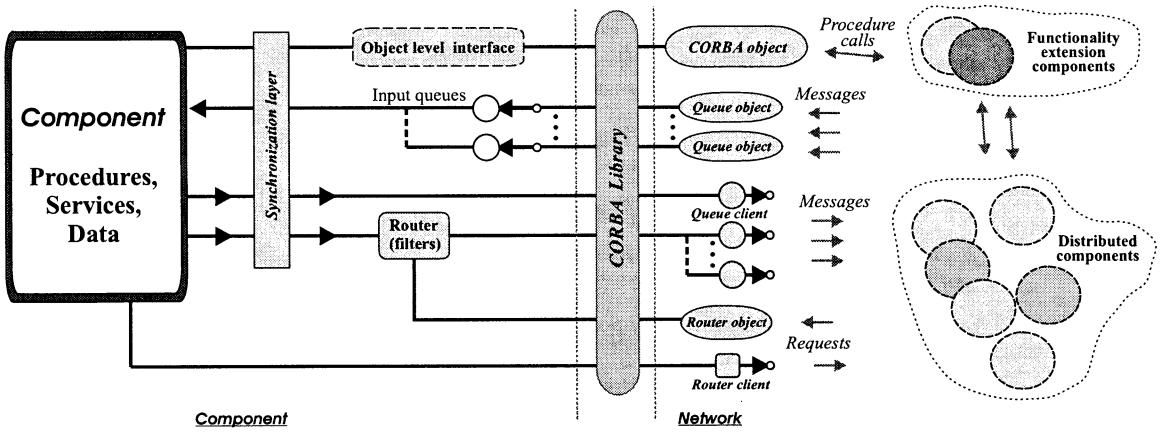


Figure 2: Distributed communication architecture.

uses object communication level to export low-level motor control routines, encoder and sensor readings. So another more specialized robot controller can be developed and included into the system without having to deal with hardware issues. The object communication level is used only when tighter coupling between system modules is required.

The greater extent of information exchange is performed on the messaging communication level. The system components are encouraged to use messaging level because it is more flexible. The modules on this level communicate by exchanging messages. Each message consists of the header and message data. The message header includes information about message context (category), message identifier, message recipient or sender, message priority, message marker. This information allows system communication algorithms to handle messages more efficiently keeping network use to the minimum. Figure 2 shows distributed communication architecture that are available to the module.

Messages are delivered to the modules with the help of queues and routers (see Fig. 2). Queues are used to receive the incoming messages and sort them according to their priority for later extraction and processing by the component. The routers provide configurable message broadcasting in case of many recipients. Queues and routers are exposed to the distributed network environment through the corresponding CORBA objects. Each queue or router object is given unique name identifier that is used by CORBA libraries to locate the appropriate queue or router on the network.

Queues receive messages from the other distributed components on network through their CORBA queue object interfaces. The control system component can have as many queues as needed. To send a message to the corresponding queue the distributed component uses COBRA queue client. The queue client uses CORBA object name service to locate the queue by its name on the network and puts the message in it. A static one-to-one communication channel between two components may be formed by posting messages to the opposite queue object.

Routers in our system provide flexible message broadcasting and dynamic one-to-one communication channels. For example, the broadcasting allows to send updates of the robot state information from robot controller to any number of system modules. Similar to the queues, the routers are exposed to distributed network environment through CORBA router objects that can be accessed by CORBA router clients. The router object maintains dynamic register of subscribed components that wish to receive messages from the component that owns the router.

The recipient component subscribes (using router client) by sending the name of the queue object to which it wishes to receive routed messages. After subscription the component can also set up the set of message filters which the router will use to select and/or discard outgoing messages based on the message header information. The messages are delivered to the recipient queues by queue clients. The router object uses CORBA libraries to check if the subscribed modules are still running and in case of conflicts updates the register accordingly. The module may use all communication mechanisms or just some of them depending on its purpose within the control system.

There is also a problem of sharing component's resources. The modules can have resources (functions, services or data) that cannot be accessed by many modules simultaneously. To handle this problem modules use synchronization mechanisms to pro-

vide correct sharing. In our system these synchronization mechanisms include command queues, request queues and resource locking.

4 Robot control with distributed components

Components in distributed system can operate with little or no outside supervision. For, example, the navigation or sensor interpretation module needs only periodic updates of the robot state to update their own state. Nevertheless, in order to operate as integrated control system the collection of distributed modules needs some form of central management. In our system this management is provided by so-called executive component.

The executive component integrates all distributed components together: coordinates their work, allows to program new robot actions and behaviors (we call them *activities*), executes these activities in real-time, etc. The executive communicates with other modules on messaging level and uses message message processing, event processing and threading libraries to perform its functions.

Robot activity programs utilize executive's message processing libraries to communicate with distributed system components and coordinate their work. Activities can create new message queues, subscribe for messages, use message filters and message triggers to be notified when specific message arrives. Event processing and threading libraries provide a framework for programming and executing robot activities. The activities use these libraries to sequence and coordinate their concurrent execution in response to distributed component messages and executive events. Executive events include timers, message triggers, synchronization and activity status signals, etc. Running activities can spawn child activities that can execute sequentially or in parallel with the parent activity. The executive component's kernel manages all interactions between described libraries and running activities. The executive kernel also schedules all currently executing activities.

5 Conclusions

In this paper we present our implementation of distributed control system. The distributed system provides open and dynamic framework for realizing mobile robot control and to share computational load between available computers to achieve maximum system performance. We focus our discussion on the system architecture, distributed communication mechanisms and distributed mobile robot control.

References

[1] R.A. Brooks, (1986), "A Robust Layered Control System for a Mobile Robot," In *IEEE J. on Robotics and Automation*, vol. RA-2, no. 1.

[2] W. Burgard, et al. (1999), "Experiences with an interactive museum tour-guide robot," In *Artif. Intelligence.*

[3] J. Connell, (1992), "A Hybrid Architecture applied to Robot Navigation," In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2719–2724.

[4] E. Gat, (1992), "Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots," In *Proc. of Conf. of American Assoc. of Artif. Intelligence.*

[5] K. Konolige, K. Mayers, E. Ruspini (1997), "The Saphira Architecture: A Design for Autonomy," In *J. of Experim. and Theor. Artificial Intelligence*, vol. 9, pp. 215–235.

[6] D.L. Martin, A.J. Cheyer, D.B. Moran (1999), "The Open Agent Architecture: A Framework for Building Distributed Software Systems," In *Applied Artificial Intelligence*, vol. 13, no. 1-2, pp. 91-128.

[7] T.M. Sobh, M. Dekhil, A.A. Efros (1996), "Sensing Under Uncertainty for Mobile Robots," In *ASME Series on Robotics and Advanced Manufacturing*, vol 3.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Research of the environmental recognition in a mobile vehicle

Research of the environmental recognition in a mobile vehicle
Masanori SUGISAKA and Shuichi OTSU
Department of Electric Engineering, Faculty of Engineering Oita University
700 Dannoharu Oita 870-1192, JAPAN
Tel: +81-97-554-7831 Fax: +81-97-554-7818

E-mail: msugi@cc.oita-u.ac.jp, s0932018@mail.cc.oita-u.ac.jp

Abstract

In this paper, we researched of the environmental recognition in a mobile recognition vehicle which our laboratory development. An important subject which a mobile recognition vehicle moves automatically is catching exactly information which it is surrounded. It is important that it is catching exactly information because of present position, state and deciding future action. Now using this mobile recognition vehicle, our laboratory research various application, such as mobile control and speech recognition. In this research, using the CCD camera the mobile recognition vehicle provided with, we research of vision. It is many method for image procession of rvehicle, in this research we performs image segmentation method. About the image segmentation method, gray scale of red, green and blue are extracted from image. And then computer calculate brightness signal, and it is the method of performing image segmentation within the image. This mobile recognition vehicle takes a image, and this performs image segmentation. Using image segmentation, the validity of a mobile recognition vehicle is verified.

Keyword: mobile recognition vehicle; image segmentation

1. Introduction

Now, a mobile recognition vehicle are researched various region and an important subject which that moves automatically is catching exactly information which it is surrounded. It is important that it is catching exactly information because of present position, state and deciding future action. That needs five wits, for example like eyes, ear, and so on. Therefore in this paper, we used CCD camera which changes men's eyes and that gets information. Using basic image processing to origin for the image is obtained from a CCD camera, we verify the validity of a mobile recognition vehicle.

2. Image Segmentation

In this paper, we performed image segmentation method for image processing to origin for the image is obtained from a mobile recognition vehicle. This method has advantage because information can be used as it is for use color image.

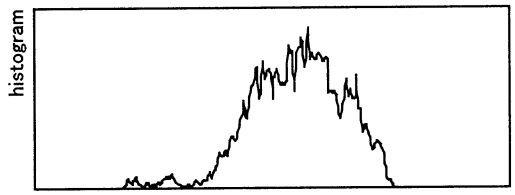
2.1. Conversion of Color Image

we abstract the data of three pixels (red(R), green(G) and blue(B)) from the color image. There are used, we calculat six value which are lightness(L), hue(H), saturation(S),brightness signal(S) and color-difference signal(I and Q). Their equations show (1).

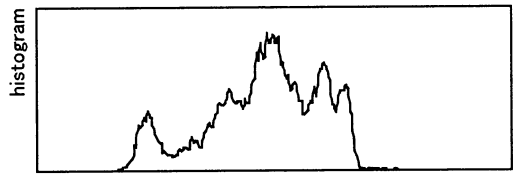
$$
\begin{aligned}
&T = R + G + B \\
&H = \tan^{-1}\frac{0.7R - 0.59G - 0.11B}{-0.3R - 0.59G + 0.89B} \\
&S = \sqrt{(0.7R + 0.59G + 0.11B)^2 + (0.3R + 0.59G + 0.11B)^2} \\
&Y = 0.299R + 0.587G + 0.144B \\
&I = 0.7R - 0.59G - 0.11B \\
&Q = -0.3R - 0.59G + 0.89B
\end{aligned}
\qquad (1)
$$

2.2. Histogram

Using nine attributes included red, blue and green, we make each histogram. If this histogram is single peaked (the number of hill is one) (Fig.1



(a) single peaked



(b) double peaked

Fig.1 histogram

(a)), we consider that this image have the number of region is one. But this histogram is double peaked (the number of hill is two or more) (Fig.1(b)), we consider that this image have the number of region is two or more, and we can perform image segmentation.

2.3. Method of deciding the threshold

It is necessary where we decide the threshold in histogram when we perform image segmentation. Fundamentally when we decide the threshold, we perform by hand. But it is not to decide by hand because a mobile recognition vehicle always act. Therefore it is necessary that it decide threshold automatically. The method is used the discriminant analyzing method for it decide threshold automatically. This method divide region between body and background, and it can decide threshold for single peaked. As the method of deciding threshold, it calculates using distribution of an overall pixel. And this method is good to divide brightness signal such as binarization. However, as seen and show in Fig. 2, threshold may be appear inside of a hill. As for the image segmentation method, it is though that it is desirable that threshold is made inside of a valley. Therefore we can not this method in case. In this paper, we perform following procedure in order to decided threshold. At first, it decides the standard, which divide a hill or valley. The standard is average which divide the gray level into the total number of pixels. If there is a histogram up the standard, this part is hill. On the other hand, if

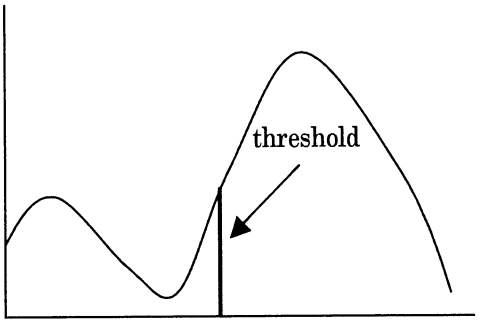


Fig.2 discriminant analyzing method

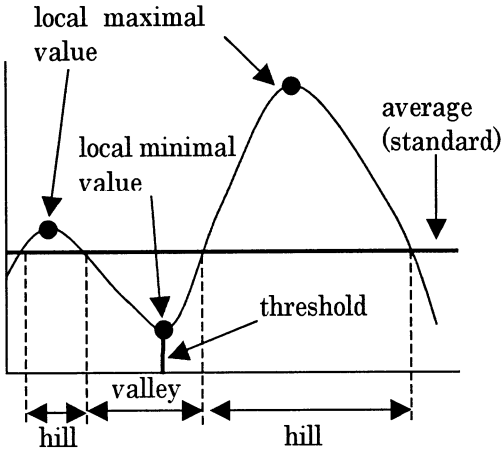


Fig3 method of deciding threshold

there is a histogram under the standard, this part is valley. At this time, it discovers the local maximal value is discovered in the part of hill and the local minimal value in the valley. And it considers the gray level which takes this local minimal value is the threshold. This situation is shown in Fig.3.

However in case that there are three or more hills (Fig.4), two or three local maximal value and local minimal value exist. As threshold exist only one, , we perform the following procedure in this case. Once again, it calculates average, and it calculates hill and valley, and it calculates local maximal value local minimal value. Next it calculate slope of a straight line which is made up of local maximal value and local minimal value. It looks for the largest slope in all them, and it considers that the gray level which takes this local minimal value which makes this straight line is threshold. Because generally it selects hill which a peak is high or width is narrow in image

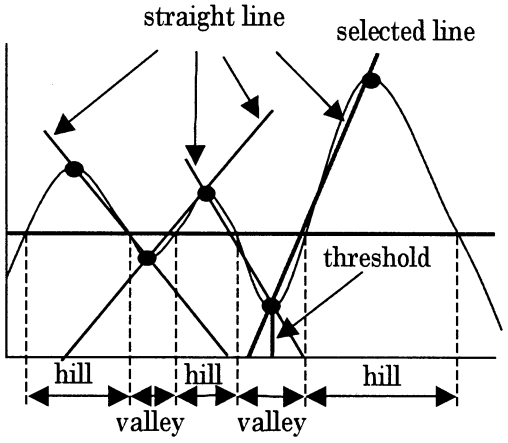


Fig.4 a lot of hill

segmentation. Therefore in this case, if a straight line is drawn from peak to valley, it think that slope is large. Such reason, we used this method. And when it selects histogram of double peaked, it looks for the largest slope in all histogram. And it selects histogram which have largest slope.

2.4 Morphological Operators

When it makes histogram, it is ideal to beautiful line. But really it can not draw it. And so if it looks for hill using that method, it finds a large number of hills. Therefore we used Morphological Operators in order to prevent this. This equation shows below.

$$\begin{aligned} &\text{dilation}: [f \oplus g^s](x) = \max_{\substack{x+u \in F \\ u \in G}} \{f(x+u) + g(u)\} \\ &\text{erosion}: [f \ominus g^s](x) = \min_{u \in G} \{f(x+u) - g(u)\} \\ &\text{opening}: f_g(x) = [(f \ominus g^s) \oplus g](x) \\ &\text{closing}: f_g(x) = [(f \oplus g^s) \ominus g](x) \end{aligned} \quad (2)$$

In this paper, we used Closing of Morphological Operators. This method fills a valley of a signal narrower than the specified width. Using this method shows Fig.5. We appreciate that parts of a large of number valleys are decrease.

3.Experiment

3.1. Outline of vehicle

The mobile recognition vehicle used by this research is what was developed uniquely at our laboratory, and experimented using this. Fig.6 is showed its diagrammatic illustration Fig.6. The

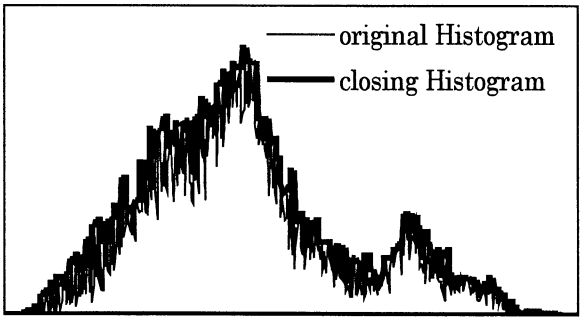


Fig.5 closing

size of this vehicle is length 440[mm], width 500[mm], and height 1300[mm], it has six ultra sonic sensors. And the vehicle has two CCD cameras. But this research we used only one came.

3.2 Experiment Method

In this research, we experimented image segmentation using image which this vehicle takes a for fundamental experiment. At the first, this vehicle takes a image using CCD camera. Using this image, it makes histogram of each attribute. After it makes histogram, using Closing of Morphology operators, it performs removal of small valleys, integration of a peak and separation, and it makes new histogram. From the histogram calculated, it distinguishes whether this is single peaked or double peaked. If all histograms are single peaked, it finishes image segmentation at this point of time. But if it exist one or more histogram of double peaked, it distinguishes best effect for dividing a image. From histogram of double peaked, it calculates local maximal value and local minimal value and it used them, it

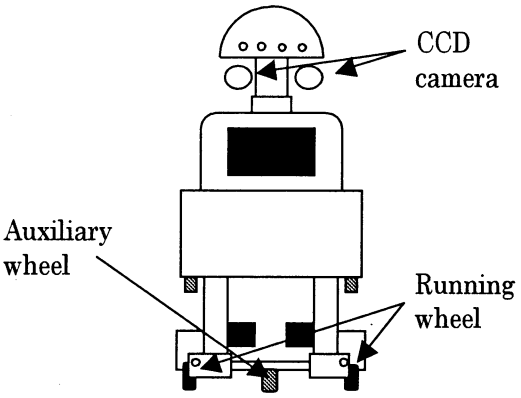


Fig.6 mobile recognition vehicle

calculates slope. This method performs all histogram of double peaked, it selects the largest slope them. From selected histogram, it decides threshold, and divides into a larger thing and a smaller thing than it. Moreover, when it performs image segmentation, noise occurs more or less. Therefore it performs smoothing after it performed image segmentation. And it continues this method until all histograms become single peaked.

3.3. Experiment Result

In Morphological operators, g (x) is delta function. It shows Fig.7 the original image which mobile recognition vehicle obtained. From this image, it obtain information of red, green, and blue, and it made histogram. It performs image segmentation. It shows left side of Fig.8 image segmentation. This shows that selected region is white, not selected region black. And from part of selected region, it draws white from the part considered to be a boundary. Also it performs smoothing in this time. It shows right side of Fig.8 image of this processing performed. Moreover, it performs image segmentation sequentially from the top. From Fig.8, we understand condition that it performed the image segmentation.

4. Conclusion

In this paper, we have ascertained validity of the automatic method of deciding threshold and the image segmentation in a mobile recognition vehicle. As a future task, when we perform image segmentation, it performs often a lot of image segmentation. We do not know causation why it performs now. So we consider that, we will make algorithm. And in this research, we perform image segmentation only. So it can not know which obtained region is obstacle or not.

Therefore we consider that we will make algorithm that mobile recognition vehicle know obstacle using image obtained image segmentation.

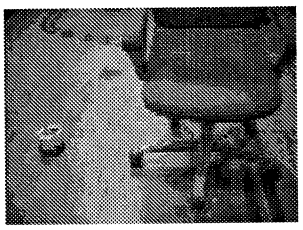


Fig.7 original image

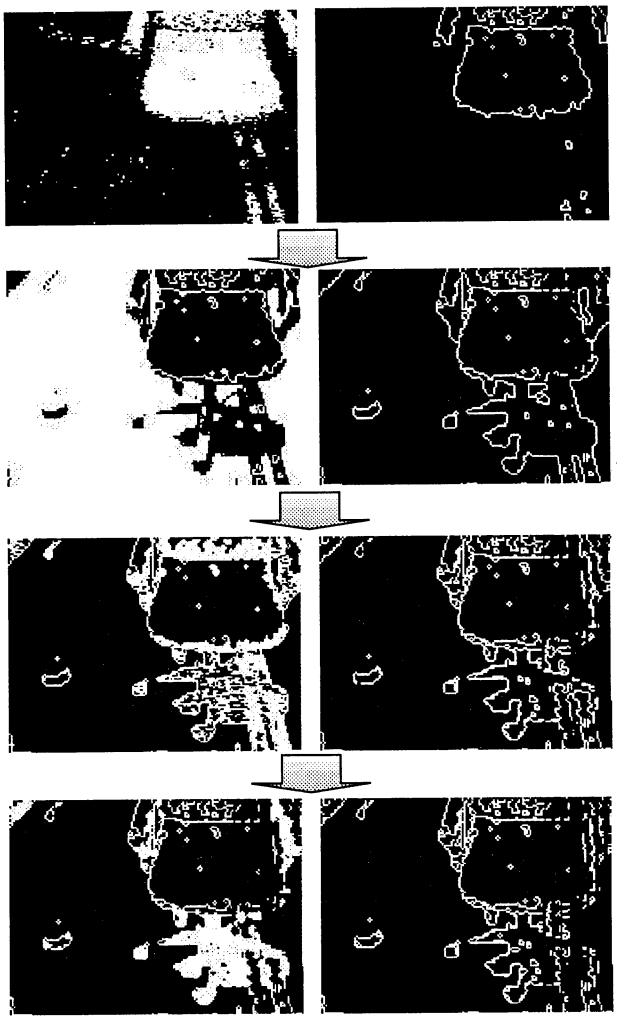


Fig.8 result of image segmentation

Reference

[1]M.YACHIDA, "Robot vision", Shokokudou(1990)

[2]T.NAGAO,T.AGUI, "Processing and Recognition of Image", Shokokudou (1992) in Japanese

[3] R.SUEMATU, H,YAMADA, "Image processing engineering" Koronasha (2000)

[4]M.SUGISAKA and T.MAEYAMA "Obstacle Avoidance in the Mobile Robot Using the CCD camera" AROB07' p238-240

[5]H.KOBATAKE "Morphology" Koronasha (1996)

[6]A.SHIJI and N.HAMADA "Color Image Segmentation Method Using Watershed Algorithm and Contour Information" IEICE, Vol.J83-D-II No.2 p593-600 Feb.2000

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

The control of the electric vehicle speed using pulse-width-modulation (PWM)

Masanori Sugisaka and MBAÏTIGA Zacharie
Department of Electrical & Electronic Engineering, Faculty of Engineering, Oita University
700 Dannoharu Oita, 870 -1192, Japan
msugi@cc.oita-u.ac.jp, zack45@cc.oita-u.ac.jp

Abstract

This paper deals with the control problem of the electric vehicle, using Pulse-Width-Modulation essentially a rapid switching on and off to vary the speed of the electric vehicle, in order to drive it to a given initial goal x-position at right angle from a given configuration. Where the speed is proportional to the pulse width modulation frequency. With the PWM the speed profile of the vehicle can be controlled by changing the rate at which "steps " are sent to the motor; so that the motor can be accelerated until some speed or position and then the control system change for instance from acceleration to constant speed and from constant speed to deceleration operation with a peak speed at about position 1500. The experimental test ascertains the merits of the proposed method and a satisfactory result have been obtained.

Key Words: Pulse-width-modulation, speed

1. Introduction

In these recent years, the control of the autonomous mobile system has become the focus of many researchers due to their importance in certain situation or places such as hospital, or helping the handicapped or elderly persons. The success of the application of autonomous vehicle [1], achieving a good control effect presents many challenging control problem due to the nonlinear dynamic of the vehicle and its difficulties of modeling the environment and its interaction with the vehicle. To solve this problem, two approaches are often used:

Quantitative approach (known as conventional one) based on analytic model of the controlled system.

Qualitative approach based on the human reasoning and learning algorithms. The first approach includes a certain number of methods as adaptive control, robust robot control, optimal control etc. It is effective when the system can be represented by an analytic model. However such representation becomes difficult when it concerns very complex system. The second approach allows to bridge the problem of the modelisation and it concerns the neural network and fuzzy control. In the field of the control of autonomous vehicle, many papers have been reported. Robust and nonlinear control strategies have been used with some success [2~5]. Neuro-fuzzy control for autonomous underwater vehicle [6], an explicit force control scheme for underwater vehicle [7], adaptive control of underwater vehicle manipulator systems [8] have been also reported. Our purpose in this paper is to make the vehicle arrive to a given goal x-position at right angle starting from a given initial configuration by controlling the speed of the vehicle. By using the pulse width modulation, the speed profile of the vehicle can be controlled by changing the rate at which "steps" are sent to the motor. So that the speed which is proportional to the PWM frequency can vary from acceleration to constant speed and from constant speed to deceleration operation in order to generate a good control trajectory to the vehicle.

2. The overview of the electric vehicle

The vehicle mark " *MINI SUZAY* " is a single rear-wheel drive; electrically powered road-vehicle produced by Daihatsu Company. The manual operator of the electric vehicle is the same as for any road vehicle. The vehicle has been retrofitted so that the brake and accelerator pedals, steering wheel and

gear can be operated by computer: because of the nature of the retrofit, the computer must control the vehicle without having direct access to the steering mechanism, drive motor, or transmission of the vehicle. That is to say, the computer must control the vehicle in the same manner as a human being would. A toggle switch located next to the steering wheel is used to change between manual and computer control. For emergencies or to disable the retrofit motors a red locking stop button is located on the right hand side of the dashboard. The computer located in the trunk-space is able to control: Two DC stepper motors and one AC servomotor, the two DC stepper motors are used to control the brake pedals (to pull it down and release) and tension the accelerator cable, while the AC servomotor is used to steer the wheel direction from left to right or vice-versa. The computer is used also for gear-changes using relays; while the speed feedback is obtained from an encoder fitted to the rear-axle. Others items which were added to the vehicle include a CCD camera with pan/tilt capability mounted on the roof for road location, a Liquid Crystal Display (LL-T1510A) touch panel mounted to the left of the steering wheel for road display. Four ultrasonic sensors located at bumper level, two at the front and two behind and four additional Liquid Emitted Diode (LEDs) placed at the dashboard. To control the car, there are two computers mounted in the trunk. These PCs are connected to the hardware via controller cards (I/O, stepper motor, servomotor drive, and frame grabber) and the serial port for camera control. They also have Ethernet cards to facilitate communication. The network message carries information about the path line seen from one PC (which have access to the frame grabber) to the PC responsible for controlling hardware. All computers use windows 98 SE Japanese version as their operating system. For development, MSVS, and the manufacturer supplied drivers are installed. The DC car batteries supply all power

3. The mechanical properties of differential drives

In this paper, a pure differential drive mobile vehicle is considered[9]. It is assume that the posture, it means that the position and the orientation of the vehicle are known at each instant. The mechanical structure of the electric vehicle (EV) is shown in Figure 2 below.

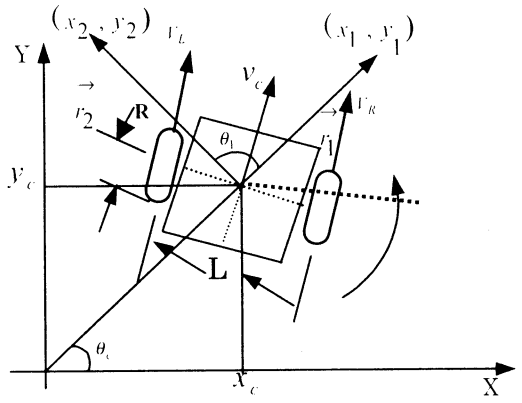


Figure 1. Modeling of differential–drive of the vehicle

Where L is the base width of the vehicle and R is the radius of the wheel. The kinematics of the vehicle can be described using Figure 1. Posture p_s and position p of the vehicle are defined as in the Formula (1) below.

$$p_s = \begin{bmatrix} x_C \\ y_C \\ \theta_C \end{bmatrix} \quad p = \begin{bmatrix} x_C \\ y_C \end{bmatrix} \tag{1}$$

where (x_C, y_C) is the position of the center of the vehicle, and θ_C is the heading angle with respect to absolute coordinates (x, y). Velocity vector is defined as shown in the formula (2), where v is the translational velocity of the vehicle and ω is the angular velocity with respect to the center of the electric vehicle.

$$s = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dfrac{V_R + V_L}{2} \\ \dfrac{V_R + V_L}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}, \frac{1}{2} \\ \frac{1}{L}, \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} \tag{2}$$

(2) gives the relation between velocity vector and the velocity of the wheel, V_L and V_R, where V_L is the left wheel and V_R is the right wheel velocity. The vehicle kinematics associated with the Jacobean matrix and the velocity vector is defined as (3).

$$\dot{p}_s = \begin{bmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = J(\theta_c)s \qquad (3)$$

To get the exact vehicle position and orientation (3) should satisfy the following nonholonomic constraint.

$$\dot{x}_c \sin\theta_c - \dot{y}_c \cos\theta_c = 0 \qquad (4)$$

Which is equivalent to

$$\frac{dy_c}{dx_c} = \tan\theta_c \qquad (6)$$

meaning that the moving direction at every instant is the same as the heading angle of the vehicle. It implies pure rolling and non-slipping as assumed. To make a vehicle move fast, not only the maximum speed of the motor but also the velocity region without slip or overturn was considered. One of the main reasons of taking caution about the slip or overturn is the large centrifugal force over the wheel friction limit. The restrictions for the vehicle are given as follows. V_m is the maximum speed of each wheel and R_m is the maximum turning speed

4. Pulse width modulation and speed

The host computer CPU Pentium II celeron 466Mhz does the control of all motors and devices connected to the driving card PCI7208 without having direct access to the drive motor or transmission. The motor use on the EV is a DC stepper motor, and the stepper motor does not rotate continuously, but turn in fixed increment, and resist a change in their fixed position. Therefore to control the speed of the vehicle, we have used the pulse width modulation to switch the power supplied to the motor ON and OFF very rapidly, the percentage of time that the power is ON determine the percentage of full operating power that is accomplished. How the P.W.M was used to control the speed? Our motor is 24 V motor, and when we took this 24 V motor and switch ON the power to it, the motor stared to speed up near the maximum speed (Figure 2. C) as the motor do not respond immediately it took some small time to reach the full speed. If we switch the power sometime before the motor reaches full speed, the motor started to slow down (Figure 2. A). If we switch the power ON and OFF very quickly enough, the motor run at some speed part way between zero and full speed, that is to say, that if the switching frequency is high enough, the motor runs at steady speed (Figure 2.B).

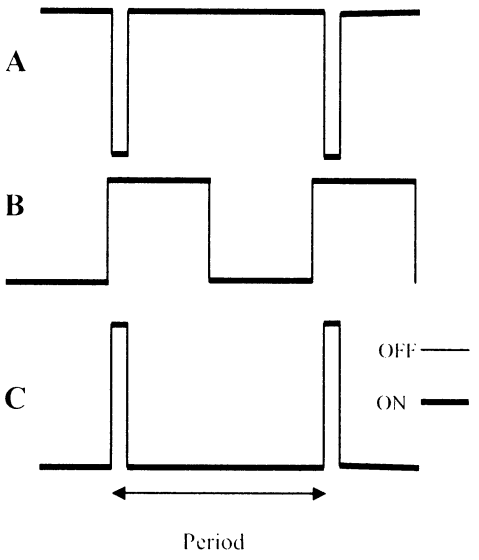


Figure 2. Pulse width modulation signal to operate the motor

Consider the wave form above, in A the switch is opens for a short time and closed for a long time, so the vehicle run at low speed. While in B the switch is ON 50% and OFF 50% that allowed the vehicle running at steady speed. The steady speed can be kept as long as possible. In C the motor is ON for most of time (about 80%) and OFF only for a while (about 20%) so the speed is near the maximum. By continuing and repeating this on-off duty cycle over and over, the voltage is changing so quickly that the on's and off's become an average voltage. Therefore, the speed of the motor can be changed by varying the amount of time the current is on and the current is off.

5. Results.

In this section, the Figure 3 below shows the variation of the speed of the vehicle that was controlled by varying the average voltage applied to the motor using the pulse width modulation, which was generated using the timer and counter and output compare registers present in the controller. The experiment test

was done for about 19 minutes and comprises three phases.

Phase 1. Acceleration

When the power supplied to the motor was ON for long time, the motor started and quickly reaches 400Hz and gains its maximum speed of approximately 2500Hz. As a motor do not respond immediately, it took about few minutes to reach the full speed with 8000 rotation per minutes.

Phase 2. Steady speed

After the motor reaches its maximum speed, we then hold that speed for about four minutes, that means that the power supplied was ON for 50% and OFF for 50% as well.

Phase 3. Deceleration

As it can be seen, after holding the speed for a while, it started to decrease slowly from 11 minutes and reach its low level at 18 minutes. That is to say the power is off for long time. The advantage of pulse width modulation is that the pulses reach the full supply voltage and will produce more torque in a motor by being able to overcome the internal motor resistances more easily

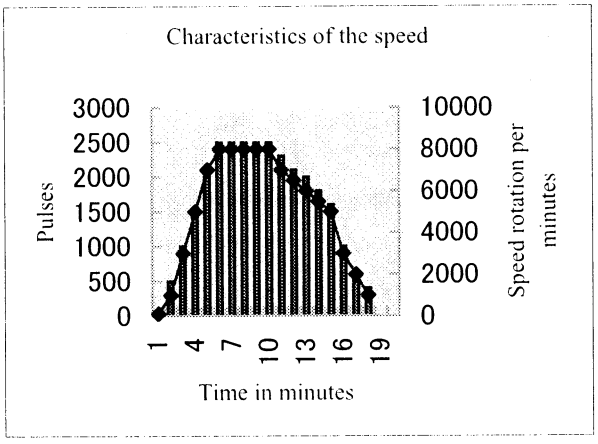


Figure 3. Three phases of the speed

6. Conclusion

The goal of this study is to find a good way of controlling an autonomous vehicle capable of traveling on a real-road vehicle without the help of a chauffeur. To make the EV arrive to a given goal x-position starting from a given initial position, we used the pulse width modulation to control the speed of the vehicle by switching ON and OFF very quickly the power supplied to the motor in order to make a variation of the speed. The control method as well as the experimental result was discussed.

References

[1] Masanori Sugisaka and Xin Wang, Ju-Jung Lee Intelligent control strategies for mobile vehicle , Applied mathematics and computation pp 91～98 (1998)

[2] Thor I.Fossen, guidance and control of ocean vehicles John Wilely and sons Ltd, 1994

[3] D.N Yoerger and J.E Slotine, Robust trajectory control of under water vehicles, IEEES ocean engineering, vol.10, No.4 pp 462～470 (19985)

[4] J. Yuh " Modeling and control of underwater robotic vehicles " IEEE Trans. Syst. Man Cybern, vol SMC-20 No.6 pp 1475～1483 (1990)

[5] Yong Cui, Tarun K. Podder, Nilanjan Sarkar " Impedance Control of Underwater Vehicles-Manipulator System" Proceedings of the 1999 IEEE/RSJ, International Conf on Intelligent Robots and Systems Vol. 1 pp148 ～ 153 (IROS'99)

[6] Jeen-Shing Wang C.S Georg Lee and Junku Yul. "Self-Adaptive Neuro-Fuzzy With Fuzzy Basis Function Network for Autonomous Underwater Vehicles Procee dings of the 1999 IEEE/RSJ, International Conf on Intelligent Robots and Systems Vol.1 pp130～135

[7] Gianluca Antonelli, Stefano Chiaverini and Nilanjan Sarkar. An Explicit Force Control Scheme for Underwater Vehicle Manipulator System. Proceedings of the 1999 IEEE/RSJ, International Conf on Intelligent Robots and Systems Vol. 1 pp136～141

[8] Nilanjan Sarkar, Junku Yul. and Tarun K. Podder," Adaptive Control for Underwater Vehicle-Manipulator subject to joint limits. Proceedings of the 1999 IEEE/IRS Int. Conf on Intelligent Robots and Syst vol.1 pp142～147

[9] Masanori Sugisaka and MBAÏTIGA Zacharie, The Controlof the Electric Vehicle Steering wheel using an AC servomotor. The SICE Kyushu Annual Conference 2002. Oita University

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Dynamics and Control of Non-holonomic Two Wheeled Inverted Pendulum Robot

D. Y. Lee, Y. H. Kim, B. S. Kim*, Y. K. Kwak
Department of Mechanical Engineering
Korea Advanced Institute of Science & Technology
373-1, Guseong-dong, Yuseong-gu, Daejon, 305-701
kyk@kaist.ac.kr
* Hanwool Robotics Corp.

Abstract

In most wheeled mobile robots, at least one wheel is the auxiliary wheel aka caster. It must move smoothly without causing the robot to be interrupted if they are to truly do their job. Looking at it practically, however, they don't work sometimes. That is, the wheels frequently slid or slip when they were dragged, even though they were designed to roll without sliding or slipping. In order to free from this problem, one of the solutions is getting rid of it. Then, the total number of wheels attached on the robot changes, moreover, the mechanical characteristic of the robot having only two driving wheels without caster will be altered to that the robot is supposed to move and balance its body with only two driving wheels. Therefore, for this inverted pendulum type robot, it is necessary to investigate whether it is valid proposal or not and what mechanical characteristics it has. For doing this, dynamics of this kind of robot was governed to provide lots of information that will be helpful for design and control. And experiments with various motions were carried out to show its practical validity.

Keywords: Wheeled mobile robot, Inverted pendulum, Non-holonomic system

1 Introduction

Several kinds of wheels are attached to wheeled mobile robot, but they can fall into one of two categories: driving and auxiliary wheels. The former ones are rotated to permit the robot to move with torque being applied to the axles of those driving wheels. On the other side, the latter ones are equipped merely to ease the movement of the robot and suspend its body, and no driving torque is applied to their axles.

In most wheeled mobile robots, at least one wheel is the auxiliary wheel. In order to be free from problems of auxiliary wheels, it would be desirable to make the operation of those better. Getting rid of them, however, would also be another feasible idea. Going a step forward, those auxiliary wheels might be replaced with something different from them.

For this two wheeled mobile robot, it is to be questioned that what will happen if their auxiliary wheel was removed instead of replacing those wheels with something different or improving its performance without taking away. One of the changes due to getting rid of those wheels would be the total number of wheels attached on the robot. Moreover, the mechanical characteristic of the robot having only two driving wheels without any other auxiliary ones will be completely altered because there are no elements that can suspend and balance the body except for the driving wheels. That is, the robot is supposed to move and balance its body with only two driving wheels. Therefore, for this inverted pendulum type robot, it is necessary to investigate whether it is a valid proposal or not and what mechanical characteristics it has.

In 1992, Konayagi it al. [1] built an autonomous self-contained inverted pendulum robot and proposed two dimensional trajectory control algorithm for that kind of robot. In 1996, Ha and Yuta [2] proposed another inverted pendulum type self-contained mobile robot. They succeeded in balancing and trajectory control using a simplified model of two dimensional inverted pendulum. In 2000, Segway [3] developed a human transporter whose speed and direction were controlled solely by the rider's shifting weight and a manual turning mechanism on one of its handlebars. In February 2002, Grasser et al. [4] presented same mechanism with three dimensional model.

Hardware design and fabrication of the robot will be presented in chapter 2. Then, the equation of motion of the designed robot is derived in chapter 3, considering its constraints. Simulation results to test the model and to design the controller are presented in chapter 4. Finally, implementing results to the real system will be showed.

2 System Design and Fabrication

As shown in Figure 1, the two wheeled inverted

pendulum robot is organized with main body, gear sets, drive wheels, motors, motor controllers, feedback sensors, and PC. The main body is made of aluminum plates and bars, and there is room to put the necessary electronic boards and sensors on the body. The body is equipped with two DC motors, and the gear of the motors is in contact with that of the wheels in the right and left sides. These motors are powered by the battery patch attached on the bottom of the robot, and they are controlled by the motion control processor. Two incremental encoders were adopted to monitor the behavior of the motors. In addition to those two encoders, gyroscope and tilt sensor were mounted on the body of the robot to measure the inclination angle and angular velocity of the robot. All these parts constitute the robot and they are controlled throughout by the host PC.

Like figure 2, the DC motors connected with the drive wheels are to provide torque to the axles for the upright balancing and navigation of the vehicle.

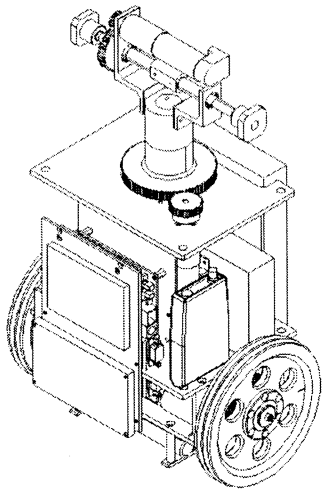


Figure 1. The designed robot

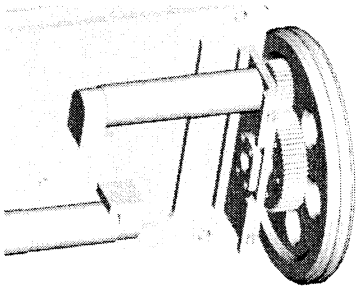


Figure 2. The interconnection between the wheel and the DC motor

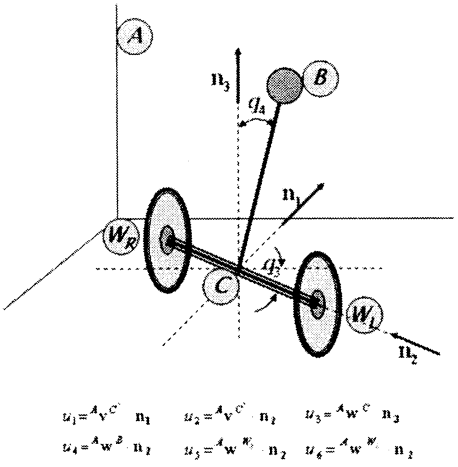
Two kinds of sensors such as tilt sensor and gyroscope were used to measure the tilt angle of the robot and its time derivative. The important considerations in selecting those sensors were its resolution, bandwidth, linear range, sensitivity, and size, as listed in Table 1. In practice, normally a physical measurement of a control system is detected using a feedback sensor, and then its derivative or integration is calculated from that detection. However, it is possible that the numerical integration of the gyroscope output implies drift error. Thus, in this control system, the tilt angle and angular velocity of the robot's body are measured independently by using tilt sensor and gyroscope, respectively.

Table 1. The feedback sensors

	Tilt Sensor	Gyroscope
Resolution	0.05 (° rms)	0.1 (°/sec)
Bandwidth	125 (Hz)	7 (Hz)
Linear Range	± 20 (°)	± 80 (°/sec)
Sensitivity	35 ± 2 (mV/°)	20.4 (mV/°/sec)
Size (mm)	19.1×47.63×25.4	37×46×18.5

3 Dynamic Model

The equations of motion need to be derived to carry out the control and further researches including dynamic analysis. However, not only the derivation itself is important, but also it is essential to establish the exact dynamic model. Two different approaches have been adopted to formulate more exact dynamic model, i.e., Kane's dynamical equations and Lagrange's equations of motion.



$u_1 = {}^{A}v^{C} \cdot n_1 \quad u_2 = {}^{A}v^{C} \cdot n_2 \quad u_3 = {}^{A}w^{C} \cdot n_3$

$u_4 = {}^{A}w^{B} \cdot n_1 \quad u_5 = {}^{A}w^{W_L} \cdot n_2 \quad u_6 = {}^{A}w^{W_L} \cdot n_3$

Figure 3. Coordinate system

As illustrated in Figure 3, the system is composed of three rigid bodies such as the right drive wheel W_R, the left drive wheel W_L, and the body B that is assumed to be a particle having total mass of the body in the point of the body's center of gravity. Each of them can be described with three Cartesian coordinates in a reference frame A, so the total number of the Cartesian coordinates of the system is 9.

Finally, for a non-holonomic inverted pendulum vehicle system possessing 3 degrees of freedom in A, 3 equations of motion is derived like followings. To test

the reliability of the equations, the results by Lgrange's equations can be compared to that by Kane's dynamical equations. The detailed comparison can be leaved out for convenience. However, the results from both approaches were completely identical. Therefore, the derived equations of motion can be said to be reliable.

$$
\begin{aligned}
&M_b\dot{u}_1+\frac{2}{R^2}I_w\dot{u}_1+2M_w\dot{u}_1+M_bL\dot{u}_4\cos q_4\\
&\quad -M_bLu_3^2\sin q_4-M_bLu_4^2\sin q_4=\frac{1}{R}(\tau_R+\tau_L)\\
&M_bL^2\dot{u}_3\sin^2 q_4-M_bLu_1u_3\sin q_4+2M_wb^2\dot{u}_3+I_b\dot{u}_3+2I_w\dot{u}_3\\
&\quad +2M_bL^2u_3u_4\cos q_4\sin q_4+\frac{2b^2}{R^2}I_w\dot{u}_3=\frac{b}{R}(\tau_R-\tau_L)\\
&M_bL\dot{u}_1\cos q_4-M_bL^2u_3^2\sin q_4\cos q_4+M_bL^2\dot{u}_4\\
&\quad +J_b\dot{u}_4-M_bgL\sin q_4=-(\tau_R+\tau_L)
\end{aligned}
\tag{1}
$$

4 Controller Design

For the proposed system, the equations of motion were established. Thus, it can be linearized and then the state-space equations can be written as in Eq. (2), where the state vector and inputs are defined as in Eq. (3) and the matrices A, B are identified as in Eqs. (4), (5) and C is 6×6 *I* matrix and D is 6×2 null matrix.

$$
\begin{aligned}
\dot{\mathbf{x}}&=\mathbf{Ax}+\mathbf{Bu}\\
\mathbf{y}&=\mathbf{Cx}+\mathbf{Du}
\end{aligned}
\tag{2}
$$

$$
\mathbf{x}=\begin{bmatrix}q_1\\ \dot{q}_1\\ q_3\\ \dot{q}_3\\ q_4\\ \dot{q}_4\end{bmatrix}\qquad \mathbf{u}=\begin{bmatrix}\tau_R\\ \tau_L\end{bmatrix}
\tag{3}
$$

$$
A=\begin{bmatrix}0&1&0&0&0&0\\ 0&0&0&0&-1.885&0\\ 0&0&0&1&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&25.77&0\end{bmatrix}
\tag{4}
$$

$$
B=\begin{bmatrix}0&0\\ 3.213&3.213\\ 0&0\\ 31.22&-31.22\\ 0&0\\ -12.19&-12.19\end{bmatrix}
\tag{5}
$$

The adopted controller is LQR, the optimal controller which asymptotically stabilizes the feedback system of the augmented system and minimizes the performance index J. The cost function J is stated in Eq. (6) and the solution of this problem is derived by the matrix Riccati equation as in Eq. (7). The state feedback gain matrix is computed as Eq. (8) and also the stability is confirmed by the fact that the closed loop poles are located in LHP as in Eq. (9). And the simulation of upright balancing was carried out with the inclined initial condition.

$$
J=\int_0^T\left(\vec{x}'Q\vec{x}+u'Ru\right)dt
\tag{6}
$$

$$
\begin{aligned}
K(t)&=R^{-1}B'P(t)\\
-\dot{P}&=A'P+PA+Q-PBR^{-1}B'P
\end{aligned}
\tag{7}
$$

$$
K=\begin{pmatrix}-3.1623&-4.1756&1.5811&0.7481&-11.7320&-2.0988\\ -3.1623&-4.1756&-1.5811&-0.7481&-11.7320&-2.0908\end{pmatrix}
\tag{8}
$$

$$
C.L.poles=\begin{pmatrix}-11.4653+7.3506i\\ -11.4653-7.3506i\\ -1.1780+0.8080i\\ -1.1780-0.8080i\\ -44.1030\\ -2.2389\end{pmatrix}
\tag{9}
$$

5 Control

The designed controller is implemented on the actual control hardware. Figure 4 shows inclination angle and its velocity during the up-right balancing. With the constant initial inclined angle, the robot started over to the stable balancing state.

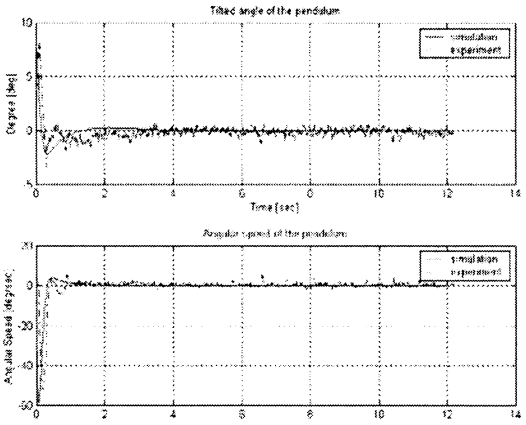


Figure 4. Up-right balancing

Figure 5 shows the results of rectilinear velocities while doing the motion. The robot follows the reference quite well, even though there exists the reaction time delay to the reference.

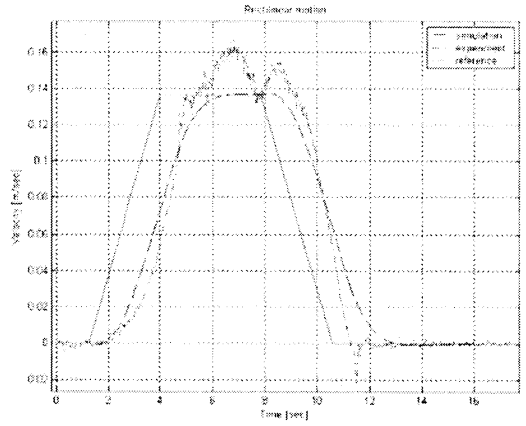


Figure 5. Rectilinear motion

Figure 6 show the results of the spinning angular velocity while doing the motion on the fixed world position. This confirms the fact that wheels without casters has a merit of mobility.

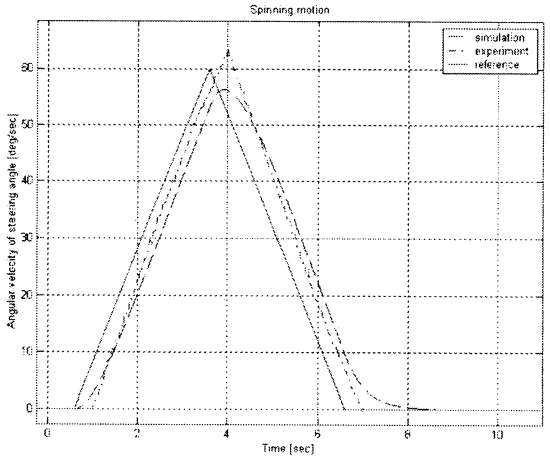


Figure 6. Spinning motion

6 Conclusion

The auxiliary wheels were removed from the general wheeled mobile robot and the system became the non-holonomic two wheeled inverted pendulum robot. The equations of motion were found by Lagrange's equations and Kane's dynamical equations, and both approaches provided the same results. Thus, the derived equations of motion can be said to be reliable. Based on the derived equations of motion, the controller was selected as LQR. Using this, the state-feedback gain matrix was found, and it was applied to the simulation. To implement the designed controller on the actual control hardware, the torque was modeled as a function of the angular velocity and acceleration of the drive wheel.

One of the two ultimate reasons why the dynamic model was derived is to construct the control system based on not trial and error but the theoretical model governing the system. To carry out more exact dynamic analysis, more exact dynamic model should be found. Therefore, the dynamic model of the two wheeled inverted pendulum robot on uneven terrain will be constructed considering the friction condition between the drive wheels and ground.

Acknowledgements

This work was supported in part by the Brain Korea 21 Project.

References

[1] E. Koyanagi, S. Iida, K. Kimoto and S. Yuta, A wheeled inverse pendulum type self-contained mobile robot and its two-dimensional trajectory control, *Proc. of ISMCR '92*, pp 891-898, 1992.

[2] Y. Ha and S. Yuta, Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot, Robotics and Autonomous Systems, pp 65-80, 1996.

[3] http://www.segway.com

[4] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, JOE: A mobile, inverted pendulum, IEEE Transactions on Industrial Electronics, Vol. 49, No. 1, pp 107-114, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Driver Intention Recognition Using Case Base Learning For Human Centered System

Toru Yamaguchi
PRESTO, Japan Science and Technology Corporation (JST)
Department of Electronic Systems Engineering, Tokyo Metropolitan Institute of Technology
Asahigaoka 6-6, Hino City, Tokyo
Japan
Email : yamachan@fml.ec.tmit.ac.jp

Chen Dayong
Department of Electronic Systems Engineering, Tokyo Metropolitan Institute of Technology
Yamaguchi lab, Asahigaoka 6-6, Hino City, Tokyo
Japan
Email : chen@fml.ec.tmit.ac.jp

Abstract— **Driver careless and lack of information about surrounding objects cause most traffic accidents. So in this paper, we propose a model of human intention recognition by case base learning. In order to recognize intention (turn left, right or straight), we detect human motion, handle condition, distance of the car to intersection. Furthermore, Improving the precision of recognition, we use case base learning. Using this method, the PC can catch driver's intention early and show dangerous information, before practical action.**

Keywords—**CFS, Case-base learning, Ontology,**

1. INTRODUCTION

ITS builds a person, a road and a vehicle as the incorporating system by utilizing cutting-edge information technology and is expected as the system which aims for the safety to improve, for the comfortableness to improve. As for the safe driving support field of ITS, a big effect to the accident reduction is expected of the traffic accident which continues to increase. To prevent an accident from happening, the preliminary dangerous warning becomes important.

Therefore, at this paper, we propose the driving support system that used the intention of the driver in human vehicle system to think much of the human nature. The human vehicle system is a human centered traffic system that regard human safety and comfortableness in ITS. It only shows the information which driver needs using the driver's intention, it can prevent confusion by the information glut and decrease driver's load. In the experiment that uses human vehicle, we detect driver's operation, stepping condition of accelerator and distance to the intersection, to recognize driver intention by intention recognition mechanism, which arranged a case node.

By arrange a case node, the recognition percentage is improved with the increase of the learning case and the inductive learning is done

2. Architecture of Intention Recognition Model

2.1 The Recall of Abstraction Concept by Case Base

In CFS (Conceptual Fuzzy Set), the most similar higher rank concept is recalled by activity of case. In fig.1, the result, which activated guppy and cat, pet is recalled. Tuna is activated as the noise, it is ignored because it is not a crowd. On the other hand, fish is recalled from guppy and tuna. Even guppy and tuna exist in both case, but we can find the concept that catch tendency is recalled respectively. This nature can be applied to intention recognition.

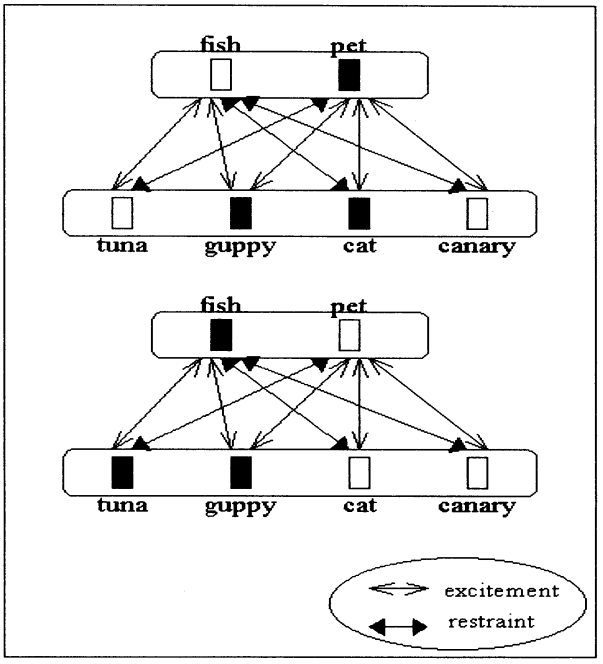


Fig.1 Concept Recall by CFS

CFS has flowing characteristics:

a) The meaning of changing concept is expressed depend on state. Moreover, because it is not the resolution of the lower rank concept, excitement of combination doesn't happen.

b) When even specified knowledge expression isn't logically made, the knowledge can be built. However, the knowledge can be specifically read in the result.

c) Because the knowledge expression is multiple structures, the denotative and connotative expression can be intermingled. When the expression with various

forms becomes possible, it is possible to do characteristic processing.

d) Reasoning, too, is expressed in activated value distribution's transmission. The reasoning that depend on the context and interactive reasoning becomes possible, too.

2.2 Architecture of Intention Recognition Model

In this paper, as the technique of the knowledge architecture about intention of the human being that is an ambiguous concept, we show the research, which used knowledge expression of CFS.

The intention recognition model that recognizes 3 basic operation intention of turning left, going straight, turning right is explained in Fig.2.

This model is loaded into Fuzzy Association memory system and association reasoning is executed. The model is composed of 3 layers. The lowest layer is entry layer and expresses the characteristic quantity of each operation by the member ship value of fuzzy label. The middle layer arranges the case node that shows case to use in the process of the learning. Because for N cases, its learn 3 operation, so its learn 3N pattern in amount. In the middle layer, when the characteristic of each operation is inputted, the node of human being that has the most similar characteristic is strongly activated. By the activated value distribution, we can understand the operation is similar with which case in each part. However, only use activated value, which appears in the middle layer, the operation intention can't be specified. This problem is solved by the introduction of context. The top layer shows operation intention, and consists of three nodes match off against turning left, going straight, turning right. The top layer is combining with all nodes of middle layer.

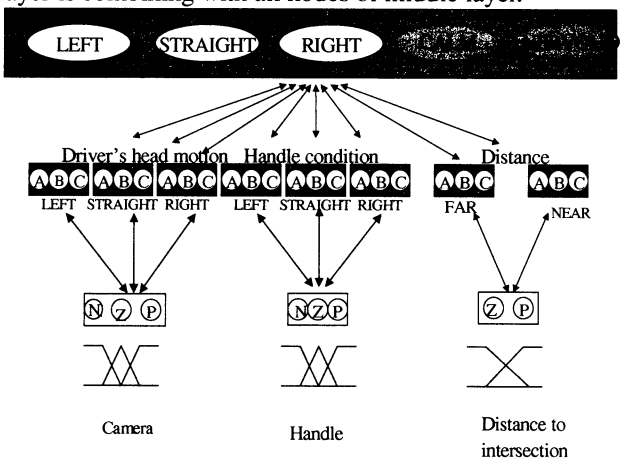


Fig.2 The model of Intention Recognition

This model memories operation intention of instance as fuzzy set, when giving input, the reverberation repeats between low layer and middle layer, middle layer and top layer. Activated value distribution converges on the condition that doesn't contradict context and model gets a recognition result. Context is standard pattern that is gotten from case of operation intention. The combination of each operation characteristic that resembled pattern is promoted, on the other hand, combination of characteristic that doesn't suit pattern is restrained, model does context sensitive recognition. Recognition result is shown by active value of three nodes that are equivalent to three operations intention in the top layer.

3. Architecture of Ontology

In usual knowledge expression, complicated concept subdivides to small concept and is expressed. The one have subdivision of the small concept, systematization of characteristic is technological ontology model, and it attempts to build common part of the knowledge structure. There is a research that expresses " state" and intention "label" by interaction with human being and outside world. In these researches, concrete case from the abstract expression by the interaction with human being, it is opposite, abstraction expression forms from concrete instance. It builds association database. Figure3 shows basics architecture of ontology by human observation of intelligent agent. By using supporting ontology, the ontology about the instruction of human being from agent is generated. It is shown in figure.4.

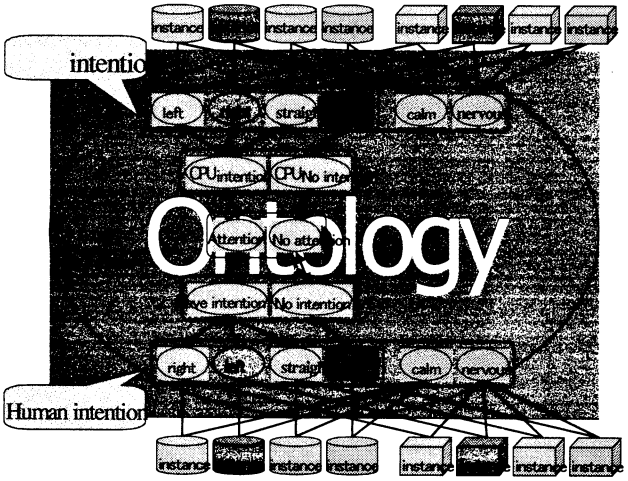


Fig.3 Generation of Ontology by Human Observation

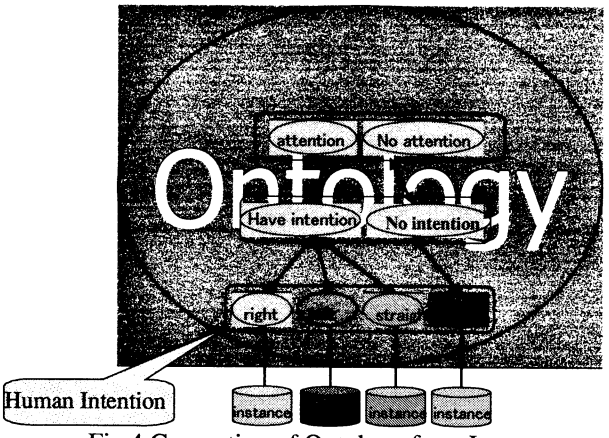


Fig.4 Generation of Ontology from Instance

The model of intention recognition is used in this experiment, the knowledge about operation with remarkable personality can be gained by arranging case node in the middle layer, and new context that subdivides knowledge can be generated. For example, it is classified into operation by the feeling of driver, too. If one learning instance is nervous operation case, like shown in Fig.4, the new nervous node is generated. In this way, using new matching expression of context and text, ontology is formed. The technique of such concept description and concept forming is extended to the dialog among intelligent agents, information sharing, becomes possible with matching of concept have been formed.

4. Experiment of Intention Recognition

4.1 Experiment System

We use the system shown in Fig.5 to do the experiment of intention recognition.

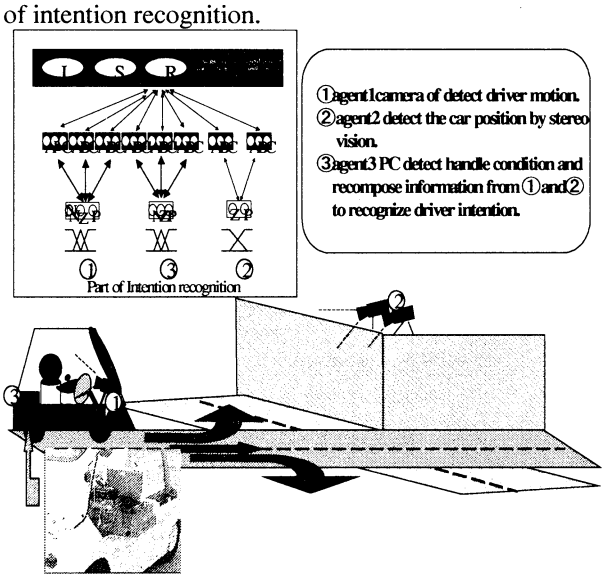


Fig.5 Experiment System

This system includes CCD cameras (agent 1,agent2) that detect driver's head motion and vehicle distance to intersection, and PC (agent3) that detect handle condition. These agents send data to sever that has intention recognition model. In this system, agent3 detect operation of handle directly, agent1 detect face direction and position by information from CCD camera. Agent2 is used in the same way.

4.2 Detection of Action

In this part, we explain that color attribute detects the head motion and vehicle position. Fig.6 shows the image that CCD camera track color objects. In fig.6(a), the color object is the marker on the hat, in fig.6(b), it is color of vehicle. In 640 X 480 pixel NTSC signal picture, each pixel is expressed by 5bits using RGB every about 100 [*mnec*], respectively. And human's skin color area is distinguished by filtering pixel within fixed error value with RGB value of a human's skin color (RGB band pass filter), and clustering the area.. The red mark is color object area. Drive's head direction and position is detected by position relation of mark.

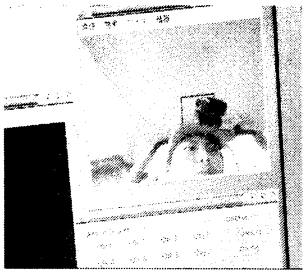


Fig.6 (a) detecting driver's head motion

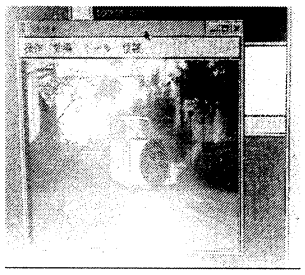


Fig.6 (b) detecting vehicle position by stereo vision

Fig.6(a) shows image of detecting driver's head motion by the mark on the his head. Fig.6(b) shows the image of detecting the color of vehicle, calculate position of it by stereo vision.

4.3 Driving Operation and Intention Recognition

When doing some operation action, a series of operation that was decided in the degree is confirmed. The intention of operation is reasoned by detecting these series of operation. When driver want to turn right, he do such series of operation, looking straightly, seeing the front of the right, speeding down, seeing the front of the right again, then taking out the right blinker, and turning handle to right. These operation is detected and agent3 reason intention. In the step that is as early as possible, such as before turning handle, intention recognition is effective in driving support. It is very important to recognize driver's intention early, then sending this intention to other vehicle and pedestrian, and getting necessary dangerous information from other agent.

4.4 Experiment of Recognition

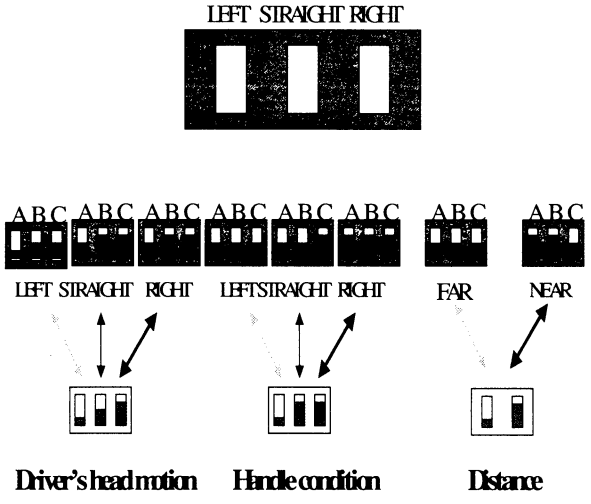


Fig.7 Result of Intention recognition (without context)

The operation data of un-learned case is inputted into the model of intention recognition that has learned 8 patterns of 3 peoples. The result that don't use context between top layer and meddle layer is shown in fig.7. In this result, the node activated value distribution of middle layer is ambiguous and can't which operation intention it is.

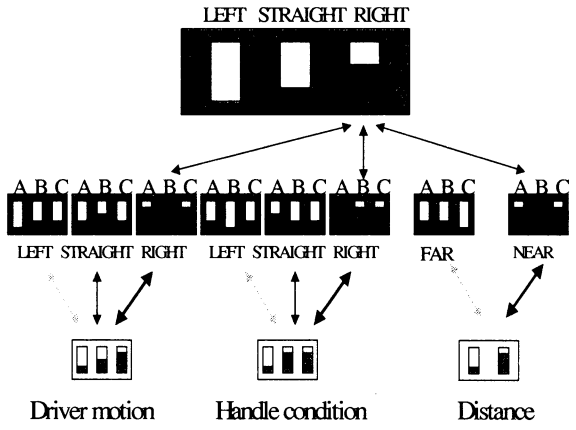


Fig.8 Result of Intention Recognition (using context)

Fig.8 shows the result that used context. The fuzzy entropy decrease by the reverberation of the interactive association memory, and the node activated value distribution converges to turn right. In this way, when even the entry value and the first activated condition are ambiguous, the activated value distribution converges on the condition that depended on context.

In this experiment, we just show the model intention recognition that used case-base learning is good method to recognize driver intention. And we will get much more operation data of driver, to show it is effective in intention recognition.

5. Conclusion

In this paper, we proposed a model of intention recognition using case-base learning; it is based on Conceptual Fuzzy Set. We selected operation data of 3 peoples, total 8 patterns, and input them to model. After learning these patterns, when one series of operation data that haven't been learned is inputted, this model can recognize driver intention (turning right, left or straight) at early step, before driver turn handle. We will continue to do experiment to show validity of this method.

In ITS (Intelligent Transport System), by this system, driver's intention is recognized early and quickly, and this information is sent to other car and dangerous information is shown to the driver. It is a valid way for driving support and reduction of traffic accident.

References

[1] T.Takagi, A.Imura, H.Ushida, T.Yamaguchi, "Conceptual Fuzzy Sets as a Meaning Representation and their Inductive Construction" *International Journal of Intelligent Systems*, Vol.10, 929-45, 1995.

[2] T.Takagi, A.Imura. H.Ushida, T.Yamaguchi, "Multi Layered Reasoning by Means of Conceptual Fuzzy Sets" *International Journal of Intelligent Systems*, Vol.11, 97-111, 1996.

[3] T.Tanihashi, M.Hagiwara, T.Yamaguchi, "Neural Network and Fuzzy Signal Treatment" *Corona inc*, 1998.

[4] T.Takagi, T.Yamaguchi, M.Sato, "Multi-Modal Information Integration by Conceptual Fuzzy Set for Interactive System" *1998 IEEE World Congress on Computational Intelligence*, 738-743, 1998.

[5] T.Takagi, S.Kasuya, M.Mukaidono, T.Yamaguchi, T.Kokubo, "Realization of Sound-scape Agent by the Fusion of Conceptual Fuzzy Sets and Ontology" *8th International Conference on Fuzzy Systems FUZZ-IEE'99, II*, 801-806, 1999

[6] T.Takagi, S.Kasuya, M.Mukaidono, T.Yamaguchi, "Conceptual Matching and its Application to Selection of TV Programs and BGMs", *IEEE International Conference on Systems, Man and Cybernetics SMC'99, III*, 269-273, *1999*.

[7] T.Takagi, K,Kawase, K.Otsuka, T.Yamaguchi, "Data retieval Using Conceptual Fuzzy Sets" *9th International Conference on Fuzzy Systems FUZZ-IEEE 2000, II*, 94-99, 2000.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Topic Stream Extraction based on Immune Network Model

Yasufumi TAKAMA
Tokyo Metropolitan Institute of Technology
PREST, Japan Science and Technology Corporation
6-6 Asahigaoka, Hino, 191-0065 Tokyo, Japan

Abstract

A method to find topic distribution from a sequence of document sets is proposed. As the Web becomes one of the most important information resources for us, the interaction between a human and a Web interface should be considered from various viewpoints, besides that of document retrieval. In this paper, we focus on the extraction and visualization of topic distribution over the Web. The proposed clustering method employs the immune network model, in which the property of memory cell is used to find the topical relation among document sets. The effectiveness of the proposed method is also shown by applying it to two sequences of online news articles.

1 introduction

A method to find topic distribution from a sequence of document sets is proposed. As the Web becomes one of the most important information resources for us, the interaction between a human and a Web interface should be considered from various viewpoints. For example, we do not only want to get certain HTML pages, but also know from the Web what is the current trends, major topics, etc., in the real world. Therefore, extracting and visualizing topic distribution over the Web space is one of the most promising applications on the Web. In particular, there are so many online-news sites on the Web and they constantly release up-to-date news articles of various topics. Our focues is on the extraction and visualization of topic stream from a sequence of online news articles.

Although the conventional systems such as Scatter/Gather by Hearst[1] and Grouper by Zamir[2, 3] are useful for users to grasp topic distribution over a document set to some extent, what they concern has been to process a single set of documents, and no sequential nature of document sets is considered. Furthermore, the clustering methods that gradually construct the structure (Fisher[4] and Lagus[5]) do not intend to detect the topic stream.

In this paper, applying *the plastic clustering method* (Takama[6, 7]) to find topic distribution from a sequence of document sets is proposed. The plastic clustering method is one of the WWW information visualization systems that are based on document clustering method, and one of its characteristic features is the generation of *keyword map* as well as document clustering. Furthermore, the keyword's activation value is calculated based on the *immune network model* (Anderson[8] and Jerne[9]), which is also useful as the visualization metaphor to improve the understandability of the keyword map (Takama[7]).

In this paper, the model of *memory cell* is proposed and incorporated into the plastic clustering method, so that it can find the topical relation among different document sets. Experimental result shows that the plastic clustering method with the proposed memory cell can find the topic stream from a sequence of online news-article sets.

The plastic clustering method is described in Section2, and the property of memory cells is introduced in Section 3, followed by the experimental results in Section 4.

2 Immune Network-based Plastic Clustering Method

The plastering clustering method is proposed to generate a keyword map[1] , while performing a document clustering. In the plastic clustering, all the documents in the same cluster have to contain at least one keyword, i.e., a *cluster identifier*. As the cluster identifier is expected to represent the significant topic found on the document set, it is also useful as a *landmark* on the keyword map that helps users to grasp the topic distribution over the document set. There-

[1] A keyword map is a 2-D space on which the keywords extracted from documents are arranged according to their similarities.

fore, the plastic clustering method can be viewed as the extraction method of a set of keywords (i.e., cluster identifiers = landmarks).

The algorithm of the plastic clustering method is as follows:

1. Extraction of keywords from a document set with using the morphological analyzer[2] and the stop-word list.
2. Construction of the keyword network by connecting the extracted keywords k_i to other keywords or documents d_i:
   (a) Connection between keywords k_i and k_j:
   Strong connection (SC): the number of documents D_{ij} containing both keywords is equal to or more than T_k.
   Weak connection (WC): The D_{ij} is more than 0 and is less than T_k.
   (b) Connection between k_i and a document d_j:
   SC: the term frequency TF_{ij} of k_i in d_j is equal to or more than T_d.
   WC: The TF_{ij} is more than 0 and less than T_d[3] .
3. Calculation of keywords' activation values on the constructed network, based on the immune network model (Eq. (1)–(5)).
4. Extraction of the keywords that activate higher than others as cluster identifiers.
5. Generation of document clusters according to the cluster identifiers.

As for the immune network model in Step 3, one of the simplest models proposed in the field of computational biology (Anderson[8]) is adapted.

$$\frac{dX_i}{dt} = s + X_i(f(h_i^b) - k_b), \quad (1)$$

$$h_i^b = \sum_j J_{ij}^b X_j + \sum_j J_{ij}^g A_j, \quad (2)$$

$$\frac{dA_i}{dt} = (r - k_g h_i^g) X_i, \quad (3)$$

$$h_i^g = \sum_j J_{ji}^g X_j, \quad (4)$$

$$f(h) = p \frac{h}{(h+\theta_1)} \frac{\theta_2}{(h+\theta_2)}, \quad (5)$$

here X_i and A_i are the concentration (activation) values of B-Cell i and antigen i, respectively. The s is a source term modeling a constant cell flux from the bone marrow and r is a reproduction rate of the antigen, while k_b and k_g are the decay terms of the antibody and antigen, respectively. The J_{ij}^b and J_{ij}^g indicate the connectivity between the antibodies i and j, and that between antibody i and antigen j, respectively. The influence on antibody i by other connected antibodies and antigens is calculated by the proliferation function (5), which has a log-bell form. The p is the maximum proliferation rate.

Applying a non-monotonic activation mechanism of immune network model enables to satisfy the following contradictory conditions for a cluster identifier.

- A cluster identifier should connect to a certain number of keywords.
- There should not exist any connection between cluster identifiers.

Experiments are performed based on the questionnaires (Takama[7]), and the results show that the quality of clusters generated by the plastic clustering method is comparable with or slightly better than the k-means clustering method.

Furthermore, as the cluster identifier suppresses the related keywords on the constructed keyword network, this relationship among keywords is also useful as the metaphor to improve the understandability of keyword map (Takama[7]).

3 Introduction of Memory Cell for Context Preservation

3.1 Application of Plastic Clustering Method to Document-Set Sequence

The information visualization systems based on document clustering method assume that documents contained in the same set, i.e., the news articles released on the same day or the html documents retrieved with a single query, should have the relation to each other from a certain viewpoint. Furthermore, it is assumed in this paper that the document sets that are given sequentially have a certain relation to each other. Examples of a sequence of document sets are online news articles, which are released day by day, and that of the retrieval results obtained through a series of retrieval processes by a user.

In particular, we aim to find the following topic stream from a sequence of document sets.

[2] As the current system is implemented to handle Japanese documents, Japanese morphological analyzer *Chasen* (http://chasen.aist-nara.ac.jp/) is used to extract nouns.

[3] T_k=3 and T_d=3 are used in this paper.

- The topics that survive through several document sets correspond to the mainstream topics of the user's current retrieval task.

- When the topics, which have missed by the user in the early stage of browsing, appears again after the several document sets, the user can reevaluate such missing topics based on the background knowledge obtained during the browsing processes.

To find mainstream topics and missing topics through a sequence of document sets, the same cluster identifier should be used for the similar topics contained in different document sets. By analogy with the immune system, such preferential keywords can be realized with the property of a*memory cell.*

To incorporate the property of a memory cell into the plastic clustering method, several types of memory cells are proposed in Takama[10]. In this paper, we employ the memory cell that has lower decay term than a normal antibody. The idea behind this model is that the activation region $R^i_{act} \subset R^+$ of an antibody i, which is defined as $R_{act} = \{h^b_i | f(h^b_i) > k_b\}$, can be broadened by decreasing the decay term.

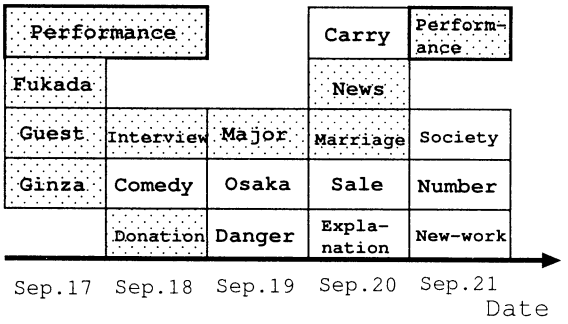
4 Experimental Results

4.1 Topic Streams Extracted from Short Sequence

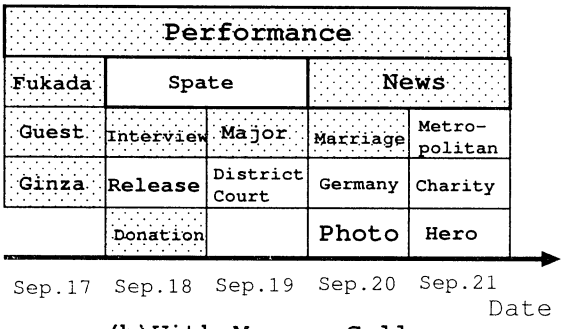
The plastic clustering method equipped with the proposed memory cells is applied to a sequence of news article sets. The news articles contained in the same set are issued on the same day. News articles of entertainment topics that were issued from 17 September 2001 to 21 September 2001 are corrected from Yahoo! Japan News.

In Fig.1, which shows the cluster identifiers extracted from a sequence of news articles, three topic streams, 'Performance', 'Spate', and 'News', can be found with using memory cell, whereas only one topic stream can be found without memory cell.

The topic concerning the tragedy in N.Y. was a mainstream topic of the period when the target news articles were released. The plastic clustering with memory cell can find the related topics ('Performance', 'Spate', etc.) from all document sets, while the clustering without memory cell cannot find such topics from the document set issued on 19 September 2001.



(a) Without Memory Cell



(b) With Memory Cell

Figure 1: Cluster Identifiers Extracted from a News-Article Sequence (Translated from Japanese). The keywords extracted in both experiments are indicated with dotted texture, and topic stream is indicated with a thick boarder

4.2 Topic Stream Extraction from Long Sequence

Table 1 shows the distribution of the landmarks that are extracted from the long sequence, according to the number of occurrence. From this table, it is shown that the ratio of landmarks that are extracted more than once in all the landmarks by using memory cell (39%) is about 2.3 times as much as that without memory cell (17%).

Regarding the landmarks that are extracted more than twice, their streams over the long sequence are shown in Fig. 2. In Fig. 2, a solid line indicates the landmark extracted with using memory cell, while a dotted line indicates those extracted without memory cell. A thin solid line indicates that the same landmark is also extracted without memory cell.

Except the 'Release' extracted from the article set of 5 February 2002, the plastic clustering method with memory cell never miss the landmarks that are extracted without memory cell as well. That is, once a

keyword becomes a memory cell, it is extracted as a landmark whenever the same one is extracted without memory cell. It is also confirmed that this fact is applicable to the landmarks extracted more than once.

Although the long sequence does not contain the specific topic such as N.Y. tragedy of the short sequence in Section 4.1, it can be seen in Fig. 2 that there exist mainstream topics concerning dramas ('Co-star' and 'Director/Manager'), musics ('Singer'), the Release of something new such as CD ('Release'), Open of some events or ceremonies ('Open'), and the marriage of the celebrities ('Marriage'). In particular, the topic indicated by the landmark 'Marriage' can also be viewed as the missing topic, because there exists a gap between its first occurrence and the second one.

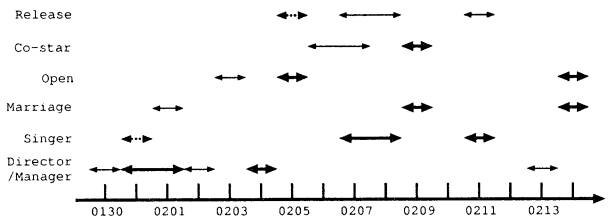


Figure 2: Topic Stream Extracted from Long Sequence

5 Conclusion

A method to find topic distribution from a sequence of document sets is proposed. The plastic clustering method with memory cell is applied to two sequences of online news articles, and the results show a memory cell can play a role to find the topic stream through a sequence of document sets. The proposed method is expected to be a basis for the information visualization systems that handle a sequence of document sets.

Table 1: Distribution of Landmarks Extracted from Long Sequence

# of occurrence	No Memory Cell	With Memory Cell
1	44(83%)	23(61%)
2	7(13%)	9(24%))
3≥	2(4%)	6(16%)
Total	53(100%)	38(100%)

References

[1] M. A. Hearst and J. O. Pedersen, "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results," Proc. 19th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96), pp. 76-84, 1996.

[2] O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," Proc. 8th Int'l WWW Conference, 1999.

[3] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibility Demonstration," Proc. SIGIR '98, pp. 46-54, 1998.

[4] D. H. Fisher, "Knowledge Acquisition Via Incremental Conceptual Clustering," in Machine Learning, 2, Kluwer Academic Publishers, pp. 139-172, 1987.

[5] K. Lagus, T. Honkela, S. Kaski, T. Kohonen, "Self-Organizing Maps of Document Collection: A New Approach to Interactive Exploration," 2nd Int'l Conf. on Knowledge Discovery and Data Mining, pp.238-243, 1996.

[6] Y. Takama and K. Hirota, "Consideration of Presentation Timing Problem for Chance Discovery," 5th World Multiconference on Systems, Cybernetics and Informatics (SCI2001), 8, pp. 429-432, 2001.

[7] Y. Takama and K. Hirota, "Immune Network-based Clustering for WWW Information Gathering/Visualization," Proc. of Japanese Society of Artificial Intelligence, SIG-FAI/KBS-J, pp. 61–66, 2001 (written in Japanese)

[8] R. W. Anderson, A. U. Neumann, A. S. Perelson, "A Cayley Tree Immune Network Model with Antibody Dynamics," Bulletin of Mathematical Biology, 55 (6), pp. 1091-1131, 1993.

[9] N. K. Jerne, "The Immune System," Sci. Am., 229, pp. 52-60, 1973.

[10] Y. Takama, K. Hirota, "Visualization of Topic Distribution from Document Sequence on Web," in P. Sincak et al. Ed., Intelligent Technologies – Theory and Applications – New Trends in Intelligent Technologies, IOS Press, pp. 189–195, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Ubiquitous Haptic Interfaces in Intelligent Space

P. T. Szemes, J. H. Lee, N. Ando, and H. Hashimoto
Institute of Industrial Science
The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, JAPAN 153-8505

Abstract

The aim of the Intelligent Space research is to bring together intelligent systems to provide convenient and useful environment for the human, who exist in this environment. This paper introduces the Human Machine Interface of the Intelligent Space. The Human Machine Interface incorporates the philosophy of Ubiquitous Computing, especially to achieve that kind of interface which the human is able to communicate naturally with the Intelligent Space. The Ubiquitous Human Machine Interface (UHMI) realizes three communication channels to the user: video, audio, and haptic. This paper focuses on the haptic interaction between the User and the UHMI. This paper introduces the policy of the UHMI to handle haptic interaction. Two demonstrations are introduced: one the disturbance rejection properties of the UHMI, and the other is the realization of a simple dynamics.

1 Introduction

The motivation of the Intelligent Space is bring together many intelligent systems, into an environment to use those systems and their services more comfortably and efficiently [1]. The system of the Intelligent Space is difficult and complex. Human Machine Interface (HMI) is necessary to hide this complexity and provides easy and natural communication interface to the user.

We applied the results of the ubiquitous computing in the design process of the HMI. We named our HMI, *Ubiquitous Human Machine Interface* (UHMI), because our motivation in this research is close to the aim of the Ubiquitous Computing: embed the technology in our daily life with the hidden layer of the complexity and an easy-to-use surface interface [2]. The concept of Ubiquitous Computing first appeared at Xerox PARC, from Mark Weisner [3].

The UHMI consists of a mobile robot platform, caring sensors and actuators for Human Machine Interaction (Fig. 1.). This paper focuses on the Haptic Interface of the UHMI.

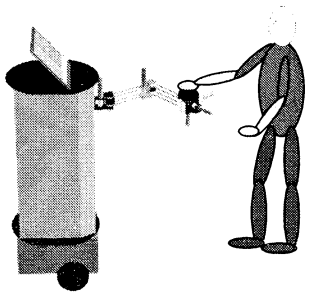


Figure 1: Ubiquitous Human Machine Interface interact with the User.

The aim of the UHMI's Haptic Interface is to realize large amount of data, which is associated with a physical object or environment. Up to nowadays the computer monitor and the visualization were the only efficient way to realize large amount of data, such as CAD or CG models. However, there is other way to realize large amount object data: via sense of touch.

The following observation is applied for the design of the UHMI. The Human hand is well-organized structure. The Human beings uses his/her arm and hand together to realize simple and also difficult movements. However, the arms and hands have different aim and task. The arm has less degree of freedom (DOF), and its main task is to make large volume movements with or without relative big force or caring load. The hand has more degree of freedom, compared to arm, and its main task is to realize small and gentle movement, with smaller forces. The Haptic Interface of the UHMI also applied the above mentioned movement separation principle.

The Haptic Interface can be separated into two subsystems. One subsystem named, Haptic Arm has 2 DOF, and it realizes the planar translation movements. At the end of the Haptic Arm, another subsystem is connected, which realizes the rotational movement along three perpendicular axes of the space. This

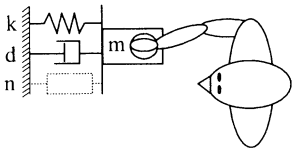


Figure 2: Ubiquitous Human Machine Interface represents absolute Virtual Impedance to the User.

subsystem named Haptic Ball.

This paper is organized as follows. In the Introduction section the basic features of the Haptic Interface of UHMI is introduced. The following section gives a detailed description of the UHMI's policy to represent virtual objects. The actual UHMI has several non-idealities, and parameter uncertainties. This disturbances are compensated by Sliding Mode Based Disturbance Controller. The short summary on this compensation techniques is given in section 3. The effectiveness of this disturbance rejection techniques is shown in the demonstration section. Representation of simple dynamics with Haptic Interface is also shown.

2 Virtual Impedance Approach to Realize Large Volumetrical Data

In this section, application of virtual impedance to realize vector force feedback is introduced. The vector force feedback is acting on the human arm's and hand's muscle. This section is divided into two parts. The first part describes the communication modes of the Haptic Interface with the user. The following part introduces the communication and data exchange between the UHMI and the Intelligent Space. The linear dynamical features of the real objects can be modelled with imaginary spring-mass-damper (SMD) system. The spring can store potential energy, so it is associated with the potential types of forces, such as electrostatic or gravitational forces. The mass can store kinetic energy. The damper represents dissipative parts of the system. The abstraction of the spring-mass-damper system is widely used method in the physic, electronic, material science and so on. In this paper, we introduce one possible application of this abstraction. The linear models are extends to any type of non-linear model.

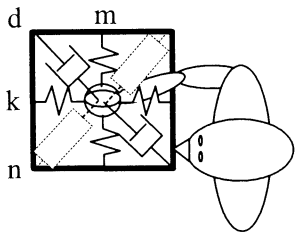


Figure 3: Ubiquitous Human Machine Interface represents relative Virtual Impedance to the User.

2.1 Communication with the User

The Fig. 1 shows the configuration scheme of the UHMI and the user. The user has direct contact with the Haptic Ball. The Haptic Interface represents the current dynamics of the virtual object, which belongs to a specified position and time. In our Intelligent Space configuration, the dynamic properties of the virtual object can be described in two ways. One possible way is that when the Haptic Interface realizes dynamical properties in absolute coordinate system. This configuration is shown on Fig. 2. The imaginary dynamical parameters are modelled in absolute coordinate system. This type of modelling is useful to statically linked virtual object.

The other type of the representation of dynamic properties of the virtual object is shown on Fig. 3. In this configuration, an imaginary frame with mass, m connected via spring, k, damper, d, and other (nonlinear) elements, n to the user's hand. This type of representation is useful for smooth operation. The primary operation of this mode is for the telemanipulation, where smooth movements are required.

2.2 Communication with the Intelligent Space

The Intelligent Space stores the dynamical features of the virtual object. The dynamical features can be distributed temporal and spatial. The aim of the UHMI is to realize those dynamical features according to its position in the Intelligent Space and the relative position compared to the human. For example, if the UHMI should represent a virtual wall toward to the user,then the virtual wall is represented as a dramatically increasing of stiffness of the imaginary spring, when the human's hand reaches a certain position. The block diagram of the representation mechanism is seen on Fig. 4. The Intelligent Space provides two services to the UHMI. One is the database of the virtual ob-

ject. This database contains the dynamical properties of the virtual object. These properties are imaginary mass, stiffness, and damping ratio, in linear case. The database extends with non-linear properties such as friction term or backlash properties. The other service is the location service [4] and [5]. This service provides the position of the human and the robot.

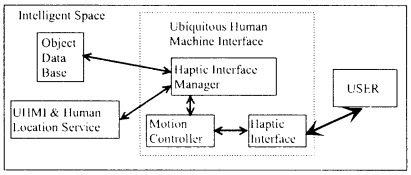


Figure 4: Ubiquitous Human Machine Interface interact with the User.

3 Virtual Impedance Controller with Sliding Mode Based Disturbance Observer

In this section, we briefly introduce how the dynamical properties of the virtual object are realized. Sliding Mode Based Disturbance Observer (SMBDO) is applied to reject the disturbance generated inside the actuators. The disturbances can be parameter uncertainties and internal or external forces. In this paper, we show that features of the SMBDO, which is important in our case. The design of SMBDO is found in [6]. The connection diagram is shown on Fig. 5. The Sliding Mode Control is a non-linear control scheme. In the design stage of the sliding mode control (SMC), an ideal trajectory is defined, which satisfies the control requirement. This trajectory called sliding surface. The aim of the sliding mode controller is to force the system toward the sliding surface and keep the system on it. When the system is on the sliding surface, it can be modelled as a reduced order system [6].

The Actuator, where the disturbance is generated, is parallel connected with a Reference Model (Fig. 5.). The Reference Model is the idealistic case of the system, without parameter uncertainties and external disturbances. The output of the real and ideal plant is compared. The output of the comparison is the input of the Sliding Mode Controller.

The disadvantage of the sliding mode control is the chattering phenomenon. The chattering is a high frequency, small amplitude vibration of the system trajectory abound the sliding surface. This vibration is

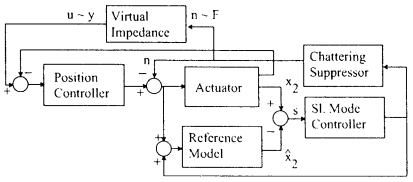


Figure 5: Sliding Mode Based Disturbance Observer with Virtual Impedance.

caused by the non-idealities of the controller. In simple case, the sliding mode controller contains a relay type of non-linearity. This type of non-linearity contains high frequency components, which may excite the non-modelled dynamics of the real system. The Chattering Suppressor filters the output of the Sliding Mode Controller, to avoid the excitation of the non-modelled dynamics in the real plant. However, this filter results a so-called pseudo-sliding mode for the real plant. The Chattering Suppressor gives a continuous approximation of the non-linear, non-continuous control signal, which cause performance looses. There are two loops on Fig. 5.: one is the SMC and the Reference Model, where the ideal sliding mode occurs. The other is, the Actuator, SMC, and Chattering Suppressor loop, where the pseudo-sliding mode is occur.

4 Demonstration

4.1 Disturbance Rejection Properties of the Sliding Mode Disturbance Observer

The Motion Controller (Fig. 4.) with SMBDO is robust against parameter variations and internal disturbances. One of the most significant internal disturbances is the friction inside the mechanical parts and the actuators. If the friction is not compensated, the Haptic Interface cannot follow the position command precisely, what results false representation of the imaginary object. This experiment demonstrates the robustness of the Sliding Mode Based Disturbance Observer. Fig. 6 shows the result of the experiment. In the experiment, first a position step command (0.35 radian) is given to the controller. The system follows the position command with steady state error. A disturbance torque is added to the system at t=2.5 sec, as a model of the user's force. The amplitude of the disturbance torque is 50 % of the system's nominal torque. The system responses with small position error, but some high frequency components, with small

amplitude can be observed. This is the so-called chattering. This is caused by the non-idealities, such as time delay of Sliding Mode Based Disturbance Observer.

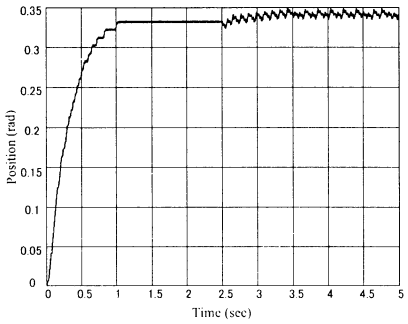


Figure 6: Disturbance Rejection Experiment of the Position Control System Equipped with Sliding Mode Based Disturbance Observer.

4.2 Representation of Dynamics Properties of Virtual Object

In this experiment, an imaginary spring-mass-damper (Virtual Impedance box on Fig. 5.) is connected to the SMBDO (Fig.5.), as introduced in section 2. In this experiment, the reference position, y for the Motion Controller is a position answer to a step-like force, input F. The Motion Controller controls the haptic interface to represent this waveform to the user. The waveform is shown on Fig. 7. First, the Haptic Interface reaches its steady position (0.35 radian) with steady state error. Then, at t=1.5 sec, a force input, F which is proportional with the disturbance signal, ν is applied to the imaginary spring-mass-damper system. The Haptic Interface follows the output of the Virtual Impedance (Fig. ??), with small error. The compensation signal, ν is added to the motor input voltage, u but this voltage is limited.

5 Conclusion

In this paper, Ubiquitous Human Machine Interface (UHMI) is introduced. The present state of the research is introduced, according to our schedule. The idea of the realization of Large Volume Virtual Object is introduced, and one DOF experimental result is demonstrated. In this paper, the robustness of the Sliding Mode Based Disturbance Observer (SMBDO) and a simple case of dynamical representation are

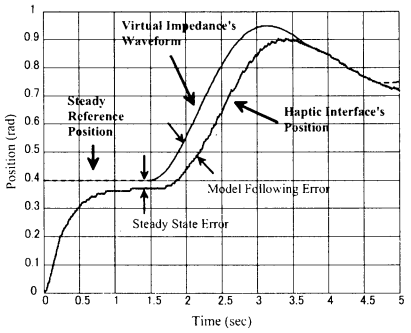


Figure 7: Virtual Impedance Representation with Haptic Interface.

demonstrated. The SMBDO also applied to measure the user's force. The next step of our schedule is the extension of the idea into all DOF of the Haptic Interface.

References

[1] J.-H. Lee and H. Hashimoto, "Intelligent Space - Its concept and contents -," *Advanced Robotics Journal*, Vol. 16, No. 4, 2002.

[2] B. Brumitt, J. Krumm, B. Meyers and S. Shafer, "Ubiquitous Computing and the role of geometry", *IEEE Personal Communications Magazine*, October, 2000

[3] M. Weisner, "Some Computer Science Issues in Ubiquitous Computing", *Communications of the ACM*, July 1993 36:7 75-84

[4] T. Akiyama, J.-H. Lee, and H. Hashimoto, "Evalua-tion of CCD Camera Arrangement for Positioning System in Intelligent Space",*International Symposium on Artificial Life and Robotics*, 2001.

[5] K. Morioka, J.-H. Lee, H. Hashimoto, " Mobile Robot Control for Human Following in Intelligent Space", *International Conference on Control*, Automation and Systems, pp.55-58, 2001.10.

[6] P. Korondi, I. Nagy, S. Kerekes, K. Zaban, T. Gajdar "Sliding Mode Based Disturbance Observer for Motion Control", *ELECTROMOTION* Vol. 5 (1998), No. 3. 119-124.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Haptic Expression of Figures Using A Surface Acoustic Wave Tactile Display Mouse

Masaya Takasaki*, Takeshi Mizuno
Department of Mechanical Engineering,
Saitama University
255, Shimo-Okubo,
Saitama City, 338-8570, Japan
e-mail: masaya@mech.saitama-u.ac.jp

Takaaki Nara
Mathematical Informatics Research,
Foundations of Informatics Research Division,
The National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, 101-8430, Japan

* PRESTO, Japan Science and Technology Corporation (JST), Japan

Abstract

A SAW tactile display mouse to indicate roughness sensation has been developed. The display was controlled so that an alternative shear force according to operator's action could be provided to his/her finger skin. Using the mouse, they could enjoy roughness sensation when they rub on a solid surface with roughness. In this research, demonstration to express a 2-dimensional figure through operator's mouse dragging action was carried out. Displaying rough area for black part and smooth area for white part, a draft could be expressed through tactile sensation. We could feel as if a rough surface would be shaped into the figure. But we could not distinguish a circle from a square or a triangle. In the future, objective evaluation of this demonstration will be carried out statistically.

Keywords: tactile display, surface acoustic wave, computer interface, virtual reality, haptic device

1 Introduction

Reproduction of human haptic sensation has lately attracted attention for various fields, such as virtual reality, remote control of robots, computer interfaces and so on. Physiologically, haptic sensation is divided into two parts. One is proprioception, which is sensation of weight, resistance, or the approximate shape of an object. The sensation is perceived in muscles. The other is tactile sensation, which is a sense of roughness, friction, or the otherwise variegated texture of an object's surface. The sensation is perceived at mechanoreceptors in our skins.

Proprioception displays have been reported by a lot of groups and a few displays of them have been applied to actual device productions[1]. For tactile displays, some methods using various actuators[2][3][4][5] have been proposed. However, the actuators need some volume to build in. So, surface acoustic wave (SAW) was focused for the actuator of the tactile display[6]. Properties of SAW are high operating frequency of more than a few MHz, thin structure, simple fabrication, easy installation of a transducer, high energy density and so on. A thin tactile display with high performance can be developed using SAW properties.

Methods to display tactile sensation of roughness by using SAW were reported[6][7][8]. In these reports, two types of SAW tactile displays, "Passive Type"[6] and "Active Type"[7][8], were proposed. Previously, a tactile display on a PC mouse button based on "Passive Type" has been developed[9]. Using the computer interface and an experimental setup, we could feel the sensation as we rub on solid surface with roughness by our fingers, and then could determine the grade of roughness.

In this research, our purpose is to express some figures using the computer interface with SAW tactile display. A user drags on computer screen through the tactile display mouse, and then feels tactile sensation as if the screen has some rough area. If the tactile display is controlled according to shape of the rough area, he/she can recognize the figure through tactile sensation. We developed an experimental setup and could distinguish a circle and a cross.

2 SAW Tactile Display Mouse

2.1 Surface Acoustic Wave

Figure 1 indicates the excitation of Rayleigh wave, a kind of SAW. An interdigital transducer (IDT) is arranged on a piezoelectric substrate. The IDT consists of a metal strip array. When AC driving voltage is applied to the IDT, Rayleigh wave is excited and propagates on the substrate surface in the direction indicated by the arrows in the figure. The frequency of the driving voltage is decided according to the size of the IDT, particularly pitch of the IDT electrodes, namely wavelength. In the case of $LiNbO_3$ 128° Y-cut substrate for the piezoelectric material, the wavelength is 400μm at the operating frequency of 10MHz. The substrate is also used as an elastic media on which Rayleigh wave propagates.

In the progressive Rayleigh wave, the surface particles go along elliptical locus as shown in Fig. 2. The

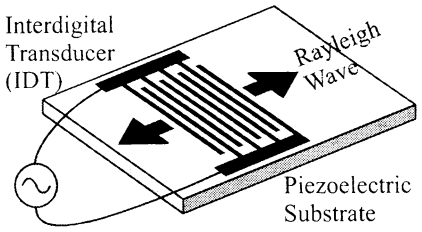


Fig. 1 Excitation of Rayleigh wave.

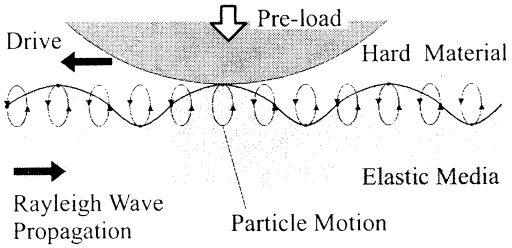


Fig. 2 Particles motion on the surface in the progressive Rayleigh wave and a steel ball driven by the wave.

vibration is distributed only in the media surface. More than 99% of vibration energy flows within 2λ depth from the surface. Therefore the piezoelectric substrate is easily fixed by cement and so on. If a hard material with a smooth surface like a steel ball is put on the wave as shown in Fig. 2, the material contacts the SAW media on the upper locus due to the very fast elliptical vibration of the surface. Therefore, vibration velocity of the top of the locus drives the material in the direction indicated in the figure. In this manner, a shear force is generated on the surface of SAW media. The shear force was applied for our novel tactile display. This phenomenon has been already applied for an ultrasonic motor[10][11].

2.2 Tactile Display Principle

Figure 3 describes a basic structure of a Passive Type SAW tactile display. It consists of a piezoelectric substrate with two IDTs and a pad. Display users put on their fingers on the pad and feel shear forces generated according to the ultrasonic motor principle as mentioned above. With a control, the forces are felt as tactile sensation. The force is proportional to both the vibrating velocity of the Rayleigh wave and the pre-load applying to the pad. The vibration velocity is proportional to the driving voltage applied to the IDT. Controlling the driving voltage amplitude and wave propagating direction, strength and direction of the shear force can be arranged. Therefore, temporal distribution of alternative shear force can be generated by switching the driving voltage (on / off control) as shown in Fig. 3. The force distribution spreads as vibration and innervates mechanoreceptors in the finger skin.

Changing the switching frequency, roughness to be

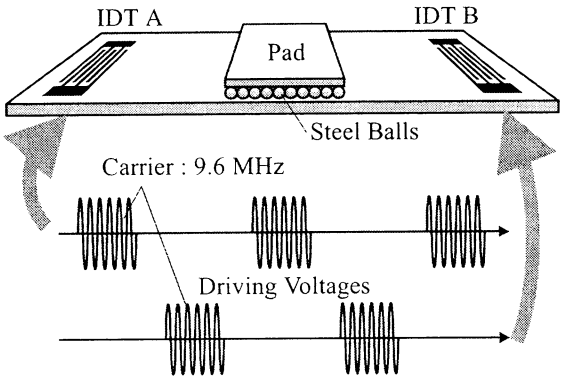


Fig. 3 Excitation of two phases of modulated Rayleigh wave.

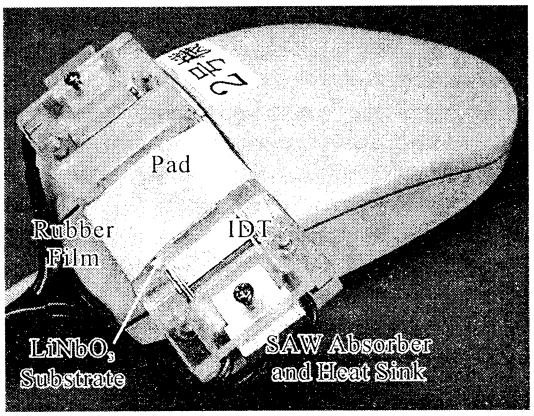


Fig. 4 A view of the SAW tactile display mouse.

rubbed virtually by the operator can be arranged, because the perceived roughness information depends on frequency of the vibration. Additionally, control of the frequency depending on the roughness information and virtual rubbing speed enhances reality of the tactile display. Therefore, both roughness information and operator's rubbing speed are very important for the control of the SAW tactile display.

2.3 SAW Tactile Display

Figure 4 shows a SAW tactile display mouse. It can be seen that a tactile display module is on the left button of the computer mouse. The tactile display module consists of a acrylic resin base, piezoelectric substrate, two SAW absorbers, two heat sinks, two covers and a pad. A $LiNbO_3$ 128°Y-cut X-prop substrate was used as a piezoelectric substrate and an elastic SAW media. The size of the substrate was 16 x 60 x 1 mm^3. To avoid SAW reflection at the edges, SAW absorbers were installed at the edges of substrate, otherwise a standing wave would be excited and shear force principle would not be available. The absorbed mechanical vibration was transformed into heat. Then the heat conducted to heat sinks coupled to the absorbers and was radiated. The

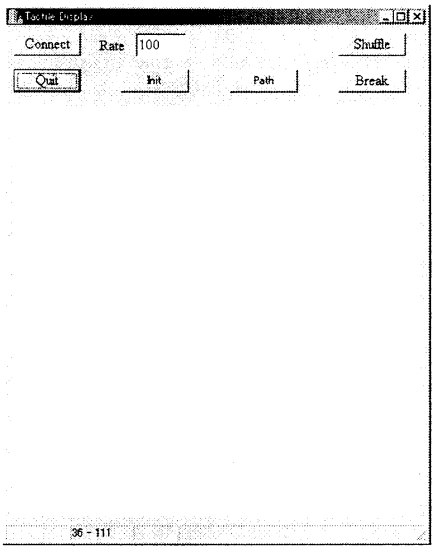


Fig. 5 A view of computer screen.

Fig. 6 A mouse pointer.

IDTs were built by aluminum vacuum evaporation and wet etching. These processes are almost same as semiconductors production processes. The pitch of the IDT was 200 μm and the strip width was 100 μm. Effective width of IDT was 12 mm. Thickness of the electrodes was 0.3 μm. The operating frequency, resonant frequency of the IDT, was 9.6 MHz. The dimension of the pad was 15 mm x 11 mm. The size was almost same as a tip of forefinger to click the mouse button. Steel balls of diameter 1.5 mm were distributed on the back of the pad. The steel balls were cemented to the pad of rubber film. The pad was suspended by a rubber film. An acrylic resin base supported both the piezoelectric substrate and the rubber film. The IDTs and heat sinks were covered by the acrylic resin covers.

Previously, demonstration to indicate roughness sensation as we rub on solid surface was carried out[9]. When we dragged on a computer screen using the tactile display mouse, we could enjoy tactile sensation and feel as if we would rub something with a rough surface drawn on the screen directly. Additionally, difference of roughness could be perceived.

3 Haptic Expression of Figures

3.1 Experimental Setup

To demonstrate haptic expression, an application software was prepared. Figure 5 shows a view of the

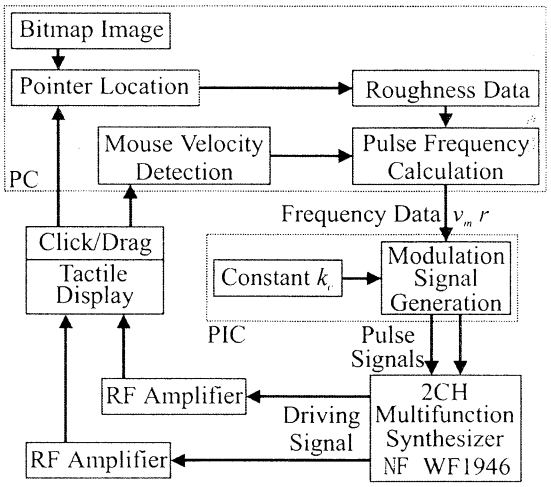


Fig. 7 A tactile display control system for demonstration.

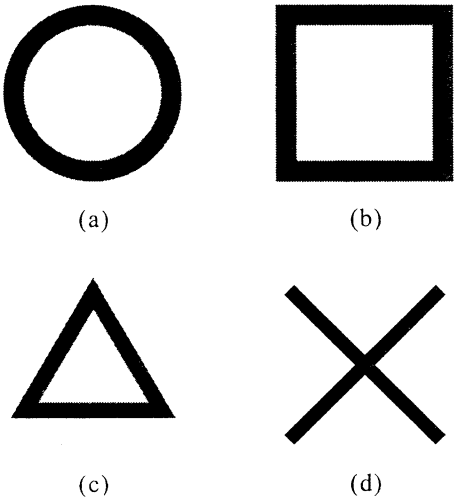


Fig. 8 Images expressed by the demonstration.

computer screen of the software. A white area was for the expression. Before the expression, a bitmap image to present a figure was assigned to the area. Each pixel of the image was black or white. The black part meant rough surface and white part meant smooth surface. But an operator could not see the image. When he/she dragged on the white area using the SAW tactile display mouse, tactile display indicated a texture (rough/smooth) under a mouse pointer shown in Fig. 6. The black area of the assigned image could be recognized as rough area. (The white area could be recognized as smooth area.) As a result, black part of the image could be recognized through tactile sensation.

To demonstrate the haptic expression, the experimental set up was controlled by a system described in Fig. 7. In a computer, one bitmap image in

Fig. 8 was selected to present the figure expressed on the white area in Fig. 5. Operator's mouse dragging speed and mouse pointer location were detected. According to the location, pixel color in the bitmap image was determined. Then 0 for white was substituted into roughness data r and r_1 for black was substituted into r (For r_1, smaller number meant rougher surface and 0 meant nothing, namely very smooth surface.) The detected mouse dragging speed v_m was multiplied by r and then outputted through 8-bit I/O port as an integer. The integer was input to a processor (Microchip PIC 16F84A). The processor was programmed to generate two pulse signals. The frequency of the signals was proportional to the inputted integer. The frequency f decision was expressed as follows.

$$f = k_c v_m r \tag{1}$$

where k_c is a constant based on resolution and the size of the screen. Duty ratio of the pulse was 12.5%. The phase difference of the pulse signals was 180 deg. The signal modulated two channels of carrier RF voltages to excite Rayleigh waves. The modulated driving voltages were amplified and applied to the IDTs. Momentary input electric power was about 100 W, averaged power was about 13 W. While demonstration, the frequency f was less than 300 Hz.

3.2 Expression

In actual use, dragging with the mouse, we could feel tactile sensation of roughness through the tactile display mouse and recognize whether the surface was smooth or rough. Dragging whole white area, the figure presented by the bitmap image could be perceived. We could discern a cross drawn in Fig. 8 (d) from other figures. But we could not distinguish one figure among Fig. 8 (a)-(c). All the three figures were felt as a circle. These results were authors subjective results. So we will discuss objective evaluation of human ability of tactile perception through statistical experiments with a large number of operators.

Using the tactile display mouse, information around a mouse pointer on the computer screen can be provided to the operator through tactile sensation. Therefore, the device can help interaction between operators and computers. It seems that more information can be presented than conventional computers using audio and visual information. Additionally, using the tactile display mouse only with assistance of sound information, computers can be operated without PC screens. There is possibility for blind people to enjoy the graphical user interface of recent computer applications.

4 Summary

A SAW tactile display mouse could indicate roughness sensation according to operator's dragging action. Demonstration to express a 2-dimensional figure through the action was carried out. We could feel as if a rough surface would be shaped into a figure. But we could not distinguish a circle from a square or a triangle. An objective evaluation of this demonstration will be carried out statistically.

References

[1] For example,
"PHANToM", http://www.sensable.com/
"HapticMASTER", http://www.fcs-cs.com/
"Cyber Force", http://www.immersion.com/

[2] R. D. Howe, W. J. Peine, D. A. Kontarinis , J. S. Son (1995), Remote Palpation Technology, IEEE Eng. In Medicine and Biology, Vol.14 No.3, pp. 318-323.

[3] Y. Ikei, K. Wakamatsu, S. Fukuda (1996), Texture Presentation by Vibratory Tactile Display, Proc. IEEE Virtual Reality Ann. Int'l Symp. (VRAIS 96), IEEE CS Press, Los Alamitos, CA, pp. 199-205.

[4] N. Asamura, N. Yokoyama, H. Shinoda (1998), Selectively Stimulating Skin Receptors for Tactile Display, IEEE Computer Graphics and Applications, Vol. 18, No. 6, pp.32-37.

[5] T. Watanabe, S. Fukui (1995), A Method for Controlling Tactile Sensation of Surface Roughness Using Ultrasonic Vibration, Proc. IEEE Int'l Conf. Advanced Robotics (ICAR), IEEE Robotics and Automation Soc., Piscataway, N.J., pp. 1134-1139.

[6] M. Takasaki, T. Nara, S. Tachi et al (2000), A Tactile Display Using Surface Acoustic Wave, IEEE International Workshop on Robot and Human Interactive Communication, Osaka, Japan, pp. 364-367.

[7] T. Nara, M. Takasaki, S. Tachi et al (2000), An Application Of SAW To A Tactile Display In Virtual Reality, IEEE Ultrasonics Symposium Proceedings, San Juan, Puerto Rico, pp.1-4.

[8] M. Takasaki, T. Nara, S. Tachi et al (2001), A Tactile Display Using Surface Acoustic Wave, IEEE International Workshop on Micro Electro Mechanical Systems, Switzerland, pp. 240-243.

[9] M. Takasaki, T. Nara, T. Mizuno (2002), A Computer Interface Using Surface Acoustic Wave Tactile Display, SICE Annual Conference 2002, Osaka, Japan, pp. 3132-3136.

[10] M. Takahashi, M. k. Kurosawa, T. Higuchi (1995), Direct Frictional Driven Surface Acoustic Wave Motor, Proc. Transducers '95 and Eurosensors IX, Stockholm, Sweden, pp. 401-404.

[11] N. Osakabe, M. K. Kurosawa, T. Higuchi et al (1998), Surface acoustic wave linear motor using silicon slider, IEEE International Workshop on Micro Electro Mechanical Systems, Heidelberg, Germany, pp. 390-39.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Emergence of un-designed behaviors of redundant systems

Kazuyuki Ito, Akio Gofuku, Mitsuo Takeshita
Okayama University, 3-1-1, Tsushimanaka, Okayama-city, Okayama, Japan
{kazuyuki, fukuchan, takesita}@sys.okayama-u.ac.jp

Abstract

Recently, flexible autonomous system that can accomplish various tasks automatically has been much attention. However most of conventional researches of autonomous system are restricted to some simple robots. In this paper, we consider an adaptive control method for redundant system, and by summarizing our previous works of QDSEGA we demonstrate that emergent behaviors that are suitable for each task and each robot are acquired using the proposed algorithm.

1 Introduction

Recently, flexible autonomous system that can accomplish various tasks automatically has been much attention. However most of conventional researches of autonomous system are restricted to some simple robots like a mobile robot [1, 2], so it cannot accomplish various tasks. On the other hand, in the field of researches of redundant robot[3, 4], the robot can accomplish various tasks using its redundancy but it requires control methods for each tasks. So it cannot be autonomous system. Considering these points, we had proposed "Q-learning with dynamic structuring exploration space based on GA (QDSEGA) [5, 6]" which is hybrid adaptive control method for redundant systems. In QDSEGA, Q-learning is applied to a small subset of exploration space to acquire some knowledge of a task, and then the subset of exploration space is restructured using acquired knowledge. So without priori knowledge, efficient search, compare to trial and error, is possible. By applying QDSEGA to redundant systems, effective movements for each task are selected from the various movements that can be realized by the redundancy of the system. In this paper, we summarize our previous works [5, 6, 7] and by comparing the results we demonstrate that emergent behaviors that are suitable for each task and each robot are acquired using the proposed algorithm.

2 QDSEGA

To realize the adaptive autonomous controller, we had proposed a hybrid method by connecting a rule-based distributed control with a centralized control based on reinforcement learning.

Fig. 1 shows the outline of our proposed learning architecture. Rule-based distributed control is employed to extract closed-subset of exploration space. Reinforcement learning is applied to the subset and some knowledge of task is obtained. To restructure

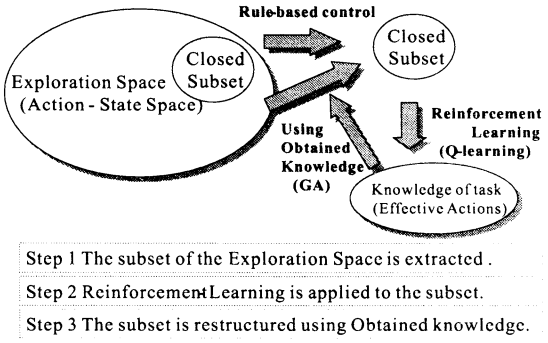


Figure 1: Outline of Learning Architecture

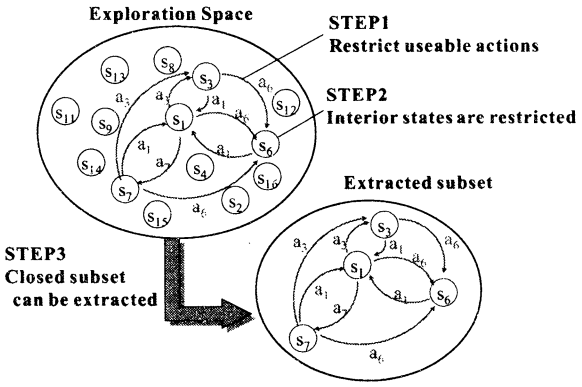


Figure 2: Extraction of Closed Subset

the subset, the acquired knowledge is utilized.

To extract the closed-subset (Fig. 2), QDSEGA has a 2-class layered structure as shown in. Fig. 3. The upper agent plans all trajectories of desired states of lower agents using reinforcement learning, and passes them to lower agents.

By extracting the closed-subset, it becomes possible to applying the reinforcement learning to the small extracted exploration space. And using acquired knowledge to restructure the subset, the search becomes more efficient. It means that the proposed method is applicable to large redundant systems. Using reinforcement learning, role of each agent for each task is assigned automatically in the changing environment. It means that the proposed method have autonomy, flexibility and adaptability. Details are written in reference[5, 6, 7].

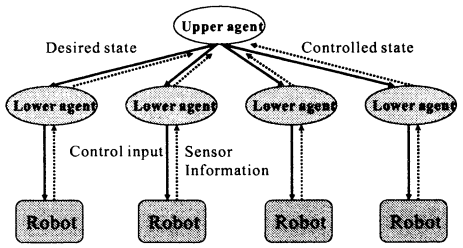


Figure 3: Layered Structure of Learning architecture

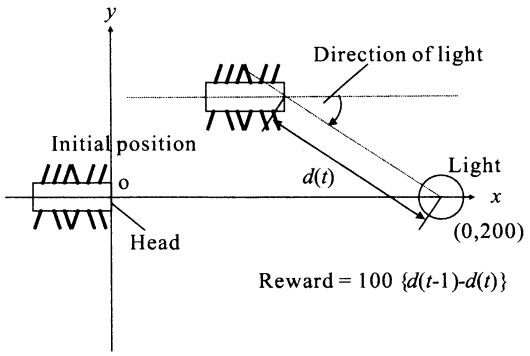


Figure 4: Locomotion Task

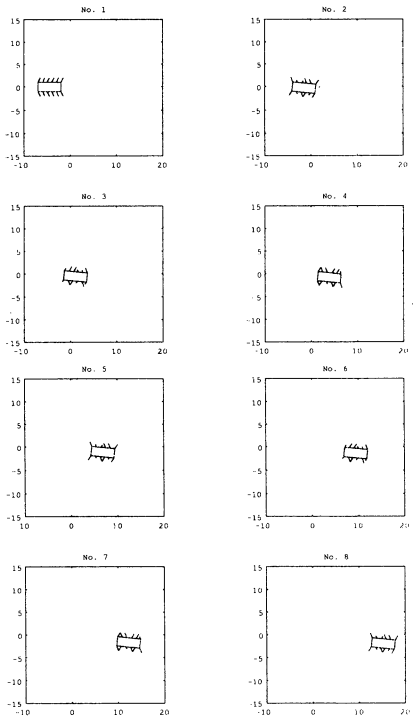


Figure 5: Acquired locomotion of the multi-legged robot

3 Application to Redundant Systems

In this section we apply QDSEGA to redundant systems.

3.1 Locomotion of multi-legged robot

3.1.1 Task

The task is how to get closer to the light source. Fig. 7 shows the outline of the task. The light source is far enough from the start position of the robot and the reward is calculated using the distance between the current position of the robot and the light source. We consider 12-legged robot and employ Minimal Simulation Model that was proposed by M. Svinin et al., [8].

3.1.2 Simulation Results

Fig. 5 shows the acquired locomotion at 200th generation. We can find that locomotion can be realized.

3.2 Locomotion of Snake-like robot

3.2.1 Task

We consider a task which is the same as the subsection 3.1.1, using snake-like robot.

In this simulation we employ a dynamic model of the snake like robot with considering friction between robot body and environment proposed by Iwasaki et al [9]. All links touch the ground and the friction of the vertical direction with respect to the robot body is larger than that of the tangential direction. Owing to this difference of friction the snake-like robot can move. The number of links is 5.

3.2.2 Simulation Result

Fig. 6 shows the acquired behavior. We can find that the winding motion is acquired and the task is accomplished. It means that proposed algorithm is effective for not only the task in the static world but also the task in the dynamic world.

3.3 Obstacle Avoidance by Manipulator

3.3.1 Task

In this subsection, we apply the proposed method to the problem of obstacle avoidance using a 50-link manipulator. Let us define the origin and coordinate as shown in Fig. 7. T he origin means the fixed end of manipulator and the first joint angle is the angle from the x-axis. The goal of the task is taking the top of the manipulator to the interior of the desired circle with avoiding the obstacle. The length of manipulator is 1.5 and the initial attitude is the straight line on the x-axis. The obstacle has a circle shape whose radius is 0.1 and its center moves on the circle trajectory whose center is (0.5, 0.5) and radius is 0.3 with constant speed. And the initial position on the circle is random.

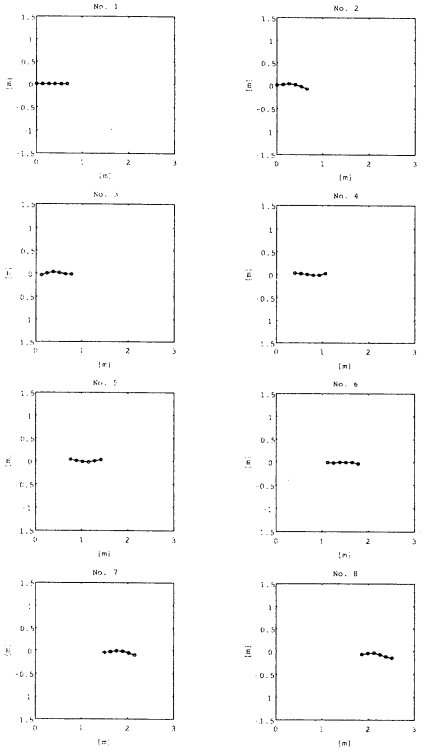


Figure 6: Acquired locomotion of the snake-like robot

The goal region is the 0.1 radius circle and the center is (0.5,0.5) and does not move. In this simulation we employ kinematic model.

3.3.2 Acquired behavior

Fig. 8 shows an acquired behavior. We can find that the effective behavior is acquired and the task is accomplished. It means that proposed algorithm is effective for large redundant systems.

3.4 Cooperative Transportation

3.4.1 Task

We consider cooperative transportation task as a typical example of multi-mobile robots' task.

Fig. 9 shows a grid world of the cooperative transportation task. The world consists of 6 × 5 grids. There are 2 loads and 10 mobile robots. G of Fig. 9 means the goal position, and in the front of goal, a door is equipped. SW means a switch to open the door. An aim of this task is to transport Load 1 to goal position G. The loads have each movable direction. Load 1 can move only vertical direction, and Load 2 can move only horizontal direction. The loads are big enough, so only one load can enter into one grid, and the robot cannot enter the grid that is occupied by the load. However the robots are small enough so all robots can share the same grid. To move the load, more than 2 robots should push it to same movable direction. Therefore, to realize the task, robots

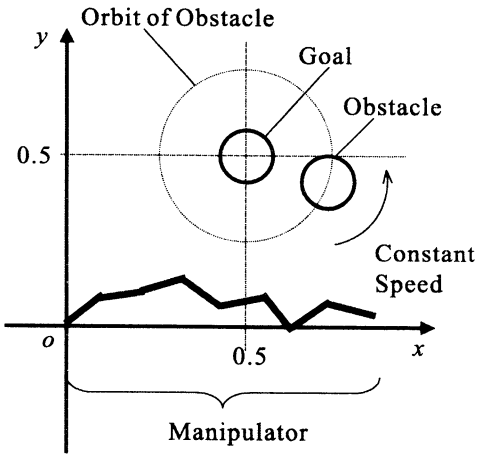


Figure 7: Obstacle avoidance using redundant manipulator

should move an obstacle (Load 2), open the door and move Load 1 to the goal in cooperation.

3.4.2 Simulation Results

Fig. 10 show a typical result. We can find that the task is completed. In this movements, the desired positions that are planed by upper agent are only 2 (i=8 and i=17 of Fig. 10), and other positions are generated by lower agent's distributed controller. We can conclude that the distributed controller reduced planning costs of upper agent, and by the planning of the upper agent, dead lock is avoided and the task is completed in cooperation.

4 Discussion

At first we consider the adaptability for the differences of the body.

From the applications of locomotion task of the multi-legged robot and the snake-like robot, we can find that difference behaviors that suitable for each body has been acquired. To acquire the behaviors, we have employed the same upper agent, which has learning algorithm, and we have not employed any knowledge for the task and robot. So we can conclude that the acquired behaviors have emerged.

Next we consider the adaptability for the differences of the task.

From the application of locomotion of the snake-like robot and the obstacle avoidance of a manipulator, we can find that using the similar bodies, different behaviors for each task have been acquired. We can also conclude that suitable behaviors for each task have emerged.

Finally, we consider the applicability of the algorithm.

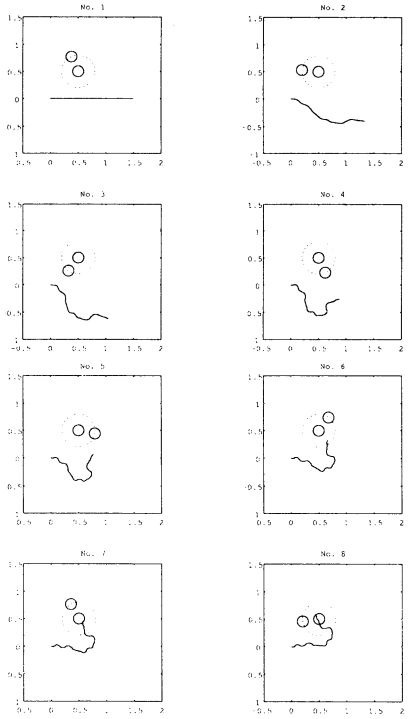


Figure 8: Acquired behavior (Task of obstacle avoidance by 50-link manipulator)

From the cooperative transportation task, we can find that the proposed algorithm is applicable not only for redundant robots but also multi-mobile robots. We can conclude that the proposed algorithm is applicable for the redundant systems that consist of multiple robots.

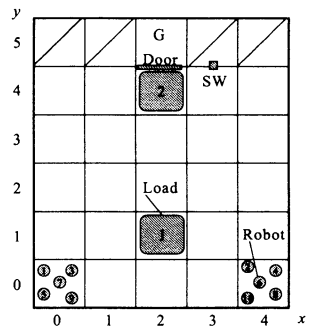


Figure 9: Cooperative Transportation Task

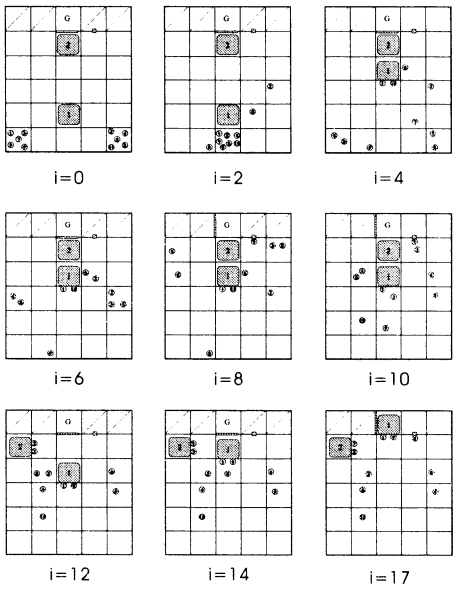


Figure 10: Acquired behavior (Cooperative Transportation Task)

5 Conclusion

In this paper, we have summarize our previous works of the QDSEGA and by comparing the simulation results we have demonstrated that emergent behaviors that are suitable for each task and each robot have been acquired using the proposed algorithm.

References

[1] T. Minato and M. Asada. Image feature generation by visio-motor map learning towards selective attention. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1422–1427, 2001.

[2] K. Yamada, K. Ohkura, M. Svinin, and K. Ueda. Adaptive segmentation of the state space based on bayesian discrimination in reinforcement learning. In *Proc. of the 6th Int. Symp. on Artificial life and Robotics*, pages 168–171, 2001.

[3] M. Yim, D. G. Duff, and K. Roufas. Modular reconfigurable robots, an approach to urban search and rescue. In *Proc. of 1st Int. Workshop on Human-friendly Welfare Robotics Systems*, pages 69–76, 2000.

[4] S. Murata, E. Yoshida, K. Tomita, H. Kurokawa, A. Kamimura, and S. Kokaji. Hardware design of modular robotic system. In *Proc. of 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages F–AIII–3–5 (CD-ROM), 2000.

[5] K. Ito and F. Matsuno. A study of Q-learning: Dynamic structuring of exploration space based on genetic algorithm. *Transactions of the Japanese Society for Artificial Intelligence*, 16(6):510–520(in Japanese), 2001.

[6] K. Ito and F. Matsuno. A study of reinforcement learning for the robot with many degrees of freedom -acquisition of locomotion patterns for multi legged robot-. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 3392–3397, 2002.

[7] K. Ito, A. Gofuku, and M. Takeshita. A study of reinforcement learning for redundant systems -extend qdsega for multi-agent system-. In *Proc. of the 6th Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pages 25–32, 2002.

[8] M. Svinin, S. Ushio, K. Yamada, and K. Ueda. Emergent systems of motion patterns for locomotion robots. In *Proc. of Int. Workshop on Emergent Synthesis*, pages 119–126, 1999.

[9] M. Saito, M. Fukaya, and T. Iwasaki. Serpentine locomotion with robotic snakes. *IEEE Control Syatems Magazine*, 22(1):64–81, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Floorplan Design Problem Using Improved Genetic Algorithm

Kenichi IDA Yosuke KIMURA
Department of Information Engineering
Maebashi Institute of Technology
Maebashi, 371-0816, Japan

Abstract

Genetic Algorithm (GA) attracts much attention because of its applicability to various kinds of optimization problems. It has been applied to many kinds of difficult combination problems. It is known that GA can find the global solution rapidly if the population holds both diversity and similarity sufficiently. However, it is difficult to satisfy both requirements at the same time, because they often trade-off each other. In this paper, from a point of practical view, we propose a new GA for the floorplan design problem (FDP), which aimed at improving the speed of calculation, the maintenance of the solution's population diversity and simplicity of parameters. We applied it to a MCNC benchmark. The experimental results showed that the proposed method performed better than the existing methods.

Key words: floorplan design problem (FDP), genetic algorithm (GA)

1 introduction

The floorplan design problem is an important problem, which is seen in the layout design of VLSI (very large scale integration), the machine layout in a factory, the room arrangement in a residence, etc. Although many algorithm for the automatic design have proposed, a lot of time is needed to perform a floorplan design, due to complications with the restricted conditions and whether the optimization problem had a strong demand. Thus, full automation would be difficult, and in many cases, very time consuming.
Generally speaking, most floorplan design problems are too complex for calculating the strict optimal solution. However, the methods of using a GA etc. to get an approximate solution to these problems are attracting attention [1]. This paper considers the application of a GA to a floorplan design problem.
On a practical floorplan design problem, in many cases we have to take complicated demand and restricted conditions into consideration. Moreover, two or more targets may have to be clearly treated as a multiple-purpose problem. For example, on the layout design problem of VLSI, one target is to minimize the layout area, although simultaneously the wires must shorten in length. Depending on the case, restrictions on wiring length may be given in order to protect timing restrictions, or restrictions on arrangement of elements may be given in order to maintain the circuit character. A GA that performs a multi-point search using various solutions, is effective in a problem with such restrictions and a multiple-purpose problem. It can be used as an applicable method for a floorplan design problem.
However, GA has a risk of falling into premature convergence. Therefore, many improved methods were proposed. Someya et al [2] adjusted the search area in adaptation to an unknown searching stage, and achieved a high performance. On the other hand, Shigehiro et al [3] used a "sequence-pair" for genotype coding in order to represent any possible floorplan and was able to search efficiently.
In this paper, after basing it on such efficiency, an algorithm aimed at "the convergence speed of the solution candidate", "improvement in the accuracy of a solution by maintenance of diversity" and "simplification of a parameter setup" is proposed.

2 floorplan design problem

In this paper, a set $r_i(1 \leq i \leq n)$ of rectangular blocks lie parallel to the coordinate axes. Each rectangular block r_i is defined by a tuple (h_i, w_i), where h_i and w_i are the height and the width of the block r_i, respectively. A placement (x_i, y_i) is an assignment of coordinates to center the rectangular blocks such that there is no two rectangular blocks overlapping. That is, the restrictions to $x_i, y_i(1 \leq i \leq n)$ are such that the following equation is consistent to any

$i, j (1 \leq i \leq n, 1 \leq i \leq n, i \neq j)$.

$$|x_i - x_j| \geq \frac{(w_i + w_j)}{2} \quad (1)$$

$$|y_i - y_j| \geq \frac{(h_i + h_j)}{2} \quad (2)$$

The cost function we use for a placement consists of two parts. One is the area of the smallest rectangle that encloses the placement. This is expressed with the following equations.

$$s = \{\max_i(x_i + \frac{w_i}{2}) - \min_i(x_i - \frac{w_i}{2})\} \times \{\max_i(y_i + \frac{h_i}{2}) - \min_i(y_i - \frac{h_i}{2})\} \quad (3)$$

The other is the interconnection cost between rectangular blocks. In this case, we use the Manhattan distance between the central coordinates of each rectangle block as an approximation. On the other hand, the interconnection cost between rectangular blocks r_i and r_j is expressed as m_{ij}. The interconnection cost is represented with the following equations.

$$l = \sum_{i=1}^{n}\sum_{j=1}^{n}\{(|x_i - x_j| + |y_i - y_j|) \times m_{ij}\} \quad (4)$$

Finally, $e = s + \lambda l$ becomes a objective function. This problem aims at minimizing this function. Here, λ is the weight of the interconnection cost in this function.

3 The Proposed Method

3.1 improved positions

1. This technique has adapted the thinking of a hash method. It makes only one individual exist with one kind of gene in a population, using a function that accepts a hash key, and returns a hash result with uniform distribution. When a collision happens, the fitness of the individual that was stored there and the individual that was newly produced, is compared. The one with the higher fitness remains. This maintains diversity.

2. Genetic operations (for example, crossover, mutation) have been performed based on the genotype on the floor plan design problem as a general method. However, this method operates these modules using coordinates that are equivalent to the phenotype. Thus, similar solutions are generated easily, and a local search is performed.

3. The proposed technique uses units (this unit is called meme after this) that have information which population will be operated by which genetic operation. This is prepared beforehand. The effective meme is bred, and the low meme of an effect is extinguished, like an individual.

4. This method prepares two kinds of populations. Each population shares search as before and maintenance of diversity. The individual that maintains diversity is called a heroic individual in this paper. Almost all individuals are a copy of one of the heroic individuals. The outstanding individual created as a result of the search can replace only the heroic individual that became an ancestor. We will call the information in which a heroic individual is an ancestor, blood after this.

5. The outstanding initial individual is obtained using Hayashi's Quantification Third Method [4].

6. A Hill Climbing local search is performed intentionally. It is uses for almost all mutations. And two kinds of search are performed by two kinds of fitness. One simply uses an evaluation function to provide fitness, and the other distributes the solution in landscapes like a Sharing [5]. The Hill Climbing local search was usually repeated until processing of the target individual was completed. But the amount of calculation in this method is adjusted thanks to it's halting, after evaluating a fixed number of individuals.

3.2 proposed algorithm

Step1 Recognition of restriction conditions
Input the objective function and restriction conditions into the system. If there are parameters, which control them, input these also.

Step2 Determination of the genotype
Express genes as sign sequences. Here, we use the already proposed sequence-pair.

Step3 Creation of the initial population
Copy outstanding heroic individuals generated in the past. If a heroic individual did not exist, generate these one at random. Give peculiar blood to each hero individual. Generate an individual group, choosing method by equal probability between randomness and using the Hayashi's Quantification Third Method.

Step4 Creation of meme
Generate a fixed number of memes that have in-

formation on which individuals will be operated by which genetic operations.

Step5 Shift from heroic individuals to individuals
Choose a heroic individual. It can choose by equal probability whether to leave it as it is or to change it for an individual with the lowest fitness.

Step6 Creation of individuals
Generate offspring by crossover and mutation. Each individual are given the parents' blood. Store the generated individuals by hashing, and breed effective memes.

Step7 Hill Climbing local search that maintains diversity
Perform the Hill Climbing local search and distributes the solution into the landscapes in the first stage of search.

Step8 The straightforward Hill Climbing local search
Perform the straightforward Hill Climbing local search at the end of the search.

Step9 Shift from individuals to heroic individuals
Choose whether to leave it in a population, or shift from heroic individuals about each blood which exists in a population

For the fixed generations do Step5 to Step9

4 Numerical Experiment

To answer the question of whether or not the proposed Genetic Algorithm (pGA) better than the existing methods, we made a numerical experiment. We compared pGA with Someya's method and Shigehiro's method. In this section, these two methods are called GSA and GA after this. These methods were applied to two MCNC examples: ami33 and ami49. The terminal with three or more Net Degree was divided into the terminal with two Net Degree. All the aspect ratios were fixed to 1. We calculated for 30 minutes. And we ran experiment 10 times on ami33, 30 times on ami49.

Average convergence process to ami49 is shown in Figure 1. Each example of result of these methods is shown in Figure 2, 3, and 4.
Best, worst and average costs of each method are shown in Figure 5 and Table 1. In Figure 5 and Table 1, minimum values of two results of problems are 0 and standard deviation of two results of problems are

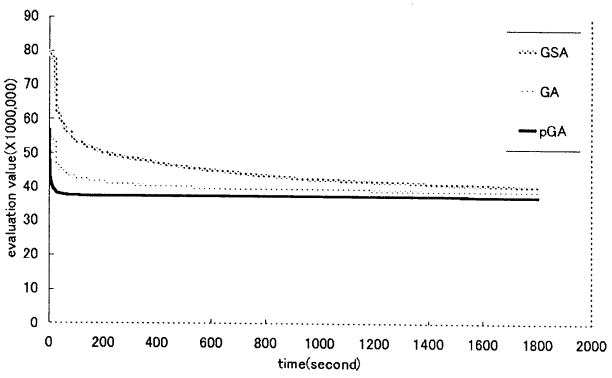


Figure 1: Convergence process

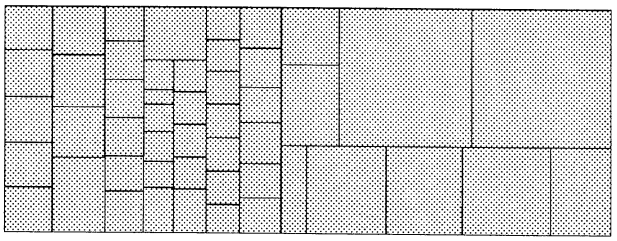


Figure 2: Result of GSA (ami49)

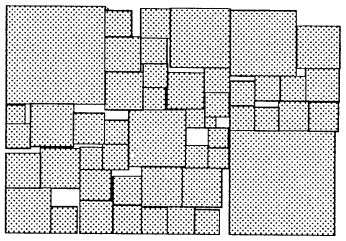


Figure 3: Result of GA (ami49)

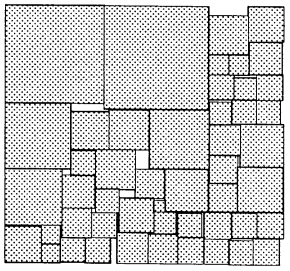


Figure 4: Result of pGA (ami49)

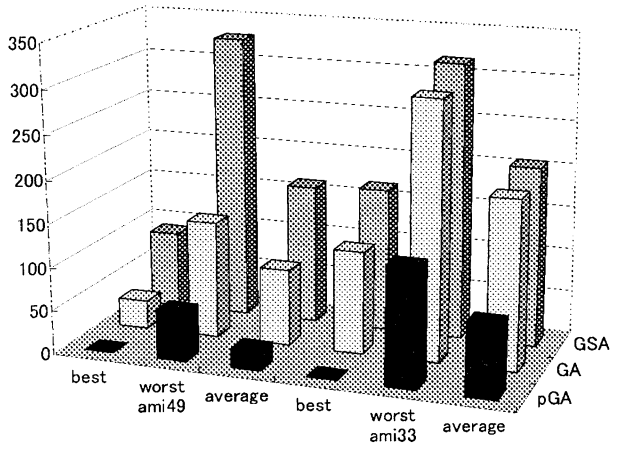


Figure 5: Experimental result by each method

Table 1: Experimental result by each method

	ami49			ami33		
	best	worst	average	best	worst	average
GSA	90	329	162	166	317	207
GA	33	135	88	118	296	194
pGA	0	55	22	0	135	82

100.
Acculate and efficient convergence of the pGA is shown in Figure 1. Solution of GSA may still be improved, but it takes long time to calculate. It is supposed that GA will not obtain a better solution than pGA. Figure 2, 3, and 4 show that pGA has the largest area of reduced deadspace. And Figure 5 shows that pGA gets the minimum cost in arbitrary items.

5 conclusions

In this paper, we aimed new GA, which aimed at "the convergence speed of the solution candidate", "improvement in the accuracy of a solution by maintenance of diversity" and "simplification of a parameter setup". The algorithm can't always calculate the strict optimal solution. But the experimental results showed that the proposed method performed better than the existing methods. And the algorithm is practical, because it does not need a complicated parameter setup.

Acknowledgements

References

[1] Gen M, Cheng R (1997), Genetic Algorithms and Engineering Design, *John Wiley & Sons*, New York.

[2] Someya H, Yamamura M (1999), A Genetic Algorithms for the Floorplan Design Problem with Search Area Adaptation along with Search Stage (in Japanese), *T.IEE Japan*, Vol.119-C, No.3, pp.393-403.

[3] Shigehiro Y, Yamaguchi S, Inoue M, Masuda T (2001), A Genetic Algorithms Based on Sequence-Pair for Floorplan Design (in Japanese), *T.IEE Japan*, Vol.121-C, No.3, pp.601-607.

[4] Tanaka Y, Wakimoto K (1998), *Methods of Multivariate Statistical Analysis* (in Japanese), Gendaisuugakusha, pp161-171.

[5] Kitano H (1995), Genetic Algorithms 2 (in Japanese), *Sangyoutosho*, pp37-38.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Nonlinear Side Constrained Transportation Problem and Two Spanning Tree-based Genetic Algorithms: A Logistic Container Terminal Application

Admi Syarif[1, 2]
[1]Dept. of Indust. & Inform. Systems Engg.
Ashikaga Institute of Technology
Ashikaga 326-8558, Japan

Mitsuo Gen[1] and X.D. Wang[1]
[2]Dept. of Mathematics
Lampung University,
B. Lampung. Indonesia

Abstract

In this paper, we discuss an extended transportation problem called nonlinear side constrained transportation problem (nscTP). With this side constraint, however, the difficulty of the problem increases significaltly and it becomes imposible to be solved by using conventional linear programming software package (*i.e.* LINDO). We proposed two hybridized spanning tree-based genetic algorithm approaches (hstGA1 and hstGA2) to solve the problem. In this methods, we adopt a Prüfer number to represent the candidate solution to the problem, design a new local search technique called displacing Prüfer and use fuzzy logic controller (FLC) to dynamically control the GA parameters. Several computational experiment results and comparisons with other conventional methods are given to show the effectiveness of the proposed methods.

keywords: side constraint, transportation problem, Prüfer number, spanning tree-based Genetic Algorithm, fuzzy logic controller, local search

1 Introduction

The first formulation and discussion of a planar transportation model was introduced by Hitchcock [1]. For some real-world applications, it is often that the transportation problem is extended to satisfy several other additional constraints. For example, Sun [2] introduced the problem called the transportation problem with exclusionary side constraint. In our real life, this kind of problem represents many applications. To solve this problem, he developed a Tabu Search method [2] and two branch-and-bound methods [3]. The Genetic Algorithm approach for solving this problem was also developed by Syarif and Gen [4].

Another similar problem called the transportation problem with nonlinear side constraint (nscTP) was introduced by Cao [5], [6]. To solve it, he proposed a Tabu search method and branch-and-bound method. In this nscTP, some pairs of destinations given as the side constraint cannot be served by a source center simultaneously. With this side constraint the difficulty of the problem becomes enermously increase, while its applications to the real world also increase significantly. Moreover, since this side constraint is nonlinear, it is impossible to solve this problem using a traditional linear programming software package (*i.e.* LINDO).

The purpose of this paper is to describe two new techniques called hst-GA1 and hst-GA2 to solve nscTP. We adopt the Prüfer number representation as it is known to be efficient way to represent network graph [7]. We briefly discuss the main characteristic of the GA such as a representation of solutions, method for handling the constraint, mechanism to create the initial population and genetic operations. Further, in order to increase the performance of the algorithm, we also develop a local search technique called displacing Prüfer number and adopt the automatic fine tuning for the crossover ratio and mutation ratio using fuzzy logic controller (FLC) [8].

2 Mathematical Model

The transportation problem with nonlinear side constraints (nscTP) that we consider in this paper has many applications in our real world. As one of examples, the posing of nscTP is as follows [5], [6]: In container terminal which is divided into several areas (indexed by i) and arriving containers are classified into several categories (index by j and k) according to certain criteria. The problem of assignment of the storage positions for arriving container is to find a reasonable assignment strategy so that the costs of operations (searching for and/or loading containers) can be minimized. The side constraint here represent some of necessary conditions (*e.g.*, the limitation of the space in the storage so that some pair of different categories

of containers can not be stacked in the same areas. In other word, the source i cannot serve two destinations j and k simultaneously).

The nscTP with given m sub-areas and n categories of container can be formulated as follows:

$$\min \quad \mathbf{Z} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \cdots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \cdots, n \tag{3}$$

$$x_{ij}x_{ik} = 0 \text{ for } (j,k) \in D_i \; i = 1, 2, \cdots, m \tag{4}$$

$$x_{ij} \geq 0, \quad \forall\, i, j \tag{5}$$

where x_{ij} is the unknown quantity of container categories j to be assigned at the sub-area i, c_{ij} will be a cost function of assigning one unit of container with

categories j to the sub-area i. Sub-area i, $i = 1, 2, \cdots, m$ has a capacity of a_i unit and the number of container j is b_j units. D_i is the set of pairs of destination (j, k) that cannot be served by source i at the same time, given in practice for all i.

3 Design of the Algorithms

Genetic algorithm is known one of stochastic search algorithms based on the mechanism of natural selection and natural genetics. To our knowledge, there are at least two types of chromosome representation for transportation problem. The first type is a matrix-based representation [7], [9] which is the most common representation used by researchers. Another representation is known as tree-based represntation.

3.1 Spanning Tree-based Representation

The effectiveness of Prüfer number representation for various network problems including the transportation problem has been shown Gen and Cheng in [7], [10]. This Prüfer number encoding procedure is belong to the class of vertex encoding. When generating a Prüfer number, there will be possibility that it cannot be adapted into a transportation network graph. In our previous work [4], we have developed the feasibility criteria and repairing procedure for the Prüfer number. A feasible Prüfer number can be decoded into a transportation tree by using the decoding procedure given in [4].

3.2 Handling for Side Constraint

After generating a feasible Prüfer number, another important issues is how to handle the side constraints. In the GA process, we proposed two algorithms for checking the exclusion of side constraints in nscTP. We describe both procedures as follows:

Algorithm 1
Initially, nscTP is solved without considering the constraint (4). After decoding the Prufer number, the developed tranportation network is checked for the side costraints. If it is not satisfy the side constraints, the chromosome is rejeceted by adding a pinalty value to the objective function. It is done by the following procedure:

procedure: Checking for Side Constraints 1

for$(n = 1; n <= nc; n++)\{$
  $j = sc(n, 1);$
  $k = sc(n, 2);$
  for$(i = 1; i <= m; i++)\{$
    $pinalty = 1000 \cdot x_{ij} \cdot x_{ik};$
  $\}$
$\}$

Algorithm 2 In this algorithm, the procedure for checking the side constraints is included into the decoding procedure. We add the side constrained checking procedure after determining the amount to be assigned at edge (i, j). The procedure for checking the side constraint in this algorithm is given as follows:

procedure: Checking for Side Constraints 2

for$(k = 1; k <= nc; k++)\{$
  if$((j = sc(n, 1))$ and $(x_{i,sc(n,2)} > 0)$ or
  $(j = sc(n, 2))$ and $(x_{i,sc(n,1)} > 0))\{$
    $x_{ij} = 0;$
  $\}$
$\}$

3.3 Genetic Operators

3.3.1 Crossover

The crossover is done for exchanging the information of two parents and providing a powerful exploration capability. we employed a one-cut-point crossover, which randomly selects a one-cut-point and exchanges the right parts of two parents to generate offspring.

3.3.2 Mutation

It is well known that the mutation operator is very important for the success of the genetic algorithm process. It is usually done by modifying one or more of the gene values of an existing individual to increase the variability of the population so that the GA process does not stuck in locally optimal solution. We used here an inversion mutation that always generates feasible offspiring (Prüfer number).

3.3.3 Evaluation and Selection

For this problem, we use the objective function as the fitness falue of each chromosome. In the selection procedure, we use the mixed strategy with $(\mu+\lambda)$–selection and roulette wheel selection. This strategy selects μ best chromosomes from μ parents and λ offspring. If there are no μ different chromosomes available, then the vacant pool of population is filled up with roulette wheel selection. So this selection method can always enforce the best chromosomes into the next generation.

3.4 Local Search Using Displacing Prüfer number

Combining GA with local search has been shown to be effective way to move out of a local optimum and carefully search the near optimal region. Here, we develop a new simple local search operation called displacing Prufer number that can keep the similarity of the chromosome. The Prüfer number resulted by this operation is guaranteed to be feasible since the appearance number for each node remains the same.The concept of displacing Prüfer number is stated as follows: select one number in the Prüfer number randomly; And, exchange it with the first number of the Prüfer number.

3.5 Fuzzy Logic Controller

One of problem in the GA applications is how to determine the value of GA parameters to reach a fine satisfactory solution. To deal with this problem,' recently, several authors proposed fuzzy logic controller to dynamically control the GA parameters. In our implementation of fuzzy logic controller, we used the Wang *et al.*'s concepts [8] to regulate automatically the GA parameters, crossover ration p_C and mutation ratio p_M.

The heuristic updating principles for the crossover and mutation ratio are to consider changes in the average fitness of the populations. For simplicity, in the implementation, the input values are respectively normalized into integer values in the range [-4.0,4.0] corresponding maximum/minimum values. After determining the control action $Z(i,j)$ value, the change in the crossover and mutation rations are computed as follows:

$$\Delta c(t) = 0.02 \times Z(i,j) \quad (6)$$

$$\Delta m(t) = 0.002 \times Z(i,j) \quad (7)$$

The value of crossover ratio for the next generation is calculated as follows:

$$p_C(t) = p_C(t-1) + \Delta c(t) \quad (8)$$

$$p_M(t) = p_M(t-1) + \Delta m(t) \quad (9)$$

4 Numerical Experiments

For our numerical experiments, the proposed algorithms (hst-GA1 and hst-GA2) were implemented in Visual C language and run on Pentium 700 PC. To confirm the effectiveness of the proposed methods, we also developed the traditional matrix based GA (m-GA) and traditional spanning tree-based GA (st-GA). We set the initial parameter $p_C = 0.4$ and $p_M = 0.2$ which have been reported as good parameters [10]. We tested all algorithms by using 4 test problems. As the test problems we modify the benchmark test problems given in [12]. The design of our numerical experiment is given in the following Table 1:

Table 1: Design of the test problems

Test problem	m	n	nc	d_s	pop_size
1	4	6	2	17	20
2	8	12	7	15	30
3	10	10	8	16	50
4	8	32	30	24	100

To be fair, for each computational experiment, all algorithms were run with the same t_max. The results of these numerical experiments are summarized in the following Table 2.

Here, we computed the optimal value of their linear problems by using LINDO. It is clear that the optimal value of nscTP should be equal to or greater than its linear optimal solution. In general, we found that hst-GA2 performs better best fitness value and average fitness value than those of the other methods. In contrass to the other algorithms, in hstGA2, the side constraints are checked during the decoding of the chromosome. So the offspring would be feasible for the side constraints. Thus this method would be more effective.

To investigate the convergence behavior of the algorithms in the evolutionary process, we did another numerical experiment by using the test problem 4 with $pop\_size = 100$ and $max\_gen = 3000$. The following Figure 1 shows the average fitness value of the algorithms in the evolutionary process.

Table 2: Numerical experiment results

Problem	t_max	m-GA		st-GA		hst-GA1		hst-GA2		LINDO results
		mean	best	mean	best	mean	best	mean	best	
1	3	99.35	99.35	99.35	99.35	99.35	99.35	99.35	99.35	99.35
	5	99.35	99.35	99.35	99.35	99.35	99.35	99.35	99.35	
2	25	271.76	267.6	270.68	267.6	268.79	267.6	267.91	267.6	266.7
	40	270.53	267.3	269.75	267.6	268.12	267.6	267.43	266.7	
3	40	250.1	236	253.8	236	246.9	236	241.1	234	228
	60	247.6	234	245.3	234	243.2	234	238.2	234	
4	100	1294.8	1112	1263.5	1117	1238.6	1101	1194.7	1057	861
	150	1187.3	1097	1162.8	1086	1137.4	1072	1105.2	1057	

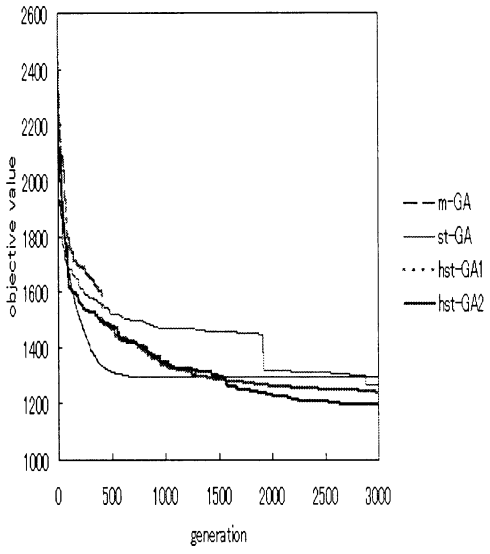


Figure 1: The objective function in the generation

5 Conclusion

In this paper, we proposed new approaches called hst-GA1 and hst-GA2 for solving nonlinear side constrained transportation problem. In order to improve their effectiveness, proposed methods were hybridized with local search and fuzzy logic controller. The numerical results show the superiority of the proposed method (hst-GA2) over the existing traditional methods for solving this nscTP.

References

[1] Hitchcock, F. L., The Distribution of a Product from Several Soources to Numerous Localities, *Journal of Mathematical Physic*, Vol. 20, pp. 224-230, 1941.

[2] Sun, M., A Tabu Search Heuristic Procedure for Solving the Transportation Problems with Exclusionary Side Constraints, *Journal of Heuristic*, 3: 305-326, Kluwer Academic Publisher, 1998

[3] Sun, M., The Transportation Problems with Exclusionary Side Constraints and Two Branch-and-Bound Algorithms, *European Journal of Operational Research*, 140: 629-647, 2002

[4] Syarif, A. and Gen, M. Solving Exclusionary Side Constrained Transportation Problem by Using A Hybrid Spanning Tree-based Genetic Algorithm, *Journal of Intelligent Manufacturing*, to appear.

[5] Cao, B., Transportation Problems with Nonlinear Side Constraints: a branch-and-bound approach, *Zeitschrift Operations Research*, 36, pp. 185-197, 1992.

[6] Cao, B. and G. Uebe, Solving Transportation Problems with Nonlinear Side Constraints with Tabu Search, *Computers Ops. Res.*, Vol. 22, No. 6, pp. 593-603, Britain, 1995.

[7] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.

[8] Wang, P. T., G. S. Wang and Z. G. Hu, Speeding Up the Search Process of Genetic Algorithm by Fuzzy Logic, *Proc. of the 5th European Congress on Intelligent Techniques and Soft Computing*, pp. 665-671, 1994.

[9] Michalewicz, Z., G. A. Vignaux and M. Hobbs, A Non-Standard Genetic Algorithm for the Nonlinear Transportation Problem, *ORSA Journal on Computing*, Vol. 3, No. 4, pp. 307-316, 1991.

[10] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.

[11] Dossey, J., A. Otto, L. Spence and C. Eynden, *Discrete Mathematics*, Harper Collins, 1993.

[12] http://www.in.tu-clausthal.de/ gottlieb/ benchmarks/fctp/

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Active Solution and Active Solution Space on Job-shop Scheduling Problem

M. Watanabe K.Ida T. Horita
Department of Imformation Engineering
Maebashi Institute of Technology
Maebashi, 371-0816, Japan

M. Gen
Department of Industrial and Information Systems Engineering
Ashikaga Institute of Technology
Ashikaga, 326-8558, Japan

Abstract

In this paper we propose a new searching method of Genetic Algorithm for Job shop scheduling problem (JSP). In the ordinal representation with a priority in JSP, an active schedule is created by using a left shift. We define an active solution, it is individual which can create an active schedule without using a left shift, a set of it called an active solution space. We propose an algorithm which can search the active solution space effectively.

1 Introduction

Genetic Algorithm had been applied to many combinatorial optimization problems[1]. Genetic Algorithm searches a better solution by leaving a good gene information in evolution. So many researchers have been studied, and proposed many coding techniques, crossover techniques and mutation techniques[1]. Especially to design a coding is important for solve a difficult combinatorial optimization problems.

Job shop scheduling problem is one of the difficult to solve combinatorial optimization problems. It has been argued about a multiple functions and UV-structure solution space[2]. Moreover it have many condition of coding for create a fisible solution (schedule). Therefore many coding methods have been proposed. We selected an ordinal representation with a priority as the coding method. It can have much information about the scheduling. And it is easy to use and can create an active schedule by using a left shift. However the evaluation is differ whether using a left shift or without using a left shift, therefore it could not said properly that individual had good gene information even if evaluation is good. it is not preference for GA because it search a better solution by leaving a good gene information in evolution.

In this paper we define an active solution and an active solution space at first. The active solution is an individual (genotype) which can create an active schedule (phenotype) without using a left shift. And we define a set of active solution is an active solution space. Next we propose an algorithm, which change a semi-active solution into an active solution while individual evaluated, in order to search the active solution space effectively. By applying the proposed method to some benchmark problems, we show its effectiveness.

2 Job-shop Scheduling Problem and Genetic Algorithm

2.1 job-shop scheduling problem

In general, Job-shop scheduling problem (JSP) is described as follows[3]. There are n jobs to be processed on m machines exactly once. A purpose of JSP is to find an optimum schedule, which has a minimum makespan. Subject as following. The technological ordering (the prosessing machine sequence) with processing times are prescribed, and all operation is not interrupted. Table1 shows an example of 3×3 JSP.

Table 1: example of JSP

job	(technological ordering, processing time)		
j_1	(1, 3)	(2, 3)	(3, 2)
j_2	(1, 1)	(3, 5)	(2, 3)
j_3	(2, 3)	(1, 2)	(3, 3)

2.2 genotype (ordinal representation with a priority)

To design a coding is needed to solve by the genetic algorithm, and it is important. In this paper, we use an *ordinal representation with a priority*, which arrange a job number using technological ordering [4]. In this method, a chromosome denoted by $n \times m$ genes.

Table 2: pacentage

	percentage					
problem	ex	ft06	la01	la06	ft10	ft20
solution space/size	3×3	6×6	10×5	15×5	10×10	20×5
active	43.93	0.82	0.02	0.00	0.00	0.00
semi-active	56.07	99.18	99.98	100.00	100.00	100.00

And a feasible schedule can be represented by a string of job numbers with each number appeared m times.

A job-machine list is created from the chromosome by using the prescribed technological ordering, so arrangements for placing to Gantt chart, is completed. Fig1 shows an example of chromosome and a job-machine list and arrangements for placing to Gantt chart on the example of Table on 3×3 JSP.

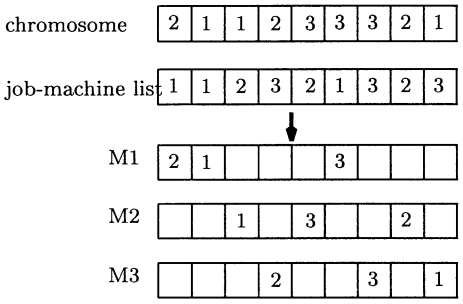


Figure 1: job-machne list

2.3 phenotype (schedule)

To change a phenotype (which means about a schedule in JSP), from a genotype is needed to obtain a makespan, and the Gantt chart is used in order to create a schedule. Then, it is important whether created schedule is an active schedule or a semi-active schedule. A feasible solution is to be satisfied the conditions which is technological ordering And a semi-active schedule is shortest schedule without exchanging of job sequence. And an active schedule is shortest schedule with exchanging of job sequence. In the ordinal representation with a priority, the gene placed on Gannt chart from head gene, then there is case which place on right side even if it placed on more left side. Therefore in ordinal representation with a priority, left shift is used in order to placed on more right side. This schedule which used left shift, is called active schedule. The example of semi-active schedule (without used left shift) and active schedule (used left shift) shows Fig2 and Fig3.

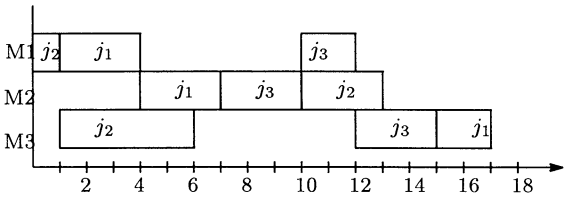


Figure 2: semi-active schedule

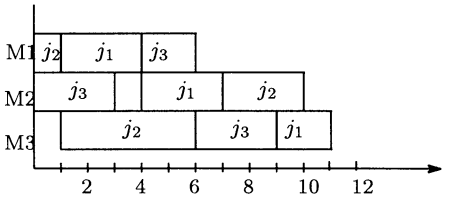


Figure 3: active schedule

3 Active Solution and Active Solution Space

3.1 define an active solution and an active solution space

As you see above description, it is possible to create an active schedule every time if it used a left shift. However the evaluation differ whether using a left shift or without using a left shift. In the other words, if it use a left shift then active schedule was created, but if it did not use a left shift then active schedule was not created so the evaluation differ. Therefore it could not say properly that individual had good gene information even if evaluation is good. In method of GA, that can search a better solution by leaving a better gene information for offspring and do generation, it is not preference.

Therefore, we define an active solution at first. It is individual that can create an active schedule without using a left shift. And non-active solution called semi-active solution. Fig4 shows relationships of the active solution and the active schedule. The solution space is considered, as it exist different space on genotype and

phenotype. At the semi-active solution space, if it used a left shift then active schedule created, and if it did not use, semi-active schedule is created. At the active solution space, active schedule was created both using a left shift and without using a left shift. Table2 shows the percentage of appearance active solution and semi-active solution in created solution randomly on some benchmark problems[5].

From this table, in a small size problem as a 3×3 JSP, it is even percentage that create a active solution and semi-active solution. However large size problems as 6×6 or more, semi-active solution was created almost every time.

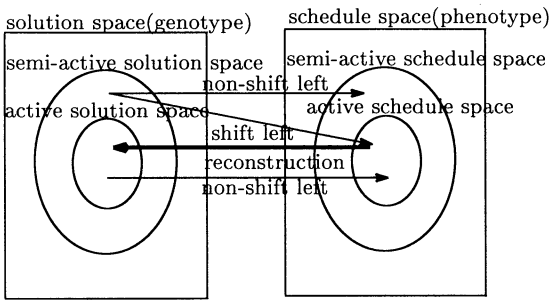


Figure 4: active solution space and active schedule space and reconstruction

3.2 searching the active solution space

We proposed an algorithm that we considered to search the active solution space effectively. When solution evaluated, change to active solution from semi active solution is performed, that is, if the left shift was used on placing on Gannt chart, and then solution is constructed. That algorithm as followed and Fig4 shows image of reconstruction of solution.

step1 Job-machine list create and $i = 0$

step2 Following step repeat until placing on Gannt chart all gene completed.

step3 Read ith gene (job) and decide a machine number which should be processed on read-job from job-machine list.

step4 Read-job place on Gannt chart most early time.

step5 If left shift is not used at step4 then $i = i + 1$ and go back to step2.

step6 Decide a gene which correspond to most early job in genes which changed processing ordering. And the position of gene set to j.

step7 ith gene is inserted in the position of jth gene. And do right shift after jth gene. Similarly data of job-machine list moved together. Go back step2.

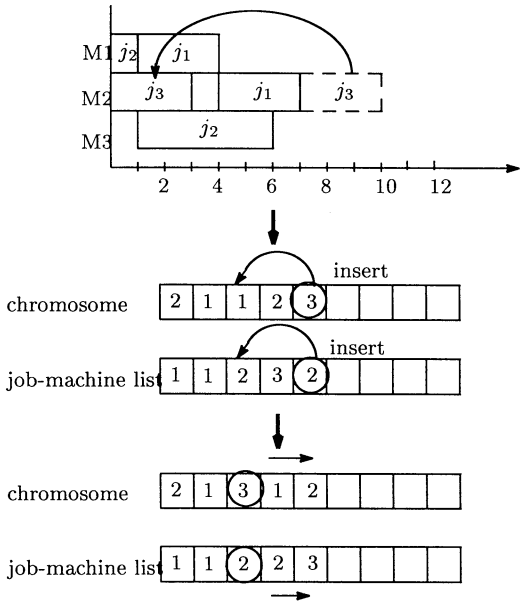


Figure 5: example of reconstruction

4 Experiment

4.1 Set up

Proposed algorithm can use additionally to former GA. So we experimented to compare a GA (before) with a GA which added proposed algorithm (after). The set up of experiment shows as follow. We applied to benchmark problems (ft10, ft20, abz5, abz7, abz8, la21 and la22)[5]. Genetic algorithm (Generation Alternation Model and Genetic operator[6]) was based on Ref[7]. it is simply and widely used. Parameters are set up as Population size is 100, and Crossover ratio and Mutation ratio are fixed 0.8 and 0.2. Each algorithm implemented in C language and run 100 times for trials on CPU *Pentium* 733MHz.

So many better genetic algorithm has been proposed these days[6][8], but purpose of this experiment and this paper, is to confirm an effectively of proposed

Table 3: experimental results

	Before				After			
	Best	Worst	Ave	Time(sec)	Best	Worst	Ave	Time(sec)
ft10	937	982	957.33	34	930	967	949.25	90
ft20	1184	1234	1207.47	39	1173	1199	1182.38	96
abz5	1234	1253	1246.15	34	1234	1250	1241.58	90
abz7	716	744	731.85	99	700	726	715.57	600
abz8	742	769	754.66	99	720	746	733.61	600
la21	1082	1130	1104.61	48	1074	1109	1092.10	176
la22	948	986	970.85	48	940	970	953.52	176

algorithm and to show meaning of an active solution, so this set up is full.

4.2 experimental results and examination

Table3 shows results of experiment. Best shows best solution, Worst shows Worst solution, Ave shows average of solution and Time shows average of CPU time. Number of evaluated solution, that is, number of created solution, was about 300,000 pieces on each method methods and problems.

From this table, we can see that proposed algorithm had good results on all problem and all results (Best, Worst and Ave). Therefore, meaning of an active solution is shown from these results.

5 Conclution

In this paper we defined an active solution and an active solution space. And we showed meaning of an active solution by experimental results of proposed an algorithm which can search an active solution space by creating an active solution effectively. In the future study, we apply this algorithm to better Genetic algorithm in order to search an optimum solution. And this study is positioned in a coding methods, so we will design a genetic operator based on this study.

References

[1] M. Gen and R. Cheng, *Genetic Algorithm & Engineering Optimization*, J.WILEY, 1999.

[2] K. Ikeda and S. Kobayashi, "UV Phenomenon on Genetic Algorithms, and UV Structure Hypothesis", *JASI* Vol.17. No.3. pp239–246, 2002 (in Japanese).

[3] I. Nabeshima, *Theory of Scheduling*, Morikita shuppan, 1974 (in Japanese).

[4] H. Hiromi,"Genetic Algorithm with Cluster Averaging Method for Solving Job-shop Scheduling problems, *JSAI*, Vol.10, No.5, pp.796–777, 1995 (in Japanese).

[5] Mattfeld, D.C. and Vaessens, R.J.M: OR-Library, http://mscmga.ms.ic.ac.uk/jeb/orlib/jobshopinfo.html

[6] H. Sato, I. Ono and S. Kobayashi, "A New Generation Alternation Model of Genetic Algorithms and its Assessment", *JSAI*, Vol.12. No.5. pp.734–744, 1997 (in Japanese).

[7] K. Ida, M. Watanabe and M. Gen,"A Genetic Algorithm with Modified Crossover Operator and Search Area Adaptation for Job-shop Scheduling problem,"*Proceedings of the 30th International Conference on Computers and Industrial Engineering* ,Vol.1, pp.389–394, 2002.

[8] H. Someya and M. Yamamura, "A Genetic Algorithm without Parameters Tuning and its Application on the Floorplan Design Problem, Proc. *GECCO'99*, pp.620–627,1999.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Evolutionary Network Design Technique Based on Genetic Algorithm

Mitsuo Gen[1] Admi Syarif[1,2] and Jang-Hyoung Kim[3]

[1]Department of Industrial & Information Systems Engineering,
Ashikaga Institute of Technology, Ashikaga 326-8558, Japan
E-mail: gen@ashitech.ac.jp
[2]Department of Mathematics, Lampung University,
Bandar Lampung, Indonesia, 35148
[3]Department of Information Engineering ,
Cheju National University, Korea

Abstract: Genetic Algorithms (GA) are known as one of powerful heuristic methods for solving various optimization problems. Over the past few years, there have been extensive research works on applying GA for networks design problems including minimum spanning tree problem, local area network (LAN) design, location allocation problem and so on. In this paper, we summarized recent research works on network design problems by using GA approach.

Keywords: Genetic Algorithm, network design, minimum spanning tree problem, transportation problem

1. Introduction

Over past view years, GA has been extensively used for solving hard combinatorial optimization problems. Many researchers have reported the success of GA in variety of applications domain including engineering, finance, economics, agriculture, business and so on. Network design problems are known to be one of problems that have wide applications in our daily life. However, with the increase of problem scale, the problems become more complicated and difficult to be solved effectively and efficiently by using conventional techniques. The purpose of this paper is to give an overview of GA applications for various network design problem including local area network (LAN) design, minimum spanning tree problem, location allocation problem and so on.

2. Minimum Spanning Tree Problem

Minimum spanning tree problem (MST) is one of best-known network optimization problem for designing backbone network It usually consist of finding the best way to link n nodes at different location. A special case of MST is a degree-constrained minimum spanning tree (dc-MST) that requires to satisfy additional constraint such as capacities on edges or nodes. The dc-MST is an NP-Hard and traditional heuristics have had only limited success in solving small-to-mid size problems [1]. Figure 1 illustrates an example of unconstrained spanning tree with degree 5 and a spanning tree with maximum degree 3.

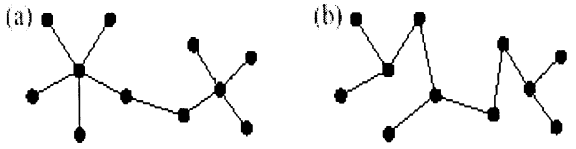


Figure 1. An example of unconstrained spanning tree and a spanning tree with maximum degree 3

Consider an undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of nodes. For a subset of nodes S define $E(S)$ be the edges whose end points are in S. For all edges $(i, j) \in E$, define $x_{ij} = 1$ if the edge (i, j) is selected for spanning tree and otherwise $x_{ij} = 0$. Let w_{ij} be the fixed cost related to the edge (i, j), The dcMST can be formulated as follows:

$$\min \quad z(x) = \sum_{i=1}^{n-1} \sum_{j=2}^{n} w_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{n-1} \sum_{j=2}^{n} x_{ij} = n - 1$$

$$\sum_{i \in s} \sum_{j \in s, j > 1} x_{ij} \le |S| - 1, S \subseteq V \setminus \{1\}, \qquad |S| \ge 2$$

$$\sum_{j=1}^{n} x_{ij} \le b_i, \qquad i = 1,2,..., n$$

$$x_{ij} = 0 \text{ or } 1, \qquad i = 1,2,..., n-1, \quad j = 2,3,..., n$$

where b_j. is the constrained degree value for node i.

There are several research works on this problem. Zhou and Gen [2] firstly reported GA applications for solving this problem. They used a two–dimension structure to encode a spanning tree with degree constraints: one dimension for node permutation and the other for degree constraint. Raidl [3] proposed a method for representing spanning tree in evolutionary algorithm directly as a set of their edges. They demonstrated the usefulness of the edge-set encoding for dcMST problem. They compared several

algorithms that encode spanning tree cia Blob Code, with network random keys and as string of weight. It was shown that their method is better especially for larger instances. Gotttlieb *et. al.* [4] noted several spanning tree encoding methods for dcMST problem such as Zhou and Gen's method [2], Palmer and Kershenbaum's [5 Krishnanoorthy's method [6] and Raidl's method [7. They reported that Raidl's method (edge list representation) perform best for this problem.

3. Local Area Network Design Problem

Recently, local area networks (LANs) are commonly used as the communication infrastructure that meets the demands of the users in local environment. Figure 4 shows an example of simple LAN architecture with 5 centers and 23 terminals. Since the use of transparent bridges in computer network requires loop-free path between LAN segments, the spanning tree topology can also be used as active LAN configuration when designing a computer network system.

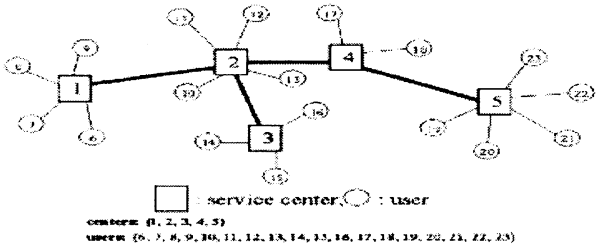


Figure 4. Simple LAN Architecture

Define the $n \times n$ service center topology matrix X_1, whose element x_{ij} represents whether the centers i and j are connected. Assume that LANs are partitioned into n segments (service centers). The users are distributed over those n service centers. The $n \times m$ clustering matrix X_2 specifies which user belongs to which center, whose element x_{2ij} means whether user j belongs to center i. Define the n $(n+m)$ matrix X called the spanning tree matrix ($[X_1,X_2]$). The $M/M/1$ model is used to describe a single cluster (LAN segment) behavior. Then we can formulate the bicriteria LAN topology design (bc-LAN) problem as the following nonlinear0-1 programming model.

$$\min \quad \frac{1}{\Gamma}\left[\sum_{i=1}^{n}\frac{c_i(\mathrm{X})}{C_i - c_i(\mathrm{X})} + \sum_{i=1}^{n}\sum_{j=1}^{m}\beta_{ij}\cdot d_{ij}(\mathrm{X})\right]$$

$$\min \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{m} w_{1ij}\cdot x_{1ij} + \sum_{i=1}^{n}\sum_{j=1}^{m} w_{2ij}\cdot x_{2ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{2ij} \le g_i, \qquad i = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{2ij} = 1, \qquad j = 1,2,\ldots,m$$

$$c_i(\mathrm{X}) < C_i, \qquad i = 1,2,\ldots,n$$

Where Γ is the total offered traffic, $c_i(\mathrm{X})$ the total traffic at center i, $d_{ij}(\mathrm{X})$ the total traffic through link (k,l), C_i the traffic capacity of center i, β_{ij} the delay per bit due to the link between centers i and j, g_i the maximum number which is capable of connecting to center i, w_{1ij} the weight of the link between centers i and j, and w_{2ij} is the weight of the link between center i and user j.

As the solution of this problem can be regarded as a spanning tree, where all users are terminal or leaf nodes and all centers are internal nodes. Kim *et. al.* [8] proposed a spanning tree-based GA for solving this problem. Prufer number was used to encode all internal nodes and all leaf nodes are not included in the encoding. GAs are used to search for Pareto optimal solutions and TOPSIS method was used to compute the compromise solution among Pareto optimal solution.

Gottlieb *et. al* [4] adapted NetKeys technique and compared the results with Prufer number representation. They used four instances derived from real-world problem whose nodes represent location throughout Germany.

4. Shortest Path Problem

The shortest path problem (SPP) is known as one of the most common problems encountered in analysis of networks. This problem appears in many applications involving transportation, routing and communication [9]. In this problem, one should find a path between two designated nodes having minimum total length or cost.

The used of GA for solving this problem was first proposed by Gen and Cheng [10]. They considered the SPP with two objective functions and proposed a priority based encoding method to overcome several difficulties of encoding path to a graph as follows: (1) a path contains variable number of nodes and (2) a random sequence of edges usually does not correspondence to a path. They used the position of a gene to represent a node and the value of the gene to represent the priority of the node for constructing a path among candidates. The illustration of their chromosome representation is given in the following figure.

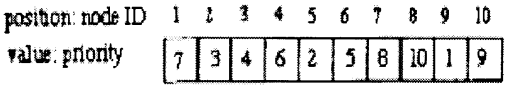


Figure 5. An example of priority based encoding.

To determine the solutions that are closest to the ideal solution, they assign a weight value to each objective to signal the different degrees of importance. The experiment was done by using a non-planar and undirected graph with 100 nodes and 473 edges. Their compromise solutions were compared by the ideal solution obtained by running Floyd-Warshall algorithm twice.

5. Multi-stage Production/Distribution Problem

Logistics problems are known as the problems in operations management and operations research that

require us to make decisions in several stages. It is defined as the process of anticipating customer needs and wants; acquiring the capitals, material, people, technologies and information necessary to meet those need and wants; optimizing the goods- or service-producing network to fulfill customer requests; and utilizing the network to fulfill customer requests in a timely way [11]. The common objective is to find strategic options for improving the efficiency in the distribution performance that meets the demand at minimum cost or fills demand for maximum profit.

In this recent year, researches in multi-stage logistics problems including multi echelon location/allocation problems have taken great interest of many researchers. Pirkul and Jayawarman researched the network design problem for two stages, multiple plants and multiple capacitated distribution centers [12]. The similar problem was solved by Sim *et. al.* by using heuristic method based on Lagrangian relaxation [13].

Syarif *et.al* [14] considered the multi-stage logistic system problem formulated by 0-1 mixed linear programming model. In this model, they give the maximum number of facilities (plants and distribution centers) to be opened as the constraints. These constraints increase the difficulty of the problem, yet the relevance for the real world applicability also increases significantly. The design tasks of this problem involve the choice of facilities (plants and distribution center) to be opened and the distribution network design to satisfy the customer demand with least cost. This kind of problem can be viewed as the combination of multiple-choice Knapsack problem with the location-allocation problem simultaneously. So this problem is known to be NP-hard [10]. They assumed that the logistic system process for the problem is organized according to a three-sequence of stages as illustrated in the following figure [15]

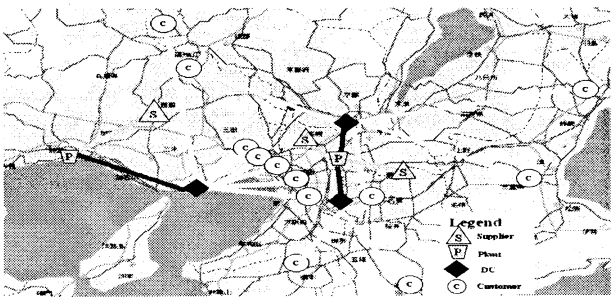


Figure 6. The illustration of three-stage logistics system

As the solution method, we proposed a spanning tree-based genetic algorithm (st-GA). They adopted the Prüfer numbers and designed the feasibility of the chromosome. To handle the infeasible chromosome, they also developed a repairing procedure. With this repairing procedure, they showed that this method can be applied for relatively large size problems. The effectiveness and efficiency of the proposed method is demonstrated by comparing numerical experiment results of the proposed method with those of traditional matrix-based genetic algorithm.

6. The facility location problem

The facility location problem is known as one of the important problem faced in industry. There are many variations of this problem dealing different models, relevant to various situations. However, in most cases, this problem is classified according to the capacity of the facilities. When the facilities have certain capacity, the problem is referred as a capacitated location allocation problem. In this problem, a number of potential facilities with certain limit on capacity such as service centers, plants, distribution centers (DCs) are given and the problem is to assign facilities to the location in such a way that the sum of the fixed cost of opening facilities and variable cost of transporting the customer demand from facilities is minimized. On the other hand, when it is assumed that the facilities have no limit on capacity then the problem is referred as an uncapacitated location/allocation problem (uLAP).

In uLAP, some facilities are located among n possible sites and the objective is to satisfy all demand at m given location with least cost. The cost here usually consists of both the fixed cost for establishing the facilities and the cost for fulfilling the demand (transportation/ distribution cost). It has been shown that this problem is NP-hard problem [16]. A homogeneous product to be produced in n possible sites, and given m customers, at known location, characterized by required level of demand. A cost function c_{ij} is associated with distribution cost of serving customer j from facility i. The fixed cost, f_i, is the cost of establishing facility at site i. The problem can be mathematically stated as follows:

$$\textbf{min} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n} f_i y_i$$

s.t.

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in J$$

$$x_{ij} \le y_i \qquad \forall i \in I,\ j \in J$$

$$x_{ij},\ y_i \in \{0, 1\} \qquad \forall i \in I,\ j \in J$$

where

$x_{ij} = 1$ If the customer j is served by facility i.
Otherwise $x_{ij} = 0$.
$y_i = 1$ If facility $i$is established. Otherwise $y_i = 0$.
$I = \{1, 2, \ldots, n\}$
$J = \{1, 2, \ldots, m\}$

In the above model, the equation 1 represents the total cost of establishing the facilities and fulfilling the demand (transportation/distribution cost) to be minimized. The constraint 2 ensures that the demand of each customer is fulfilled by only one facility (i.e. no partial fulfillment of demand is allowed).

The body of literature on various location/allocation problems is large, however, most of them deal with capacitated location allocation problem. Al-Sultan and Al-Fawzan [17] presented a Tabu search algorithm for solving

uLAP. They developed a Net Benefid Heuristic (NBH) algorithm for the uLAP. The improvement of this algorithm is given by Al-Fawzan [18]. Another well known heuristic procedure for solving this problem is also given by Kuehn and Hamburger [19]. Sule [20] developed a heuristic method based on the idea of net saving resulting from the reallocation of facilities. This method has some similarities with Al-Fawzan's method.

Gen *et. al* [21] proposed a parallel GA approach to solve uLAP. The fuzzy logic controller (FLC) approach was adopted to auto-tune the GA parameters. The proposed algorithm is tested by using some standard test problems taken from literature [22, 23]. The computational results of the proposed algorithm are compared with the known optimal solution taken from literature.

Conclusion

With the development of modern society, network design becomes an important part human being's life. How to solve network design problem effectively and efficiently will be a great research issue in this century or even further future. In this paper, we just gave a brief review about our research works and other recent research works on network design problems by using GA. It is shown that GA has a great potential power to cope with network design problem such as: minimum spanning tree problem, local are network (LAN) design, location allocation problem and so on.

References

1. Narula S. C. and Ho CA, (1980) Degree-constrained minimum spanning tree problem, Computer and Operational Research, Vol 7, pp. 239-249.
2. Zhou G. and Gen M. (1997) Approach to degree-constrained minimum spanning tree problem using genetic algorithm, *Engineering Design and Automation,* 3 (2), pp. 157-165.
3. Raidl R. Gunther, (2000) An afficient evolutionary algorithm for degree-constrained minimum spanning tree problem, in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation* , Carlos Fonseca, Jong-Hwan Kim and Alice Smith, Eds, pp. 104-111, IEEE Press.
4. Gottlieb J, Raidl, G. R, Julstrom B and Rothlauf, (2001) Prufer number: A Poor Representation of Spanning Trees of Evolutionary Search, Proc. Of Generic and Evol. Comp. Conference, pp. 343-350
5. Palmer C. C. and Kershenbaum A. (1994) Representing Trees in Genetic Algorithms., in *Proceedings of the first IEEE conference on Evolutionary Computation*, David Schaffer, Hans-Oaul Schwefel and David B. Fogel Eds. pp. 379-384.
6. Krishnamoorthy A., Ernst T. and Sharaiha. Y. M. (1999) Comparison of algorithm for degree constrained minimum spanning tree. Technical Report, CSIRO Mathematical and Information Science, Clayton, Australia.
7. Raidl G. R. and Justrom B. A. (2000) A weight coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. *Proc. Of the 200 ACM symposium on Applied Computing*, pp. 440
8. Kim J. R. Gen M and Ida K, (1999), Bicriteria network design using spanning tree-based genetic algorithm, Artificial Life Robotic, 3, pp. 65-72
9. Sancho N. G. (1986) A multi-objective routing problem, *Engineering Optimizatio*n, 10, pp. 71-76.
10. Gen, M. and Cheng, R. (1997) *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York.
11. Tilanus B. (1997) Introduction to information system in logistics and transportation, pp. 7-16, in Tilanus. B. *ed*: *Information Systems in Logistics and Transportation*, Elsevier Science Ltd..
12. Pirkul H. and Jayaraman V. A. (1998) Multi-commodity, multi-plant, capacitated location allocation problem: formulation and efficient heuristic solution, *Computer and Operation Research*, 25,10: pp. 869-878
13. Sim E., Jang Y. and Park J. (2000) A study on the supply chain network design considering multi-level, multi-product, capacitated facility, *Proceedings of Korean Supply Chain Management Society.*
14. Syarif A, Yun Y and Gen M. (2002) Study on multi-stage logictic chain network: a spanning tree-based genetic algorithm approach, Computer and Industrial Engineering, 43, pp. 299-314.
15. Yu. H. (1997) ILOG in the supply chain, *ILOG technical report.*
16. Shmoys, D. B., Tardos, E. and Aarday, K. I. (1997) Approximation algorithm for facility location problem, *Proceedings of 29th Annual ACM Symposium on Theory of Computing*, pp. 265-274.
17. Al-sultan, K. S. and Al-Fawzan, M. A. (1999) A new approach to the Uncapacitated Facility Location Problem, *Annal. Of Operations Research*, 86, pp. 91-103
18. Al-Fawzan, M. A., (2001) An improved heuristic for the uncapacitated facility location problem, *International Journal of Industrial Engineering*, 8 (2), pp. 115-121.
19. Kuehn, A. and Hamburger, M. J. (1963). A Heuristic program for locating warehouse, *Management Science*, 9 (4), pp. 643-666.
20. Sule, D. R. (1988), *Manufacturing Facilities*, PWS-KENT Pub. Co., Boston Massachusetts.
21. Gen M , Syarif A and Wang X. (2002) Solving Uncapacitated Facility Location Problem by Using Hybridized Parallel Genetic Algorithm, *Proc. Of second international workshop on Intelligent Techniques for Industrial Engineering,* Beijing, pp. 34-38
22. Beasley, J. E.m (1990). OR Library: distribution Test Problem by Electronic Mail. *Journal of Operational Research Society*, 41 (11). pp. 1069-1072.
23. http://www.ms.ic.ac.uk/jub/pub

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Communication device to use acceleration sensor for the serious disabled

Y.Fukuda*, H.Tanaka*, K.Yoshimochi* and T.Ishimatsu*
* Department of Mechanical System Engineering, Nagasaki University, Nagasaki-city,Nagsaki852-8521, Japan
(Tel : +81-95-847-3842; Fax :+81-95-847-3842; E-mail:yoshio@welcome.mech.nagasaki-u.ac.jp)

Abstract: Communication devices to use acceleration sensor for the serious disabled is proposed. This device is mainly for the disabled people such as ALS or a cerebral infarction, they have communication difficulty because of physical disability. A feature of this device is that the sensor can be readily mounted on the user's body. Mounting of the sensor on the body is quite easy, not like the conventional touch sensor. Another feature is that even though physical ability of the user is serious, he can operate this device only by moving a part of his body slightly. In addition, we developed an adaptive word processing software for seriously disabled people. A feature of this software is that the arrangement and size of the button and window on the display can be optimized based on the user's physical ability. In the field test we confirmed that our communication device to incorporate this adaptive word processing software was applicable.

Keywords: Disabled, Welfare device, Communication, ALS, Acceleration sensor

1. Introduction

These days, the increase of the aged people is one of the notable social problems, and it should be noticed that many of them are disabled in some meanings. Therefore, the development of assistance equipments for these people is highly required. It should be also noticed that home care of aged people is becoming more popular than before because of worse financial situation of medical insurance policy. In fact many aged people and disabled are required to return to home.

Communication with other people is very important for the serious disabled people such as ALS (Amyotrophic Lateral Sclerosis) or cerebral infarction. These people often use the communication device specially designed for the disabled. The communication device usually includes the touch sensor and the push button as an input device. However, as user's physical ability becomes serious, it often occurs that the user cannot use the touch sensor. One reason is that the user is unable to move his body or finger to the position of the touch sensor. One typical example is a case of a ALS patient who lives in Nagasaki. Because of the lack of muscle power he lives by the help of artificial respirator. He lies on the bed all days and is not able to utter voice, and to move his body. All he can do is to move his eyes and to move legs slightly. The care giver tested him to use a touch sensor. But the result was unsuccessful.

In this paper we propose a communication device to use acceleration sensor as an input switching device. We focused on the disabled who can move a part of body slightly, even if his physical disability is serious. The communication device detects a slight motion of his body by the acceleration sensor and the sensor signal is converted to an control signal of the communication device. In addition, the word processing software for the disabled is proposed. An advantage of this software is that the input screen of the text can be readily rearranged considering the physical ability of the user. This adaptivity is preferable for the disabled user whose physical ability may verify in a near future.

We applied this communication device to the patient of ALS and a cerebral infarction, and confirmed its applicability.

2. Communication device to use acceleration sensor

Our communication device is composed of a personal computer, original word processing software, One-chip CPU (PIC), wireless alarm, remote control switch and acceleration sensor (ADXL202) which detects a motion of a user's body as shown in Fig.1.

The user wears an acceleration sensor at a part of his body, where a slight movement can be generated. When the user want send a control signal to communication device, he is requested to move this part of his body consciously. The movement of his body is detected by an acceleration sensor. One-chip CPU send the acceleration data to the personal computer through serial communication line. The computer recognizes the control signal and operates word processor, wireless alarm and programmable remote controller. The wireless alarm switch is used to call the family member to the bedside in case some emergency occurs. The programmable remote controller can operate the TV and the air-conditioner.

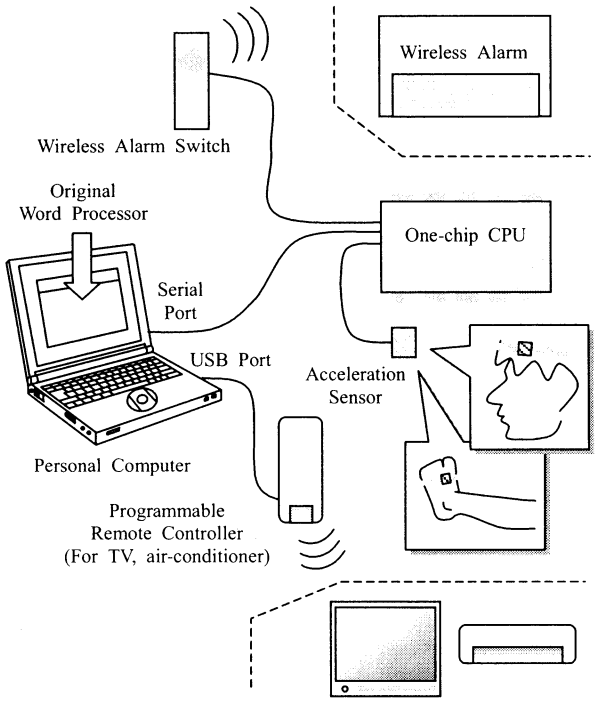


Fig.1 Communication device to use acceleration sensor

3. Adaptive communication device

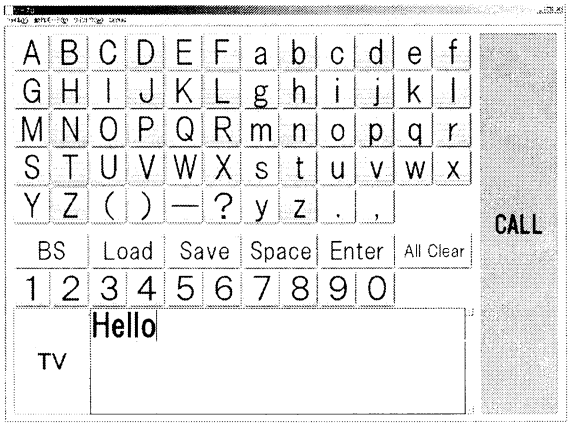
We developed a word processing software designed for the disabled to make this communication device adaptive to various disabilities of the disabled user. A feature of this software is that button arrangement and size can be optimized considering user's physical ability. Especially, this function is important for ASL patients and muscular dystrophy patients, because of their physical ability easily changes.

In addition, the user can operate two kinds of output devices, programmable remote controller and wireless alarm by the same operation as word processor.

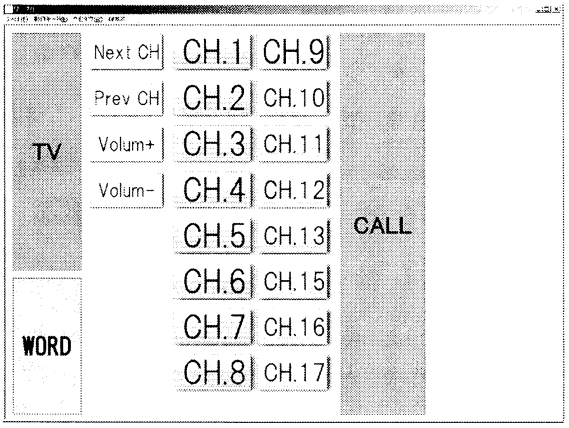
An output screen of our communication device is shown in Fig.2. To select one target button on the screen, the user is requested to send control signals twice to this software. The procedure to select one button on the screen is as follows. Fig2-1) shows the screen image at the starting point. On this screen a vertical line moves from the left to the rightward. The user can select the desired horizontal position by moving his body with the acceleration sensor. After the selection of the horizontal position, a horizontal line starts to move from the top to downward. The user can select the vertical position of the target button by moving his body with the acceleration sensor.

In the word processing mode as shown in Fig.2-1), a character or a text can be allocated to the button. In the remote control mode as shown in Fig.2-2), remote controlled devices; TV or air-conditioning machine et al. can be allocated. In the figure, TV channel are allocated.

It should be noticed that arrangement and size of those buttons can be edited by a simple computer pointing device operation.



1) Word processing mode



2) Remote control mode

Fig.2 Output screen of adaptive communication device

4. Acceleration sensor

In our communication device, we introduced an acceleration sensor as input device. By using the acceleration sensor, the care giver needs special cares about the position of input device like the touch. In case of the touch sensor, the sensor position should be not too far and not too near from the user's body. If the sensor is too near to the user's body, the sensor often emits wrong control signals. If the sensor is too far, the user is unable to

touch the sensor. But in case of the acceleration sensor the arrangement of the sensor on the body is not so sensitive to the control signals.

The acceleration data of the user's body is transmitted to one-chip CPU as PWM signals as shown in Fig.3 On this one-chip CPU, the PWM signals is changed into numerical data. This numerical data is sent to the personal computer through the serial port, and this personal computer detects the control signals as shown in Fig.4. In our communication device the control signal is recognized when the absolute value of acceleration of gravity becomes larger than the pre-specified value as shown in Fig.4, (a) (c). It is important to note that once the user moves his body toward one direction, the user's body tends to return to the original position as shown in Fig.4, (b) To cope with this problem, an interval where the detection of a movement is ignored is introduced.

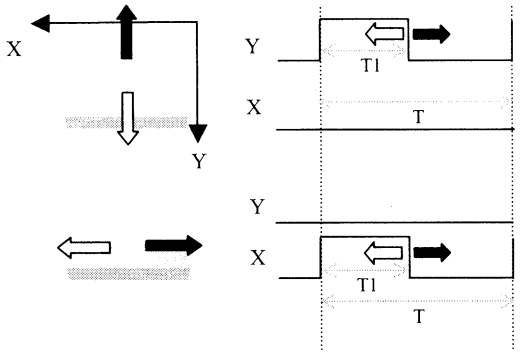


Fig.3 PWM signals from acceleration sensor

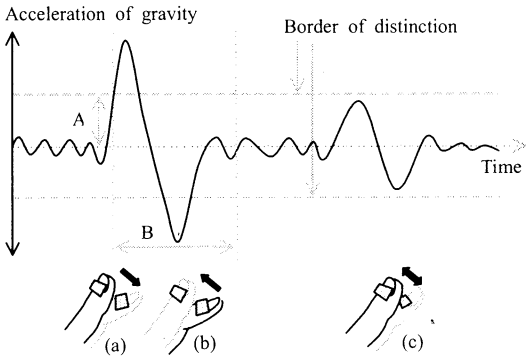


Fig.4 Detection of control signal

5. Application results

We applied our communication device to an ALS patient and a cerebral infarct patient in Nagasaki. And we evaluated this device.

(1) Case of ALS patient

First, we applied our device to an ALS patient (Fig.5). He lies on the bed all days and is not able to utter voice and to move his body. But he can move his eyes and legs slightly. We attached an acceleration sensor for input device on his tiptoe as shown in Fig.6. Although movement of his leg is very slight, the movement was detectable enough by turning up the sensitivity of the acceleration sensor as shown in Fig.7

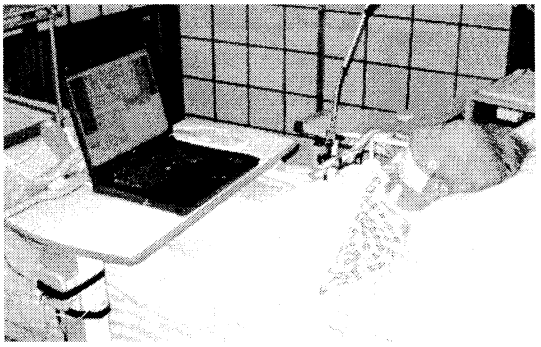


Fig.5 ALS patient with communication device

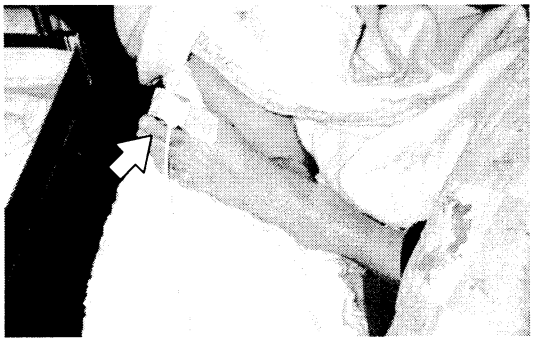


Fig.6 Acceleration sensor on tiptoe

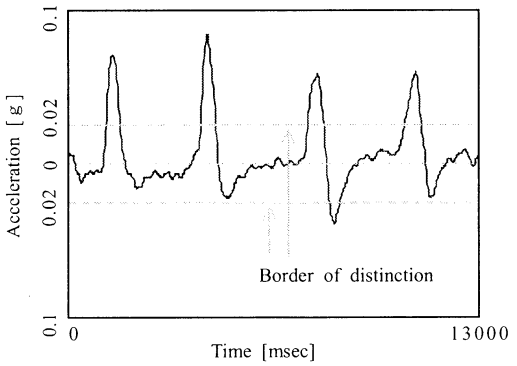


Fig.7 Actual acceleration data of tiptoe

The acceleration data in Fig.7 is obtained while the patient move his tiptoe.

(2) Case of cerebral infarct patient

Next, we applied our device to cerebral infarct patient as shown in Fig.8. He lies on the bed all days and is not able to utter voice, and to move his body too. But, he can move his head up and down slightly. We attached an acceleration sensor as input device on his forehead as shown in Fig.9. Since the movement of his head is clearly compared with the previous case, we turning down the sensitivity of the acceleration sensor to avoid an incorrect signal as shown in Fig.10. As a result, he succeeded to select the target button comfortably.

Fig.8 User with communication device

Fig.9 Acceleration sensor on forehead

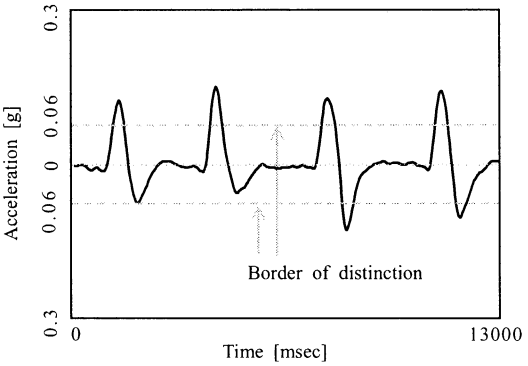


Fig.10 Accutual acceleration data of forehead

6. Conclusions

We developed the communication device to use acceleration sensor for the serious disabled like ALS and cerebral infarction patients. A feature of the communication device is that the sensor can be attached easily without special attention. In two application, the sensor is attached at the tiptoe and forehead. From these application we concluded that even though physical ability of the user is serious, our communication device is applicable. In addition we developed an adaptive word processing software considering the various situations of the disabled. A feature of this software is that button arrangement and size can be optimized.

We are now going to apply our communication device to many other disabled people.

References

[1] T.Ochiai,T.Ishimatsu (1997), Computer Input devic e for handicapped using vision sensor, Proc.3rd Int.sy mposium on artificial life and robotics,Vol.2.pp.634-63 7

[2] T.Ochiai,T.Ishimatsu (1999), Computer Input devic e for physically disabled person using head movement, Special Issue of International Journal of System Scien ce,Vol.30-1,pp.131-134

[3] H.Kawakami (1988), Human Interface for the seve rely Physically Handicapped, Proc.REIS'88, pp.15-22

[4] O.Sueda (1992), Supporting Technology for Worki ng and Independent Living of the Disabled in Japan, Proc.REIS'92, pp.47-58

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Muscle Stiffness Sensor to Control Assisting Device for Disabled

S. Moromugi*, Y. Koujina**, S. Ariki*, A. Okamoto*, T. Tanaka***, Maria Q. Feng**** and T. Ishimatsu*

*Faculty of Engineering, Nagasaki University, 1-14 Bunkyo-machi, Nagasaki City 852-8521, Japan
(Tel: 81-95-847-3842; Fax: 81-95-847-3842; E-mail: smoromug@net.nagasaki-u.ac.jp)
**DAIHEN Co., 2-1-11 Tagawa, Yodogawa, Osaka 532-8512, Japan (koujina@daihen.co.jp)
***Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan (E-mail: ttanaka@mce.uec.ac.jp)
****Department of Civil & Environmental Engineering, University of California Irvine,
Engineering Gateway, E4139 University of California Irvine, CA 92697-2175 (E-mail: mfeng@uci.edu)

Abstract: An innovative sensor was developed to control a mechanical glove for a patient who has disability on his fingers. The sensor is attached on the human body and non-invasively detects activity of a specific muscle. The patient can operate the mechanical glove and achieves the grasping function with desired force by activating the muscle on which the sensor is attached. The sensor is developed based on the fact that the muscle gains its stiffness as it is activated. The level of the muscle activity is estimated by measuring the level of the muscle stiffness. Here, the "stiffness" means the level on which the muscle is stiffened. It is different from meaning of the elastic property of material. In some researches, the similar sensors that mechanically measure stiffness information of muscle have been developed as the man-machine interfaces to control mechanical devices. However, those sensors are not robust enough against the external disturbance such as contacting with something from outside. In this study, a muscle stiffness sensor that is highly robust against such external disturbance was developed based on an original principle and reported its superior performance shown through experiments with the patient.

Keywords: Stiffness of muscle, Assist for disabled, Mechanical glove, Man-machine interface, Muscle stiffness parameter

1. Background

This study started with a request from a medical student who got disability on his body by damaging his cervical code at an accident. He is disabled on whole body under his chest and all fingers of both hands. A mechanical glove is developed to help his study at school and assist his activities in the daily life. Fig.1 shows the outline of the system developed for him. The grasping function was achieved with the mechanical glove and the muscle stiffness sensor (Moromugi et al [1]). The glove is controlled by the user's activity of a specific muscle extracted through the sensor. However, the stiffness information from the sensor was still slightly affected by the external disturbance. The more robust and accurate method to extract stiffness information of muscle from the sensor data was required to make it more reliable system.

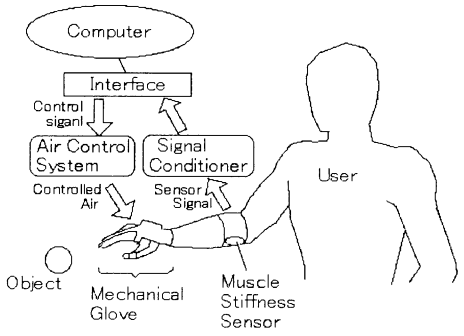
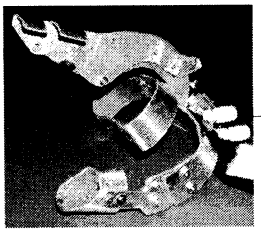


Fig.1 Outline of system



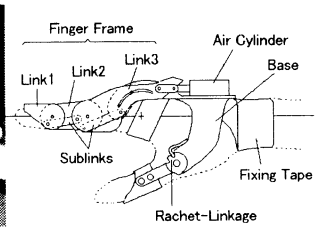


Fig.2 Mechanical glove Fig.3 Structure of mechanical glove

2. Mechanical glove

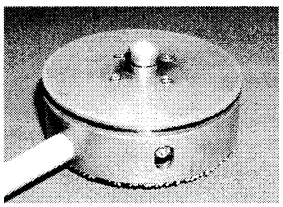
Fig.2 and Fig.3 show the developed mechanical glove and its structural sketch respectively. The grasping function is achieved by bending user's forefinger toward his fixed thumb. The mechanical glove is composed of a finger frame to bend a forefinger and a base to be attached on the hand and an air cylinder mounted on the base. The finger frame consists of three links, link 1, link 2 and link 3 as shown in the Fig.3. Each link is connected by small sub-links each other. All joints of the finger fame are driven simultaneously with the actuation of an air cylinder. The theoretical maximum value of the grasping force is 40[N] from kinematics and the capability of the air cylinder. The total weight of the device is 110[g].

3. Muscle stiffness sensor and stiffness parameter

The Electromyogram (EMG) is widely used to detect the user's intention to control wearable devices such as artificial limbs. However, the EMG is not comfortable enough to use in daily life because it is easily affected by noise and also requires annoying installations. As another trial, the sensor to mechanically measure stiffness information of human muscle was proposed and tested by several researchers. Yamamoto et al [2] developed wearable device to assist nursing operation and sensors to measure stiffness of muscle were used for controlling his device. One of the merits of the stiffness sensor is that the activities of each muscle can be measured independently. It is hard to extract activity of a small muscle located together with a lot of muscles by using EMG. Also the stiffness sensor is easily attached/detached to the human body and the sensor signals can be easily handled than that of EMG.

In this study, the sensor to measure muscle stiffness was developed with original principle. The appearance of the sensor is shown in Fig.4. This sensor has two components, a flat disk (3.8cm in diameter) and a button (6mm in diameter) in the center of the disk. The size of the button is selectable from 4, 5 or 6 mm in height. The casing is made of brass. The whole weight of the sensor without the cable is 48g. Two small pressure sensors are used to measure the force loaded on the button and the entire sensor.

Fig.5 illustrates forces loaded on the sensor. The flat dist is pushed on the skin surface and the button is indented into the skin more than the disk when the sensor is mounted on the human body. The forces loaded on the components of the sensor have the relation, $F_T = F_B + F_D$, where F_T is the total force loaded on the sensor. F_B and F_D are the forces loaded on the button and disk from the skin respectively. As the muscle under the skin gains stiffness, the button is pushed back and the ratio of F_B/F_T increases. That means that the ratio, F_B/F_T has correlation with the activity of the muscle. Considering these mechanisms, F_B/F_T was firstly defined as the parameter to represent the muscle stiffness and this parameter was used to control the mechanical glove at the early stage of this study.



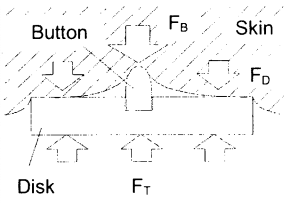


Fig.4 Muscle stiffness sensor Fig.5 Forces loaded on sensor

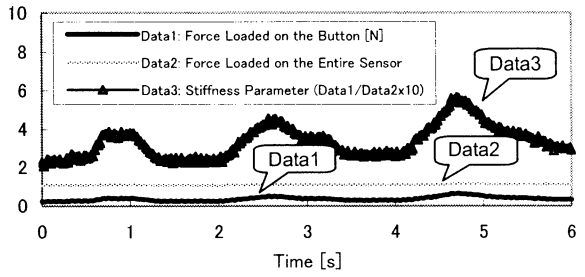


Fig.6 Measured data

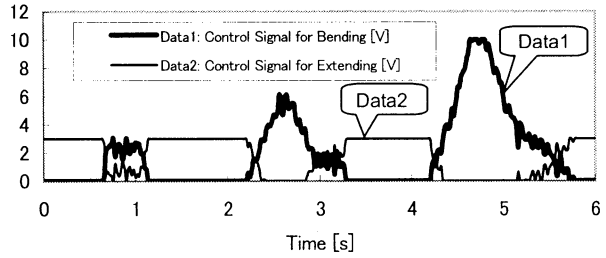


Fig.7 Control signal for valves

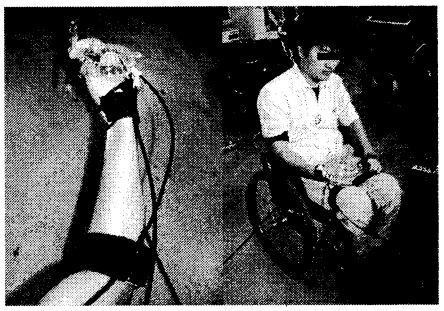


Fig.6 Experiment

4. Control of Mechanical Glove

A double-acting air cylinder is used to drive the mechanical glove thus the two pressures inside the air cylinder, to pull and push, are controlled using the stiffness parameter described in Section 3. The control algorithm is chosen as follows.

$$V_B = (S - L_S) \times G_V \quad (1)$$

$$V_E = \begin{cases} V_P - V_B & (S > L_S) \\ V_P & (S \le L_S) \end{cases} \quad (2)$$

V_B: The control signal for the valve to bend the finger.

V_E: The control signal for the valve to extend the finger.

S: Stiffness Parameter, ($10xF_B/F_T$)

L_S=3.2:The maximum stiffness to keep extending the finger.

G_V=10.0: Gain to set the sensitivity.

V_P=3.3[V]: The output signal to keep extending the finger.

Fig.6 shows the measured sensor data. The sensor is mounted on the muscle, Flexor Carpi Ulnaris, which acts for bending the wrist for this experiment. The calculated output signals from the data for the two valves are shown in Fig.7. The control signal for vending the finger increases as the stiffness does. The control signal for extending the finger keeps a constant value, V_P at the ordinary state and starts decreasing when the stiffness parameter exceeds L_S.

The patient operated the mechanical glove and picked up a pen by himself. Fig. 8 shows the photos of the experiments. The applicability of the both glove and sensor was shown through the experiment. However, it was found out that the stiffness parameter, F_B/F_T slightly fluctuates as the F_T changes especially when the muscle is highly stiffened. The measured stiffness parameter against F_T is shown in Fig.9. The stiffness parameters were measured from the patient's muscle, Flexor Carpi Ulnaris, under 4 different levels of activities. The EMG signal is also measured and used as a scale to keep the same level of the muscle activities through the measurement. In addition, the measured

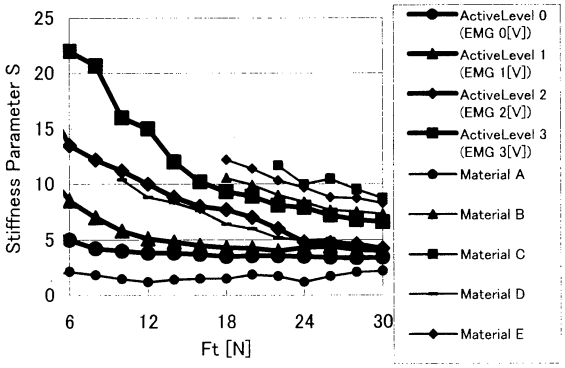


Fig.9 Measured stiffness parameter

data of several kinds of soft material are plotted for comparison in the graph. It seems that the stiffness data of both the muscle and material pieces change as F_T increase following similar tendencies.

5. Modification of stiffness parameter

To eliminate the influence of the changing F_T, the stiffness parameter was modeled experimentally in a specific range and modified based on the results. 6 through 30 [N] were set as an effective range of F_T. The stiffness parameter S was modeled as hyperbola,

$$S = \frac{a}{F_T - b} + c \tag{3}$$

where, a, b and c are parameters to decide the curve of S. These parameters were experimentally derived as functions of F_T shown as following.

$$\begin{aligned} a &= \alpha \times S + \beta \\ b &= \chi \times S + \delta \\ c &= \varepsilon \times S + \phi \end{aligned} \tag{4}$$

$$\begin{aligned} \alpha &= 0.0823F_T{}^2 - 0.0509F_T + 11.646 \\ \beta &= -0.120F_T{}^2 - 2.883F_T - 17.058 \\ \chi &= -0.0012F_T{}^2 + 0.0005F_T - 0.1641 \\ \delta &= 0.0023F_T{}^2 - 0.001F_T - 0.6718 \\ \varepsilon &= -0.0014F_T{}^2 + 0.0004F_T - 0.1951 \\ \phi &= 0.0033F_T{}^2 - 0.001F_T + 3.4682 \end{aligned} \tag{5}$$

Substituting a pair of F_T and S into the equation (4) and (5), the three parameters, a, b and c are decided. It means that one curve of S is derived by putting a pair of measured data, S and F_T into the equation (3). Now, the stiffness parameter at a specific F_T can be estimated using the curve. Through these procedure, the derived S at F_T=6[N] is defined as the modified stiffness parameter. Fig.10 shows the graph of both the modified and unmodified stiffness parameters against F_T measured under the condition that the activity of the muscle is kept constant. It is shown that the modified stiffness parameter keeps the same level under the constant activity of the muscle even if the F_T changes.

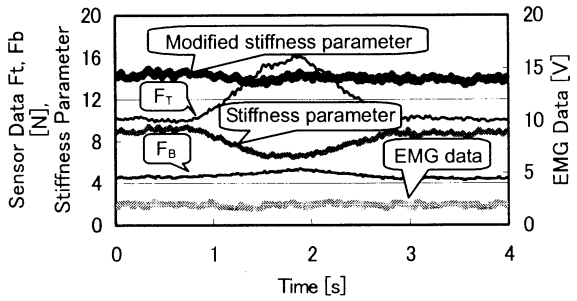


Fig.10 Modified stiffness parameter

Conclusion

The sensor to measure the level of the muscle's stiffness is developed to control the mechanical glove for the disabled. That allows the user to control the glove by activating the muscle on which the sensor is attached. It was succeeded to eliminate the influence of the external disturbance on the measurement. By using the developed sensor, the user can operate the assisting device properly and safely even if the sensor contacts to something outside. The safety and reliability are very significant factors especially for devices to be worn by human. It seems that the muscle stiffness sensor can be widely applied for controlling other kinds of assisting device as the effective man-machine interface.

Reference

[1] Moromugi S, Izumi H, Yoshimochi A et al (2002), Device for Assisting Grasping Function, Proc. International Conference on Control, Automation and Systems, Muju, Korea, pp.1250-1254

[2] Yamamoto K, Hyodo K and Matsuo T (1997), Powered Suit for Assisting Nurse Labor Employing Muscle Sensor and Sliding Rotary Actuator, Proc. 5th International Symposium on Fluid Control, Measurement and Visualization, SICE, J., Vol-1, 497-501

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

3-D Analysis of Functional Instabilities of the Ankle using Digital Still Cameras

Kang Sui Jung*, Shigeki Yokoyama**, Nobuou Matsusaka**,
Nanae.Hatano**,Takahiro.Kobayashi**,Rie.Touma**,Takakazu Ishimatsu*
*Dept. Mechanical Engng. Nagasaki Univ., Nagasaki 852-8521,Japan, E-mail:ishi@net.nagasaki-u.ac.jp
**Dept. Medical. Nagasaki Univ., Sakamoto-machi Nagasaki 852-8102,Japan

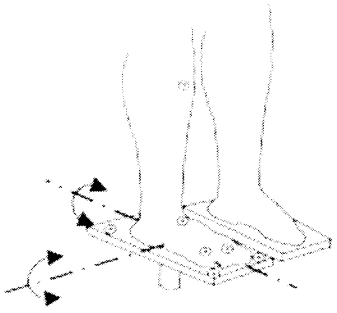
Abstract

A technique to analyze the functional instability of the ankle is proposed. The technique is based on the measurement to use two digital still cameras. Using this technique we had experiments. Subjects are requested to stand on an experimental table whose inclination angle is adjusted. Comparing the postures on both feet, we could conclude that subjects to have experience of the ankle sprain have functional instabilities on their ankles.

Key words: Functional instability, ankle sprain, image processing, digital still camera

1. Introduction

Ankle sprain is a kind of diseases those can be often found on the ankle of sports players. Functional instability of the ankle is recognized as one of the most common residual disability after an acute ankle sprain. Due to this functional instability of the ankle, recurrent ankle sprains are reported. Several causes for this functional instability have been proposed: mechanical instability, peroneal muscle weakness, and a proprioceptive deficit. The proprioceptive deficit means an improper sense of joint angle. Freeman et al.[1] proposed that proprioceptive deficit of the injured leg leads to an impaired postural control and coordination of the legs after some experiments. They measured the degree of impairment of postural control of the legs and ankles using a mechanical device. It is highly desired that the measurement employed in this experiment should be non-contact from the sensor.[2] One promising non-contact technique is the image processing technique. One problem related with the image processing is the accuracy since the conventional video camera has limited accuracy. Fortunately, high resolution digital still cameras are available as an image input device. The experiment to analyze functional instabilities of the ankle is executed as follows. Subjects are requested to stand with his left foot on an inclined table and the right foot on the other table. The inclination angle of left table is pre-specified to a certain values. The right table has two-degree of freedom. The subject is requested adjust the inclination angle of the right table so that postures of both tables coincident each other as shown in Fig.1.



(One table is fixed. The other table is free.)

Fig.1 Foot on the inclined table

However, if the subject has experiences of ankle sprains, there exists some deviation between the actual posture and the posture recognized. Difference of the posture between the right and left feet is admitted as an index to represent proprioceptive deficit or functional instabilities. The postures of the feet should be measured with accuracy. In addition, the sensing technique should be non-contact since even a tactile feeling on the

foot and leg effects the postural control of the ankle. .

In this paper we propose an image processing technique to analyze the functional instabilities of the ankle using two digital still cameras. Feature of the technique is that the posture of the leg and foot can be measured accurately without any contact. Two digital cameras are used to acquire two different images and realize the principle of stereo vision. The computer is used to analyze the image and measure the posture of the leg, foot and inclined plate. It should be noticed that the system can be readily realized with a low expense. In addition to the stereo vision composed of two digital still cameras, we also propose a system to incorporate two digital still cameras with a liquid crystal video projector. By projecting a structured light pattern onto the target object, a 3 D scanner can be readily realized. The system is also cost effective. A feature of this system is that 3-D image can be obtained accurately with low expense.

In the followings we explain how we realized a 3-dimensional measuring system and a experimental results.

2. Measuring system based on stereo vision

Three-dimensional posture of the leg is measured as shown in Fig.2. The system is composed of two digital still cameras and one computer, which executes the image processing and obtains three-dimensional data. In order to simplify the measuring, we attached special marks on the ttarget body. Every mark is a white round plate, which is easy to recognize on the target. Detection of the target position on the images is executed manually. While manual operations are required, reliable results are obtained.

Procedures of the measuring are as follows.

Step1) At the beginning of the experiment, the geometry of the measuring system needs to be calibrated. As a practical calibration technique, we put a square box inside the visible region and obtain two images from two cameras. From these two images we reads raster coordinates of six corners of the square box using an image application program. Allocating the coordinate origin at one corner and x,y and z axis along the edges of the square box, we can get the geometrical relationship easily. The technique is wellknown.

Step2) Once the geometrical relationship between the cameras and coordinates, we can obtain the three-dimensional coordinates of the focused points readily.

Three-dimensional coordinate $p=(x,y,z)^T$ of the target position can be obtained by

$$p = (B^T B)^{-1} B^T d$$

where B and d are defined using raster coordinates

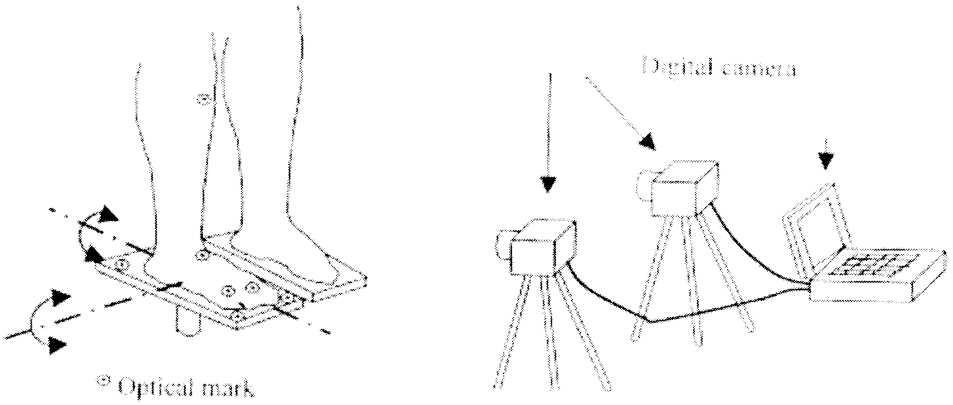


Fig.2 Measuring system to use two digital still cameras

of two cameras $(u,v)_L$, $(u,v)_R$ as follows

$$B=\begin{pmatrix} T_{31}u_R-T_{11} & T_{32}u_R-T_{12} & T_{33}u_R-T_{13} \\ T_{31}v_R-T_{21} & T_{32}v_R-T_{22} & T_{33}v_R-T_{23} \\ S_{31}u_R-S_{11} & S_{32}u_R-T_{12} & T_{33}u_R-S_{13} \\ S_{31}v_R-S_{21} & S_{32}v_R-S_{22} & S_{33}v_R-S_{23} \end{pmatrix}$$

$$d = (T_{14}-u_R, T_{24}-v_R, S_{14}-u_L, S_{24}-v_L)^T$$

Parameters Tij and Sij are geometrical parameters of two cameras determined by the calibration.

3. Measurement of posture

In Fig.3 one image of digital still camera is shown. Seven target points can be recognized. Three of those points ⑤、⑥、⑦ are allocated on the inclined table, ② is at the front side of the ankle, ③ is at the right-hand side of the ankle.① is allocated at the front side of the knee and, ③ is allocated at the right-hand side of the knee.

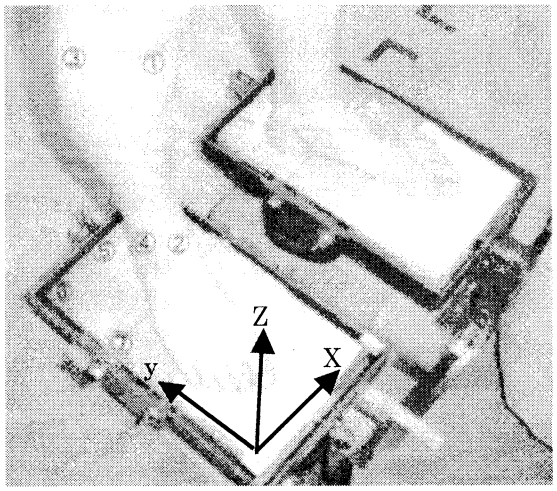


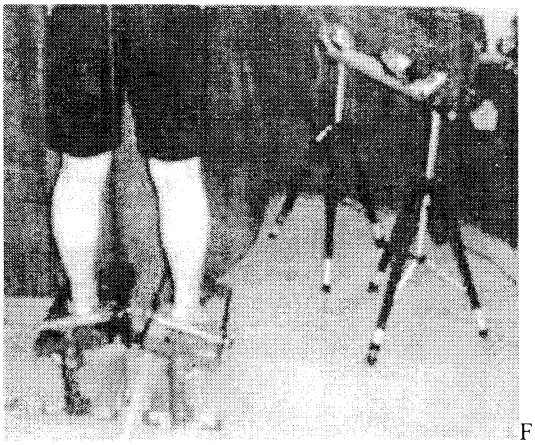
Fig.3 Seven target points on the leg

From the three-dimensional coordinates of these seven points, posture of the leg and foot is represented by the angle θ -inv (inversion angle) which is defined as inclination angle of projected line of a straight line to connect point ① and ② onto x-z plane. The other angle to represent the posture is θ -plan (planter flexion) which is defined as inclination angle of projected line of a straight line to connect point ③ and ④ onto y-z plane.

4. Experiment

In the experiment, two group of the subject are examined. Members of the first group have experiences of ankle sprain at least twice in the last six months. And members of the second group have no such experiences. Both goups have four members. Subjects are requested to stand on an plate with his left foot on an inclined table and the right foot on the other table. The inclination angle of left table is settled to various specified values in advance. The right table has two-degree of freedom as shown in Fig.1. The subject is requested to incline his right table to the inclination angle that he feels coincides to the inclination angle of the left taable. Difference between those two inclination angles is recognized as an index to represent proprioceptive deficit. All subjects stood on the table with bare-foot and were requested to close their eyes .

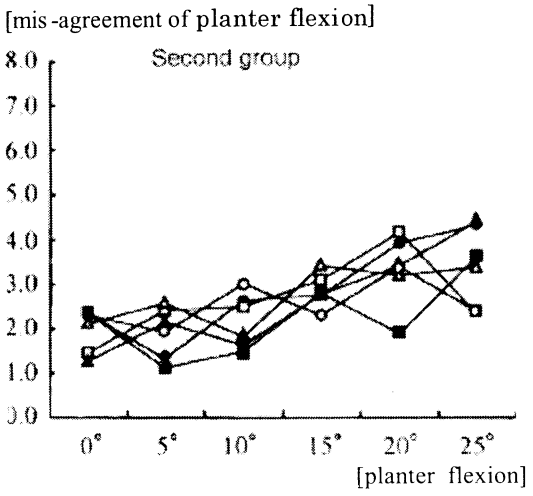
Experiments are executed. Fig.4 shows the geometry of two cameras and the subject.



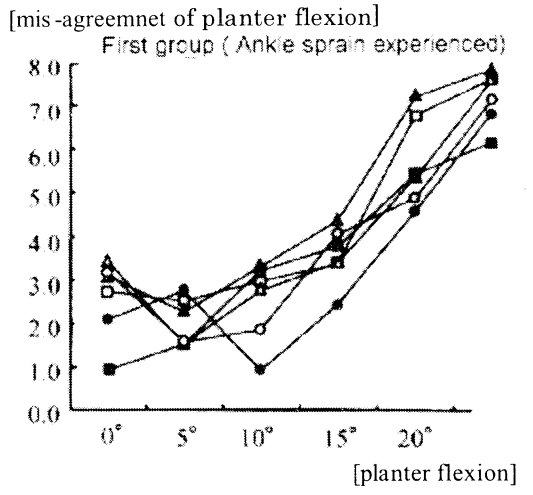
F

Fig.4 Geometry of measuring system

Experimental data are represented in Fig.5, where subjects who have experience of ankle sprain have wide range of mis-agreement of planter flexion between both ankles.



(b) Group without ankle sprain experience



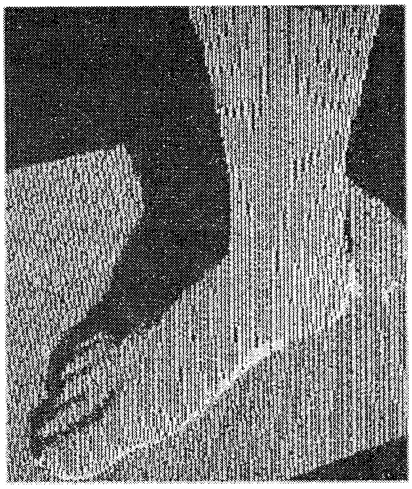
(a) Group with ankle sprain experience

Fig.5 Comparison of two groups

5. 3-D scanner to use digital still camera

We propose a technique to measure the 3-D geometry of leg and foot on the experimental table in detail. The technique is called as 3-D scanner, which is composed of one or two digital still cameras, a liquid crystal video projector and a computer[3]. Using the digital still camera as an image input device, high accuracy 3-D and low-cost measurement became possible.

In this measurement, similar procedures to that in Section two are executed. One difference is that the geometrical relationship of the video projector needs to be calibrated. In our system, a box whose size is known is settled inside the visible region. Measuring the raster coordinates of all corners of the box in the image enables to calibrate the camera system. Projecting the spot light onto the corners of the box enables to calibrate the video projector. After these calibrations, 3-D measuring starts by projecting a series of structured light onto the target. One experimental data are represented in Fig.6. Using this system, functional instabilities of the ankle could be also analyzed.



7. Conclusions

A technique to analyze functional instabilities of the ankle by using two digital still cameras is proposed. Due to the high accuracy of the digital camera, accurate data were obtained.

References

1)KB.Freeman,J.Bemstein, "The etiology and prevention of functional instability of the foot", J.Bone Joint Surg., 47B(1965)pp.678-685

2)N.Matsusaka,S.Yokoyama,"Effect of ankle disk training combined with tactile stimulation to the leg and foot", The American Journal of Sport Medicine, Vol.29, No.1, (2001),pp.25-30

3)C.Rocchini,P.Cignoni;"A low cost 3D scanner based on structured light",Proc.Euro Graphics, Vol.20, No.3,(2001)

4)C.K.Wu. and D.Q.Wang,"Acquiring 3-D Spatial Data of a Real Object",CVGIP,Vol.28,(1984), pp.126-131

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Evolutionary Control Systems with Competitive-Cooperative Neural Network for a Mobile Robot

M. Tabuse
Dept. of Computer Science and Systems Engineering Miyazaki University Miyazaki, 889-2192, Japan

T. Shinchi
The Center of Educational Research and Practices Miyazaki University Miyazaki, 889-2192, Japan

T. Kitazoe
Dept. of Computer Science and Systems Engineering Miyazaki University Miyazaki, 889-2192, Japan

A. Todaka
Dept. of Computer Science and Systems Engineering Miyazaki University Miyazaki, 889-2192, Japan

Abstract

This paper describes a new approach to control systems for an autonomous mobile robot by using sandwiches of two different kinds of neural networks. One is a neural network with competition and cooperation and used for recognizing sensor information. The other is a neural network with adaptive in the creature and used for self-learning of wheel controls. In a computer simulation, we are successful to obtain a typical type of a robot with good performance when going along a concave wall. The robot also shows robust behaviors in a real environment.

Key words: evolutionary control systems, neural network, competition and cooperation

1 Introduction

Many attempts have been focused on developing autonomous robots inspired by animals and humans which have robust adaptation and stable behavior in changing environments. One approach on this line is to use neural networks between input from senores and output to controllers and to adapt synaptic couplings in the networks to the environment. Many authors have proposed evolutionary robot control systems by using evolutionary adaptation of neural network, genetic programming and a classifier system. However, there are some parts in the brain which do not include learning procedure, just inherited from parent when born. Some stages of image processing do not need learning procedures. Stereovision neural networks, for example, do not have learning procedure since we have automatic focusing ability without training. Even if it is thought that early vision neural networks themselves have been developed through genetic algorithms, we may assume that their synaptic couplings are considered fixed when mobile control systems of robots are trained.

The strategy of the present paper is, therefore, that the neural networks are divided into two parts. One is sensor recognition processor in which sensor input data are processed with cooperation and competition, reducing noise from environment and giving definite decision for sensor data, which we call competitive and cooperative neural network (*CCNN*). The network parameters in this part are always fixed. The other is processor with self-training ability called a neural network controller (*NNC*). *NNC* processes the data from sensor recognition part and outputs motor control. The synaptic couplings are trained to adapt to the environment by means of, for instance, genetic algorithms.

The idea of neural network with competition and cooperation originated initially from a stereovision pattern recognition. The famous neural network model for stereovision was studied by Amari and Arbib[1] which is called a primitive competitive model. Reimann and Haken[2] proposed neural network with cooperation and competition. Kitazoe et al.[3] presented neural networks which were able to give stereovision recognition for moving objects. In the neural network with competition and cooperation, competition makes only one neuron active and cooperation maintains the active state. In a real robot, competition is performed among neural activities corresponding to different sensors, while cooperation takes place among time delayed neural activities of the same sensor. Thus, the robot recognizes the nearest object in surrounding environments and keeps up this recognition under small fluctuations of sensor values and behaves correctly in dynamically changing environments.

After the processing through *CCNN*, we use evolutionary algorithms for *NNC*. In general, evolutionary processes require a large population size and a number of generations. Thus, experiments for evolutionary robotics are usually carried out in computer simulations which were helpful to train and test robot control systems. In this case, the simulator must include appropriate noises as a real robot will have.

The task for a mobile robot employed in the present paper is to go along walls with concave shapes and under noisy environment. We choose this particular task not only to show how well our neural network system works under noisy environment but also for us to be able to develop the future realistic applications.

2 Two-Layer Neural Network with Competition and Cooperation

We explain a two-layer *CCNN*[3], which processes sensor values of a mobile robot. The two-layer *CCNN* equations are given as

$$\frac{d}{dt}\alpha_u^a(t) = -\alpha_u^a(t) + A\lambda_u^a - B\sum_{a'\neq a} g(\xi_u^{a'}(t)) + D\sum_{u'} g(\xi_{u'}^a(t)), \quad (1)$$

$$\frac{d}{dt}\xi_u^a(t) = -\xi_u^a(t) + f(\alpha_u^a(t)), \quad (2)$$

where t denotes an internal processing time and u is an actual external time step to control a robot, where one step of u is carried after dozens of internal steps of t done in the numerical calculations. $\alpha_u^a(t)$ is a neural activity of a first layer and $\xi_u^a(t)$ is a neural activity of a second layer. λ_u^a is an input sensor value to this neural network. $f(x)$ is a well known sigmoid function and $g(x)$ is a function given by

$$f(x) = \frac{1}{2}(\tanh(x - h) + 1), \quad (3)$$

$$g(x) = x^+ = \frac{1}{2}(x + |x|). \quad (4)$$

A, B, D, h are positive constants which are to be chosen appropriately. The neural network for these equations has a two-layer structure as shown in Figure 1. In equation (1), the third term represents a competition with other neuron activities $\xi_u^{a'}(a' \neq a)$ and the fourth term measures a cooperation with past time neural activities $\xi_{u'}^a$ ($u' = u-\ell,\ldots,u-2,u-1,u$). Let us consider the control of a miniature robot Khepera by using this neural network. Khepera has eight infrared proximity sensors (six in front and two in rear, see Figure 2). In our experiments we suppose that the neural activities related to Khepera's sensors compete with each other and the time sequence of the neural activity for each sensor cooperates with each other. Thus a indicates a sensor number (a=0,1,2,...,5). The input value λ_u^a is normalized by −1 to 1 as

$$\lambda_u^a = 2.0 \cdot \frac{S_u^a}{1023.0} - 1.0, \quad (5)$$

where S_u^a is a value of #a sensor at an actual time step u.

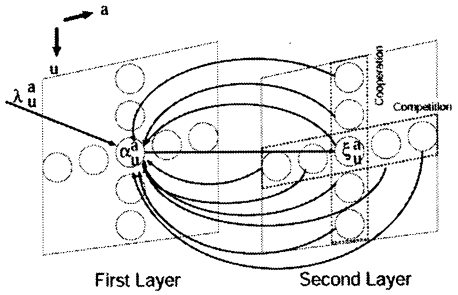


Figure 1: Two-layer neural network.

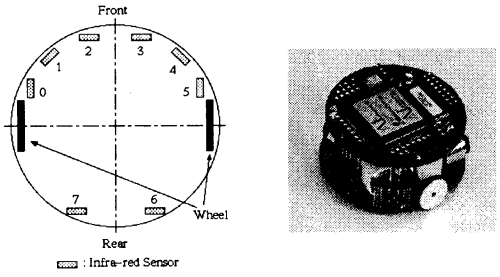


Figure 2: Mobile Robot Khepera.

When Khepera with *CCNN* is applied to a real environment, *CCNN* gives a clear cut signal processing. While each raw sensor returns value simultaneously as shown in Figure 3(a), almost only one neural activity returns high value after processing with *CCNN* as shown in Figure 3(b). Therefore, Khepera with *CCNN* has an ability to decide or choose the sensor value for which Khepera should react.

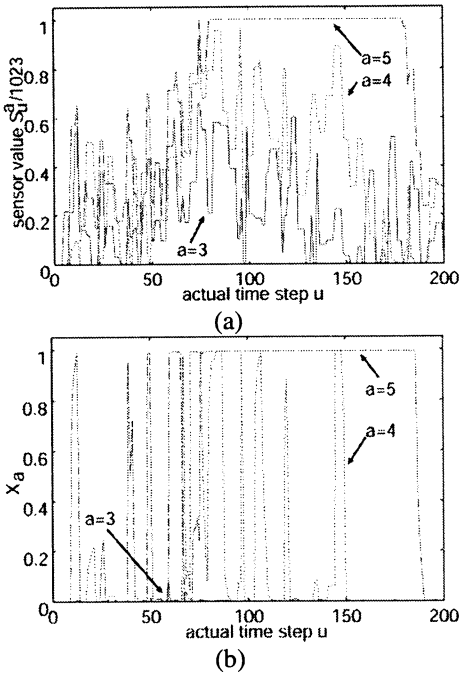


Figure 3: (a) Raw sensor values of Khepera in a real environment. $S_u^a/1023$ for a=3,4,5 are drawn as solid line, dotted line and broken line, respectively. (b) Values after processed with *CCNN* in real environment. x_a for a=3,4,5 are drawn as solid line, dotted line and broken line, respectively.

3 Evolutional Adaptation

To train robot control systems, we perform adaptation under a computer simulative model of a robot and its environment. As shown in Figure 4, the data $x_0, x_1, \ldots, x_5$ outputted from *CCNN* are fed to the second neural networks with self-training abilities. Then the synaptic couplings in *NNC* are revised by genetic algorithms. The control signals to right wheel and left wheel, R and L, and a speed of right-motor and left-motor, V_R and V_L, are given by

$$R = F(\sum_{i=0}^{5} W_{Ri} \cdot x_i + W_{R6}), \quad (6)$$

$$L = F(\sum_{i=0}^{5} W_{Li} \cdot x_i + W_{L6}), \quad (7)$$

$$F(x) = \tanh(x), \quad (8)$$

$$V_R = V_{\max} \cdot R, \quad (9)$$

$$V_L = V_{\max} \cdot L. \quad (10)$$

W_{Ri} and W_{Li} (i=0,1,...,5) are synaptic couplings as shown in Figure 4, which connect data from *CCNN* with the control of each right or left motor. W_{R6} and W_{L6} are thresholds, which adjust values of sigmoid function (8). $V_{\max}$ is a maximum speed of right and left motors.

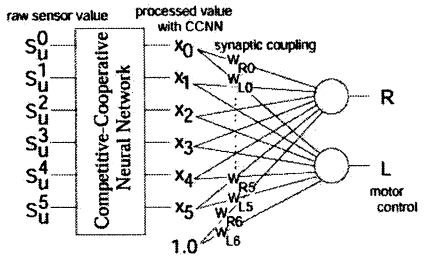


Figure 4: *CCNN* and *NNC*. The data $x_0, x_1, ..., x_5$ outputted from *CCNN* are fed to *NNC*.

We determine W_{Ri} and W_{Li} by using the genetic algorithms. The algorithms to obtain the best genes are as follows.

1) We make N_1 robots with randomly generated synaptic couplings and let them run in an area surrounded with walls for a certain time period.
2) Make new N_2 robots from old N_1 robots by using genetic algorithms in which synaptic couplings of new robots are generated with real-coded genetic algorithms[4], in which α of BLX-α is set 0.5. Then, let them run for the same period as 1).
3) Measure the whole robots $N_1 + N_2$ by a given evaluation function and choose best N_1 robots with high scores. If total score of N_1 robots exceed a given threshold, stop the loop, otherwise go to 2).

4 Experiments

We investigate movements along a wall in order to estimate capability of the control of Khepera by the combined use of *CCNN* and *NNC* with genetic algorithms. #5 sensor has a crucial role for counterclockwise movement along a wall. If x_5 is active, it is near a wall so that it should go along the wall. Since we do not need independent information of each sensor in the task of going along a wall, we can degenerate variable x_i to X_i which are defined as

$$X_0 = x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4, \quad (11)$$

$$X_1 = x_5, \quad (12)$$

where $\oplus$ means that $X_0 = x_0 + x_1 + x_2 + x_3 + x_4$ for $x_0 + x_1 + x_2 + x_3 + x_4 \le 1.0$ and $X_0 = 1.0$ for $x_0 + x_1 + x_2 + x_3 + x_4 > 1.0$. Let us substitute eqs.(11) and (12) into eqs. (6) and (7) as

$$R = F(\sum_{i=0}^{1} W_{Ri} X_i + W_{R2}), \quad (13)$$

$$L = F(\sum_{i=0}^{1} W_{Li} X_i + W_{L2}), \quad (14)$$

The evaluation function in genetic algorithms is given as

$$g = \frac{|V_R + V_L|}{2.0} \cdot (1 - \frac{|(V_R + 1) - V_L|}{2V_{\max} + 1.0}) \cdot (1.0 - \frac{1}{5}\sum_{a=0}^{4} \frac{S_u^a}{1023.0}) \cdot (\frac{V_R + V_L + 2V_{\max}}{4V_{\max}}) \cdot \frac{S_u^q}{1023.0}. \quad (15)$$

Each term in eq.(15) evaluates the robot performance from different points of view, measuring Khepera speed, a counterclockwise rotation, movements without obstacles on the X_0 side, going forward and going along a wall on the right side. Then the evaluation function g has high value if a robot goes along a wall without collision and if it runs forward as fast as possible. We set A=8.0, B=4.0, D=2.0 and $\ell = 6$ in eq.(1) and h=1.0 in eq.(3), in which the features of *CCNN* are satisfied. We take $N_1 = N_2 = 20$ and $V_{\max} = 8$. The evaluation function is calculated for each 2000 step run of a robot.

4.1 Behavior Type by Evolutional Adaptation

As a result of the evolutional adaptation in a computer simulation, coupling values of the best 5 robots are shown in Table 1 together with fitness value which is a sum of g for 2000 steps. It is interesting to see that these best 5 robots have almost the same behaviors, which are shown in Table 2. The data x_a (a=0,1,...,5) outputted from *CCNN* show that all x_a have almost zero or that one of x_a has almost one and others have almost zero. Therefore we can assume that X_0 and X_0 take 0 or 1.

Table 1: Coupling values of the best 5 robots after 100 generations.

No.	W_{R0}, W_{R1}, W_{R2}, W_{L0}, W_{L1}, W_{L2}	Fitness
1	1.06, 1.18, -0.05, -2.54, 0.09, 1.50	7685.05
2	0.96, 1.13, 0.02, -2.65, 0.15, 1.52	7679.29
3	1.02, 1.18, -0.04, -2.46, 0.01, 1.47	7636.54
4	1.11, 1.20, -0.04, -2.85, 0.56, 1.48	7623.72
5	4.80, 1.11, -0.03, -3.11, 0.65, 1.51	7620.40

Table 2: Typical behaviors of the best 5 robots in Table 1.

X_0 \ X_1	0	1
0	Sharp Right	Wide Right
1	Sharp Left	Sharp Left

4.2 Robustness of a robot behavior in noisy environment

We examine robustness of a robot behavior with *CCNN* in noisy environment. A random noise is added to the sensor value of a robot, where it takes a value ranging between 0 and the maximum value $V_{\max}$. If a sensor value is over 1023, it is set to 1023. Since the amount of noise is represented by $V_{\max}$, we measure average raw value of #5 sensor by changing $V_{\max}$. In Figure 5, we find that the average #5 sensor value of a robot without *CCNN* goes down rapidly, while one with *CCNN* keeps up high value and goes down slowly. Since the #5 sensor value represents the distance between a right-hand side of a robot and a wall, a high value of #5 sensor means a robot behaves going along a wall very well. The actual trails of a robot with and without *CCNN* under noisy environment are also shown in Figure 6(a) and Figure 6(b), respectively. A robot with *CCNN* has good behavior of going along a wall for noise $V_{\max}$=0,150,160. On the

other hand, a robot without *CCNN* fails in the movement task to go along a wall for v_{max} =150,160. These results show that a robot with *CCNN* has robust behavior in noisy environment much better than the one without *CCNN*.

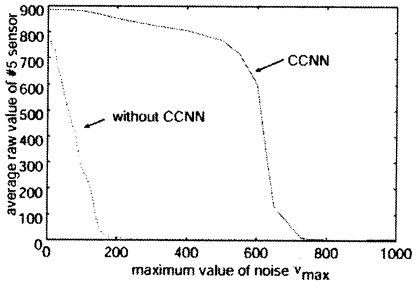
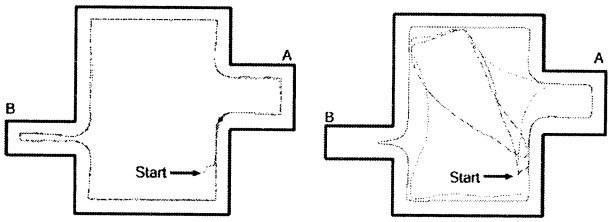


Figure 5: The average raw value of #5 sensor corresponding to the maximum value of noise v_{max}. The solid and dotted lines represent the values of the robot with and without *CCNN*, respectively.



(a) (b)

Figure 6: The trails of the robot with *CCNN* (a) and ones without *CCNN* (b), which take about 1200 simulation steps to go around. The solid, dotted and broken lines denote the trail for v_{max} =0, 150, 160, respectively.

4.3 Experiments in the Real Environment

We investigate Khepera's behaviors in a real environment by using synaptic couplings obtained from the simulation. All parameters of *CCNN* and *NNC* are set to the same values as the simulation. Figure 7 shows the real environment. Khepera is controlled by a workstation (Sun UltraSparc 450MHz) through a serial line. Figure 8(a) and 8(b) show the trails of Khepera with and without *CCNN* in the real environment, respectively. The solid line and the dotted line indicate the trail under the natural light and the light of fluorescent lamps on the ceiling. The light of fluorescent lamps fluctuates temporally, so that it gives noise to Khepera's sensors. Figure 3 shows sensor values of Khepera, whose fluctuation is the effect of the light of fluorescent lamps and reaction to the walls.

Khepera with *CCNN* goes along a wall and enters both wide and narrow spaces, *A* and *B*, very well under both the natural light and the light of fluorescent lamps. On the other hand, Khepera without *CCNN* cannot enter the narrow space *B* under both lights and doesn't go along a wall correctly under the light of fluorescent lamps. These results are similar to those in the simulation and we think Khepera with *CCNN* has a better performance both in the simulation and in the real environment.

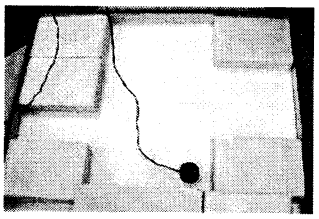
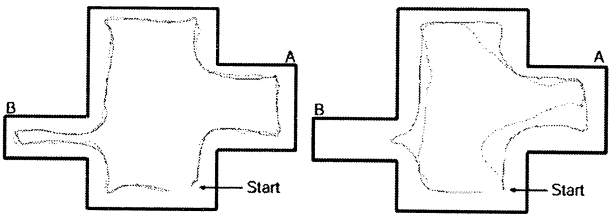


Figure 7: The real environment.



(a) (b)

Figure 8: The trails of Khepera with *CCNN* (a) and ones without *CCNN* (b) in the real environment, which take about 3000 actual time steps to go around. The solid and dotted lines show the trails under the natural light and the light of fluorescent lamps, respectively.

5 Conclusion

We have presented the control system for mobile robots by using competitive and cooperative two-layer neural network (*CCNN*) and using self-adaptive neural network (*NNC*). The competitive term makes only one neuron activity ξ get active for large input values and the cooperative term makes ξ keep up the active state for a small fluctuation of the input value. As a result, self-learning neural network created going along a wall behavior by using ξ outputted from *CCNN*. We found that Khepera controlled by *CCNN* and *NNC* had robust behaviors against noises when going along a concave wall both in a computer simulation and in a real environment.

References

[1] S. Amari, M.A. Arbib, "Competition and Cooperation in Neural Nets", *Systems Neuroscience*, pp.119-165, 1977.

[2] D. Reimann, H. Haken, "Stereo Vision by Self-organization", *Biol. Cybern.* 71, pp.17-26, 1994.

[3] T. Kitazoe , J. Tomiyama, Y. Yoshitomi, T. Shii, "Sequential Stereoscopic Vision and Hysteresis", *Proceedings of Fifth International Conference on Neural Information Processing*, pp. 391-396, 1998.

[4] L.J. Eshelman, J.D. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata", *Foundations of Genetic Algorithms 2*, pp.187-202, 1993.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Performance Evaluation System for Probabilistic Neural Network Hardware

o Noriyuki Aibe	Ryousuke Mizuno	Masanori Nakamura	Moritoshi Yasunaga	Ikuo Yoshihara
Doctoral Program of System and Information Sciences, University of Tsukuba, Tsukuba, Ibaraki. 305-8573, Japan.	Doctoral Program of System and Information Sciences, University of Tsukuba, Tsukuba, Ibaraki. 305-8573, Japan.	College of Information Sciences, University of Tsukuba, Tsukuba, Ibaraki. 305-8573, Japan.	Institute of Information and Electronics, University of Tsukuba, Tsukuba, Ibaraki. 305-8573, Japan.	Faculty of Engineering, University of Miyazaki, Miyazaki, Miyazaki. 889-2192 Japan.

Abstract

The probabilistic neural network (PNN) is one of the promising neural networks, and is now applied to some real world applications. In order to speed up the PNN calculation considerably, we have developed the PNN hardware system for video image recognition. In this paper, we develop a performance eveluation system for the PNN hardware, and the performance of the PNN hardware can not be evaluated precisely until the evaluation system is completed.

Key words: Pattern Recognition, Image Processing, Neural Networks, PNN, Evaluation System, Hardware.

1 Introduction

The probabilistic neural network (PNN)[1] gives us high recognition accuracy for practical applications such as a satellite cloud classification[2], facial image recognition[3], etc. However, it requires long computation time in the recognition and especially parameter learning. In these computation, a large number of sample patterns are required for each category, and more sample patterns are required to achieve high recognition accuracy. This is the reason why we develop a PNN hardware[4][5] by using the novel architecture (which is called sigma parallel architecture (SPA)[6]). Then, our PNN hardware works independently, it means it does not require any Host computers, and it makes the size of system to be compact (the first prototype board is 240 × 180 mm scale). However, the performance evaluation system is indispensable to evaluate the PNN hardware precisely, and the computer is useful for it. In this paper, we connect the conventional PC with the PNN hardware efficiently and provide a useful measurement environment for the PNN hardware. Using the evaluation system, the PNN hardware can be evaluated in precise, and its results can be fed back to the next design quickly.

2 Probabilistic Neural Network Hardware

PNN is based on the Bayes' theorem, thus the high recognition accuracy close to the theoretical one may be obtained as the number of sample patterns increases. Then, PNN consists of feed-forward three layers, such as kernel layer, summation layer, and decision layer. Each neuron in the kernel layer corresponds to the kernel function, where $\mathbf{X}$ is the input pattern (unknown pattern) and $\mathbf{S}_j^i$ is the jth pattern in the category C_i. The superposition of the kernel functions that is executed in the summation layer, constructs the estimator of the probabilistic density function $p(\mathbf{X}|C_i)$ of the category C_i, that is,

$$\hat{p}(\mathbf{X}|C_i) = \frac{1}{N_p} \sum_{j=0}^{N_p} K\left(\mathbf{X} - \mathbf{S}_j^i\right), \qquad (1)$$

where N_p is the number of sample patterns in each category and N_p is regarded as 1 because it can be assumed to be identical in all categories. The neuron in the decision layer selects the highest and its category C_i is output as the answer (i.e., predicted category) to the input pattern (unknown pattern) $\mathbf{X}$.

Our first prototype of probabilistic neural network(PNN) hardware is shown in Fig.1, and the specification is shown in Table.1. This prototype board is customized for video image recognition, and the purpose is to verify the circuit of the pre-processor and the PNN processor.

In this prototype, we use three FPGA chips, Xilinx XCS30XL-4PQ240C for PNN calculation, XCV800-4HQ240C for image pre-processing, and XCV300-4PQ240C for parameter learning and process control. This board works without a

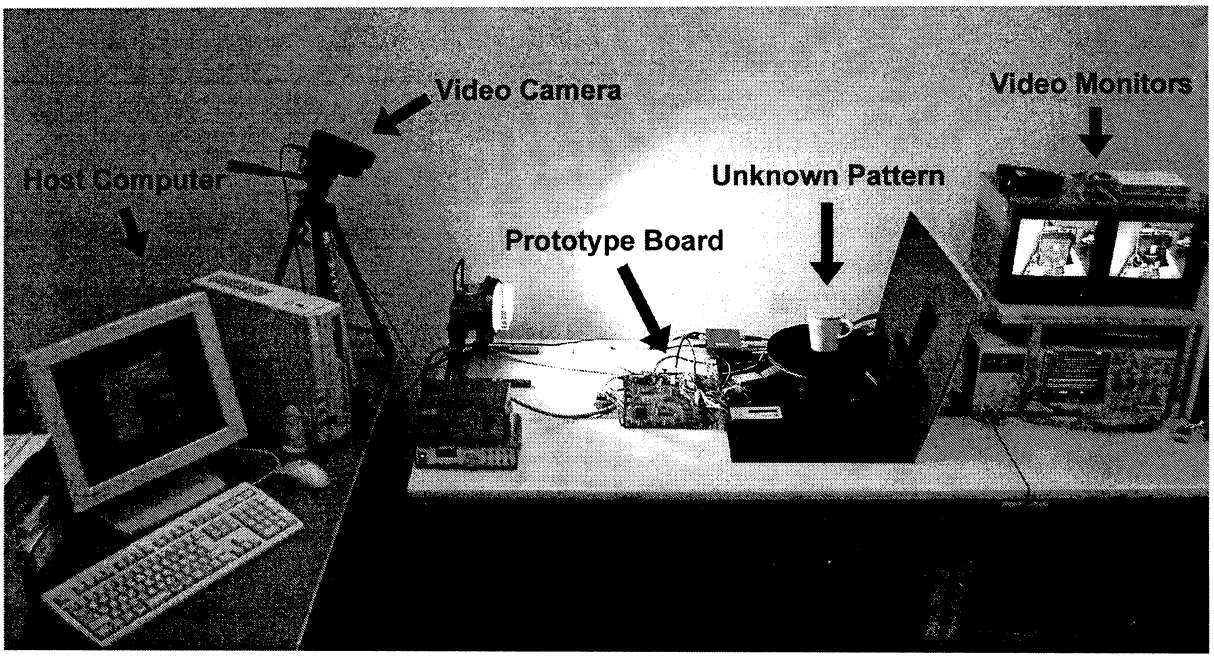


Figure 2: Photograph of Probabilistic Neural Network Hardware.

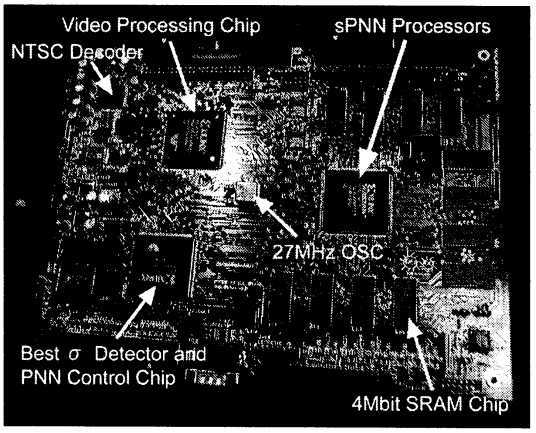


Figure 1: Photograph of First Prototype Board.

host computer, because it has NTSC video decoder/encoder, SRAM memories for sample patterns, and useful user interface such as PS/2 Keyboard Input on the board. Thus, this board can be carried on moving machines such as a car or a robot that requires real time image recognition.

3 Configuration of Performance Evaluation System

The performance evaluation system is shown in Fig.2, and its configuration is shown in Fig.3. Unknown image patterns (target patterns) from video

Table 1: Specification of First Prototype Board.

Base Clock Freq.	27MHz(PNN: 13.5MHz)
Number of PNN Proc.	3(Max)
Memory Capacity	48Mbit (4Mbit × 12)
Memory Data Width	96bit (8bit × 12)
Board Size	240 × 180 mm (4 Layer)
Interfaces	NTSC Video In/Out, PS/2 Keyboard Input, RS-232C, ATA, Analog-RGB Out.

camera are input to the pre-processor, and their feature extractions such as brightness adjustment and conversion to mosaic images are executed in it. Currently, we are developing additional feature extraction circuits using principal component analysis (PCA) in it. The unknown patterns in characteristic space are compared with sample patterns in the PNN processors. The PNN processor has a memory to store sample patterns, and it calculates the probabilistic density for each category using these sample patterns. Then the probabilistic neural network in PNN processor chooses the category that has the highest probability of the unknown pattern's category.

The host computer of the performance evaluation system is connected to three bus lines, which

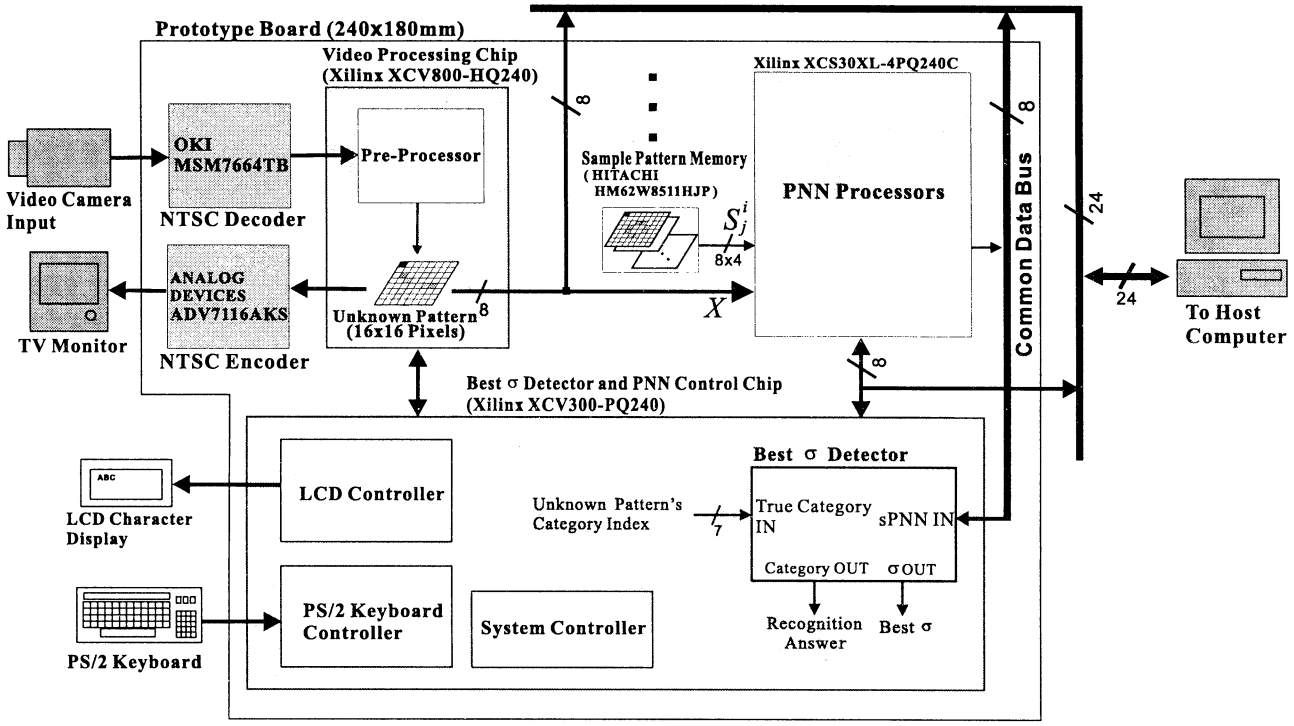


Figure 3: System Configuration.

are not related to each other on the prototype board (Fig.3). This is the feature point of this evaluation system that is considered the independent bus lines to be one bundled bus line. Each bus line consists of 8 bits, so the bit width between the PNN hardware and Host computer is 24-bit by using three bus lines (Host computer has a parallel interface card "Interface PCI-2703"). In order to achieve this technique, 1)all outputs (that the bus lines consists of) have to be try-state outputs, 2)the system must have a bus master and the other outputs are slaves. These necessary conditions make what only one output on the bus is active and the others are high impedance. Fig.4 shows the instruction and data out timing in the case of data transfer from the prototype board (Board) to the host computer (Host). The first and second signals show data bus and sync line state, the third and fourth signals show master output states (In this case, we decided that Host computer (Host) is the master.), the fifth and sixth signals show slave output states. At first, the master outputs some instructions and sync signals, then the slave, which is selected in the instruction, outputs some data and sync signals. By this technique, we have overcome the problem of the pin insufficient.

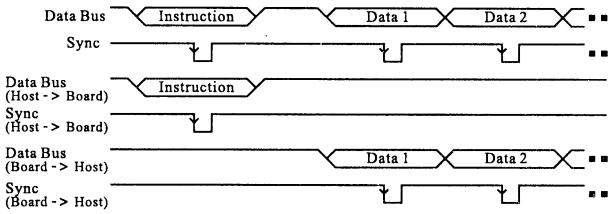


Figure 4: Instruction & Data Output Timing.

4 Control Instructions on Host Computer

We make a set of 256 instructions for the system. Some of them for controlling the PNN hardware are listed in Table.2. The operation code length is thus 8-bit. All 8-bit operation code is send by the host computer (Master), but the operation land is not always send by it. The graphical user interface (GUI) on Host computer is developed to send/receive instructions/data (Fig.5). The GUI is developed using MS-Visual C++ on MS-Windows 2000. The GUI helps us to send the instructions to the pre-processor, and it also shows the data from them visually as well as in text format (In Fig.5, a mosaic image from the pre-processor is shown in one window and its text formatted data is also shown in another window).

Table 2: The parts of instructions.

Operation	Opcode	Operand Length(bit)
Data Receive Mode 1 (Board -> Host)	21h	-
Data Receive Mode 2 (Board -> Host)	22h	32
Data Transmit Mode 1 (Host -> Board)	23h	-
Data Transmit Mode 2 (Host -> Board)	24h	32
Data Address Counter Set	26h	32
Data Address Counter Read	27h	-
Number of Category Set	29h	32
Number of Sample/Cat. Set	2Ah	32
Data Address Jump to Top Sample of Next Category	2Bh	-
Data Address Jump to Top Sample of the Category	2Ch	32

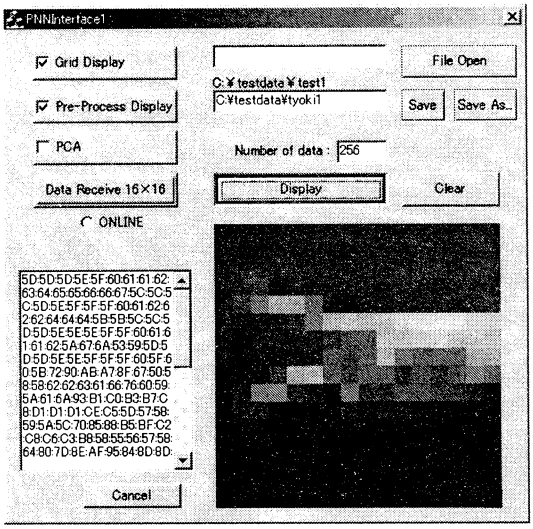


Figure 5: Control Interface on Host Computer.

5 Conclusions

We developed a performance evaluation system for PNN hardware. We proposed a new bus-line technique to measure the multiple circuit blocks without additional bus-line to them. With this technique, performance of the PNN hardware can be measured precisely and quickly overcoming the pin-bottle-neck problem. This tecknique is useful for other performance evaluation systems, e.g. the evaluation system for multi processor computer. The evaluation system has some advantages mentioned above, However, it can be only applied to the system having bus-shared mechanism, currently.

Acknowledgements

This research is supported in part by the 2002 grand of Japan Society for the Promotion of Science (Grant No.13450163, 14015206).

References

[1] D.F.Specht, "Probabilistic Neural Networks", *J,Neural Networks*,Vol.3, pp.109-118,1990.

[2] Bin Tian, Mahmood R.Azimi-Sadjadi, Thomas H.Vonder Haar, and Donald Reinke, "Temporal Updating Scheme for Probabilistic Neural Network with Application on Satellite Cloud Classification",*IEEE Trans. Neural Networks*,Vol.11, No.4, pp.903-920 2000.

[3] K.Z.Mao, K.-C.Tan, and W.Ser, "Probabilistic Neural-Network Structure Determination for Pattern Classification", *IEEE Trans. Neural Networks*, Vol.11, No.4, pp.1009-1016, 2000.

[4] N.Aibe, M.Yasunaga, I.Yoshihara, and J.H Kim, "A Probabilistic Neural Network Hardware System Using A Learning-Parameter Parallel Architecture", *Proceeding of IJCNN2002 on IEEE WCCI2002*, CD-ROM, Honolulu, Hawaii, May.11-17, 2002.

[5] N.Aibe, M.Yasunaga, and I.Yoshihara, "Self-learning Probabilistic Neural Network Hardware Using Reconfiguable LSIs", *AROB 6th '01*, Vol.1, pp.89-92, Tokyo, Japan, Jan.15-17, 2001.

[6] N.Aibe, M.Yasunaga, "An Architecture for Parallelized High-speed Probabilistic Neural Network Calculation and Its Application to Image Recognition Systems", *IPSJ Trans. HPS*, Vol.43, No.SIG6 (HPS5), pp.206-218 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A Multi-modal Neural Network with Single-state Predictions for Protein Secondary Structure

H. ZHU
Graduate School of Engineering, Miyazaki University, 1-1 Gakuen Kibanadai Nisi, Miyazaki city, 889-2192, Japan

I. YOSHIHARA
Faculty of Engineering, Miyazaki University, 1-1 Gakuen Kibanadai Nisi, Miyazaki city, 889-2192, Japan

K. YAMAMORI
Faculty of Engineering, Miyazaki University, 1-1 Gakuen Kibanadai Nisi, Miyazaki city, 889-2192, Japan

M. YASUNAGA
Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki, 305-8573, Japan

Abstract

Prediction of protein secondary structure is considered as an important step towards elucidating its three-dimensional structure, as well as its function. We have developed a multi-modal neural network for predicting protein secondary structure. Our work includes two steps: first, several neural networks are developed to predict a single secondary structure state of proteins independently: α-helix, β-stands and coil; second, a multi-modal neural network is developed to combine the single-state prediction to obtain an overall prediction on three states. The multi-modal neural network gives an overall accuracy of 68.0% when using seven-fold cross-validation on a database of 126 non-homologous proteins. Moreover, majority decision is introduced to each network of single-state prediction for the higher accuracy, which improve the overall accuracy to 68.7% with corresponding Matthews' correlation coefficients C_H=0.58, C_E=0.50.

Keywords: Multi-modal neural network, protein secondary structure, single-state prediction, majority decision

1 Introduction

Proteins are large, complex molecules made up of amino acid sequences. The amino acid sequence is called its primary structure. Different regions of the sequence form local regular secondary structures, such as α-helices and β-strands. Since the increasing of the number of sequences in public database is much faster than our ability to solve their structures experimentally, the prediction of protein secondary structure from the primary amino acids sequence is considered as a very challenging task, and the problem has been approached from several angles. In particular, many different neural networks have been applied to this task, such as NNPREDICT [1], PHD [2] and PSIPRED [3]. The traditional prediction method is based on a local input window of consecutive amino acids. The window slides along the sequence and the corresponding output is defined to be the secondary structure of the center residue of the input window.

One of the characteristics of our work is to predict the three secondary structure states independently. The results of them are input into a multi-modal neural network to get an overall prediction. In order to improve the prediction accuracy, majority decision is introduced to the single state prediction. When predict the single-state of the proteins, several neural networks do the prediction in parallel and the results are determined by majority rule. By using majority decision, the overall accuracy of the multi-modal neural network can be improved to 68.7%.

2 Data for prediction and prediction accuracy

2.1 Compiling data for experiments

We used the data information recorded in HSSP (homology-derived structures of proteins) files [4] to be experiment data for prediction, which can be downloaded from

ftp://ftp.ebi.ac.uk/pub/databases/hssp/.

Several data information are recorded in one HSSP file as following:

1. Amino acid sequence. There are 20 kinds of amino acids, which are represented by 20 capital letters.
2. Secondary structure. The secondary structure of the protein is most often assigned based on the hydrogen bond. By DSSP [5], eight kinds of secondary structure classes are distinguished. These eight classes are often grouped into three states: H=helix, E=strand and L=coil structure. Typically, H includes H (α-helix), G (3_{10}-helix) and I (π-helix). E includes E (extended strand) and B (residue in isolated b-bridge). L includes T (turn), S (bend) and (blank=other). Because HSSP adopted secondary structure assignment in DSSP, the classification in HSSP is the same.
3. The profile of multiple sequence alignment, which gives the relative frequencies of each kind of amino acid at each position of the protein chain.

When using neural networks to predict the

secondary structure, the estimation of their performance is greatly influenced by the choice of protein database. To avoid the misleading result of homologous proteins, protein sequences are usually required to have a low pair-wise identity. We adopted a database of the 126 non-homologous protein sequences used by Rost and Sander, 1993 [2]. This is a set of non-redundant PDB [6] chains. In this set, for the chains with a length of more than 80 residues, the mutual pair-wise similarity is less than 25%. The total number of residues is 23333, in which 32% H, 23%E, 45%L are included.

Table 1. The database of non-homologous proteins

1	1acx	1azu	1bbp_A	1bds	1bmv_1
	1bmv_2	1cyo	256b_A	2aat	2ak3_A
	2alp	3ait	6acn	8abp	8adh
	9api_A	9api_B			
2	1cbh	1cc5	1cdh	1cdt_A	1crn
	1cse_I	1eca	2cab	2ccy_A	2cyp
	3blm	3cla	3cln	4bp2	4cms
	4cpa_I	4cpv	5cyt	6cpa	6cpp
	6cts	7cat_A			
3	1a45	1dur	1etu	1fc2_C	1fdl_H
	1fkf	1fnd	1fxi_A	1g6n_A	1gd1_O
	1gp1_A	1iqz	2fox	2gbp	2gls_A
	2gn5	3ebx	5er2_E	6dfr	
4	1gdj	1hip	1il8_A	1l58	1lap
	1lmb_3	2hmz_A	2i1b	2lhb	3hmg_A
	3hmg_B	3icb	4grl	5hvp_A	5ldh
	6hir	7icd	9ins_B		
5	1mcp_L	1ovo_A	1paz	1pyp	1r09_2
	2ltn_A	2ltn_B	2mev_4	2orl_L	2pab_A
	2pcy	2phh	3pgm	4pfk	5lyz
	9pap				
6	1mrt	1ppt	1rbp	1rhd	1s01
	2mhu	2rsp_A	3rnt	3sdh_A	4rhv_1
	4rhv_3	4rhv_4	4rxn	4sgb_I	7rsa
7	1bks_A	1bks_B	1sh1	1tgs_I	1tnf_A
	1ubq	2sns	2sod_B	2stv	2tgp_I
	2tmv_P	2tsc_A	2utg_A	2wrp_R	3tim_A
	4ts1_A	4xia_A	6tmn_E	9wga_A	

Furthermore, to exclude a potential dependency of evaluated accuracy on the particular test set chosen, we use seven-fold cross-validation testing to estimate the performance of the method. The 126 protein sequences are separated into seven groups. Six groups are used to train the neural networks and the remaining group is used for testing. It is necessary to do the test cyclically seven times until each group of proteins is used once for testing. The average value over all seven tests shows a reasonable estimation for prediction.

2.2 Measure of accuracy for prediction

Several measures of prediction accuracy have been suggested to estimate the performance of the prediction methods. The most common measure is the overall accuracy on three states Q_3 defined as the ratio of the total number of correctly predicted residues to the total number of residues in the database [2]:

$$Q_3 = \frac{1}{N}\sum_{i=1}^{3} p_i * 100\ \% \qquad (1)$$

where, c_i is the number of the residues predicted correctly in state i (i=H, E or L), and N is the total number of residues in the database. Q_i gives the percentage of correctly predicted residues in state i:

$$Q_i = c_i / N_i * 100\% \qquad (2)$$

where, N_i is the number of observation residues in state i. Another complementary measure of prediction accuracy is the Matthews' correlation coefficients for each type of predicted secondary structure [2]:

$$C_i = (p_i n_i - u_i o_i)/\sqrt{(p_i+u_i)(p_i+o_i)(n_i+u_i)(n_i+o_i)} \qquad (3)$$

where p_i is the number of correctly predicted residues in assigned state i; n_i is the number of those correctly predicted residues in not assigned state i; u_i is the number of underestimated residues and o_i is the number of overestimated residues in state i. The closer this coefficient is to a value of 1, the more successful the method for predicting a residue in state i.

3 Neural networks for prediction

3.1 Neural networks for single-state prediction

The first step of our work is to develop several single-state neural networks to predict three states of secondary structure independently.

The structure of the single-state neural network for H state is shown in Figure 1. The networks for E and L are same. The input layer is constructed of a string of local consecutive residues. The window width w is set to be 13 (w is 5 in the figure 1). The profile of the multiple sequence alignment is used to represent amino acid residues, which can be taken from a HSSP data bank. For 20 kinds of amino acids, the frequency of occurrence of each amino acid at the position in the alignment is computed. These 20 numbers are used to represent one residue of the window. Moreover, additional information about the alignment, conservation weight, is used. The conservation weight is defined by placing a higher weight on positions that are particularly well conserved. The conservation weight can also be taken from a HSSP data bank. In Figure 1, each square of the input layer represents one residue of the window, thus, it includes (20+1) numbers. Therefore, for w=13, the whole input pattern of the first layer extends to (20+1)*13. The output layer includes one node, which is supervised by the secondary structure of the central residue of the window. In Figure 1, the assigned structure is simplified to be only two statuses: H and not H (* is not H). The window is shifted residue by residue along the sequence. However, because the target output of the network is defined to be the central residue of the window, there will be no sufficient residues in both terminuses of each protein chain. In order to smooth over the default of the insufficient part, in the part of extending over the terminus, the 20 numbers to represent amino acids are set to be 0 and the conservation weight is set to be 1.

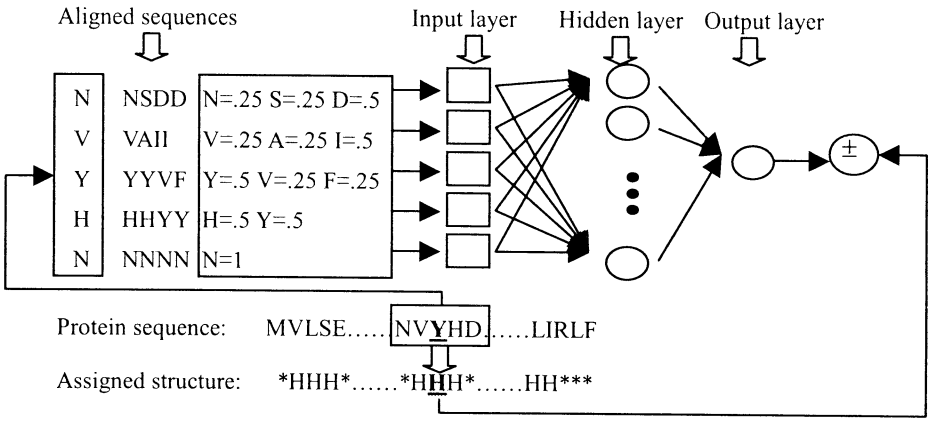


Figure 1. A three-layer neural network for single-state prediction

3.2 Multi-modal neural networks for combination

To get the overall prediction, the second step of our work is to develop a multi-modal neural network (MNN) to combine the single-state predictions.

The structure of the MNN is the same as Figure 1., except that there are three notes in the output layer, which are supervised by the three states of the assigned structure of the central position of the window. The input layer is constructed of a segment of assigned secondary structure of proteins. The window width w is also set to be 13. Three secondary structure states are represented by three binary values as following: H →(1 0 0); E →(0 1 0); L →(0 0 1). Therefore, the whole input pattern becomes 3*13.

The purpose of developing a MNN is to combine the single-state predictions to calculate the overall accuracy. In the procedure of combination, the result of single-state prediction are arranged in order of H, E and L to define the input pattern of the test group, and the network will give the overall prediction of the three states. According to our work, when the assigned structure of the protein sequences of the test group is input, which is not for prediction, but for test, the network can give the corresponding structure state with 100% accuracy. That means the higher the single-state prediction accuracy is, the better overall prediction the network can give.

3.3 Enhancement of MNN with Majority Decision

In order to improve the single-state prediction, majority decision is introduced, i.e., several neural networks are used to predict the single-state of proteins independently and the final result is decided by majority decision. We call the MNN with majority decision the enhanced MNN.

Figure 2 gives a sketch map of majority decision for H-state prediction. E-state and L-state predictions are the same. SNN means a neural network for single-state prediction. The total number of SNN n is set to be 5. The results of all SNN are only two statuses: 0 and 1, namely, H or not H. We add the results and judge whether the sum is larger than (n/2) or not. If the sum is larger than (n/2), the final result is 1, i.e., the state in the position is predicted to be H. Otherwise the result is 0, i.e., not H.

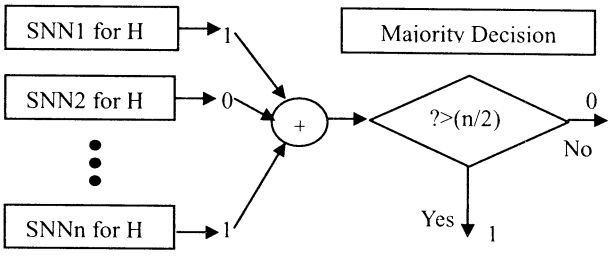


Figure 2. Majority decision for single-state prediction

The majority decision has been proved to be an effective tool to improve the prediction from the view of probability. Moreover, since the initial connection weights of the neural networks are taken randomly, by doing the prediction several times independently, majority decision can greatly reduce the contingency of prediction. Therefore, using majority decision is also advantageous to improve the stability of prediction.

4 Model validation using real data

In the experiment, the 126 non-homologous protein sequences shown in Table 1 are predicted by using both MNN and enhanced MNN to estimate the performance of the neural networks. According to the seven-fold cross-validation testing, the experiments are repeated seven times. Figure 3 and Figure 4 give the prediction accuracy of MNN and enhanced MNN respectively. And the main accuracies of the enhanced MNN are compared with the MNN in Table 2. As we see, by using majority decision, the Q_3 is improved from 68.0% to 68.7%. Although 0.7% is not a great value, the accuracies of Q_H and Q_E increase greatly. The average correlation coefficients of C_H and C_E are improved to 0.58 and 0.5 respectively. Moreover, we note that the deviation values to the average accuracies

of our methods is about $\pm 3\%$, which shows that according to different data, prediction by our methods are stable.

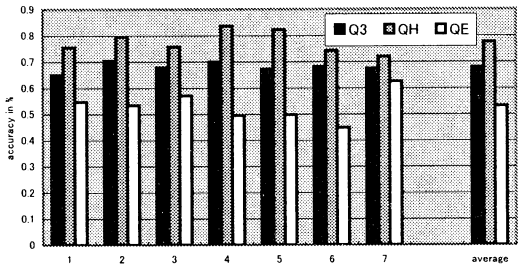


Figure 3. Prediction accuracy of MNN

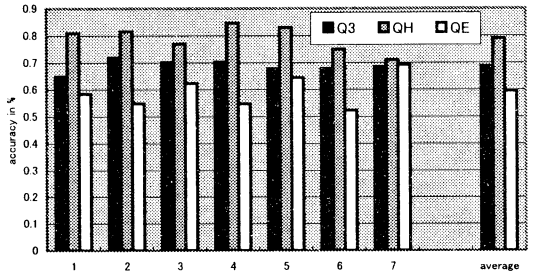


Figure 4. Prediction accuracy of enhanced MNN

Table 2. Comparison of the enhanced MNN with the MNN

	Q_3 (%)	Q_H (%)	Q_E (%)	C_H	C_E
MNN	68.0	77.5	53.1	0.57	0.47
Enhanced MNN	68.7	79.0	59.4	0.58	0.50
	+0.7	+1.5	+6.3	+0.02	+0.03

Moreover, we compare the accuracy of our methods with the claimed accuracy of conventional prediction methods. Figure 5 shows the comparison of their overall prediction accuracy Q_3. The accuracy of our methods: the MNN and the enhanced MNN are better than GOR1 [7], GOR3 [8], NNPREDICT [1] and SIMPA [9].

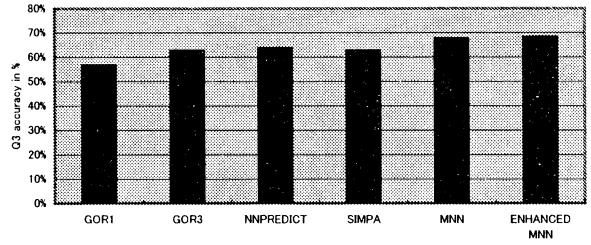


Figure 5. Comparison with other methods

5 Conclusion

We developed a multi-modal neural network to predict protein secondary structure. There are two steps in our work: first is to develop neural networks for single-state prediction; second is to develop a MNN to obtain the overall prediction. To enhance the prediction ability of MNN, majority decision is introduced to the single-state prediction. By introducing majority decision, the overall accuracy of the enhanced MNN is improved to 68.7%.

We compared our MNN and enhanced MNN with some other prediction methods. Our method is better than the conventional methods such as GOR3, NNPREDICT and SIMPA. But they still have not exceeded 70%, which is a traditional barrier of prediction accuracy upon three states.

The neural networks show a great potential for predicting the secondary structure of proteins. As we believe the MNN involves strong possibilities for better prediction, we are continuing to work to achieve greater accuracy.

Acknowledgements

This research is supported in part by the 2002 grant of the Ministry of Education, Culture, Sports, Science and Technology in Japan (Grant No. 14015206).

Reference

[1] Kneller, D. G., Cohen, F. E. and Langridge, R., "Improvements in protein secondary structure prediction by enhanced neural networks", *J. Mol. Biol.*, Vol. 214, pp. 171-182, (1990)

[2] Rost, B. and Sander, C., "Prediction of Protein Secondary Structure at Better than 70% Accuracy", *J. Mol. Biol.*, Vol. 232, pp. 584-599, (1993)

[3] David, T. Jones, "Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices", *J. Mol. Biol.*, Vol. 292, pp. 195-202, (1999)

[4] Sander, C., Schneider, R., "Database of homology-derived structures and the structural meaning of sequence alignment", *Proteins: Struct. Funct. Genet.* Vol. 9, pp. 56-68, (1991)

[5] Kabsch, W. and Sander, C., Dictionary of Protein Secondary Strcutre: Pattern Recognition of Hydrogen Bonded and Geometrical Features, *Biopolymers,* Vol. 22, pp. 2577-2637, 1983

[6] Bernstein, F. C., et al., "The Protein Data Bank: a computer based archival file for macromolecular structures", *J. Mol. Biol.,* Vol. 112, pp. 535-542, (1977)

[7] Garnier, J., Osguthorpe, D.J., Robson, B., "Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins", *J Mol Biol.*, Vol. 120, pp. 97-120, (1978)

[8] Gibrat, J. F., Robson, B., Garnier, J., "Further developments of protein secondary structure prediction using information theory", *J Mol Biol.*, Vol. 198, pp. 425-443, (1987)

[9] Levin, J. M, "Exploring the limits of nearest neighbor secondary structure prediction", *Protein Engineering*, Vol. 10, pp. 771-776, (1997)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

3-D Perception for Monochromatic Surface by Self-organization Neural Network

Xijun Hua
Faculty of Engineering
Miyazaki University
Miyazaki, 889-2192, Japan

Yibing Tang
Faculty of Engineering
Miyazaki University
Miyazaki, 889-2192, Japan

Masahiro Yokomichi
Faculty of Engineering
Miyazaki University
Miyazaki, 889-2192, Japan

Tetsuro Kitazoe
Faculty of Engineering
Miyazaki University
Miyazaki, 889-2192, Japan

Abstract

Stereo matching is one of the most active research topics in computer vision. In this paper, the stereo correspondence problem for two images of monochromatic surface is considered. Even if there exist some hints, it is not easy to reconstruct the correct 3-D scene from two images because it is ill-posed problem. We modify our previous competitive and cooperative neural network model so that we can percept monochromatic surface which is enclosed by two vertical stripes. The modification consists of two factors: (1) combination of the parameterized multiple inputs (similarities), (2) extending the cooperative terms of neural network equation. Effect of the proposed model is examined by experiments with artificial and real images. For the real images, a segmentation method is proposed to deal with the similarity maps.

Keywords: stereo vision, stereo correspondence, disparity, neural network

1. Introduction

Stereo vision is a direct and passive method of obtaining 3-D structure of the visual world which makes it attractive for applications like robot navigation and object localization. Stereo matching, which is to find corresponding points in two or more images of the same scene is the most important and difficult issue of stereo vision and it is an ill-posed problem. The computational theory of stereo vision was first proposed by Marr and Poggio [1] [2]. Recently, many algorithms such as feature-based, area-based, phase-based, multi-view stereo vision, neural network and data fusion have been researched.

However, in stereo vision, it is not easy to reconstruct 3-D scene for a monochromatic surface by stereo disparity computation. There is no clue to make correspondence between stereo image pairs for the monochromatic surface. But human visual systems are capable of perceiving 3-D scene for a monochromatic surface depending on the mechanism of visual psychological completion("filling-in" phenomena) as shown in Fig.1. The 3-D sketch map viewed from Fig.1 is shown in Fig.2.

left right

Fig.1 Random dot stereogram

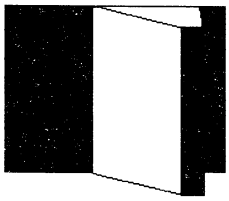


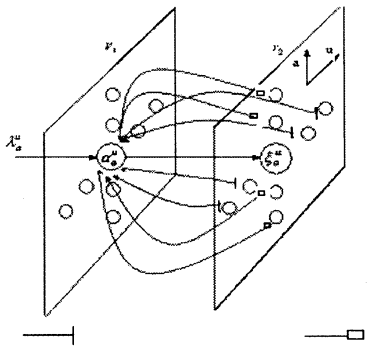
Fig.2 3-D map from Fig.1

Though various algorithms for the correspondence problem have been proposed, the importance of competitive and cooperative neural network was shown through interactions mediated by excitatory and inhibitory axon collateral pathways in rat visual cortical slice[3]. It has been suggested that cells which respond to similar disparities might reinforce one another, while cells with different disparities might inhibit one another (Julesz 1971, Marr and Poggio 1976, Blake and Wilson 1991). Amari and Arbib (1977) proposed neural network model for the stereo matching [4]. Poggio (1985) solved the ill-posed problem by the application of regularization theory[5]. Reimann and Haken (1994) proposed a new mathematical model to solve the correspondence problem by a dynamical self-organizing process [6]. Kitazoe et al. (1998) researched sequential stereo vision with hysteresis in neural network [7].

In this paper, we use two layered self-organization neural network model to simulate the competitive and cooperative interaction of binocular neurons. We define a special similarity function as input based on the stereo images. In order to reduce the region with high similarity in the image-disparity space, we consider similarity maps

in the several reference frames parameterized by ε in which two images are to be fused and we obtain a narrow region by taking logical AND calculation of multiple similarities maps. We extend the neural network equations by adding two cooperation terms so as to take broad cooperation from neighboring region. In our previous work [8] [9], we have made a perception for the monochromatic surface enclosed by random dots. In order to research the recognition problem of monochromatic surface, such as mobile robotic vision for white corridor, we choose monochromatic surface enclosed by two vertical stripes to research. Effect of the proposed model is examined by experiments with artificial and real images. A segmentation method is proposed to deal with the input similarity maps of real images and also compared with the method of Marr and Poggio.

2.Neural Network Model by Self-organization



Excitatory connection Inhibitory connection

Fig.3 Neural network with two layers (1D model)

The neural network model is a competitive, cooperative network shown as Fig.3. a represents disparity, and u represents pixel position. In the neural network, the membrane potentials of binocular neurons corresponding to each possible disparity interact each other and specific neurons win by self-organizing process, resulting in a stereo disparity perception.

The neural network equations are written as Eq.(1) - Eq.(3). α_{ab}^{uv} is membrane potential of neuron. ξ_{ab}^{uv} is output membrane potential of the neuron. A, B and C are positive constants which have to be chosen appropriately.

$$\tau_1 \dot{\alpha}_{ab}^{uv} = -\alpha_{ab}^{uv} + A\lambda_{ab}^{uv} - B\sum_{a' \neq a} g(\xi_{a'b}^{uv}(t)) + C \sum_{u' \in U_u, v' \in V_v} g(\xi_{ab}^{u'v'}(t)) \quad (1)$$

where U_u and V_v denote certain neighborhood of u and v respectively .

$$\tau_2 \dot{\xi}_{ab}^{uv}(t) = -\xi_{ab}^{uv}(t) + f(\alpha_{ab}^{uv}) \quad (2)$$

$$\lambda_{ab}^{uv} = \frac{1}{2} - \left| l(u - \varepsilon a, v) - r(u + (1-\varepsilon)a, v + b) \right| \quad (3)$$

here $\tau_1 << \tau_2$, $g(u)$ is a function given as:

$$g(u) = \begin{cases} u, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (4)$$

and $f(x)$ is a sigmoid function. For simplicity, we consider the case where the two eyes are located horizontally so that we can neglect "b" in Eq.(1), Eq.(2) and Eq.(3).

In Eq.(1), the input data λ_{ab}^{uv} are given in formula (3). Where $l(u, v)$ and $r(u, v)$ are the intensity of left and right images, respectively. α_{ab}^{uv} has inhibitory connection for a competition(the third term) and excitatory connection for a cooperation(the fourth term). We should note that the input data λ_{ab}^{uv} contains a parameter ε $(0 \leq \varepsilon \leq 1)$. When ε =0 (or ε =1), the similarity of two images is calculated based on the left (or right) image plane. The case where 0< ε <1 corresponds to their intermediate image plane.

3. Disparity Perception Simulation of Surface Enclosed by Two Stripes

3.1 Extending the Cooperation Effect

In the neural network (shown in Fig. 3), we take the cooperation effect of neighbor region into account, so as to make the cooperation of neural network more stronger and the competition more weaker relatively. We modify the neural net equations by adding two cooperation terms (the D and E terms) as:

$$\tau_1 \dot{\alpha}_{ab}^{uv} = -\alpha_{ab}^{uv} + A\lambda_{ab}^{uv} - B \sum_{a' \notin [a-k:a+k]} g(\xi_{a'b}^{uv}(t)) + C \sum_{u'=u-l}^{u+l} g(\xi_{ab}^{u'v}(t))$$
$$+ D(t) \sum_{a''} \sum_{u'=u-l}^{u+l} g(\xi_{a''b}^{u'v}(t)) + E \sum_{v'=v-m}^{v+m} g(\xi_{ab}^{uv'}(t)) \quad (5)$$

where $a'' \in [a-k : a-1] \cup [a+1 : a+k]$.

Here, the parameters are chosen appropriately as follows: τ_1 =0, k=1, l=2, m=2, A=1.0, B=0.83, C=0.23 . E=0.20

If the parameter D is not small relative to C, the convergence of α and ξ becomes slow and a few ξ's survive for some (u, v)'s. Thus we choose such D decay exponentially as formula (6).

$$D(t) = C\, exp(-t/100) , \quad (6)$$

here t denotes the time variable.

3.2. Disparity Perception of Surface Enclosed by Two Artificial Stripes

We discuss the monochromatic surface between two vertical stripes. The similarity maps for several ε 's are depicted in Fig.4. In Fig.4, two strips whose width is 3 pixels are located between the 28th pixel and the 75th pixel. From its similarity map of ε =0 and ε =1(see Fig.4 -a and Fig.4 -b), it is difficult to compute the disparity by the neural network since there are larger area of same dots than the previous example. So we also consider the similarity map of ε =1/2(Fig.4 -c). The action of ε =1/2 is to make the peaks dissolving gradually by simulating the relative fusion of left and right images in front of observer's eyes, when two images like Fig.1 are viewed. The final AND map of ε =0, ε =1 and ε =1/2 is shown as Fig.4 -d.

The dynamic evolution of ξ_a^u (t) calculation is shown in Fig.5. The stereo graph of final ξ is shown as Fig.6. The decline disparity surface simulates the monochromatic surface between two stripes well.

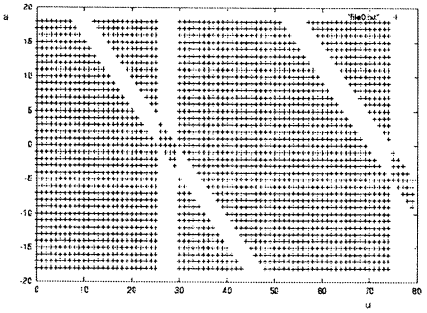


Fig.4 –a Similarity map (ε =0)

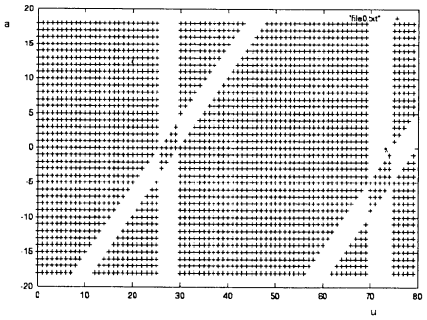


Fig.4 -b Similarity map (ε =1)

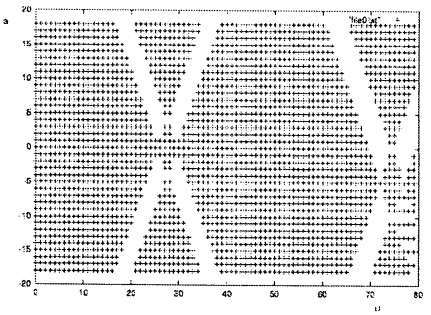


Fig.4 -c Similarity map (ε =1/2)

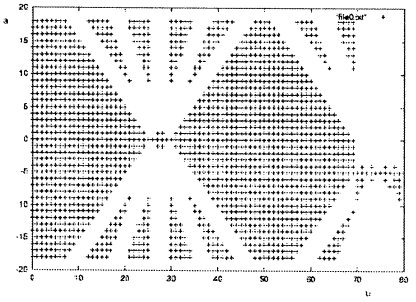
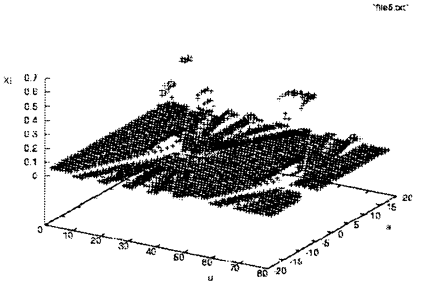
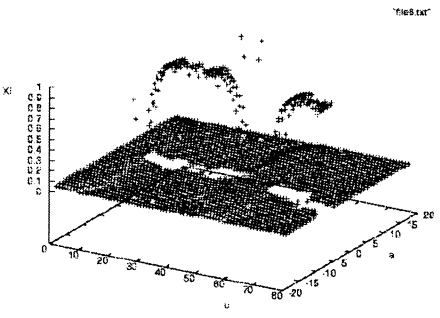


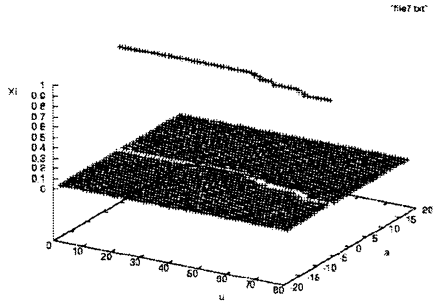
Fig.4 -d Similarity map
(ε =0 AND ε =1 AND ε =1/2)



(a) ξ_a^u (20)



(b) ξ_a^u (50)



(c) $\xi_a^u(500)$

Fig.5 The process of single line ξ calculation

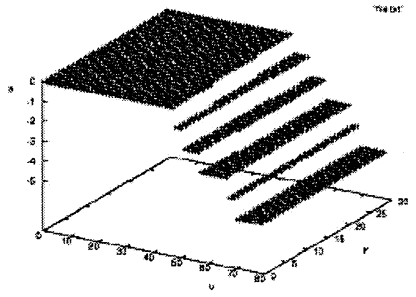
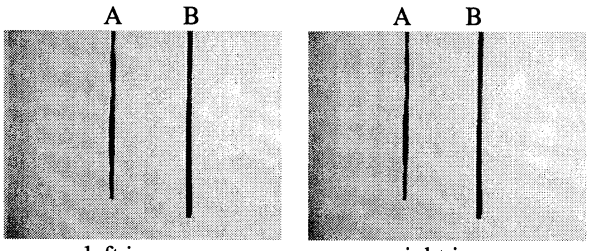


Fig.6 ξ stereo graph

4. Experiment for Real Images

We set two black stripes as marking lines on an white inclined wall in order to localize the white wall (shown in Fig.7). The left and right images are acquired by two CCD cameras. The line B is nearer to the cameras than the line A. It is necessary to keep two cameras located horizontally and the epipolar lines of two cameras parallel. The size of camera images is 320×240.



left image right image

Fig.7 Real images

In the formula (7), the data of R, G, B is gotten through the real images. The element values of similarity map vary in the range [0, 1] because the grey values of image pixels change in the range [0,255]. If we use this similarity map for neural net calculation straightly, the competition term is always depending upon the range of *a*. So the relation between the cooperation and the competition of neural net is also sensitive to the range of *a*.

$$L(u,v), R(u,v) = (R+G+B)/(255\times 3) \qquad (7)$$

For the real images in which the intensity value of different pixels is always varying in some range, we propose a segmentation method to deal with the similarity map. We use the formula (8) to filter the similarity map so as to get binary similarity map. The filtering result is shown in Fig.8. In the formula (8), the parameter S is very important to get a satisfactory similarity map. In the left and right images, there are small radiance difference for the same points in the real scene due to the effect of photographing condition. If S is too large or too small, there may be miss-correspondence or multiple correspondence problems appeared. Here, S is chosen as 0.7.

$$\lambda = \begin{cases} 1, & \lambda_i > S \\ 0, & \lambda_i \le S \end{cases} \qquad S: \text{Threshold gate} \qquad (8)$$

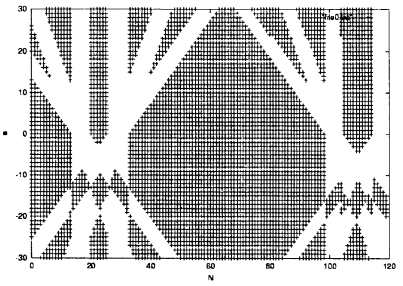


Fig.8 Similarity map

For simplicity we select the middle area (120×20) of real image to discuss. In Fig.9, the decline surfaces is well represented by the disparity perception result of the monochromatic surface between two stripes.

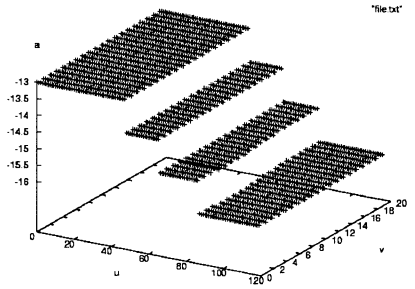
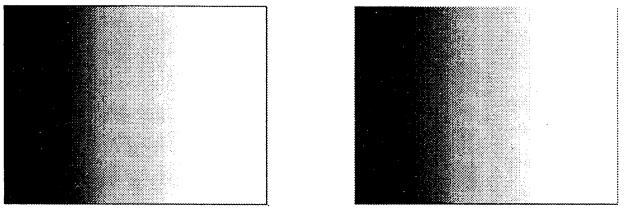


Fig.9 ξ stereo graph

Next we try to make a comparison with the feature-based method of Marr and Poggio. The algorithm proposed by Marr and Poggio for solving correspondence problem is to obtain a primitive description(such as zero-crossings) of the intensity changes present in each image, and then to mach these descriptions [1]. However, for the

surface in which pixel intensity varies in linear rule (shown as Fig.10), because there exists no zero-crossing with the LOG edge detection method, the two stereo images can not be matched efficiently by the Marr-Poggio-Grimson method[10]. But with our method, the similarity map can be obtained as shown in Fig.11. The ξ of the surface is obtained successfully as shown in Fig.12, in which a dense disparity map is gotten.



left right

Fig.10 Surface with no zero-crossing

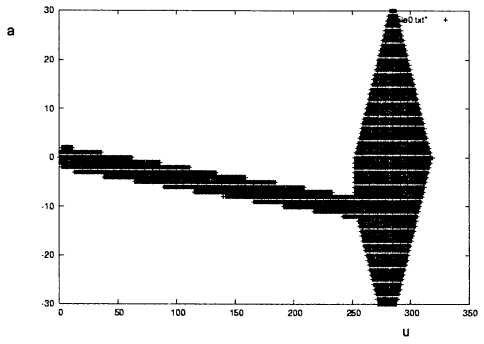


Fig.11 Similarity map

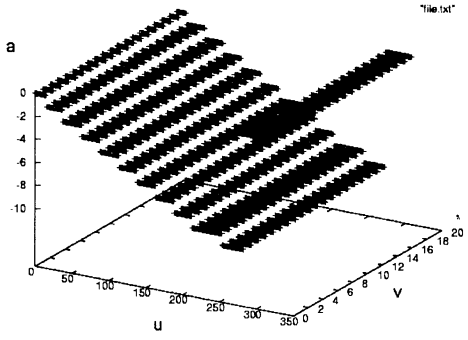


Fig.12 ξ stereo graph

5. Conclusion

We use two layered self-organization neural network model to make disparity perception for monochromatic surface. The combination of multiple similarities is used efficiently. We extend our previous neural network model by extending the cooperation of neighbor region. The calculation process of neural network shows that the depth perception of the monochromatic surface proceeds by getting a cooperation from edge region which has a correct depth perception and also by erasing incorrect depth map region through a competition. As a result we are successful to get stereo disparity perception for monochromatic surface enclosed by two vertical stripes. The experimental results for real images show that the monochromatic surface between two black vertical stripes is recognized efficiently. For the real images, it is necessary to use the segmentation method to deal with the similarity maps. In the future, we will continue to research the stereo correspondence problem with new similarity maps of different R, G, B stereo images. The neural network perception approach proposed in this paper are hoped to be improved and applied in mobile robot stereo vision for monochromatic surface environments.

References

[1] D.Marr, T.Poggio(1976), Cooperative computation of stereo disparity, Science, 194,pp.283-287

[2] D.Marr, T. Poggio(1979), A computational theory of human stereo vision, Proc.R.Soc.Lond.B.204, pp.301-328

[3] Keisuke Toyama , Manabu Tanifuji(1996), Imaging a computational process in the visual cortex, Neural Networks,9-8,pp.1351-1356

[4]S.Amari, M.A.Arbib(1977), Competition and cooperation in neural nets. Systems Neuroscience, pp.119-165

[5] Tomaso Poggio, Vincent Torre, Christof Koch(1985), Computational vision and regularization theory, Nature, .317-26,pp.314-320

[6] D.Reiman, H.Haken(1994), Stereo vision by self-organization, Biological Cybernetics, Springer-Verlag , pp.17-26

[7] T.Kitazoe, J.Tomiyama, Y.Yoshitom(1998), Sequential stereoscopic vision and hysteresis, Proceeding of 5th International Conference on Neural Information Processing, pp. 391-396

[8]Xijun Hua,Tadashi Mitsugi, Yibing Tang, etal(2002), Stereo disparity perception for monochromatic flat slope based on neural net dynamical model. Proc. of the seventh international symposium on artificial life and robtics(AROB7th), pp.606-609.

[9]Xijun Hua,Yibing Tang, Masahiro Yokomichi , et al,(2002) Stereo Vision for Monochromatic Surface Recognition Based on Competitive and Cooperative Network, Proc. of international conference on control, automation and systems, pp. 259-263

[10]W.Eric. L.Grimson(1985), Computational experiments with a feature based stereo algorithm. IEEEPAMI 7(1) pp.17-34

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Quantitative Comparison of Defect Compensation Schemes for Multi-Layer Neural Networks with Flip-Link Defects

Kunihito YAMAMORI
Faculty of Engineering
Miyazaki University

Kin'ya TAKAHASHI
Graduate School of Engineering
Miyazaki University

Ikuo YOSHIHARA
Faculty of Engineering
Miyazaki University

{yamamori@cs,takahasi@taurus.cs, yoshiha@cs}.miyazaki-u.ac.jp

Abstract

This paper addresses quantitative performance of defect compensation schemes for multi-layer neural networks. We compared following three defect compensation schemes; Partial Retraining (PR), whole network Back-Propagation (BP) retraining and Fault-Tolerant BP (FTBP) training. The recognition rate of training patterns and test patterns on XOR problem are used to evaluate these three defect compensation schemes after defect compensation.

keywords: Neural network, Defect compensation, Flip-link defect, Fault-tolerance, Wafer scale integration

1 Introduction

Multi-Layer neural networks have been used for many applications in recent years. However large scale problems which need hundreds or thousands neurons make the training time unacceptable. Neurons in a neural network can work in parallel when they receive required signals, parallel computing has also been investigated[1, 2]. In recent years, wafer scale integration (WSI) has been attracted since neural networks consist of neurons which have the same structure. In addition, WSI can reduce the communication overheads among processing elements. Since WSI uses a silicon wafer as a system, some defect compensation mechanisms are required.

Tan[3] proposed fault-tolerant BP (FTBP) training that tries to obtain fault-tolerant weights through the training process for 0-stuck weight defects. Khunasaraphan[4] introduced a weight shifting technique to compensate the 0-stuck weight defect. Tohma[5] showed that defective neural networks can be compensated by the whole network retraining. In these researches, only the 0-stuck weight defect is considered as broken link defect model because no signal has come from the broken (open) link. However, this assumption should not be applied for recent digital VLSIs because these devices have high input impedance and there is no guarantee that signal '0' always comes from the broken link.

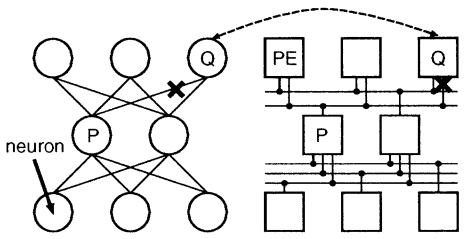


Figure 1: Parallel implementation model of multi-layer neural network.

In this paper we propose a new defect link model called on 'flip-link defect'. In this model signals through the broken link change 0 or 1 at random. To compensate the flip-link defect, three defect compensation methods based on BP algorithm are employed; Partial Retraining (PR), whole network BP retraining and FTBP training. These methods are evaluated on the training pattern recognition rate to discuss on the performance of defect compensation, and the generalization ability after (re)training.

2 Defect Compensation Schemes

Multi-Layer neural networks have some inherent parallelisms. In this paper we assume the neuron parallel model that each neuron is implemented in a processing element (PE) on a wafer as shown in Figure 1. In this model, the PEs for the neurons on the same layer can work in parallel.

In this paper we focus on the broken link defect as shown in Fig.1 that illustrates the link between the hidden neuron P and the output neuron Q is broken. Previously the broken link in the neural network is addressed as that

any signal through the broken link becomes 0. However recent digital VLSI with high input impedance does not guarantee that the signals through the broken (open) link are always 0. On the contrary, the signal from the open link is addressed as 'unsettle'. So we introduce a new broken link defect model called as "flip-link defect" model. Signals through the broken link change to 0 or 1 at random on the flip-link defect model. In the following discussion, we assume the neural network has one or two broken links with the flip-link defect. And we also assume that all links between PEs are bidirectional.

2.1 Definition of the Defect

When BP training has successfully finished, the neural network guarantees Equation (1) ,

$$\max_{i,p} \sqrt{\left(x_{i,p} - t_{i,p}\right)^2} < \varepsilon, \qquad (1)$$

where ε is the acceptable error level, $x_{i,p}$ denotes the output of the output neuron i for the training pattern p, and $t_{i,p}$ denotes the training signal of the training pattern p for the output neuron i. In this paper we define a defect of the neural network when Equation (1) is not satisfied by any reason after training.

2.2 Whole Network BP Retraining

Whole network BP retraining was proposed by Tohma[5], and this method is the simplest way to compensate the defective neural network. In this method, the defective neural network is trained from the beginning. Through the training process, weights will be adjusted to reduce the effects from the defects. This is very simple, but takes long retraining time.

2.3 Partial Retraining

Partial retraining is introduced by an author[6]. Figure 2 shows an example of the broken link defect between the neuron M and the neuron N. In this case only the neuron N is affected by the broken link. It means that this defect could be compensated by adjusting the weights belonging to the neuron N by 2-layer BP algorithm. In Figure 2 the network consisting the neuron N and the hidden neurons is called as a partial network.

If a hidden neuron has a defective input from a broken incoming link or a broken input neuron, PR is applied on this hidden neuron. After retraining, however, the activation values of retrained hidden neuron will be different from the original values of the defect-free network. This difference may lead large error on the output neurons. Then partial retraining is applied again on all output

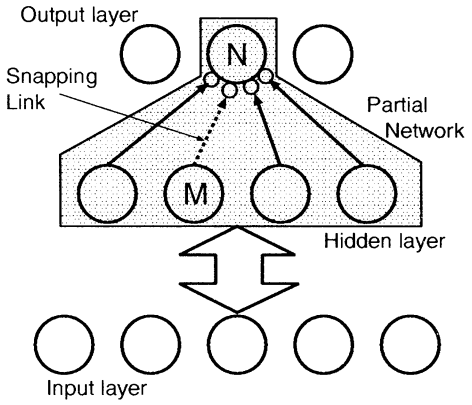


Figure 2: An example of a broken link defect.

neurons because all output neurons receive the modified activation values from the retrained hidden neuron.

2.4 Fault-Tolerant BP Retraining

Fault-Tolerant BP (FTBP) retraining was proposed by Tan[3]. Whole network BP retraining and PR adjust the weights after defects have occurred. On the other hand, FTBP training tries to obtain robust weights for some kinds of defects in advance.

FTBP training minimizes the Equation (2),

$$E'_p = \sum_f \sum_p \sum_i \left(x_{fip} - t_{ip}\right), \qquad (2)$$

where x_{fip} denotes the output of the output neuron i for the training pattern p on the neural network including the defect f. This means that the FTBP training minimizes the both errors from the difference between the output of neural network and the training signals, and from the defects. As a result, the neural network has robust weights for the assumed defects at the FTBP training process. Because FTBP training assumes that this training process is done on the defect-free system, it has to be executed on conventional computer system as a software process.

3 Experiments and Discussions

3.1 Conditions of Experiments

For the quantitative evaluation among three defect compensation methods for multi-layer neural networks, XOR problem is employed. Simulation conditions are showed as follows;

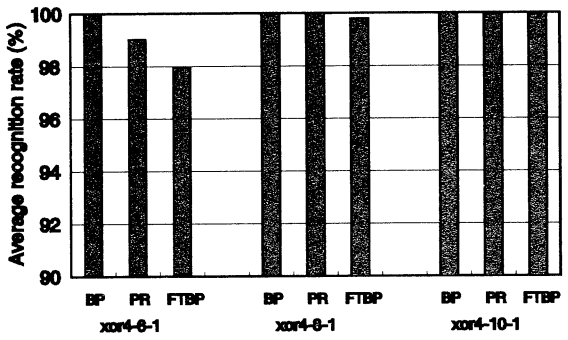


Figure 3: Average training pattern recognition rate at the last training iteration on the neural network with a single flip-link defect.

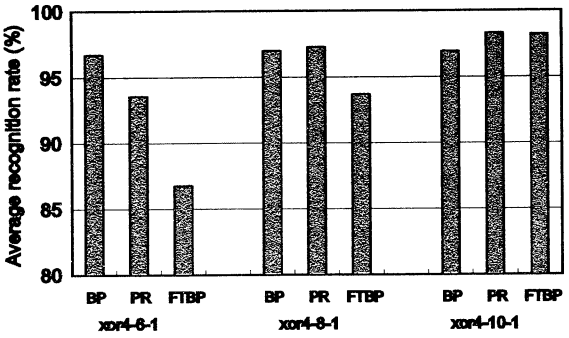


Figure 4: Average training pattern recognition rate at the last training iteration on the neural network with two flip-link defects.

Network size: 4-6-1, 4-8-1, 4-10-1.
$L - M - N$ means the network size that L input neurons, M hidden neurons and N output neurons. The inputs are multiplexed because the influence of defect on the link between an input neuron and a hidden neuron is too large. For example, the input (0, 1, 0, 1) is presented for the neural network as the input (0, 1). In other words the use of four input neurons are redundancy for the broken link between the input neurons and the hidden neurons.

Maximum iterations: 10,000 epoch.

Acceptable error level: $\varepsilon \leq 0.1$.

Defect model: Flip-link defect.

Number of defects: 1 or 2.

Number of simulations: 50 sets. Since BP retraining and PR select the defective links at random, $5 \times {}_{N_w}C_{N_p}$ defective patterns are simulated for each set. Here N_w denotes the total number of links in a neural network and N_p denotes the number of defective links (1 or 2). For FTBP, all defect pattern in a neural network are tested in a set.

Test pattern: 25 sets of training patterns with white noise, so total number of test patterns is 100. The noise level is $\pm 10\%$ of the average input signal level.

3.2 Training Pattern Recognition Ratio

Figure 3 shows the average training pattern recognition rate at the last training iteration on the neural network with a single link defect. BP retraining could recognize all training patterns on all network sizes. PR also showed 100% recognition rate except the 4-6-1 network. FTBP training only succeeded in training when the network size was 4-10-1.

BP retraining starts the training process from the beginning, any defect-free neuron can alternate the neuron with defective link. On the other hand, PR only adjusts the weights belonging to the neuron with defective incoming link, then the defect compensation process is close in a neuron. If the defective link is a critical one, PR cannot compensate with this defect. On FTBP training, the neural network has to minimize the error for all training patterns and for all defect patterns. Then it seems that the 4-6-1 network is too small to recognize so many training patterns.

Figure 4 illustrates the training pattern recognition rate at the last training iteration on the neural network with two flip-link defects. All methods could not achieve 100% recognition rate. The recognition rate by BP retraining was about 97% for any network sizes, PR and FTBP training showed higher recognition rate when the network size became larger. The 4-6-1 network is large enough for BP retraining, but PR and FTBP training need more redundancy (neurons) to compensate multiple flip-link defects.

3.3 Generalization

To evaluate generalization ability of the compensated neural networks, noisy inputs are presented. The average test pattern recognition rate on the defect-free neural network is about 55% when $\varepsilon \leq 0.1$.

Figure 5 shows the average test pattern recognition rate on the neural network with a single flip-link defect. The test pattern recognition rate by BP retraining does not depend on the network size (redundancy). In contrast to BP retraining, FTBP training strongly depends on the redundancy in the neural network, and showed the higher recognition rate than that by defect-free 4-8-1 and 4-10-1 networks. PR showed the intermediate performance among

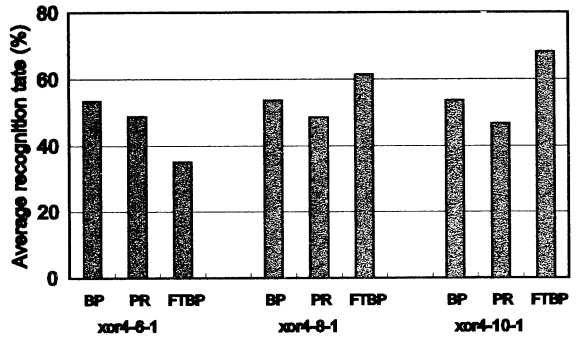


Figure 5: Average test pattern recognition rate (generalization) on the neural network with a single flip-link defect.

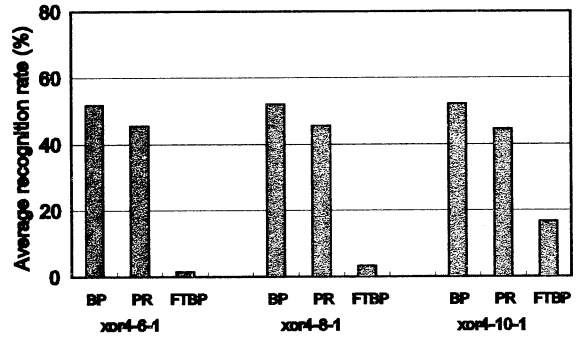


Figure 6: Average test pattern recognition rate (generalization) on the neural network with two flip-link defect.

three defect compensation schemes, and the test pattern recognition rate was about 6% less than that by defect-free neural network.

Figure 6 shows the average test pattern recognition rate on the neural network with two flip-link defects. Different from the case on the single flip-link defect, FTBP training showed poor recognition rate. When the neural network has multiple defects, the number of training pattern of FTBP training becomes the product of the original number of training patterns and the number of all combination of defective parts. Therefore, the training process on FTBP training did not converge in many case. In other words the network size is too small to train so many patterns. In fact, the test pattern recognition rate by FTBP training was increasing as network size increases.

4 Conclusions

In this paper we introduced a new defective link model on the multi-layer neural network. And we evaluated the quantitative performance of typical defect compensation methods; BP retraining, PR and FTBP training. Simulation results by XOR problem said that BP retraining from the random initial weights is the reasonable way to compensate flip-link defect. For large scale neural networks with much redundancy, PR will be the second choice since BP retraining will need huge retraining time. In most cases, the training process of FTBP training did not converge when the neural network had multiple defects.

Simulations by other defect model and the study on hardware design remain as future works.

Acknowledgment

A part of this research is supported by the Grant-In-Aid for Scientific Research from Japan Society for the Promotion of Science #13780218.

References

[1] K. Yamamori, T. Abe and S. Horiguchi, "Theoretical learning-speed evaluation of parallel backpropagation algorithms", *Systems Research and Information systems*, Vol. 9, pp. 121–148, 2000.

[2] M. Witbrock and M. Zagha, "An implementation of backpropagation learning on GF11, a large simd parallel computer", *Parallel Computing*, Vol. 14, No. 3, pp. 329–346, 1990.

[3] Y. Tan and T. Nanya, "Fault-tolerant backpropagation model and its generalization ability", *Proc. 1993 int. Joint Conf. on Neural Networks*, Vol. 2, pp. 2516–2519, 1993.

[4] C. Khunasaraphan, V. Vanapipat and C. Lursinsap, "Weight shifting techniques for self-recovery neural networks", *IEEE Trans. on Neural Networks*, Vol. 5, pp. 651–658, 1994.

[5] Y. Tohma, "Self-repair and self-resumption in intelligent information processing", *Proc. Artificial Intelligence and Neuro-Computers, 6th Symposium on Universities and Science*(in Japanese), pp. 204–208, Kubapro Co., 1992.

[6] K. Yamamori, T. Abe, S. Horiguchi and I. Yoshihara, "An efficient defect compensation scheme for multilayer neural networks on wsi devices", *Proc. International Joint Conference on Neural Networks*, pp. 1056–1061, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Genorobotics

Syed I.Ahson
Department of Computer Science
Jamia Millia Islamia
New Delhi-110025
Email : mehar@ahson.org

Abstract: In this paper we wish to speculate on robotic application that could result from the understanding of DNA sequence and cellular function at the genomic level Understanding in detail how biological systems store and retrieve information and control development, fabricate structural elements will let us build autonomous adaptive robots.

Introduction

Robotics has undergone a revolution over the past few years. The availability of less expensive hardware, faster personal computers with powerful software tools for simulation and program development environments has renewed much interest in mobile robotics. Behavior-based approaches have emphasized a direct perception-to-action mapping without modeling the physical world [1]. The past few years has seen a flurry of research activities into the behavioral control of robots. Connectionist models and evolutionary ideas have further contributed in the race to build more intelligent autonomous robots. Tasks are, however, still limited to wall following, obstacle avoidance, etc.

A series of models related to higher-order cognition has recently been proposed. These models are deeply entrenched in neurobiology and hence are supposed to be less controversial than their predecessors which used artificial neural networks, and thereby lacked correspondence to the human brain. These models derive their information about the structure and function of the human brain which is plentiful with the advent of such techniques as fMRI and PET. However, the current understanding of the human brain is still insufficient in order to reverse-engineer its attributes of robustness and functional behavior although there have been concerted effort in recent years toward the development of mechanistic models of mind [2]. In this paper we consider the impact of the understanding of DNA sequence and the cellular function at the genomic level. Understanding in some detail how biological systems store and retrieve information, control development and fabricate structural components as was rightly predicted by Wiener in 1943 [3]:

"If an engineer were to design a robot, roughly similar in behavior to an animal organism, he would not attempt at present to make it out of proteins and other colloids. He would probably build it out of metallic parts, some dielectrics and many vacuum tubes. The movements of the robot could readily be much faster and more powerful than those of the original organism. Learning and memory, however, would be quite rudimentary .In future years, as the knowledge of colloids and proteins increases, future engineers may attempt the design of robots not only with a behavior, but also with a structure similar to that of a mammal. The ultimate model of a cat is of course another cat, whether it be born of still another cat or synthesized in a laboratory"

What is Life?

Life and consciousness are hard to define and difficult to understand. My position is that you have to be alive to be conscious. And as pointed out by Ian Glynn [4] in his recent book "the origin of life though still puzzling it is less puzzling than the origin of consciousness within living organisms." There are of course many features that are common to living things:

(a) Living things reproduce themselves.
(b) The reproduction is autonomous, i.e. the organism itself provides all the information needed for reproduction.
(c) A living thing has enzymes and smaller molecules for carrying out the chemical changes.
(d) The entire behavior of an organism promotes its survival.

(e) The organism appears to have purpose.
(f) Behavior is largely the product of natural selection.
(g) All known forms of life store their information as long chain of nucleotides in DNA.
(h) There is a machinery in all living things to synthesize protein.

The specific sequence of four-letter DNA in a cell determines the variations in living organisms and the chemical processes involved in the development of the organism[6] .If chemical process do not work properly then life will not succeed. Hence life and consciousness cannot be easily separated from matter. At the same time matter alone cannot explain life ,mind and consciousness There is something left over. Theologists view this as soul – as a 'vital force'. When defining consciousness we must be aware of the vast timescale of evolution. Worms appearing 600 - 700 million years ago exhibit a clearly functioning neuronal system. During the same period the first vertebrates developed. The leap from these time periods to primates appearing about 5 - 8 million years ago is enormous. We know now the rest of the story: Brains have developed very fast during the last million years but the biochemical background of the neuronal tissue is most probably the same leading to a stepwise development from the same building blocks which are the foundation of all biology. Then when would consciousness or experience have jumped into the biological system? Although consciousness is still evading proper explanation in spite of many elegant theories we must assume that it has been present all the way during evolution if we accept the reductive view. Our task would be to find biological compounds, molecules, which are able to store information also in very primitive organisms. In fact such molecules have been found like bacteriorhodopsin a protein from a prokaryocyte Halobacterium halobium with an estimated age of 3.5 billion years. It can interact with light in a way that most proteins cannot. Corresponding data storing molecules in the brain system including vision would enhance our understanding of consciousness as an archaic element in the biological system.

The robots of the mid 21st century will have silicon components and certainly a bunch of other materials and polymers and structures that we do not know. The technology that lets us understand biology at the molecular level will be required. Recently scientists have started turning towards synthesis—transferring molecular biology into engineering [5]. Work is being done on culturing cells and controlling their growth so as to develop replacement organs in vitro. Some researchers have already started building robots that are actuated by mouse muscles – muscles that can be grown in vitro from a single cell. A small microprocessor controls the robot sending command signals that actuate the artificial biological muscles. Another interesting work is the E.Coli robots at MIT that sense molecules that can be absorbed through the cell wall.

Summary

Stephen Hawking also says "We must develop as quickly as possible technologies that make possible a direct connection between brain and computer, so that artificial brains contribute to human intelligence rather opposing it". A similar warning has been given by Hugo de Garis in his "Species Dominance War" where one billion may die in a "gigadeath artillect war". This vision of an unavoidable war is based on the support that there will be no effort in integrating the advances made in AI technologies ,Robotics and Genomics [6]. As our understanding of biological systems improves reverse-engineering their attributes of robustness, reliability and nonlinear behavior would become possible in the near future.

References

[1] Brooks R A (1986),A robust layered control system for mobile robots.IEEE Journal of Robotics and Automation 2(1), pp.14-23

[2] Ahson S I (2000), Daydreaming robots, 5th AROB Conference, Oita, Jan 28-31, pp. 342-346.

[3] Rosenblueth A, Wiener N and Bigelow J (1943), Philosophy of Science, vol.10, pp.18-

[4] Glynn I (1999), An Anatomy of Thought-The Origin and Machinery of the Mind, Oxford University Press, New York.

[5] Brooks R (2002), Flesh and Machines: How Robots will change us, Pantheon Books, New York

[6] Mount D W (2001) Bioinformatics-Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, New York

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

CAM-Brain Machines and Pulsed Para-Neural Networks: Toward a hardware for future robotic on-board brains

Andrzej Buller
ATR International, Human Information Science Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
buller@atr.co.jp

Abstract This paper concerns Pulsed Para-Neural Networks (PPNN) developed for artificial brain building. PPNNs use only three kinds of units: paraneuron, mexor (multi-input exclusive OR) and delayer. All of them receive/send spikes at discrete moments of time called clocks. The paraneuron adds the weighted sum of pulses received at clock t to the state of its internal counter (weights can be either + or −) and when the obtained value is beyond a defined range, the counter discharges at clock t+1, which may be followed by the emission of a pulse at clock t+2. The mexor emits a pulse at clock t if and only if it received one and only pulse at clock t-1. The delayer is a single-input device, such that it emits a pulse at clock t if and only if it received a pulse at clock t-d. Using these units, one can define any spiketrain-to-spiketrain function. In cellular 3-dimensional PPNN every paraneuron, mexor every 1-clock-delay segment of every delayer can be represented by a point in a discrete 3-dimensional space, where pulse transition is possible between closest-neighbor points only. Any instance of such a PPNN segmented into modules of 24×24×24 cells can be run on the CAM-Brain Machine (CBM). The CBM is designed to support up to ca. 75,000,000 paraneurons. The NeuroMaze software tool facilitates rapid prototyping of such CBM-style PPNNs.

1 Introduction

One day we will want to equip robots with intelligent on-board brains. Every cubic millimeter in a robot's head will be worth its space in gold [1]. General-purpose-microprocessor-based devices with gigabytes of unwanted memory and unwanted libraries seem not to be a prospective solution. Hence, Pulsed Para-Neural Networks (PPNN) that use only as many processing units as really necessary are being developed [2]. This paper presents the current state and anticipated development of PPNN theory and engineering.

The term *para-neural* has been introduced because the key concept—*paraneuron*—is functionally simpler than McCulloch-Pitts' formal neuron [10]. The values of its weights can equal only +1 or −1. On the other hand, PPNN generalizes the CoDi modules invented during the search for evolvable hardware for the ATR's artificial brain [7].

2 Toward a PPNN theory

The PPNN theory is to provide formal definitions of PPNN and related notions, as well as a set of theorems useful for development of methods and tools for PPNN-engineering. Although a number of concepts has already been formulated [2] this theory is still in its infancy. In the following subsections a draft PPNN taxonomy with necessary definitions, as well as a set of theorems will be presented. The theorems are being used in practical PPNN synthesis, nevertheless they still wait for their formal proofs.

2.1. Nomenclature and notation

$\mathbf{B} = \{0, 1\}$;
$\mathbf{E}^n f$ – n-th echo of function f;
$\mathbf{S}$ – space of spike trains;
$\mathbf{T}$ – space of clocks, i.e. integers referring to consecutive discrete moments in time;
$\mathbf{U}^N$ – space of N-element streams of spike trains;
$\mathbf{W} = \{+, -\}$;
$\mathbf{Z}$ – space of integers;
Δ – paraneuron;
$\square$ – mexor;
$d\leftarrow$ – delay of d clocks;
Note: "$1\leftarrow$" can be written "$\leftarrow$", while "$0\leftarrow$" can be omitted in formulas;
$fu_0u_1\ldots u_N$ – value of the function f for its arguments $u_0, u_1, \ldots, u_N$ (a shortened notation);
$[\ ]$ – function returning 1 when its argument equals *true* (Examples: $[2 > 1] = 1$; $[1 = 0] = 0$).

2.2. Basic definitions (adopted from [2])

Definition 1. Stream of spike trains. *Spike train* is a function $u : \mathbf{T} \rightarrow \mathbf{B}$. All possible spike trains constitute space $\mathbf{S}$. *N-element stream of spike trains* (^{N}SST) is a vector $\boldsymbol{u}^T$, $\boldsymbol{u} \in \mathbf{S}^N$, $N \in \mathbf{Z}^+$. All possible ^{N}SSTs constitute the space $\mathbf{U}^N$.

Definition 2. Weight vector and association. $\mathbf{w} \in \mathbf{W}^N$ is *weight vector* of the length N. An *Association* of $\mathbf{w}$ =

$(w_0, w_1, \ldots, w_{N-1}) \in \mathbf{W}^N$ and $\boldsymbol{u} = (u_0, u_1, \ldots, u_{N-1}) \in \mathbf{S}^N$ is a vector $(\mathbf{w}\boldsymbol{u})^T = (w_0u_0, w_1u_1, \ldots, w_{N-1}u_{N-1})^T$.

Example:

$\boldsymbol{U} = ($ +0010000001000000000000000...,
+0011000001000000000000000...,
+0000001001000000000000001000...,
+0000001011000000000000000...,
-0000000000001010000010000...,
-0000000000001000000100000...,
-0000000000001000000110000...,
-0000000000001000010011000...$)^T$.

Definition 3. *Activation* α: $(\mathbf{W} \times \mathbf{B})^N \to \mathbf{Z}$, $N \in \mathbf{Z}^+$ is a function such that for every $\mathbf{w} = \{w_0, w_1, \ldots, w_{N-1}\} \in \mathbf{W}^N$ and $\mathbf{u} = \{u_0, u_1, \ldots, u_{N-1}\} \in \mathbf{B}^N$, $\alpha\mathbf{w}\mathbf{u} = \sum_{i=0..N-1} w_i u_i$.

Definition 4. *A paraneuron* is a function

$$\Delta^P : \mathbf{W}^N \times \mathbf{U}^N \to \mathbf{S},$$

such that for every $\mathbf{w} \in \mathbf{W}^N$, $\boldsymbol{u} \in \mathbf{U}^N$

$(\Delta^P\mathbf{w}\boldsymbol{u})_{t+1} = [\, x_t \le A'' \,] \vee [\, x_t \ge B'' \,]$,

$x_{t+1} = [x_t \le A']\, A^\circ + [x_t \ge B']\, B^\circ$
$+ [A' < x_t < B']\, x_t + \alpha\mathbf{w}\boldsymbol{u}_t$,

where $P = (A'', A', A^\circ, B^\circ, B', B'') \in \mathbf{Z}^6$ is a constant vector parameter, while $x : \mathbf{T} \to \mathbf{Z}$ is an auxiliary function which can be interpreted as a paraneuron's internal state.

Example: For $\mathbf{P} = (-3, -4, -1, 2, 4, 6)$, $x_0=0$, for the association U taken from the example to Definition 2 the paraneuron returns spike train $\Delta^P U$ as shown below. Note: an underlining of a character is to be interpreted as multiplying it by –1.

αU = 00210020141140100100331000...
x = 00023335237222333414410 00...
$\Delta^P U$ = 00000000000100001111011000...

Definition 5. *A mexor* is a function $\square : \mathbf{U}^N \to \mathbf{S}$, such that for every $\mathbf{u} = \{u_0, u_1, \ldots, u_{N-1}\} \in \mathbf{U}^N$, $(\square\boldsymbol{u})_{t+1} = [\, \sum_{i=0..N-1} u_{i,t} = 1 \,]$.

Example:
p = 00100010100001000...
q = 00001000100001000...
r = 00000000001010 00...
$\square$pqr = 00010100000010000...

Definition 6. *A delay* is a function $d\leftarrow : \mathbf{S} \to \mathbf{S}$, such that for every $u \in \mathbf{S}$, $(d\leftarrow u)_{t+d} = u_t$.

Example:
u = 00100010100001000...
$3\leftarrow$u = 00000100010100001000...

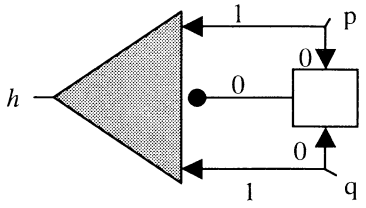
Definition 7. A PPNN (Pusled Para-Neural network) is a function $f : \mathbf{U}^{M'} \to \mathbf{U}^{M''}$ composed of paraneurons, mexors and delays only, where $M', M'' \in \mathbf{Z}^+$.

Example: $h$pq = $\Delta^P(+1\leftarrow$p, $-0\leftarrow\square$pq, $+1\leftarrow$q).

Note: (i) A PPNN can be defined using formulas, as well as graphs. In graphs, paraneurons are red or grey triangles, mexors are yellow or white squares and delays are arrows labeled with numbers. The ending of a given arrow can be sharp or not sharp to represent a positive weight or a negative weight, respectively. (ii) Character '+' does not have to be written. (iii) Delay parameter 1 does not have to be written. If delay parameter d equals 0, the symbol '$\leftarrow$' does not need to be written. Hence, h can be defined more elegantly as

$$h\text{pq} = \Delta^P(\leftarrow\text{p}, -\square\text{pq}, \leftarrow\text{q})$$

or as



Definition 8. Echo. An n-th echo of function $f : \mathbf{U}^{M'} \to \mathbf{U}^{M''}$ is any function $g : \mathbf{U}^{M'} \to \mathbf{U}^{M''}$ such that for every $u \in \mathbf{U}^{M''}$ and for every $t \in \mathbf{T}$: $(gu)_t = (fu)_{t+d}$. A set of all n-th echoes of a function f is denoted $\mathbf{E}^n f$.

Examples:

$\Delta^P(2\leftarrow$p, $-\square$pq, $\leftarrow$q) $\notin \mathbf{E}^n h$ for all $n \in \mathbf{Z}$.

$3\leftarrow\Delta^P(\leftarrow$p, $-\square$pq, $\leftarrow$q) $\in \mathbf{E}^3 h$.

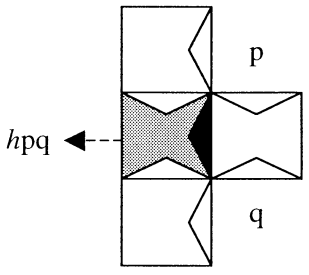
$\Delta^P(4\leftarrow$p, $-3\leftarrow\square(\leftarrow$p, $\leftarrow$q), $4\leftarrow$q) $\in \mathbf{E}^3 h$.

Definition 9. A PPNN belongs to the class of nD-cellular-automatic PPNN denoted CnD PPNN if it can be represented as n-dimensional cellular automaton, where a given cell can be occupied by either a paraneuron, a mexor or a single-clock delay and a given wall of a given cell can serve as either a single input or as a single output.

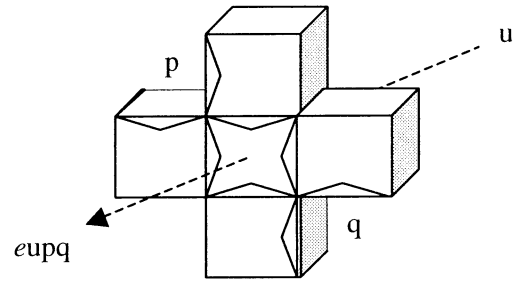
When dealing with CnD PPNNs it has appeared convenient to represent their elements as squares or cubes of identical size, where paraneurons are red or grey, mexors are yellow or white and 1-clock delays are are blue or white. Small white or black triangles symbolize positively or negatively weighted inputs.

Examples:

Function h such that $h$pq = $\Delta^P(\leftarrow$p, $-\square$pq, $\leftarrow$p) belongs to the class C2D PPNN as shown below.



Function e such that $e$puq = $\square$uppqq doesn't belong to the class C2D PPNN, however it belongs to the class C3D PPNN as shown below.



2.3. Special PPNNs (adopted from [2])

Definition 10. Modular PPNN. A PPNN belongs to the class of modular PPNNs denoted *m*C3D PPNN if it is C3D PPNN and:

1. it is segmented into $l_X \times l_Y \times l_Z$ -cell modules;
2. every module has:
   a. a limited set U of input-to-module points;
   b. a limited set Y of output-from-module points;
   c. a limited set Π of cells that can be occupied by paraneurons;
   d. a set of wraparounds passing spikes between respective points on opposite walls.

Definition 11. A PPNN belongs to the class of μ3D PPNN if it is C3D PPNN, $l_X = l_Y = l_Z = 24$; and assuming that:

i. module cells are identified using the coordinates $(x, y, z) \in \mathbf{L} = 0..23 \times 0..23 \times 0..23$;

ii. cell contact points are identified using coordinates $(c_x, c_y, c_z) \in \mathbf{C} = \{(x + .5v_x, y + .5v_y, z + .5v_z) \mid (x, y, z) \in \mathbf{L}, (v_x, v_y, v_z) \in \mathbf{V}^3, \mathbf{V} = \{ \{-1, 0, 1\} \mid \mid v_x \mid + \mid v_y \mid + \mid v_z \mid = 1 \},$

iii. $U \cup Y = \{(x + .5v_x, y + .5v_y, z + .5v_z) \mid ((x-1) \bmod 3 = y = (z-1) \bmod 3 = 0, (v_x, v_y, v_z) = (0, -1, 0)) \vee (x = 23, (y-1) \bmod 3 = (z-1) \bmod 3 = 0, (v_x, v_y, v_z) = (1, 0, 0)) \vee ((x-1) \bmod 3 = (y-1) \bmod 3 = 0, z = 24, (v_x, v_y, v_z) = (0, 0, 1))\}$

$Y = \{(13, 0, 1) + .5(0, -1, 0), (1, 0, 13) + .5(0, -1, 0), (23, 1, 13) + .5(1, 0, 0), (1, 13, 23) + .5(0, 0, 1)\},$

$\Pi = \{(x, y, z) \in \mathbf{L} \mid (x-2) \bmod 3 = 0, (y-1) \bmod 2 = 0, z \bmod 2 = 0\}.$

Definition 12. $^{\beta(a)}$PPNN. A function $f: \mathbf{U}^{M'} \rightarrow \mathbf{U}^{M''}$, $M', M'' \in \mathbf{Z}^+$ is $^{\beta(a)}$PPNN if it is composed of functions from the set $\{\Delta, \Box, \leftarrow\}$, where for every $\boldsymbol{u} \in \mathbf{U}^N$, $\mathbf{w} \in \mathbf{W}^N$

$(\Delta \boldsymbol{u})_{t+1} = [\, x_t \geq 2 \,]$, $x_0 = 0$, $x_{t+1} = [a < x_t < 2]\, x_t + \alpha \mathbf{w} \boldsymbol{u}_t$,

$(\Box \boldsymbol{u})_{t+1} = [\, \sum_{i=0..N-1} u_{i,t} = 1 \,]$, and for every $u \in \mathbf{S}$,

$(d \leftarrow u)_{t+d} = u_t$.

Notes: (i) When defining $^{\beta(a)}$PPNN's symbol of paraneuron, it does not need to be labeled with **P** which has the default value $(-\infty, a, 0, 0, 2, 2)$. (ii) The CAM-Brain Machine (CBM) supports μ3D $^{\beta(-7)}$PPNNs.

2.4. $^{\beta(a)}$PPNN-based devices (adopted from [2])

Definition 13. Delayed AND and delayed OR.
$\&pq = \Delta(\leftarrow p, -\Box pq, \leftarrow q)$. $||pq = \Delta(\leftarrow p, \Box pq, \leftarrow q)$.

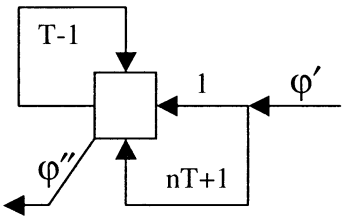
Theorem 1. If $\&$ and $||$ are $^{\beta(a)}$PPNNs, then $\forall_{a<0,\, t \in \mathbf{T}}$ $(\&pq)_{t+3} = p_t \wedge q_t$, $(||pq)_{t+3} = p_t \vee q_t$.

Definition 14. Flexible AND. $\&_\Phi pq = \varphi'' \leftarrow \Delta((\varphi'+1) \leftarrow \varphi^\circ \leftarrow p, -\varphi' \leftarrow \Box(\varphi^\circ \leftarrow p, \varphi^\circ \leftarrow q), (\varphi'+1) \leftarrow \varphi^\circ \leftarrow q)$, where $\Phi = (\varphi^\circ, \varphi', \varphi'')$.

Theorem 2. If $\&_\Phi$ is $^{\beta(a)}$PPNN , $\Phi = (\varphi^\circ, \varphi', \varphi'')$ and $n = 3+\varphi^\circ+\varphi'+\varphi''$, then $\&_\Phi = \mathbf{E}^n \&$.

Note: An algorithm for replacing given delays with proper wildcards would be useful for automatic generation of echoes of a given PPNN.

Definition 15. Single-pulse-triggered **timer** producing pulses with frequency 1/T for the period nT.
$\omega u = \varphi'' \leftarrow \omega^* u$.
$\omega^* u = \Box(\leftarrow \varphi' \leftarrow u, (nT+1) \leftarrow \varphi' \leftarrow u, (T-1) \leftarrow \omega^* u)$.



3. Brain Builder's Toolkit

While a PPNN engineer is believed to one day build artificial brains made up of small cubic PPNN modules (as if he were playing with LEGO blocks), currently, apart from the infant PPNN theory, he has a software package known as NeuroMaze to his disposal. Neuromaze is a software tool for rapidly prototyping μ3D $^{\beta}$PPNNs that includes a library of ready-to-use PPNN structures. In addition to NeuroMaze, he also has the CAM-Brain Machine—a dedicated hardware system that supports designs created under NeuroMaze.

NeuroMaze provides its user with a planar worksheet on which, using mouse, he can create a defined layer of his μ3D PPNN via locating red, yellow and blue cells in allowed places. At any stage, the design can be seen on 3D display in a separate window and moved, resized and rotated. Also at any stage of the design, it may be tested via providing pulses to desired cells and simulating their propagation. Simulation sessions can be recorded as .avi

files. Any part of the design can be cut, copied, and pasted into another place or sent to the library. The flag feature of the tool is a heuristics for automated creation of delay paths of desired length between desired cells. Since the class of PPNN supported by the NeuroMaze requires segmentation of working space onto modules of 24×24×24 cellular automata cells, this tool offers several ways of designing a network of module interconnections. The source-point/destination-point pair can be defined in a table or using the MMHD (Multi-Module High-Level Definition) langue [9]. *NeuroMaze 3.0 Edu* with its User's Guide can be downloaded from `http://www.his.atr.co.jp/ecm/n_maze`.

The Brain-Building Library is a repository of useful PPNN modules and clusters of modules, as well as extracted structures to be reused. A desired structure can be downloaded and pasted in the design that is open under NeuroMaze. As for the modules, a double 32-input OR, a 64-input OR, adjustable timer [3], object location recognizer of a robotic arm [3], a spiking integrator and an associator are available. The spiking integrator non-linearly cumulates positive/negative excitation and responds with a changeable frequency of produced pulses [4]. The associator learns associating each of its 6 input channels with each of its 8 output channels trying several possibilities that are all the time either rewarded or punished [4]. As for the clusters, a processing unit for a distributed working memory [5], a motor drive for a simulated legged robot [6] and a vision system for a mobile robot are available. As for the extracted structures, a delayed AND/OR, as well as various oscillators and limited-length time-series generators can be downloaded from the library.

The CAM-Brain Machine (CBM) is the first dedicated hardware developed to process μ3D $^{\beta}$PPNNs. supporting up to ca. 75 million paraneurons. In the CBM's FPGA-based working memory all 24^3 cells of a given PPNN module are updated in parallel. The speed of the update is 130 billion cells per second [8]. Nevertheless, multi-module structures are processed module-by-module. Next generations of the CBM are intended to process all involved modules in parallel.

4. Concluding remarks

Lack of a truly parallel dedicated hardware is an obstacle in the development of real-time PPNN-based models. Another obstacle on the road to a PPNN-based artificial brain is still underdeveloped methodology of creating very-large-scale networks, consisting of tens of millions of paraneurons, as well as the still not satisfactory automation of module creating. Nevertheless, owing to the libraries of predefined structures and the heuristics for automated delay-path creation, a heuristics for PPNN-to-3D PPNN conversion should be invented soon.

The directions of considered further development of PPNN engineering include (i) Searching theorems helpful in automated PPNN synthesis, automated PPNN-to-3D PPNN conversion and automated module interconnection networks, (ii) practical testing of usefulness of various genetic algorithms and other heuristics in PPNN engineering, and (iii) development of new dedicated hardware for PPNNs including CBM-compatible chips to be implemented as small separate cubes to be rapidly interconnected by an industrial robot. The robot himself could have a PPNN-based brain. In such case his job would be just a reproduction.

Acknowledgement: This research was conducted as a part of the *Research on Human Communication* supported by the Telecommunications Advancement Organization of Japan (TAO).

References

[1] Buller A (2002) In Quest of an Artificial Brain, *Proceedings of the Fifth International Conference on Human and Computer (HC-2002), September 11-14, 2002, Aizu, Japan*, 195-201.

[2] Buller A. (2002) Pulsed Para-Neural networks (PPNN): Basic concepts and definitions, *Technical Report TR-HIS-0007, ATR Human Information Science Laboratories, Kyoto.*

[3] Buller A. Eeckhaut H. & Joachimczak M. (2002) Pulsed Para-Neural Network (PPNN) Synthesis in a 3-D Cellular Automata Space, *The 9th International Conference on neural Information Processing, November 18-22, 2002, Singapore.*

[4] Buller A, Joachimczak M, Bialowas J (2002) NeuroMaze: A New Method of Pulsed Neural Network Synthesis, *The Seventh International Symposium on Artificial Life and Robotics (AROB 7th '02), January 16-18, 2002, Beppu, Japan*, 648-649.

[5] Buller A. & Shimohara K. (2002) Kansei Processes in a Psychodynamic Agent, Proceedings, *The 6th World Multiconference on Systemic, Cybernetics and Informatics, July 14-18, 2002, Orlando, Florida, USA, Volume VIII*, 133-138.

[6] Buller A. & Tuli T.S. (2002) Four-Legged Robot's Behavior Controlled by Pulsed Para-Neural networks (PPNN), *The 9th International Conference on neural Information Processing, Nov. 18-22, 2002, Singapore.*

[7] Gers F., de Garis H. & Korkin M (1997) CoDi-1Bit: A Simplified Cellular Automate based Neuron Model, *Evolution Artificiele 97, 22 Octobre 1997, Nimes, France*, 211-229.

[8] Korkin M, Fehr G & Jeffrey G (2000) Evolving hardware on a large scale, *Proceedings, The Second NASA / DoD Workshop on Evolvable Hardware, July 2000, Pasadena*, IEEE Comput. Soc., 173-81.

[9] Liu J (2002) *NeuroMaze User's Guide, Version 3.0*, ATR HIS, Kyoto.

[10] McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity, *Bulletin of Math. Bio., 5*, 115-133.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Handcrafting Pulsed Neural Networks for the CAM-Brain Machine

Hendrik Eeckhaut and Jan Van Campenhout
Department of Electronics and Information Systems (ELIS)
Ghent University
St.-Pietersnieuwstraat 41, B-9000 Ghent, Belgium

Abstract

The CAM-Brain Machine (CBM) is a hardware implementation of a brain-inspired, recurrent, digital neural network. It is an experimental machine composed of reconfigurable (evolvable) hardware, capable of training and evaluating cellular automata based neural network modules directly in silicon.
The networks of the CBM were originally intended to be built with a genetic algorithm. However, currently the implemented genetic algorithm is not powerful enough to evolve satisfactory networks for applications of a meaningful complexity. To that end, the training technique should be considerably enhanced.
This paper addresses the problem of using the CBM more efficiently, still based on genetic evolution, but using a much more efficient gene pool. The paper focuses on the identification of frequently used primitive functions, and the hand crafting of these functions into efficient basic network patterns. Functions range from simple delay lines over logic gates, to adjustable timers and switches. Furthermore a technique is introduced to build neural structures that can generate arbitrary pulse trains.
Eventually these basic structures will be combined in a library, that will serve as a potent gene pool.

1 CAM-Brain Machine architecture

The CAM-Brain Machine (CBM) is a hardware implementation for large, recurrent, digital neural networks [1]. The networks inside the CBM are built from two types of cells: transport cells and neuron cells. Cells communicate with pulses (one-bit signals) also called spikes. The cells are periodically tiled on a regular three-dimensional lattice, and are each connected with six neighbors (Von Neumann Neighborhood). All cells are physically assembled into interconnected modules of $24 \times 24 \times 24$ cells.

The cells are updated in discrete time steps; they can only change their states when the clock ticks.
Information is not only encoded in the occurrence of these instantaneous pulses, but also in the timing pattern of these pulses. Timing is essential in the CBM's pulsed model. Pulses need some time to travel through the neural network and the time interval between pulses also contains information. Seen over time, the cells generate pulse trains, which are the true carriers of information. These can be converted in analog values in various ways [2].
As already stated, cells come in two flavors: transport cells and neuron cells. On each of their six faces, transport cells have either an input or an output. Pulses flow from one cell to another only if the first cell has an output and the adjacent cell has an input. Each cell has a 1-bit state. The state of a cell becomes one (excited) at the end of a clock period only if the number of incoming pulses during this period is exactly one (spike blocking). In all other cases the state becomes zero. If a cell's state is one, it fires (emits a pulse) through all its outputs during the next clock period.
Neuron cells are a bit more complex than transport cells. They also have 6 in- or outputs, but now the inputs also have a weight ($+1$ or -1). The neurons take a weighted sum of their inputs during one cycle and add the result to their (4 bit) accumulator at the end of this cycle. If the accumulator becomes greater or equal to its threshold (in the current implementation this is fixed at 2) the firebit is set and the accumulator is reset to zero. If, however, the accumulator becomes less than -7, the neuron is also reset to zero but the firebit is not set to one. If its firebit is one, a neuron cell fires through all its outputs. Hence the neuron cells have a firing delay of 2 clock periods.

2 Handcrafing basic functions

2.1 Notation

In this section some useful networks are presented. This document includes some simple networks of [3] and also uses its notation (figure 1).

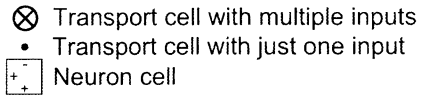


Figure 1: Basic cell symbols; the + and − indicate the weight of the neuron connections.

A distinction is made between transport cells with one and transport cells with multiple inputs because cells with exactly one input just pass on there inputs to there outputs with a one clock delay. Cells with multiple inputs in addition perform the spike blocking function.
Connections between the cells are represented by lines. Arrows, indicating the direction of the connection, are used only when necessary. Numbers labelling outputs denote the latency of the signals.

2.2 Basic routing primitives

Controlling the delay of connections The normal way to connect cells[1] (figure 2a) is to give the source cell an output and the destination cell an input. This way one can only create paths between two cells of length $ShortestPath + 2k$ with $k \in \mathbb{N}$, where $ShortestPath$ is the Manhattan distance between the cells in the 3D-grid. To create detours of odd length one needs to include a neuron in the path as in figure 2b. The extra delay is obtained thanks to the extra clock tick a neuron needs for its processing. Sometimes a similar result can be obtained by using an external input (figure 2c).
Detours are important to delay signals, so as to ensure that different signals arrive at the correct time.

[1] All networks shown in this paper can be viewed in three dimensions on our CBM-website: *http://www.elis.rug.ac.be/~heeckhau/CBM/*. The construction of these networks, the detailed configuration of all these cells (inputs, outputs, ...) was possible thanks to some free assisting CAD-environments [4].

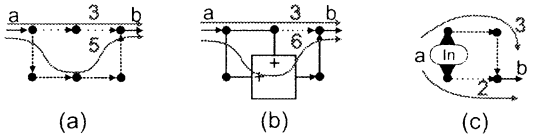


Figure 2: (a) Normal detours have path-length differences of even length. (b) A detour path-length difference of odd length. (c) Inputs can provide both.

Crossings Sometimes two paths have to cross without interfering. One could deflect one path to another plane to avoid contact of the paths but then one path gets longer. If there is enough room (2 extra cells) the technique of figure 3 can be used, introducing no extra delay in either path.

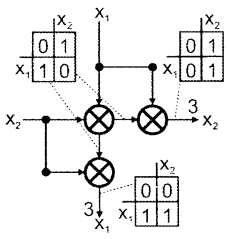


Figure 3: Two paths can cross in the same plane without interfering. Note that all structures can be rotated and mirrored.

2.3 Combinational functions

Figure 4 shows how all boolean functions of two variables can be implemented with the neural model. Note that all circuits are pipelined and may have different delays.
Circuits of more variables can be obtained by combining these functions. An example is shown in figure 5a. This multiplexer chooses one data bit from two sources (x_2 or x_3) under the control of the select input (x_1). Figure 5b shows a 3-input AND-gate. Note that this network is smaller than a naive combination of two AND-gates. Another useful example is the majority function (figure 5c) This circuit fires when the majority (two or more) of the inputs received a pulse.

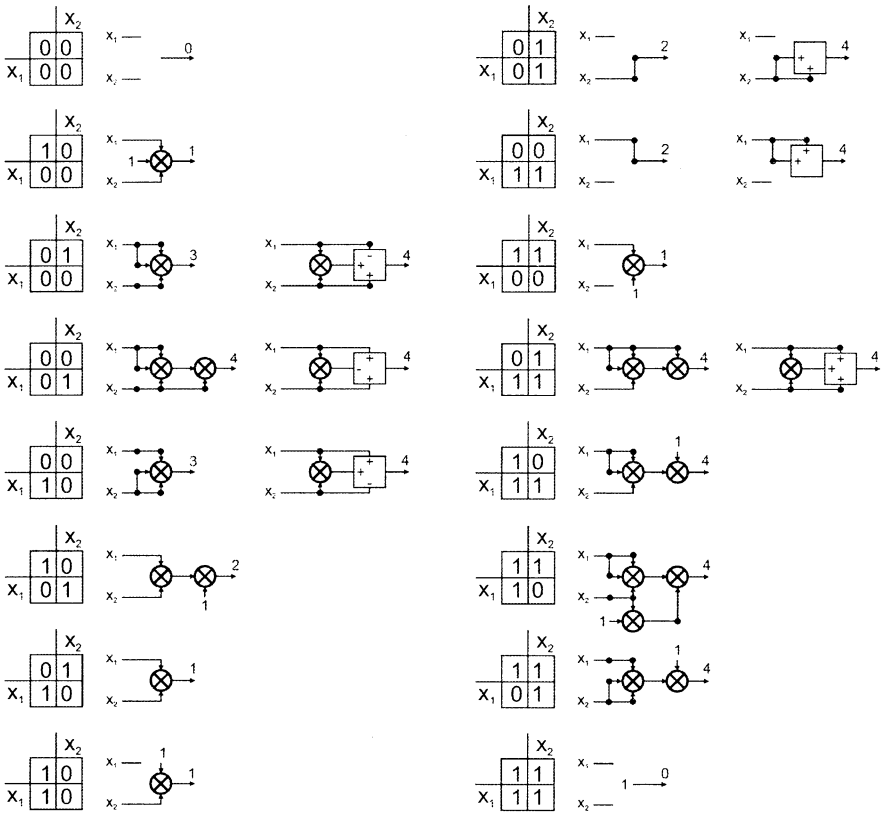


Figure 4: All boolean functions of two variables. A '1'-input represents a continuous source of pulses.

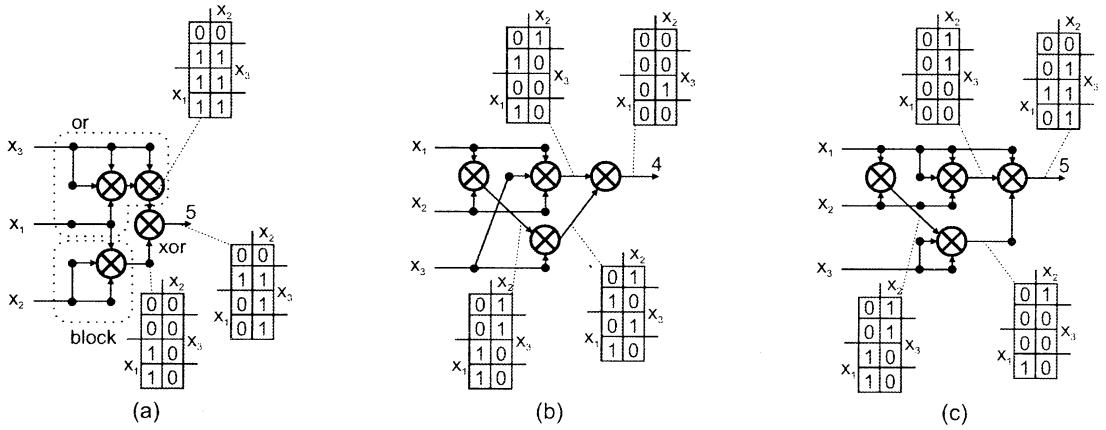


Figure 5: (a) 2-to-1 multiplexer (if (x_1) then (x_2) else (x_3) == $(x_1$ or $x_3)$ xor $(x_1$ and not $x_2))$. (b) 3-input AND function. (c) Majority function of three inputs.

2.4 Sequential functions

The **spike doublers** in figure 6 convert a single pulse in a sequence of two pulses (010 $-delay\rightarrow$ 0110). The difference is that the right version also can handle consecutive pulses on its input. The left version will convert 0110 into 01010 where the right version converts it into 01110.

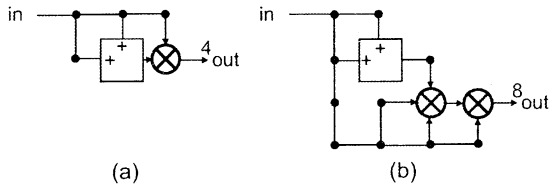


Figure 6: Two spike doublers.

Memories can be made easily with feedback loops of a length equal to the number of bits one likes to store (figure 7a). The smallest possible loop contains four cells (4 bit memory). Memories of an odd number of cells are possible by including a neuron in the loop (as in figure 2b). The content of the memory can be set/reset by injecting a pulse at the appropriate moment.
The circuit of figure 7b shows a 5 bit memory with a separate set- and reset-input. The bits are now stored in a double feedback loop.

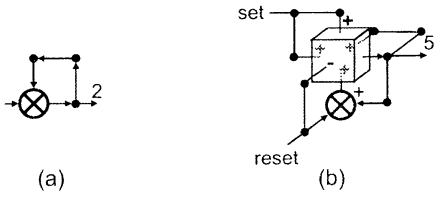


Figure 7: (a) Simple memory feedback loop. (b) A set/reset memory. Because its six faces are used, the neuron is shown in 3D.

On/Off toggles (figure 8) make it possible to toggle a continuous stream of pulses on or off.
An **On/Off switch** with separate on- and off-inputs is shown in figure 9. If two spikes arrive at the two inputs at the same time (or with a difference of 1 clock tick) the switch will be on. The minimum time-interval between two spikes on the same input should be 5 clock

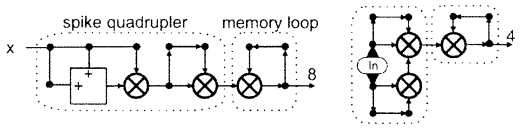


Figure 8: Two on/off toggles. Note how the right version makes use of the external input (as in figure 2c).

ticks[2] and a spike on the off-input must not arrive 2,3,4 or 5 clock ticks after a spike on the on-input, otherwise the switch will have distorted output.

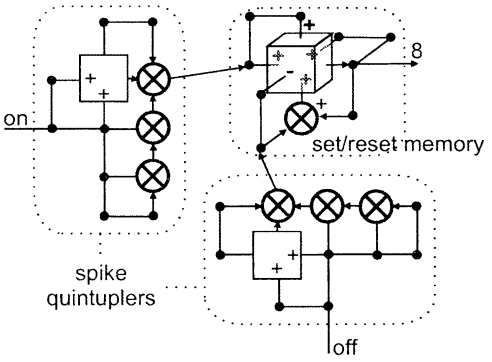


Figure 9: An on/off switch.

A simple way to have a constant stream of spikes for specified period is the **adjustable timer (one-shot)** of figure 10. The input spike is used to turn on an on/off toggle. The same spike turns the toggle back off after being detoured for the desired period.

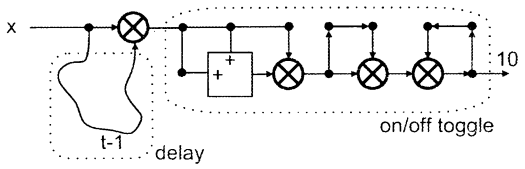


Figure 10: An adjustable timer (one-shot).

The structure in figure 11 can **retrieve the** (positive[3]) **state of a neuron** when a spike arrives at the upper input. Only if the state of the neuron is one, a spike

[2]This can be avoided by applying the technique of figure 6b.
[3]This circuit will not work correctly if the neuron has a negative state.

will be emitted on the test output. The neuron preserves its normal function as long as no spikes arrive simultaneously with the test input. Only if the neuron receives spikes during the test the results of the normal operation of the neuron will be affected.

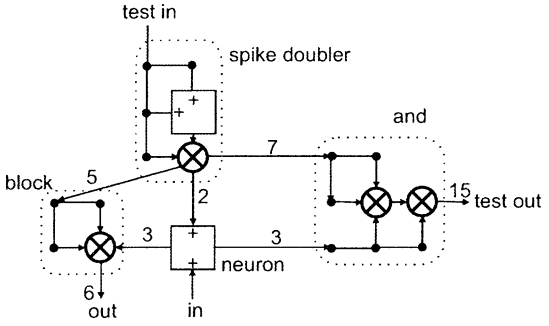


Figure 11: Test if the state of a neuron is one or zero. The numbers above the arrows indicate the length of the path.

2.5 Fixed spike trains

It is also possible to generate arbitrary spike trains with the CBM's neural model. With the specific configuration of figure 12 a predefined spike train is produced after receiving a single spike at the input.
The principle is simple: Let the pulse propagate through one chain of cells and let it split into a second chain at those points where one wants a spike in the target spike train. Because there are no diagonal connections, two different chains are needed for the spikes on even and the spikes on odd clock ticks. The spike train of the odd chain has to be delayed by one clock tick before it is combined with the spike train of the even chain. There are two ways to implement this as mentioned in 2.2. A simple example is given in figure 13.
With this technique it is possible to build modules that can follow a given target spike train for 6864 clock ticks[4]. This is remarkably much larger than 70, the result of the genetic algorithm, as mentioned in [5].

3 Conclusions

This paper has presented some reusable primitive pulsed neural network structures for the CAM-Brain

[4]Because the chains have to be folded into the module, a few bits can't be generated (5 times 2 bit).

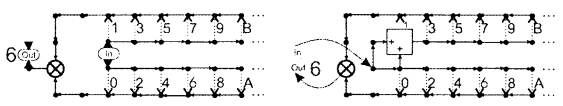


Figure 12: Generate an arbitrary spike train.

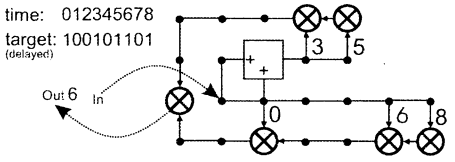


Figure 13: An example of the generation of an arbitrary spike train.

Machine. These basic structures can serve as building blocks for the design of more complex networks.
We expect that these networks can form the base of a much more powerful gene pool for the CBM's genetic algorithm. However, before we can effectively verify this, we shall have to modify this algorithm so that it can take advantage of the higher complexity of these genes.

References

[1] M. Korkin, G. Fehr, and G. Jeffery. Evolving hardware on a large scale. In *Proceedings. The Second NASA/DoD Workshop on Evolvable Hardware*, pages 173–81. IEEE Comput. Soc., July 2000.

[2] M. Korkin, N. E. Nawa, and H. de Garis. A "spike interval information coding" representation for ATR's CAM-Brain machine (CBM). *Lecture Notes in Computer Science*, 1478, 1998.

[3] H. Van Marck and Y. Saeys. Haalbaarheidsstudie van de CAM-Brain computer (dutch). Technical report, S.AI.L, 2001.

[4] A. Buller, H. Eeckhaut, and M. Joachimczack. Pulsed Para-Neural Network(PPNN) synthesis in a 3-D cellular automata space. In *The 9th Int. Conference on Neural Information Processing (ICONIP'02)*, November 2002.

[5] H. de Garis, A. Buller, T. Dob et al. Building multimodule systems with unlimited evolvable capacities from modules with limited evolvable capacities (MECs). In *Proceedings. The Second NASA/DoD Workshop on Evolvable Hardware*, pages 225–34. IEEE Comput. Soc, July 2000.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Heuristic-based Computer-Aided Synthesis of Spatial β-type Pulsed Para-Neural Networks (3D-$^{\beta}$PPNNs)

Daniel Jeliński
Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, G. Narutowicza 11/12, Gdańsk 80-952 Poland
danielj.@wp.pl, daniel@box43.pl

Michał Joachimczak
ATR International, Human Information Science Laboratories, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
(on leave from Gdansk University of Technology, Poland)
mjoach@atr.co.jp, guhru@wp.pl

Abstract This paper deals with semi-automated synthesis of 3-Dimensional Pulsed Para-Neural Networks (PPNN) in 24×24×24-cell working space of a cellular automaton. The cells exchange pulses of fixed amplitude at discrete moments of time called clocks. 3D-PPNNs are structures built of three kinds of cells: blue cells, yellow cells and red cells. Every blue cell receives pulses from one and only one neighbor cell to store every pulse until next clock and then passes it to up to five other neighbor cells. Every yellow cell can receive pulses from up to five neighbors. If at clock t it receives one and only one pulse, at clock t+1 the pulse is passed to a defined neighbor cell. Every red cell has its internal state x and receives pulses from up to n defined neighbors and passes pulses to up to 6-n other neighbors. For β-type PPNNs each red cell's output y[t+1] equals 1 if and only if x[t] > 1, while x[t+1] = x[t]·*Truth*(a<x[t]<2) + U[t], where U is a weighted sum of all pulses entering this cell with a weight of either "+" or "-". When handcrafting purposeful 3D-$^{\beta}$PPNNs under the early NeuroMaze editor, creating long paths of desired propagation times was the most tedious and time-consuming activities, especially when already existing cells constituted strenuous obstacles. Handcrafting of paths of delay of some hundreds clocks took a couple hours. Nobody even tried to handcraft delaying paths of tens of thousands cells. The heuristics we propose reduced this time to seconds, which opens a new chapter in PPNN engineering and the ATR's CAM-Brain Machine's exploitation.

1 Introduction

Pulsed Para-Neural Networks (PPNNs) are functions such that for given input streams of spike trains a stream of other spike trains is produced. They are composed of three and only three kinds of functions: paraneurons, mexors, and delays. A paraneuron counts incoming pulses and in certain circumstances may produce an output pulse and/or zero its counter. A mexor produces an output pulse when it received one and only one pulse a clock before. A delay returns a pulse after a defined time of reception of the pulse. A PPNN theory is being developed at ATR HIS, Kyoto, toward a future generation of hardware for artificial brain building [3]. An Action Drive of the ATR's Neko 1.0 Mobile Robot [5] as well as a Vision System [3] have been designed as a PPNN.

In this paper we deal with a special class of PPNNs called spatial β–type Pulsed Para-Neural Networks (3D-$^{\beta}$PPNNs) synthesized in a working space of a cellular automaton, where the cells exchange pulses of fixed amplitude at discrete moments of time called clocks. Since we use 3-dimensional cellular automaton each cell can be though of as a cube with 1 to 5 inlets and with remaining facets working as outlets.

In 3D-PPNNs paraneurons are "red cells", mexors are "yellow cells" and delays are paths made of "blue cells" (Fig. 1)

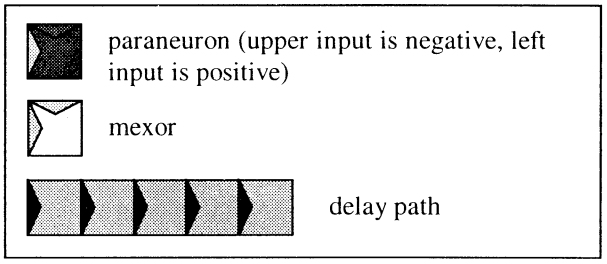


Figure 1. PPNN processing units

Every blue cell receives pulses from one and only one neighbor cell to store every pulse until the next clock and then pass it to up to five other neighbor cells. Every yellow cell can receive pulses from up to five neighbors. If at clock t it receives one and only one pulse, at clock t+1 the pulse is passed to a defined neighbor cell. Every red cell has its internal state x (counter) and receives pulses from up to n defined neighbors and passes pulses to up to 6-n other neighbors. For β-type PPNNs each red cell's output y[t+1] = 1 if and only if x[t]>1, while x[t+1] = x[t]·*Truth*(a<x[t]<2)+U[t], where U is a weighted sum of pulses entering this cell, a is the *counter's lower threshold* and a weight is either "+" or "- " [3].

The ATR's CAM_Brain Machine (CBM) has been designed based on the idea of CoDi-1bit neurons [2]. It was initially intended to support evolution and a real-time running of networks of up to 74,465,280 CoDi-1bit neurons [1][4]. On the CBM, 3D-$^{\beta}$PPNNs (such that the counter's lower threshold equals –7 for every paraneuron) segmented into 24×24×24-cell modules can be run. Since the evolutionary part of the CBM is still not ready to become a source of useful modules, a methodology of rapid handcrafting of certain class of 3D-$^{\beta}$PPNNs has been developed at the ATR HIS, Kyoto, Gdansk Artificial Brain Research Initiative (GABRI) and the Department of Electronics and Information Systems, Ghent University. As a result of cooperation between ATR HIS and GABRI, early versions of the *NeuroMaze*—a software package for rapidly prototyping useful CBM-compatible modules and module clusters—was produced. Nevertheless, even using the software, module creation was tedious work.

When using the early *NeuroMaze*, the most time-consuming activity was creation of long delay paths of desired propagation time between defined points of the 3-dimensional working space, especially when already occupied cells constituted strenuous obstacles. Not always such a path can be made of blue cells only. A length of a path between two defined points in a 3-dimensional space of cubic cells can be either an odd number or an even number. When a desired propagation time is odd, while lengths of possible paths are even, or when a desired propagation time is even, while lengths of possible paths are odd, the path must be forked and both arms of the fork must be taken to a red cell to get an extra 1-clock delay. Hence, handcrafting of paths of delay of some hundreds clocks used to take a couple hours. Nobody even tried to handcraft delay paths of tens of thousands cells. This motivated us to work out such algorithms that would reduce this time to seconds. However finding a path of desired length is an NP-hard problem and this led us to creating heuristics which can deal with the task in most cases.

In following sections we provide a formal description of CBM-compatible 3D-$^{\beta}$PPNN model, a description of the heuristics for automatic creation of a cellular path of desired delay between defined cells already implemented in the *NeuroMaze 3.0 Pro*, as well as conclusions.

2 CBM-specific 3D-$^{\beta}$PPNN model

Due to hardware limitations, some additional assumptions need to be taken to satisfy the CBM-specific 3D-$^{\beta}$PPNN model. Let us take the following notions:

$\mathbf{Z}$ - space of integers

$\mathbf{L} = \{ x \in \mathbf{Z} \mid x \geq 0, x \leq 23 \}$

$\mathbf{D}$ - is a set of coordinates that can be occupied by mexor and axon cells

$\mathbf{D} = \{(x,y,z) \in \mathbf{L}^3\}$

$\mathbf{\Pi}$ – is a set of coordinates that can be occupied by paraneuron cells

$\mathbf{\Pi} = \{(x,y,z) \in \mathbf{D} \mid (x+1) \bmod 3 = 0, (y+1) \bmod 2 = 0, z \bmod 2 = 0\}$

We say that two cells $c_1=(x_1,y_1,z_1) \in \mathbf{D}$, $c_2=(x_2,y_2,z_2) \in \mathbf{D}$ are in a neighbour relation $N(c_1,c_2)$
if and only if
$((|x_2-x_1| \bmod 22)=1, y_2=y_1, z_2=z_1)$
$\vee (x_2=x_1, (|y_2-y_1| \bmod 22)=1, z_2=z_1)$
$\vee (x_2=x_1, y_2=y_1, (|z_2-z_1| \bmod 22)=1))$

Hence the cube is wrapped around and the cells placed on the opposite facets of the cube are in a neighbour relation.

3 Automatic creation of delay paths

The heuristics APF (Axonic Path Finder) we present includes three steps. First we look for the shortest path connecting the given points (x_0, y_0, z_0) and (x_1, y_1, z_1). If the path is longer than desired, an error is reported. Then we check if an odd-length delay is needed. If so, we add one. The last step is adding even-length delays until we reach the desired length.

Each of these steps can be accomplished in many ways. This is how the creation is done in current version of NeuroMaze:

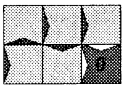
Step 1. To find the shortest path APF uses the Dijkstra algorithm. The path is a sequence of points ($p_i \in \mathbf{D}$), where $p_0 = (x_0, y_0, z_0)$ (the starting point), $p_n = (x_1, y_1, z_1)$ (the ending point), and for each $i \in \{0..n-1\}$ $N(p_i, p_{i+1})$. If many possible shortest paths exist, APF attempts to use the path that passes through a coordinate that can be occupied by a paraneuron cell as late as possible. In other words, it tries to maximize the number i so that $p_i \in \mathbf{\Pi}$ and for each $j>i$ $p_j \notin \mathbf{\Pi}$. The path is built of axon cells.

Step 2. APF checks if it needs an odd-length delay. If so, it takes the last cell from the path that can be replaced by a paraneuron. That is, we take such cell p_i that $p_i \in \mathbf{\Pi}$ and for each $j>i$ $p_j \notin \mathbf{\Pi}$. Now if there is any unoccupied cell c that is a neighbour of p_i and p_{i-2}, it changes the cell p_i to a neuron, cell c to axon, and then sets the directions so that it gets a path that is one clock longer, as seen on the figure below:

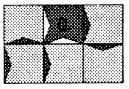
Before:  After: 

If there is no such cell, APF checks if there are three unoccupied cells s_0, s_1, s_2 such that $N(s_0, s_1)$, $N(s_1, s_2)$,

$N(s_0, p_i)$, $N(s_1, p_{i-1})$ and $N(s_2, p_{i-2})$. If it finds such cells, it extends the path to these cells and gets a path that is 3 clocks longer, as seen on the figure below:

Before:  After: 

If none of above is the case, APF takes the next cell that can be replaced by a paraneuron, and tries again. In the unfortunate case that it cannot use any of the above methods of extending the path, it tries to find three unoccupied cells s_0, s_1, s_2 such that $N(s_0, s_1)$, $N(s_1, s_2)$, $N(s_0, p_i)$, $N(s_1, p_{i-1})$ and $N(s_2, p_{i-2})$ for any $i>1$ and $i\leq n$, and cell s_1 can be replaced by paraneuron. If it finds such cells, APF extends the path, as seen below:

Before:  After: 

If APF cannot use this method it gives up and usually creates a path that is one clock longer than desired.

Step 3. Now APF is to extend the path to the desired length. For further analysis it will take the whole path $p_0..p_n$ unless it has added an odd-length delay. In this case, APF takes the path from p_0 to the place where the path forks. The extending procedure is quite simple: APF tries to find two unoccupied cells s_0, s_1 such that $N(s_0, s_1)$, $N(s_0, p_i)$ and $N(s_1, p_{i-1})$, where p_i is any of the cells that create the path. If it finds such cells APF extends the path by two axons. This is how it looks:

Before:  After: 

One of the most difficult parts of extending the axonic path is proper selection of p_i. In current version of *NeuroMaze* the last cell of the path is selected as p_i. Every next iteration i is decremented by 2. If i falls below 0, the last cell is assigned to p_i again, and so on, until the path reaches the desired length or until we run out of free cells. This method of selection of p_i results in concentrated paths, but may not always use the available space in most effective way.

4 Conclusions

The heuristics for automatic creation of axonic paths of desired length in a 3-dimensional cellular automata space we implemented in the *NeuroMaze 3.0 Pro* software tool opened a new chapter in PPNN engineering and the ATR's CAM-Brain Machine's exploitation. The most tedious and time-consuming part of the process of handcrafting Pulsed Para-Neural Networks has been automated. A visual perception system for the Neko 1.0 Mobile Robot run on the CAM-Brain Machine has been quickly prototyped under the *NeuroMaze 3.0 Pro*. This heuristics is also a key element of a future algorithm for automated conversion of $^{\beta}$PPNN designs into CAM-Brain Modules.

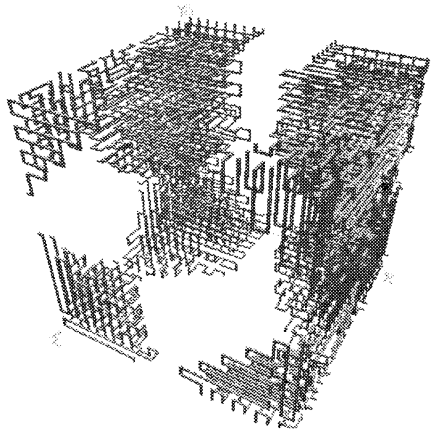


Figure 2. Example of automatically created axonic path inside a CA module, length=5000

Acknowledgements: D. Jelinski's work was supported by Gdansk Artificial Brain Research Initiative (GABRI). M. Joachimczak's research was conducted as a part of the *Research on Human Communication* supported by the Telecommunications Advancement Organization of Japan (TAO).

References

[1] Eeckhaut H & Van Campenhout J (2002) Digital Neural networks in the CAM-Brain Machine. Unpublished manuscript.

[2] Gers F, de Garis H & Korkin M (1997) CoDi-1Bit: A Simplified Cellular Automata based Neuron Model, *Evolution Artificiele 97, 22 Octobre 1997, Nimes, France*, 211-229.

[3] Joachimczak M (2002) A visual perception system for the Neko 1.0 Mobile Robot implemented on the CAM-Brain Machine. Research Memo ABG-RM#06, ATR HIS Artificial Brain Group: Kyoto.

[4] Korkin M, Fehr G & Jeffrey G (2000) Evolving hardware on a large scale, *Proceedings, The Second NASA / DoD Workshop on Evolvable Hardware, July 2000, Pasadena*, IEEE Comput. Soc., 173-81.

[5] Lee S-I (2002) Synthesis of Pulsed Para-Neural Networks (PPNNs) for an Action Drive of the Neko 1.0 Mobile Robot, Research Memo ABG-RM#04, ATR HIS Artificial Brain Group: Kyoto.

[6] Liu J (2002) *NeuroMaze 3.0 User's Guide*, ATR Human Information Science Laboratories: Kyoto.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Neko 1.0 – A Robotic Platform for Research on Machine Psychodynamics[1]

Tarun S. Tuli[2]
ATR International, Human Information Science Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
tstuli@atr.co.jp

[1] This research was conducted in cooperation of A. Buller and M. Joachimczak who also provided a critical review of this paper's draft

[2] On leave from the University of Calgary, Canada.

ABSTRACT

This paper deals with the construction and testing of a mobile robotic platform currently being used for research on machine psychodynamics. The robot, Neko 1.0 is a differential drive robot containing a wide variety of sensors and hardware that will be required for it to perform its designated research tasks. Control of the robot and sensory states retrieval is done remotely via a serial RF link. The specifics of how the sensors will be used to demonstrate the initial behaviors of Neko 1.0 will be discussed.

In addition to the physical robot being constructed, one of the requirements of Neko 1.0 was to have an associated software simulator to go along with it. This simulator matches the physical and control properties of the real robot as closely as possible. The main purpose of this simulator is to allow rapid offline evolution of the modules used to control the robot. Some of the techniques used for the simulator, as well as how the robots sensors suite was simulated will be discussed.

INTRODUCTION

Neko 1.0 is a differential drive mobile robotic platform that is being used for research on machine psychodynamics. According to the psychodynamic paradigm, an agent learns behaviors that could be helpful to reduce psychic tensions (see [1]). Neko 1.0, the physical robot, contains a wide variety of sensors and actuators that permit it to perform its designated research tasks, e.g. to interact with objects of interest (escape from a red ball, chase green ball), look around to reduce anxiety, or search for an object of interest to reduce boredom and avoiding collisions with obstacles all at the same time.

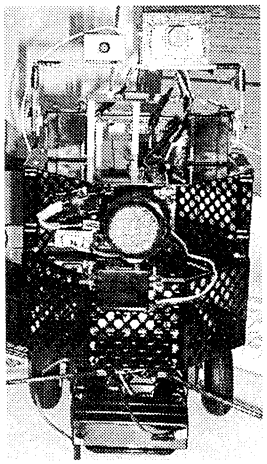


Figure 1: Neko 1.0

A host PC connected to a RF transceiver acts as the direct controller of Neko 1.0. The software contained on this PC allows other computers over a local area network (or even the internet) to connect to, and ultimately receive sensory data and control the robots actuators remotely. This allows Neko 1.0 to be controlled by a brain based on large scale Pulsed Para-Neural Networks (PPNNs).

In addition to the physical Neko 1.0, a software simulator is also being developed in parallel to mimic the actions of the real Neko 1.0. This simulator will play an important role in rapid offline evolution and testing of Neko 1.0's controllers when the physical model is not available for such tasks, or when then desire to run tests at faster than the real time scale is necessary.

HARDWARE DESCRIPTION

Neko 1.0 contains a wide variety of sensors, actuators and controllers [2]. Described in this section is a brief outline of each major component in the robot.

Mobile Platform – Neko 1.0's is based around a commercially available differential drive mobile robot platform. The platform, known as EasyBot uses two modified servomotors to propel the robot. The chassis contains three levels of room to place various electronics/etc. in the robot. Some minor modifications were done to this platform so that it would work with our desired sensor, actuator and controller suite.

Power System – The various components of Neko 1.0 all have different power requirements. The core of this power system consists of two 1.3Ah 6V gel cells (wired in series to get 12VDC) mounted in

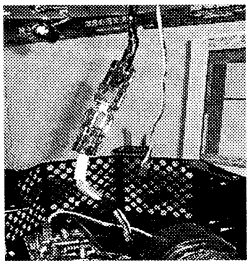


Figure 2: Upper control bus

Neko 1.0's base. Various voltage regulators are present in Neko 1.0 simply due to the widely varying power requirements of each component. For example, the ultrasonics and camera require a regulated 9VDC source, whereas the servo controller and radio transceiver need 5VDC. Power is distributed throughout Neko 1.0 by the use of a "Control-bus". This "Control-bus" is a set of DB9 connectors that carry both serial data and power to the various levels of Neko 1.0. This approach makes it very easy to dismantle Neko into its smaller sub-components since all that needs to be done is the removal of a few screws and disconnecting the "Control-bus". This is very useful for maintenance and modification that needs to be performed on the robot.

Robots Main Processor – The main processor of the robot is centered on Parallax's Basic Stamp II. The primary purpose of this processor is to distribute commands between the various devices in the robot and facilitate radio communication with a controlling host PC.

Slave Servo Controller – This controller takes servo commands from the main processor and continuously produces PWM signals for the robots servomotors. The advantage of having a slave controller doing this is so that the main processor does not need to worry itself in continually sending these signals (and wasting precious processing cycles).

Slave Sonar Controller – Neko 1.0 has a single sonar sensor mounted atop of servomotor that allows it to rotate from side to side. This sonar sensor simply returns distance values and is mainly useful for wall and large object detection. The sonar is only useful within a limited range is quite prone to noise and other anomalies. As such, bump sensors are also used on the robot for detecting walls.

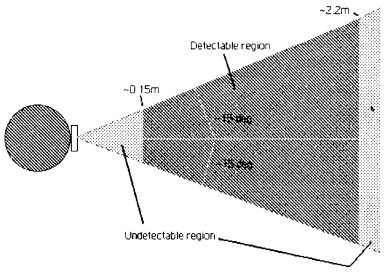


Figure 3: Sonar Useful Range

Radio Frequency Transceiver – Atop of the robot is the radio frequency transceiver. This transceiver has both a serial transmitter and a serial receiver. An identical unit is located on the PC side of the system as well. Apart from its limited data rate (1200bps), the radio frequency transceiver is also somewhat unreliable in providing communications. A software based verification scheme needs to be used to ensure exactly the correct; error free data is received at either end.

Perimeter Bump Sensors – Mounted on the robot are three perimeter bump sensors. Two are located on the front on the left and right sides. A third is mounted on the rear of the robot. These bump sensors are used in the robots wall avoidance algorithm, as well as relaying information back to the host network to act as a input into a tension generator for the psychodynamic agent.

CCD Camera – Mounted atop of Neko 1.0's head is a miniature CCD camera. This camera is connected to a host PC's frame grabber to have image processing done on it. In the case of early Neko 1.0 behaviors, it was used to find GREEN and RED balls [3].

The desired behaviors of Neko 1.0 are much too complex for its simple on-board processors to deal with. As a result, all higher-level "brain" commands of Neko 1.0 are sent to it via the RF radio transceivers. As one of the transceivers is connected to a normal PC's serial port, special software had to be designed to send the appropriate commands. This special program is DCOM enabled and allows any computer on a local area network (or even the internet) to connect to it, receive the sensory data from the robot and control its actuators. By doing this, different portions of the robots brain can be located in physically different locations.

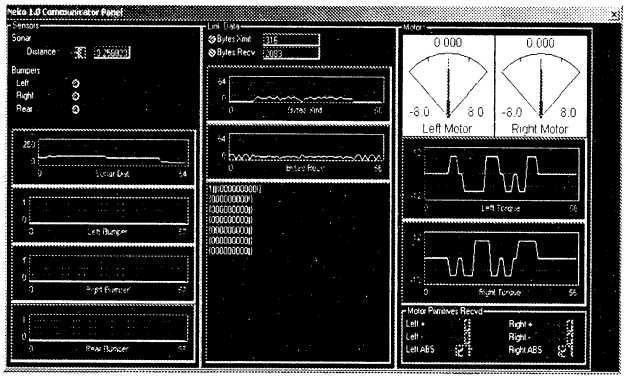


Figure 4: Communicator Panel

NEKO 1.0's INBORN BEHAVIORS

The majority of NEKO 1.0's behavior are controlled via higher-level brain components, and sent to the robot via the radio RF link. Some behaviors, such as simple wall avoidance however are contained within the robot itself. Using both the sonar and bumper sensors, the robot is able to navigate its environment without any

higher-level controls. A simple algorithm was put in place to accomplish this task.

If at any time the robot is instructed to travel in the forward direction from a higher-level control through the radio link, the robots sonar will continually rotate between the left, forward and right position taking distance measurements. A simple algorithm is then employed to determine whether the robot should change its course to avoid an obstacle.

When the sonar is facing the forward direction, it will take a distance measurement. If this distance is less than some preset minimum (in Neko 1.0's case about 25cm), it will set the *Forward* variable to TRUE. The sonar will then turn to the left and right positions and take another distance measurement. Based on this sequence, the robot will then make a decision as to whether or not it should turn based on the following rule:

If(*Forward* == TRUE) then
-Turn Left if *Left_Distance* > *Right_Distance*
-Turn Right if *Right_Distance* > *Left_Distance*
-Randomly Turn if *Right_Distance* == *Left_Distance*
AND continue turning until *Forward* == FALSE

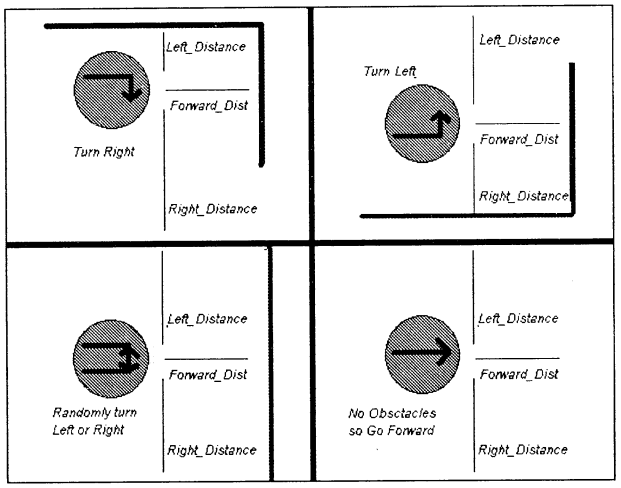


Figure 5: Wall Avoidance Algorithm

The bumper sensors are also used in the wall avoidance algorithm for situations that the wall isn't within the sonar's minimum detection range or poor reflection from the wall prevents the sonar from returning accurate values. If the Left bumper sensor is activated, the robot will turn right until it is no longer triggered. Similarly, if the Right bumper sensor is activated, the robot will turn left until it is no longer triggered. Finally, if the robot is moving in a backward direction and the rear bump sensor is activated, the robot will simply stop and await a new command.

During our testing of this algorithm, the robot was able to successfully avoid walls on its own without any higher-level intervention. The only issues were the bump sensor feelers sometimes getting caught on other objects within Neko 1.0's environment.

SOFTWARE SIMULATOR

Along with the physical Neko 1.0, a software robot simulator was also designed. This simulator serves several import purposes. The first is that it serves as a way to have multiple robots available all the time (currently there's only one physical robot, and isn't always available due to maintenance/battery charging/etc.). The second advantage is that the simulations can be run hundreds or even thousands of times faster than real time. This will be important when evolutionary algorithms are implemented into Neko 1.0's behaviors and tens of thousands of trials will need to be run in the shortest amount of time possible.

The software simulator consists of two separate interconnected (DCOM) components. The first is the Simulation Object. The purpose of this program is to simulate the actual physical characteristics of the robot including its dynamics as well as simulate the sensor suite. The second is the Simulation Board. The purpose of this module is to create the simulated environment in which the robot is to live in, and to update the Simulation Object about changing conditions in that environment (such as moving colored balls).

Simulation Board – The simulation board is activated by the robots Action Drive module, and appears to the rest of the control loop as the physical robot itself. It contains assessors for modifying the inputs into the robots simulated actuators. When the Simulation Board is activated, it creates a Simulation Object of the robot, sets its physical parameters (like weight, diameter, etc.), sets up the environment (where the walls are, colored objects are, etc.) and finally makes calls to the object when a simulation step should be executed.

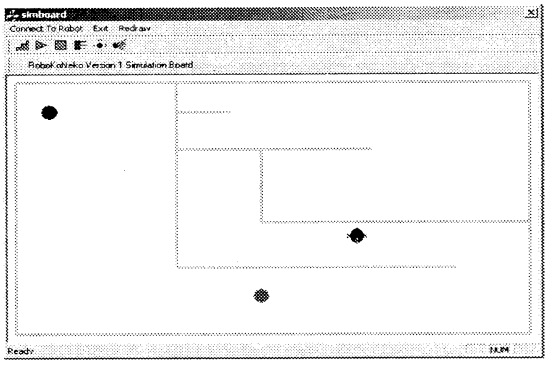


Figure 6: Simulation Board

Simulation Object – The job of the Simulated Robot object is to take motor primitives as an input and have perception data and static/dynamic data as an output. The object is instantiated by the Simulation Board object via a DCOM interface and communicates directly with this module, and only this module.

Simulating the Sensors – All of Neko 1.0's sensors were simulated using line intersection methods. Parameters

for each of the sensors are defined ahead of time (such as sonar max/min range, view angle, etc).

Sonar Simulation – To simulate the sonar, two rays exiting from the front of the robot are created. A line intersection routine is then run on these lines with all walls within the simulated environment. If the sonar ray lines intersect with any walls lines, the point of intersection is calculated. The resulting distance to the wall is the shortest of *d1* and *d2*.

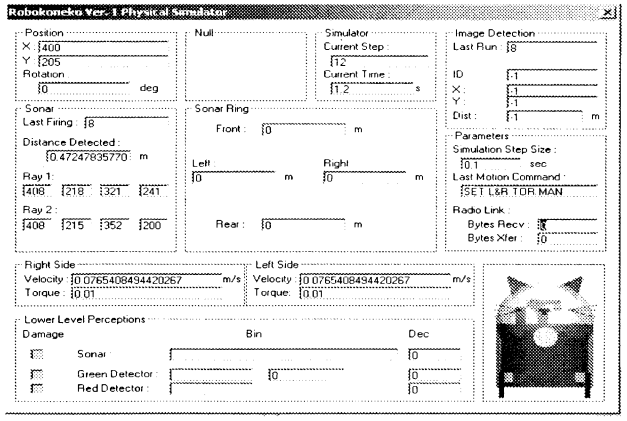


Figure 7: Simulated Robot

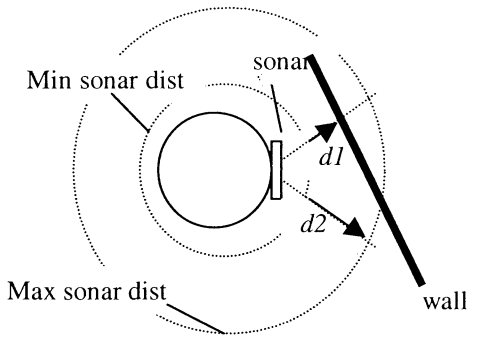


Figure 8: Sonar Simulation

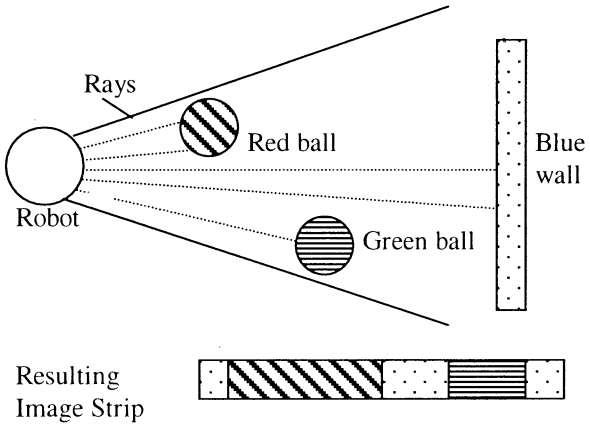


Figure 9: Camera Simulation

Image (camera) Simulation – Since simulating a traditional camera on the robot would prove to be a difficult and unnecessary task, a 1-dimensional setup was used instead. In front of the simulated camera on the robot is a set of rays emitting from it (a total of 256). These rays return the color of the first object that they run into. The same line intersection routines from the sonar were used for this. With this setup, say as the robot got closer to a green ball, the green ball would occupy more and more of this simulated visual field, and we would know what area the object was in front of the robot. This same phenomenon would be found in the physical robots camera; therefore we felt the 1-dimensional assumption to be acceptable for our purpose.

CONCLUSION and FUTURE WORK

The Neko 1.0 set of physical hardware platform and software simulator has proven to be an indispensable tool for our research on machine psychodynamics and Pulsed Para-Neural Networks (PPNNs).

A few modifications will need to be made to the robot and the simulator however to ensure it is as useful as it can be. First, the mechanical switch (somewhat unreliable) based feeler sensors should be replaced with a no-contact solution. Small infrared proximity sensors appear ideal for this task. Second, the radio link between the robot and the controlling PC should be replaced with something of higher speed and capacity. At its current 1200bps rate, commands can be delayed to arriving at the robot due to the inaccuracies inherent in radio communication and the fact that the processor ensures commands are received properly by constantly resending them. Several commercially available solutions are available for this task offering much higher data rates and reliability. Finally, the servo-based motors of the robot may be replaced with standard DC motors. The motivation for this change would be to help increase the maximum velocity of the robot. If the robot were operated in larger areas, this additional speed would be quite important.

Acknowledgements: This research was conducted as a part of the *Research on Human Communication* supported by the Telecommunications Advancement Organization of Japan (TAO). T. Tuli's work was made possible owing to the Co-op Japan Program.

REFERENCES

[1] Buller A. (2002) Psychodynamic Robot, *Proceedings, 2002 FIRA Robot World Congreess, May 26-29, 2002, COEX, Seoul, Korea*, 26-30.

[2] Tuli T.S. (2002) Neko 1.0: Physical Robotic Hardware Platform and Related Controls, *Technical Report TR-HIS-0008, ATR Human Information Science Laboratories, Kyoto.*

[3] Buller A., Harada Y., Joachimczak M., Lee S-I. & Tuli T.S. (2002) Neko 1.0. A robotic platform for a research on machine psychodynamics, *Research Memo ABG-RM#03, ATR Human Information Science Labs, Artificial brain Group, Kyoto.*

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Artificial Mind: Theoretical Background and Research Directions

Andrzej Buller and Katsunori Shimohara
ATR International, Human Information Science Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
{buller; katsu}@atr.co.jp

Abstract In this paper we present an artificial-agent-development doctrine being currently followed at the ATR Human Information Processing Laboratories, Kyoto by its Artificial Brain Group. Our objective is to build an agent that fits the spirit of Kansei technology of communication in a cyber society. We intend to make a machine think in a human-like way, learn to communicate with people and learn to extract knowledge of people's feelings. Our doctrine is based upon three key assumptions: (1) A guided epigenesis, which means that a number of computing units are interconnected into a permanently living artificial brain, where each of these units is subject to gradual improvements stimulated by its human care-taker. (2) An evolutionary short cut allowing isolated functions to be evolved separately with an advanced handcrafted structure as a starting point. (3) A psychodynamic architecture that adopts the idea that mental life is a continuous battle between conflicting psychic forces. The psychodynamic agent's "will" to act and learn comes from a hardwired principal task to always try to reduce psychic tensions. If a tension is reduced, this means a reward. If a tension stays at high level, this means a punishment. Based on such rewards and punishments, the agent's mind self-reinforces its own cognitive development. An emergence of a human-like thinking is expected when a set of complementary models of realities is successfully implemented. The set includes: Perceived Reality, Desired Reality, Anticipated Reality and Ideal Reality. It is believed that when achieving a certain level of complexity of its models, the agent will be able to autonomously develop relationships with people and understand the significance of these relationships. To accomplish our research tasks, we explore a number of areas including psychodynamic modeling, pulsed neural networks and programmable hardware.

1 Introduction

A project aimed to build a machine that thinks in human-like ways is not only an enormous scientific challenge, but also a social challenge. The building of an artificial brain of human-level performance is a sport for not only the open-minded, but also the brave and adventurous researchers who want to enjoy a thrill when venturing past something a bit magical and not commonly recognized as possible. In this paper we present an artificial-agent-development doctrine being followed currently at the ATR Human Information Processing Laboratories, Kyoto. The machine brain/mind theme is being explored by the Artificial Brain Group in the framework of the Emergent Communication Mechanism (ECM) project aimed to build an agent that fits to the spirit of Kansei technology of communication in a cyber society [25][15]. We expected the agent to increase its competence in self-initiated exploration of an environment, plan/execute actions while taking into account anticipated behaviors of people and/or other objects, and, finally, learn to communicate with people.

Our doctrine is based upon three key methodological pillars [9]:

i. **Psychodynamic Architecture (PDA)** that adopts the idea that mental life is a continuous battle between conflicting psychic forces. The psychodynamic agent's "will" to act and learn comes from the hardwired principal task to always try to reduce psychic tensions. If a tension is redueed, this means a reward. If a tension remains high, this means a punishment. Based on such rewards and punishments, the agent's mind self-reinforces its own cognitive development.

ii. **Guided Epigenesis (GEG)**, which means that a number of computing units, each representing functional blocks of the thinking agent are interconnected into a permanently living brain, where each of these blocks is subject to gradual improvements, implemented or inspired initially by its human care-taker, and later by the brain itself.

iii. **Evolutionary Short Cut (ESC)** allowing isolated functions to be evolved separately with an advanced handcrafted structure as a starting point.

In the following sections we discuss some theories that constitute the background for the doctrine, a considered scenario of artificial mind development and planned directions of our research.

2 Theoretical grounds

As it can be easily noted, the first methodological pillar of the discussed artificial mind building doctrine is strongly inspired by the psychodynamic theory of mind assumed as a reduced version of the psychoanalytic

theory of mind. The seemingly common agreement that human mental life is a continuous battle between conflicting psychic forces (wishes, fears, intentions) [20], and that most psychic processes go on out of the subject's control and awareness [19] is the most valuable legacy of Sigmund Freud. Another Freudian suggestion, that sexual impulse is the most essential of the psychic forces, as well as the concept of the Oedipus complex, have a limited, however quite significant number of adherents. While a theory of mind accepting the whole Freudian model is called *psychoanalytic theory*, a theory admitting the existence of conflicting forces and unconscious part of the psyche is called *psychodynamic theory* [18]. Although even the psychodynamic theory has poor empirical evidence, we treat it as an ingenious insight into one of the secrets that Nature protects the most jealously. Moreover, we note that lot of psychodynamic concepts can be implemented computationally.

An ability to imitate people's or other agent's behaviors is currently one of the most essential factors to qualifying a robot as an intelligent entity. It is recommended to build robots that can apply and control imitation as a part of their strategy to survive in social world, rather than being machines that are designed solely as imitating machines [17]. How though can we make a machine imitate without designing it as imitating one? We consider a tension-based solution. According to the psychodynamic view of the mind, a creature always tries to reduce its internal tensions. When a tension is being reduced, pleasure is felt (see [20:3]). We propose to hard-wire our agent's brain to create an image of itself behaving the same way as an already perceived agent, generate a tension dependent on the difference between the created and perceived image, and punish oneself when the tension stays high for too long time (cf. [15])

Freudian psychology does not provide any idea about the essence of psychic forces and mechanisms of their interactions. We propose to represent the forces as populations of identical copies of binary words representing temporarily concepts of interest. We call the binary words *memes* [13][8]. Working Memory is a theater in which populations of memes fight for domination and access to actuators. We have performed a number of simulation experiments with such kind of memory [12][14][8]. The results were comparable with psychological evidence about intrinsic dynamics of judgment provided by Andrzej Nowak and Robin Vallacher [24] and provided a psychological justification for Dynamic Fuzzy Set theory being developed to support modeling of psychodynamic phenomena [8].

The question that currently seems to divide the AI community the most is the question whether intelligent minds use a *world model* or not. While even in the newest textbooks a world model is assumed to be an indispensable part of intelligent systems [1][22], the idea of intelligence without representation is being promoted by authorities in advanced robotics [2]. Rodney Brooks and his colleagues collected psychological evidence supporting the statements that humans do not build an internal model of the entire visible scene and that there are multiple internal representations that are not mutually consistent [4]. Psychodynamic Architecture provides a compromised solution. It does not and cannot build an internal model of the entire scene because the mentioned Working Memory employs volatile swarms of memes representing only selected parts of the scene and those obtained on request from long-term memories that represent only the most necessary elements of perceived reality [6]. The model of perceived reality is being updated for its entire life and never becomes perfect. It is also never accessible as a whole because of the limited capacity of Working Memory. A modified version of the mechanisms of a child's mind development proposed by Jean Piaget [26] is being currently modeled toward machine long-term memory self-improvement. Another argument for representations and world models is the possibility to improve machine performance in pattern recognition via a mechanism proposed by Jerome Brunner, i.e. comparing an image of a newly perceived object with previous similar recognitions [5].

The results of experiments with simulated worlds populated by primitive creatures expected to evolve gave a limited hope that a thinking machine can be obtained via an evolutionary process initiated in an artificial computational environment [3]. On the other hand, there are numerous reports about successful employment of genetic programming to solve defined modest-sized problems [23]. Hence the idea of evolutionary short cut—isolated functions evolved separately with fitness function referring to behavior of an artificial agent as a whole based if necessary on a simplified simulation model of the agent and its world [9]. The functions can be coded in any way, however recently our favorite paradigm is based on Pulsed Para-Neural Network (PPNN) theory [10][11][16].

3 Artificial Mind Development Scenario

We assume that an artificial mind mind is to pass five stages in its development: (I) Reactive Loop, (II) Double Control Loop, (III) Two-Level Control System, (IV) Triple-Model-based Control System, and (V) Psychodynamic Agent. A scheme of fully developed Psychodynamic Agent is shown in Figure 1.

3.1. At stage I., **Reactive Loop**, The *Feature Extractor* and *Action Driver* constitute a primitive brain. The Feature Extractor converts the whole sensory pattern onto a smaller pattern consisting of variables called features. The Action Driver maps the features onto values of variables covering motor primitives.

3.2. At stage II., **Affective Loop**, The *Tension Handler* increments or decrements values of variables called tensions. Some tensions have internal sources, while some are induced via external stimuli. The overall brain

is hardwired to try to reduce the tensions in any available way, however it initially does not "know" what behavior can cause a reduction of which kind of tension (cf. [21]). Hence, given a tension the brain may try consecutive behaviors and treat the tension reduction as a reward. If after trying a particular behavior the tension keeps its level or increases, is interpreted as a punishment. Based on the reward and punishments, the brain can reinforce itself its learning what to do in face of a given tension. In this way, the agent can "discover" that, say, a basic "anxiety" disappears when it looks back. However, if it looks back and sees an approaching enemy, the tension "anxiety" will turn into the tension "fear" or so. It is believed that having a developed version of this mechanism the brain will self-reinforce its cognitive development.

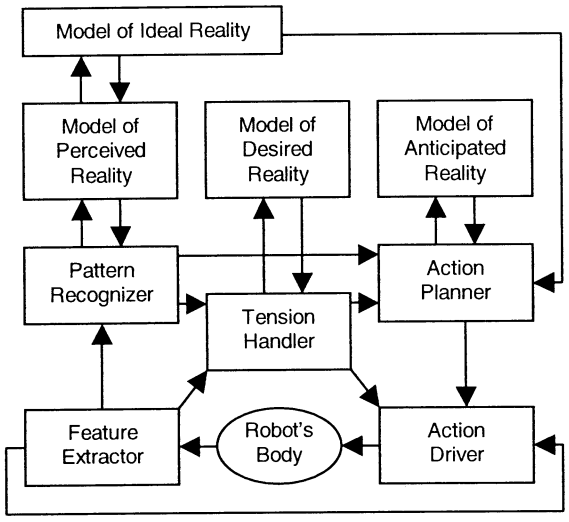


Figure 1. Psychodynamic Agent scheme (adopted from [9])

3.3. At Stage III., **Two-Level Control System**, The Tension Handler together with the *Pattern Recognizer* and *Action Planner* constitute the second level of control. The Pattern Recognizer, which sprouts from the Feature Extractor converts sets of features into complex data structures called recognitions. The Action Planner which sprouts from the Action Drive generates sequences of actions to be executed by the Action Drive toward a more effective reduction of tension volume. The Tension Handler acts at this stage based on both features and recognitions.

3.4. At stage IV., **Triple-Model-based Control System**, The *Model of Perceived Reality* sprouts from the Pattern Recognizer as a repository of previous recognitions to be compared with newly perceived objects. The *Model of Desired Reality* sprouts from The Tension Handler as a monitor presenting the environment and the agent itself in a fictitious configuration leading to a significant reduction of tension volume. The *Model of Anticipated Reality* sprouts from Action Planner as a monitor facilitating effective testing of generated candidate plans.

3.5. At stage V., **Psychodynamic Agent**, The *Model of Ideal Reality* sprouts from The Model of Perceived Reality and develops the knowledge about what is good or wrong, what is fair or unfair, what is moral or immoral. The knowledge is being acquired first of all through social interactions with the agent's caregivers. It is believed that when achieving stage 5 the agent will be able to autonomously develop relationships with people and understand the significance of the relationships.

4 Research directions

To accomplish our research task we have explore a number of areas. We have grouped the explorations onto four projects:

- **EgoBlocks Project**—for the design of a detailed block diagram of human-like Ego with its mechanisms of preventing the agent from unprofitable reactive ways of tension reduction;
- **NeuroMaze Project**—for the development of tools and methods for the construction of large-scale neural networks. These will be used in the physical implementation of an artificial Ego;
- **MemeStorms Project**—for the investigation of the dynamics of population of, so called, *memes* in distributed memories and a development of the Dynamic Fuzzy Set theory;
- **Quantrix Project**—for the development of concepts of new generation hardware for neural networks and robots' embodiments, as well as a quantum-logic version of the Dynamic Fuzzy Set theory.

The above projects should interact with one another towards such challenges as psychodynamic robots, rapid neural synthesis, distributed intellection engines and emergent thought incubators to be finally brought together as components of an artificial mind site christened *Volitron* [7] (Figure 2).

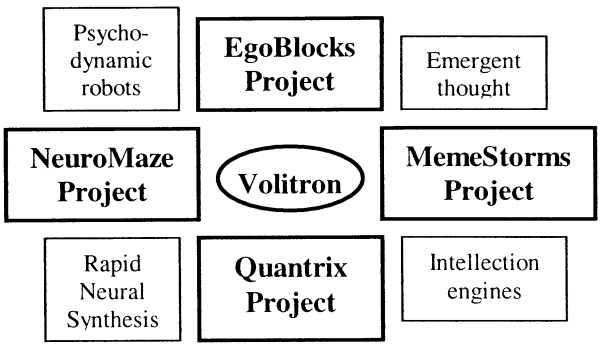


Figure 2. Research themes toward an artificial mind

Acknowledgement: This research was conducted as a part of the *Research on Human Communication* supported by the Telecommunications Advancement Organization of Japan (TAO).

References

[1] Albus J.S. & Meystel A.M (2001) Engineering of Mind: An Introduction to the Science of Intelligent Systems, New York: J.Wiley & Sons.

[2] Brooks R.A. (1991) Intelligence Without Representation, *Artificicial Intelligence Journal, vol. 47*, 139-160.

[3] Brooks R. (2002) *Flesh and Machine*, New York: Pantheon Books.

[4] Brooks R.A., Breazeal (Ferrel) C., Irie R., Kemp C.C., Marianović M., Scassellati B. & Williamson MM (1998) Alternative Essences of Intelligence, *Proceedings, 15th National Conference on Artificial Intelligence (AAAI-98), July 26-30, Madison, Wisconsin*, 961-968.

[5] Bruner J.S. & Anglin J.M. (1973) *Beyond the Information Given: Studies in the Psychology of Knowing*, New York: W.W.Norton & Co.

[6] Buller A. (2002) Psychodynamic Robot, *2002 FIRA Robot World Congress, Proceedings, May 26-29, 2002, Soeul, Korea*, pp. 26-30;

[7] Buller A (2002) Volitron: On a Psychodynamic Robot and Its Four Realities, In: Prince CG, Demiris Y, Maron Y, Kozima H, Balkenius C (Eds.) *Proceedings of the 2nd International Workshop on Epigenetic Robotics: August 10-11, 2002, Edinburgh, Scotland, Lund University Cognitive Studies 94*, 17-20.

[8] Buller A. (2002) Dynamic Fuzziness, *Proceedings, 7th Pacific Rim Conference on Artificial Intelligence, Tokyo, Japan, August18-22*, 2002, 90-96.

[9] Buller A (2002) In Quest of an Artificial Brain, *Proceedings of the Fifth International Conference on Human and Computer (HC-2002), September 11-14, 2002, Aizu, Japan*, 195-201.

[10] Buller A. (2002) Pulsed Para-Neural networks (PPNN): Basic concepts and definitions, *Technical Report TR-HIS-0007, ATR Human Information Science Laboratories, Kyoto.*

[11] Buller A. Eeckhaut H. & Joachimczak M. (2002) Pulsed Para-Neural Network (PPNN) Synthesis in a 3-D Cellular Automata Space, *The 9th International Conference on neural Information Processing, November 18-22, 2002, Singapore.*

[12] Buller A., Kaiser L., Shimohara K. (2001) Para-evolutionary Paradigm of Reasoning, In: Baba N, Jain LC, Howlett RJ (Eds.) *Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies (KES'2001), Part 1*, Amsterdam: IOS Press, 127-131.

[13] Buller A. & Shimohara K. (1999) Decision Making as a Debate in the Society of Memes in a Neural Working Memory, *The J. of 3D Images, 13* (3), 77-82.

[14] Buller A.. & Shimohara K.. (2000) Does the 'Butterfly Effect' Take Place in Human Working Memory? *The Fifth International Symposium on Artificial Life and Robotics (AROB 5th '00), January 26-28, 2000, Oita, Japan*, 204-207.

[15] Buller A. & Shimohara K. (2002) Kansei Processes in a Psychodynamic Agent, Proceedings, The 6th World Multiconference on Systemic, Cybernetics and Informatics, July 14-18, 2002, Orlando, Florida, USA, Volume VIII, 133-138.

[16] Buller A. & Tuli T.S. (2002) Four-Legged Robot's Behavior Controlled by Pulsed Para-Neural networks (PPNN), *The 9th International Conference on neural Information Processing, November 18-22, 2002, Singapore.*

[17] Dautenhahn K. & Nehaniv C.L. (2002) The Agent-Based Perspective on Imitation, In: Dautenhahn K. & Nehaniv C.L. (Eds.) *Imitation in Animals and Artifacts*, Cambridge MA/London: A Bradford Book/The MIT Press, 1-40.

[18] Drat-Ruszczak K. (2000) Teorie osobowości – podejście psychodynamiczne i humanistyczne. In: Strelau J. (Ed.) *Psychologia, T.2*, Gdańsk: Gdańskie Wydawnictwo Psychologiczne, pp. 601-652.

[19] Freud S. (1904/1995) Psychopatology of Everyday Life, In: *The Basic Writings of Sigmund Freud*, translated by Dr. A.A. Brill, New York: The Modern Library, 3-146.

[20] Freud S. (1920/1990) *Beyond the Pleasure Principle*, New York: W.W.Norton & Co.

[21] Freud S. (1923/1990) *The Ego and the Id*, New York: W.W.Norton & Co.

[22] Meystel A.M. & Albus J.S. (2002) *Intelligent Systems: Architecture, Design, and Control*, New York: J.Wiley & Sons.

[23] Nolfi S. & Floreano D (2000) *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines* Cambridge MA, London: A Bradford Book/The MIT Press, 2000.

[24] Nowak A. & Vallacher R.A. *Dynamical Social Psychology*, New York: Guilford Press.

[25] Shimohara K. (2001) Emergent Communication mechanisms as KANSEI Technologies, *Proceedings, International Symposium: Toward a Development of KANSEI Technology (KANSEI 2001), October 5-6, 2001, Muroran, Hokkaido, Japan*, 55-58.

[26] Wadsworth B.J. (1996) *Piaget's Theory of Cognitive and Affective Development: Foundations of Constructivism*, 5th edition, Addison-Wesley.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Tension-Driven Behaviors of a Mobile Robot. Early Experimental Results

Andrzej Buller, Yoshio Harada[1], Michał Joachimczak[2], Seung-Ik Lee[3] and Tarun S. Tuli[4]
ATR International, Human Information Science Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
{buller; y-harada; mjoach; silee; tstuli}@atr.co.jp

[1] Part-time at ATR, full-time at Osaka Sangyo University, Osaka, Japan.
[2] On leave from Gdansk University of Technology, Poland.
[3] On leave from the Yonsei University, Seoul, Korea.
[4] On leave from the University of Calgary, Canada.

Abstract We present the early results of research on tension-driven behaviors using a dedicated mobile robot Neko-1. His sensory system consists of a CCD camera, sonar and bumper sensors. Neko's "in-born" behaviors include chasing of "food", escaping from "enemies" and looking around from time to time. His acquired behaviors consist of exploring an area in search for new stimuli. All behaviors were aimed to reduce psychic tensions—scared, excited, anxiety and boredom. As for boredom, when no sensory stimulus was detected for a defined period of time, the action driver triggered one of hardwired behaviors, waited for some time, and when the tension level remained high, tried another trigger, and so on until the boredom dropped. In this self-reinforcement learning, the decrease of tension volume was a reward and unchanged tension was punishment. Anxiety was represented by three variables that referred to three separate directions: left, backward, and right. Every time Neko faced a direction, the value of the related variable dropped. This limited set of mechanisms appeared sufficient to make observers' feel that Neko behaved as if he were an animal in a cage.

1 Introduction

The objective of the research presented in this paper was to confirm the hypothesis that a robot's emergent behavior* can be driven by psychic tensions. According to the psychodynamic paradigm of intelligent agent building, an artificial brain should be hard-wired to continuously try to reduce tensions represented by dedicated variables [1]. The agent does not know any way a given tension can be reduced, so he must try several behaviors to learn which one is adequate in a particular case. It is believed that when tension-reduction mechanisms include appropriately developed models of perceived reality, desired reality, anticipated reality and ideal reality, the agent may possess the ability to communicate with other agents and, if they are people, understand their feelings [2][4]. We present early results in the subject. We used a dedicated mobile robot christened Neko-1 (and called simply Neko) [6] as a research tool.

We investigated four kinds of tension: scared, excited, anxiety and boredom. We assumed that behaviors leading to reduction of scared, excitation and anxiety are inborn. As for boredom, Neko had to "discover" what to do to stop being bored. Neko's brain architecture contains wiring such that every tension has a priority and only one behavior program is ever triggered at a time. These mechanisms together with several small imperfections of Neko's electrical and mechanical parts of his body caused emergence of a non-trivial, life-like behavior [4].

In the following sections we briefly describe Neko's body and environment, Neko's nervous system, the behavior Neko demonstrated, and conclusions.

2 Neko's body and environment

Neko-1 is a small differential drive mobile robot (measuring about 40cm tall and weighing just over 2kg). The robot possesses some of his own onboard processing; however the majority of its commands are relayed to it via a serial radio link connected to a host PC.

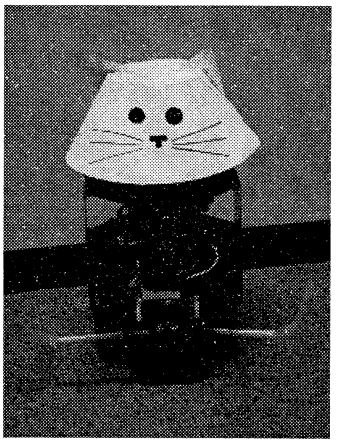


Figure 1: Neko-1

On board Neko 1.0 is a set of sensors and servo motors. The sensory system of Neko-1 consists of:

- Ultrasonic (sonar) sensor – This sensor is used primarily for wall avoidance. Mounted atop of a small servo motor, it has the ability to rotate approximately 90 degrees to either side of Neko's body for detecting the distance to large objects and walls. One limitation with this sensor has been its limited working range (minimum detection is 15cm, maximum detection is 220cm) and its inability to distinguish between different objects.
- Bump sensors – There are a total of three bump or "feeler" sensors mounted on Neko. One is located on either side near the front of the robot with the 3rd being mounted directly at the rear. The front two bump sensors are used in obstacle avoidance for objects that the sonar cannot see or missed due to its anomalies. The rear bump sensor is used incase Neko is driving backwards and bumps into a wall (which neither the sonar nor the camera have the ability to see).
- CCD Color camera – This sensor is essential to Neko's ability to live in its environment. The camera is connected to a host PC so that various image processing routines to be performed on it.

Four addition servo motors are also onboard Neko. A pair of these servos drives the left and right wheels of the robot and the remaining two allow the head/camera assembly to tilt up and down.

Neko has the ability, using his sonar and three bump sensors to avoid large obstacles and walls on his own (no control for this comes from its higher level "brain" components). These are hardwired into Neko's onboard computer and serve as its lowest level behaviors.

Neko in turn receives all of his higher-level brain commands from a host PC via a serial radio link. These commands, originating from the Action Driver portion of his brain are received by Neko's onboard processor and are processed and executing. Any commands coming from the serial radio link will have a higher priority than Neko's own built-in wall avoidance behaviors.

In addition, during our experiments, Neko was confined to a "cage". This cage, simply a wooden box measuring approximately 40cm x 150cm x 180cm was specially constructed to ensure Neko's drive wheels and sensors would operate optimally within it. It also served to confine his mobility to a small area so that observation would be easier.

3 Neko's nervous system

The brain of the robot consists of three functional blocks run on two separate but interconnected workstations. The blocks are: the Feature Extractor, the Action Driver and the Tension Handler. The first station hosts the Feature Extractor and Tension Handler, while the second station hosts the Action Driver as well as a software supporting radio-communication between the brain and Neko's on-board devices [3].

3.1. Control loops.

Neko's out-board brain together with an on-board computer constitute a "nervous system" organized into three control loops (Figure 2.) The lowest-level loop covers wall avoidance. It includes a sonar, two bumper switches, an on-board computer and an actuating block (i.e. servos providing propulsion for the right and left wheel). The next control loop was constituted by a full sensory block (CCD camera, sonar, bumpers), Feature Extractor, Action Driver and actuating block. The psychodynamic control loop is constituted by the Feature Extractor, Tension Handler and Action Driver.

3.2. Feature Extractor

In its current form, the Feature Extractor takes a sensory pattern and returns information on colored objects of interest as well as of the direction that the object is seen. The direction is provided as a binary word of length 5 in which at most one bit can equal 1.

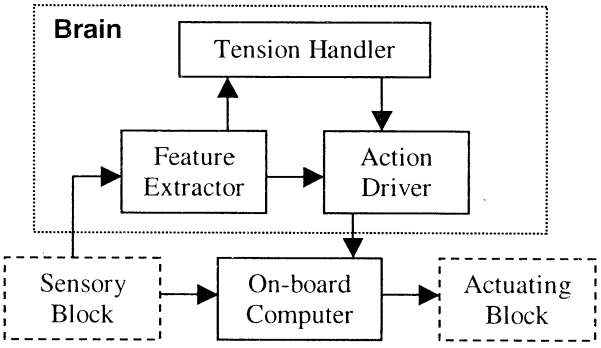


Figure 2: "Nervous system" of Neko-1

3.3. Tension Handler

The Tension Handler in its current form deals with four tensions—scared, excitated, anxiety and boredom—all represented by dedicated variables. The first two tensions come strictly from external stimuli. Neko gets scared every time he sees a red object and gets excited when he sees a green object. The next two tensions arise themselves. Anxiety is divided into three variables associated with three directions: leftward, backward and rightward. As for reduction of these tensions, the first two drop when the related stimulus disappear, a given part of anxiety drops when Neko turns to look at respective direction, and boredom drops when he notices an object of interest. The Tension Handlers task is also to decide which of the tensions is the first in line to be

reduced. Although psychologically plausible solution should adopt a kind of fighting for access to the Action Driver, in the current model we hard-wired a fixed hierarchy: scared on the top, and, further, excitation, anxiety and boredom. Hence, boredom-reducing behavior has a "green light" only when all other tension are low. Anxiety-reducing behavior is allowed when there are neither high scared nor high excitation, and excitation-related behavior may be blocked only by scare. Permission for continuing behavior related to reduction of a given tension can be at any time withdrawn in favor of more important tension. As for the three parts of anxiety, there is an additional family hierarchy, however it applies only to the priority of triggering a respective behavior in the case of conflicting needs, while permission for continuing behavior related to reduction of a given part of anxiety cannot be withdrawn in favor of another part of anxiety.

3.4. Action Driver

The Action Driver is assumed to execute a plan obtained from the Action Planner (not implemented in the current version of Neko) taking into account the feature pattern (FP) provided by the Feature Extractor and the tension pattern (TP) provided by the Tension Handler. The lack of the Action Planner dramatically reduces Neko's intelligence, but does not interfere in demonstrating a non-trivial life-like behavior. The Action Driver consists of a collection of generators of motor control signals and an associator selecting appropriate generator given the FP/TP. Every generator is responsible for one defined action. The associator is subject to reinforcement learning in which unchanged tension is punishment and discharge of a tension is a reward. Generators for the following behaviors have been implemented (Figure 2):

- The *Action Driver* consists of a hierarchy of three levels: *Behavior*, *Actions*, and *Primitive Motor Control*. The *Behavior* level, the highest of the hierarchy, has six behaviors. Each behavior is implemented by utilizing the motors directly or by the three actions defined in *Actions* level, depending on what the behavior is. The three actions in *Action* level are "turn left," "turn right" and "move forward." *Primitive Motor Control* generates motors signals for the two motors of the robot, each of which can accept positive or negative signals. This module can generate four different speed signals: high (H), Medium (M), Low (L), and Stop (S).
- The *escape behavior* module drives the robot to escape from something dangerous (red object). In order to do that, this module first drives the robot to turn 180°, and then, to move forward, and finally to turn 180° again to look to the original direction to check if he is still being followed by the red object.
- The *chase behavior* is for chasing green objects. This behavior is a sequence of reactive actions to the direction of the green object. Therefore, the chase behavior directly maps the FP to the speed controller.
- *Look left, look back, and look right behaviors* are for looking at the left side, backside, and right side respectively for a while and soon turn to the original direction. For example, the look left behavior drives the robot to turn left 90° to check if there is some object (red or green object), and then turns back to the original direction.
- The explore behavior drives the robot to explore the area. Currently it is just a moving forward behavior. However, the robot can explore the area as this move forward behavior is combined with his own internal wall avoidance behavior.

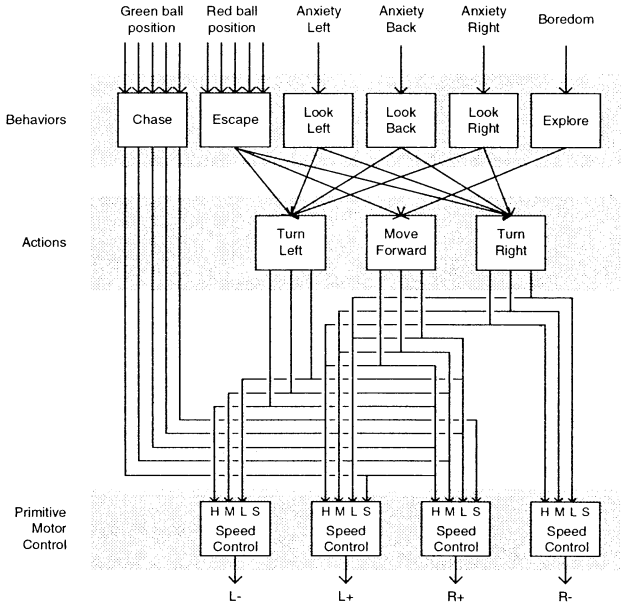


Figure 3: "Action Driver" architecture

4 Experiment

We tested both Neko's "in-born" behaviors that currently includes unconditional-reflexive avoidance of collision, escaping from "enemies" (red objects), chasing of "food" (green objects), and acquired ones driven by boredom. Our experiment were conducted in public during the ATR Open House, Nov. 7-8, 2002. Below we provide results of selected tests [3]:

Test #1.

The Tension Handler was switched off. The Neko Presenter, denoted further NP, shows Neko a green ball. Neko approaches the ball. NP gently moves the ball perpendicularly to the way Neko is facing. Neko changes

direction of his movement and chases the ball. NP quickly removes the ball form Neko's visual field. Neko stops moving.

Test #2.

The Tension Handler is switched off. NP shows Neko a red ball. Neko screams "Meow!" (in ca. 50% cases), turns left ca. 60°-150° (depending on state of batteries), moves forward for a couple of seconds, then turns right a similar angle. If a wall appeared in his way, the direction of escape was modified to avoid collision. The uncertainty about whether Neko produces a scream or not was caused by the limited capacity of radio-transmission of the Action Drive output signals.

Test #3.

The Tension Handler is switched on. "Anxiety" is disabled. No external stimuli. After ca. 5 sec. Neko screams "Meow!". After another ca. 5 sec. Neko turns left ca. 90°. After another ca. 5 sec. Neko starts going forward, however if a wall appears in his way, the direction of escape is modified to avoid collision. After the next ca. 5 sec. Neko stops, waits for ca. 5 sec., screams "Meow!", and so on.

Test #4.

The Tension Handler is switched on. "Anxiety" is disabled. No external stimuli. After ca. 5 sec. Neko screams "Meow!". After the next ca. 5 sec. Neko turns left ca. 90°. NP shows Neko the green ball. Neko chases the ball. NP quickly removes the ball from Neko's visual field. Neko stops moving. After c.a 5 sec. Neko turns left, skipping the "Meow!" previously performed in such a situation.

Test #5.

The Tension Handler is switched on. "Anxiety" is disabled. Action Driver's associator reset. No external stimuli. After ca. 5 sec. Neko screams "Meow!". After the next ca. 5 sec. Neko turns right ca. 90°. After the next c.a 5 sec. Neko starts going forward. NP shows Neko the red ball. Neko screams "Meow!" turns and escapes, and after a couple of seconds turns back and stops. After c.a 5 sec. Neko starts going forward, skipping the formerly performed "Meow!" and turning right.

Test #6.

The Tension Handler is switched on. "Anxiety" is enabled. Action Driver's associator reset. No external stimuli. After ca. 5 sec. Neko screams "Meow!". After the next couple of seconds Neko turns left ca. 180° and immediately ca. 90° right. He then starts going forward and changing direction to avoid walls. From time to time he stops and looks "nervously" around... Visitors of the ATR Open House admit that his behavior resembles an animal in a cage.

5 Conclusions

This limited set of mental mechanisms we implemented in the robot's nervous system appeared sufficient to make observers' impression that Neko behaved as if he were an animal in a cage. He also learned himself what behavior gives him a chance to reduce boredom. However, in these experiments the presenter decided when to show a boredom-reducing object. Nevertheless, in a labyrinth-like environment with hidden balls, he would seemingly "discover" that going forward gives him a better chance for meeting something interesting than "Meow!" or turning only. If Neko had to fight for survival, his emergent behaviors would be of great adaptive significance. Boredom-driven search for new stimuli gives a better chance to find food. Anxiety-driven looking around from time to time decreases the probability of being successfully attacked from behind. Currently new kinds of tensions seem to be good candidates for investigation: hunger for energy (forcing the robot to go to a recharging corner) and sadness caused by loss of companion, as well as with a joy in case of successful reunion.

Note: **Emergent behavior* here means that the overall behavior is caused by small set of internally combined simpler behaviors, as commonly admitted in robotics [5:156].

Acknowledgement: This research was conducted as a part of the *Research on Human Communication* supported by the Telecommunications Advancement Organization of Japan (TAO). T. Tuli's work was made possible owing to the Co-op Japan Program.

References

[1] Buller A. (2002) Psychodynamic Robot, *2002 FIRA Robot World Congress, Proceedings, May 26-29, 2002, Seoul, Korea*, pp. 26-30;

[2] Buller A (2002) In Quest of an Artificial Brain, *Proceedings of the Fifth International Conference on Human and Computer (HC-2002), September 11-14, 2002, Aizu, Japan*, 195-201.

[3] Buller A., Harada Y., Joachimczak M., Lee S-I. & Tuli T.S. (2002) Neko 1.0. A robotic platform for a research on machine psychodynamics, *Research Memo ABG-RM#03, ATR Human Information Science Labs, Artificial brain Group, Kyoto.*

[4] Buller A. & Shimohara K. (2002) Kansei Processes in a Psychodynamic Agent, Proceedings, The 6th World Multiconference on Systemic, Cybernetics and Informatics, July 14-18, 2002, Orlando, Florida, USA, Volume VIII, 133-138.

[5] Murphy R.R. (2000) *Introduction to AI Robotics*, MIT Press.

[6] Tuli T.S. (2002) Neko 1.0: Physical Robotic Hardware Platform and Related Controls, *Technical Report TR-HIS-0008, ATR Human Information Science Laboratories, Kyoto.*

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Synthesis of Behaviors of the Neko 1.0 Mobile Robot

Seung-Ik Lee*, Tarun S. Tuli†
ATR International, Human Information Science Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
{silee; tstuli}@atr.co.jp

Abstract This paper presents a preliminary result on the synthesis of behaviors of a psycho-dynamic mobile robot controlled by Pulsed Para-Neural Network (PPNN). Based on several building blocks defined for the synthesis, we synthesize PPNN models that map external stimulus or internal psycho-dynamic tensions to a sequence of spike-trains, which are transformed into voltages to control the robot's motors. An experiment shows the synthesis of behaviors drives the robot successfully such that the robot responses adequately to external stimulus as well as to internal psycho-dynamics.

1 Introduction

The system presented in this paper is being developed for the purpose of building a psycho-dynamic agent that has internal psycho-dynamic tensions, the brain of which should be hard-wired and continuously try to reduce tensions [1]. The Neko-1 agent used in this paper has a sonar sensor, a vision sensor (CMOS camera), and three bumpers around its body to protect itself and detect collisions as well as two motors combined with Frequency-To-Voltage Converters (FTVC). The brain of the robot consists of three main components (see Fig. 1): the feature extractor (FE), the action driver (AD), and the tension handler (TH)[2]. The FE provides situational information to the AD and TH from the robot's sensors. The TH is a reflection of psycho-dynamics and provides tensional information–currently, boredom and anxiety. The place where the synthesis of behaviors occurs is in the AD, which is in charge of ultimately driving the robot.

The control spike-trains for the motors are provided by synthesized Pulsed Para-Neural Networks (PPNNs). A PPNN is a set of simple processing units that change their states only at certain discrete instants of time known as clocks [3]. PPNN can be regarded as a graph with nodes of two different types as well as arcs that represent delays measured in clocks. Two kinds of nodes are defined. The first one is the MXOR that produces an output pulse at clock $t+1$ if and only if a single pulse entered it at clock t. The second type of node is Pulsed Para-Neuron (PNN). This node produces a pulse at clock $t+2$ if and only if the value of its internal counter at clock $t+1$ is equal to or greater than 2. The state of the counter for clock $t+1$ is the state of the counter for clock t plus the weighted sum of all the pulses incoming to the node at clock t.

Based on MXOR, PPN, and delays, this paper defines several functional building blocks that will facilitate the synthesis process, and then, describes several synthesized PPNNs of the AD to control the robot.

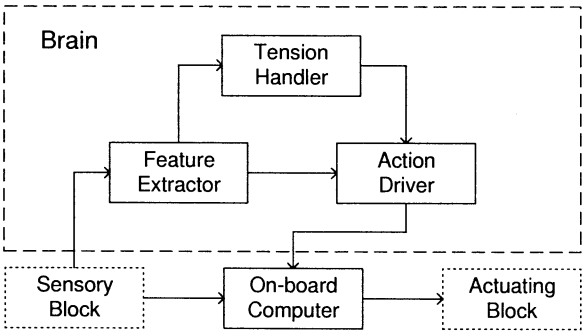
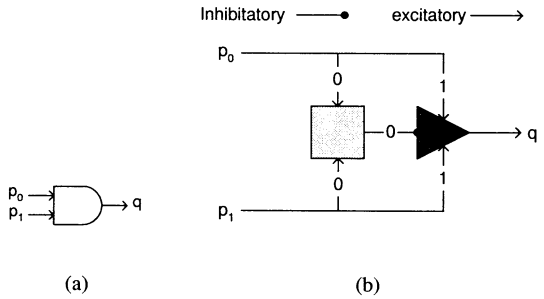


Fig. 1: Nervous system of Neko-1



(a) (b)

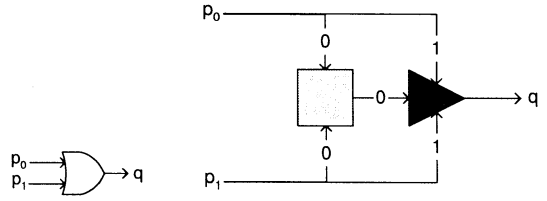
Fig. 2: An AND gate with two inputs spike-trains and one output. (a) AND symbol (b) AND implementation. If, for example, p_0 = 00101110..., p_1 = 01111000..., then q = 00000101000....

2 Basic Building Blocks

To facilitate the synthesis of PPNNs, several building blocks are defined based on MXORs, PPNs, and delays. These building blocks contribute to the construction of more complex PPNNs. Each delay can be either inhibitatory or excitatory. Through Fig. 2 to Fig. 6, it is assumed that inhibitatory delay (an arc with black circle) is with $weight = -1$ and excitatory delay (an arc with arrowed end) with $weight = 1$.

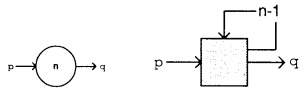
Fig. 2 shows an AND block that takes two pulses as inputs and generates a result of logical AND. The AND gate is composed of a MXOR (represented as a rectangle in Fig. 2), a PNN (represented as a triangle in Fig. 2), and several delays(the number of clocks necessary to propagate pulses from one node to other, represented as arcs with numbers). Fig. 3 shows a logical OR block that takes two pulses as input and generates a result of logical

*On leave from Yonsei University, Seoul, Korea.
†On leave from the University of Calgary, Canada.



(a) (b)

Fig. 3: An OR gate with two input spike-trains and one output. (a) OR symbol (b) OR implementation. If, for example, p_0 = 00101110..., p_1 = 01111000..., then q = 00001111110....



(a) (b)

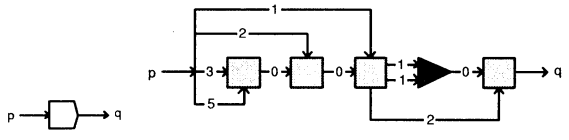
Fig. 4: An oscillator of frequency $1/n$ triggered by a single pulse. (a) Oscillator symbol (b) Oscillator implementation. If, for example, n = 8 and p = 00100000000000000000..., then q = 00001000000010000000100....

OR of the two input pulses. Fig. 4 shows an oscillator of frequency $\frac{1}{n}$. This oscillator generates a pulse every n clock cycles. Fig. 5 shows a 8-Pulser that generates eight and only eight clock–by–clock spikes triggered by a single pulse. Fig. 6 shows an ON/OFF switch that lets p pass through the block once the block is on. Once the switch is off, p cannot pass through such that q generates only null spikes .

3 Synthesis of Action Driver

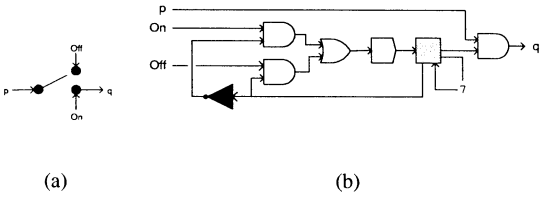
The Action Driver is assumed to execute a plan obtained from the Action Planner (not implemented in the current version of Neko) taking into account the feature pattern (FP) provided by the Feature Extractor and the tension pattern (TP) provided by the Tension Handler. The lack of the Action Planner dramatically reduces Neko's intelligence, but does not interfere in demonstrating a non-trivial life-like behavior.

The Action Driver consists of a collection of gener-



(a) (b)

Fig. 5: A generator of eight spikes with a single triggering pulse. (a) 8-Pulser symbol (b) 8-Pulser implementation. Let p =01000000..., then q =01000001111111100....



(a) (b)

Fig. 6: A On/Off switch that allows the original spike-trains (p) pass through only after it gets a triggering pulse via "On."

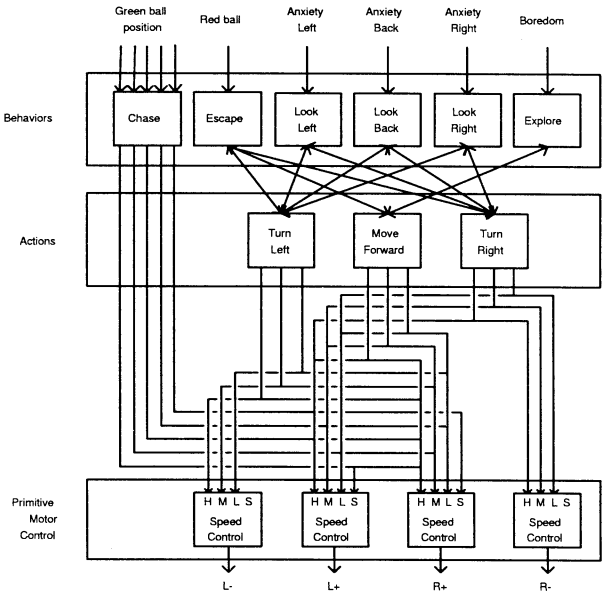


Fig. 7: Action driver architecture of Neko-1

ators of motor control signals and an associator selecting appropriate generator given FP/TP. Every generator is responsible for one defined behavior. The associator is subject to reinforcement learning in which unchanged tension is punishment and discharge of tension is a reward. Generators for the following behaviors have been implemented as in Fig. 7.

The Action Driver consists of a hierarchy of three levels: Behavior, Action, and Primitive Motor Control. The Behavior level has six behaviors. Each behavior is implemented by utilizing the motors directly or by the three actions defined in Action level, depending on what the behavior is. The three actions in Action level are "turn left," "turn right," and "move forward." Primitive Motor Control generates motors signals for the two motors of the robot, each of which can accept positive or negative signals. This module can generate four different speeds signals: high (H), Medium (M), Low (L), and Stop (S).

3.1 Behavior Level

Neko-1 has six behaviors that are "Chase," "Escape," "Look Left," "Look Back," "Look Right," and "Explore." These six behaviors utilize lower level actions and motor controllers to drive the robot. The "Chase" behavior is used to chase green objects. This behavior is different from the others in that it is a reactive behavior that directly maps FP to the motor controllers. The others cannot be implemented only with reactive control schemes so that they necessarily require a kind of internal memory

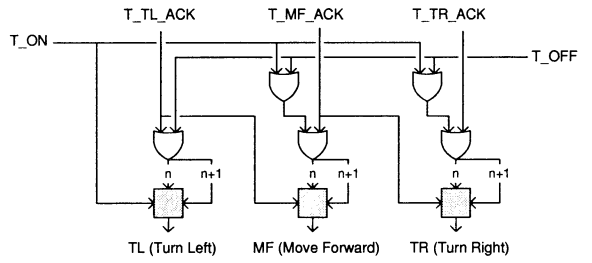


Fig. 8: "Escape" behavior's control module. Once triggered via T_ON, it immediately sends an "Enable" spike-train (10) to TL (Turn Left) action and disables the others. A signal on T_TL_ACK leads to an "Disable" spike-train (11) to TL and an "Enable" spike train to MF (Move Forward). Similarly, a signal on T_MF_ACK enables T_TR_ACK, while T_TR_ACK just disables "TR".

for the current status of behaviors and a communication mechanism with actions.

3.1.1 Escape Behavior

The "Escape" behavior module drives the robot to escape from something dangerous (red objects). In order to do that, this module first drives the robot to turn 180°, then to move forward, and finally to turn 180° again to let the robot check if still being followed by red objects. Provided a correct sequence of triggering signals timely ($T\_TL\_ACK \rightarrow T\_MF\_ACK \rightarrow T\_TR\_ACK$), the module in Fig. 8 can generate a sequence of "turn back," "move forward," and again "turn back" such that Neko-1 will escape and look back.

3.1.2 Look Left/Look Back/Look Right Behavior

Look left/back/right behaviors are for looking at the left/back/right side respectively for a while and soon turning to the original direction. For example, the "Look Left" behavior drives the robot to turn left 90° to check if there is some object (red or green object), and then turns back to the original direction. Fig. 9 shows a "Look Left" control module that generates a sequence of actions such that the robot first turns 90° left, and then turn back to the right, if timely provided with T_TL_ACK and T_TR_ACK in turn. "Look Back" and "Look Right" behaviors can be similarly implemented.

3.1.3 Chase/Explore Behavior

The "Chase" behavior is for chasing green objects. This behavior is a sequence of reactive actions to the direction of the green object such that the mapping from FP to the speed controller is very intuitive. Basically, the "Chase" behavior keeps the speed of the right motor high and low in the case of left motor, when the green object is on the left side of the robot. If the green object is in the right side of the robot, then the set-up of the motors is the reverse. In this way, the "Chase" behavior directly maps the FP to the speed controller (see Fig. 7).

The "Explore" behavior drives the robot to explore an area. Currently it just triggers "Move Forward" action. However, the robot can explore the area as this "Move Forward" behavior is combined with the robot's internal

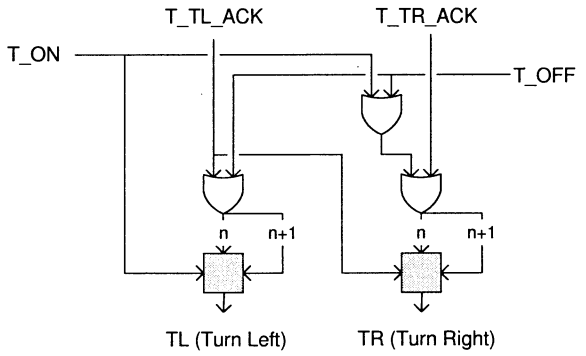


Fig. 9: "Look Left" behavior's control module. Once triggered via T_ON, it immediately sends an "Enable" spike-train (10) to TL (Turn Left) action and disables TR (Turn Right). A signal on T_TL_ACK leads to an "Disable" spike-train (11) to TL and an "Enable" spike train to TR. T_TR_ACK just disables "TR". T_OFF disables both TR and TL.

wall avoidance behavior and inherent uncertainty of the motors and sensors.

3.2 Action Level

Neko-1 currently has three actions defined as "Turn Left," "Move Forward," and "Turn Right". All of these actions have two basic functions. One is to generate appropriate spike-trains for the "Primitive Motor Control" to control the robot. The other is to send appropriate information on the current progress of action execution so that the behaviors in the upper level can control the actions in the lower level. "Turn Left" and "Turn Right" action send information on how much degrees the robot has turned to the behavior that activated these actions. In the case of "Move Forward," information on how much time has passed is sent to upper level behaviors. To make these possible, all these actions have inputs for time elspase.

3.2.1 Turn Left/Turn Right

"Turn Left" or "Turn Right" actions drive the robot to turn to left or right at a given speed once it is enabled. Three different turning speeds are predefined so that the robot can turn at high, middle, and low speed. Fig. 10 shows a synthesized PPNN for "Turn Left." It is assumed that only one of the inputs for choosing speed is triggered at a time. This PPNN generates an acknowledgment pulse every time the robot turns left 90° so that the behavior module that triggered it can control the robot appropriately , considering the current state of execution. As Neko-1 does not have any sensors that give information on its position or direction, an input for time information is periodically triggered to measure how much the robot has turned. This module has three connections with speed controllers of the lower level. Each of these connections drives the robot to turn left at a given speed. Spikes of "10" is a enable signal for the speed controllers and "11" is for disabling the controllers.

3.3 Move Forward

The "Move Forward" action is to drive the robot to move forward (see Fig. 11). Three inputs are used to control

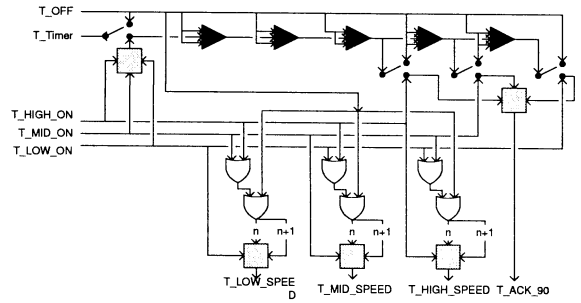


Fig. 10: "Turn Left" action's control module. T_HIGH_ON, T_MID_ON as well as T_LOW_ON are enabling pulses that drive the robot to turn left at different speed while disabling the other two. Timing pulses are periodically provided via T_TIMER such that the module can produce a T_ACK_90 pulse when the robot turns left about 90°.

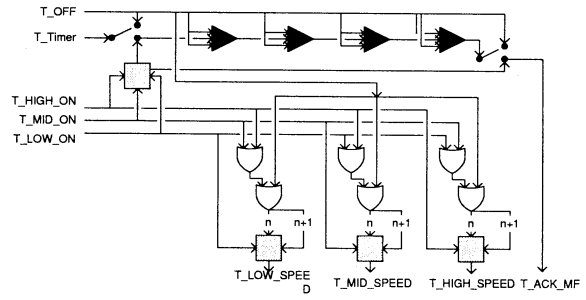


Fig. 11: "Move Forward" action's control module. T_HIGH_ON, T_MID_ON as well as T_LOW_ON are enabling pulses that drive the robot to move forward at different speed while disabling the other two. Timing pulses are periodically provided via T_TIMER such that the module can produce a T_ACK_MF pulse when the robot moves forward for a fixed time.

the robot at high, middle, and low speed, respectively. This module generates acknowledge information whenever the robot moves forward for a fixed time.

4 Experiment

Neko-1 has six defined behaviors described in section 3.1. These are escape from "enemies" (red objects), chasing of "food" (green objects), three anxiety-driven behaviors (look left/right/back), and boredom-driven explore behavior. Together with these behaviors, Neko-1 has another inherent behavior of collision avoidance to avoid obstacles or walls. The experiment was conducted in public during ATR Open House, Nov. 7-8, 2002.

Neko-1 shows appropriate behaviors when it is presented with external stimulus (presented with red objects and/or green objects). When presented with green objects, the robot follows and chases the objects successfully. With red objects, Neko-1 turns to the opposite direction of the objects, then moves forward for a while, and turns back again to confirm if the red objects still follow itself. If the objects still try to catch the robot, then the robot escapes again from the red objects until the objects do not follow the robot anymore.

Even when no external stimulus is presented, Neko-1 sometimes turns left/right/back to make sure that there is no dangerous objects, or sometimes moves looking for something interesting like green objects. As there is no external stimulus, theses behaviors are a reflection of Neko-1's internal tensions and seems like the behaviors of real-life animals who have the ability to determine their behaviors based on their internal mind.

5 Concluding Remarks

In this paper, we have conducted the synthesis of behaviors for the Neko-1's control modules. First, Basic building blocks of PPNN have been defined in order to facilitate the synthesis process. Based on these building blocks, the synthesis of the behavior modules for the robot have been conducted. The experimental results of the robot during the ATR Open House shows that the robot has psycho-dynamic behaviors based on internal tensions of the robot, as well as appropriate responses to external stimulus.

However, we have also identified several problems. First, the synthesis process is really a kind of time consuming and trial-error-based process even with several predefined building blocks. This property results from the fact that every procedure for synthesis requires a certain level of creativity. In addition, some parts of synthesized PPNN requires strict delay lengths. If delays in such parts were configured just a little bit differently (say delay length 4 instead of 3), then the whole spike-trains of the PPNN would be completely different.

It seems that this kind of time consuming process is a good candidate for the application of evolutionary methods. However, the whole search space of the PPNN synthesis process seems to be too wide to be solved only by evolutionary methods. Therefore, the combination of human's creativity and the global search capability of evolutionary methods is necessary. Furthermore, as the two psycho-dynamic tensions are insufficient for the robot to behave just like animals in real life, more work should be done on other tensions. Good candidates for new tensions can be hunger for energy or emotion.

Acknowledgments

This research was conducted as part of "Research on Human Communication" with funding from the Advancement Organization of Japan.

References

[1] Buller A., "Psychodynamic robot," *2002 FIRA Robot World Congress Proceedings*, pp.26-30, 2002.

[2] Buller A., Harada Y., Joachimczak M., Lee S.-I., and Tuli T.S., "Neko 1.0. A robotic platform for a research on machine psychodynamics," *Research Memo ABG-RM#03, ATR Human Information Science Labs, Artificial Brain Group*, 2002.

[3] Buller A. Eeckhaut H. and Joachimczak M., "Pulsed Para-Neural network (PPNN) synthesis in a 3-D cellular automata space," 9^{th} *International Conference on Neural Information Processing (ICONIP'02)*, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Can a robot empathize with people?

Hideki Kozima Cocoro Nakagawa Hiroyuki Yano
Communications Research Laboratory
Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
E-mail. {xkozima, kokoro-n, yano}@crl.go.jp

1 Introduction

This paper proposes a robotogenetic model of empathetic understanding of others' behavior as one of the capabilities required for engaging in the human social activities. The term "robotogenetic" means that we implement possible ontogeny (i.e. developmental process) of the social capability onto a robotic embodiment with a certain phylogenetic background (i.e. innate prerequisites). As a possible embodiment we built infant-like upper-torso humanoid robots, Infanoids [Kozima 02] (shown in Figure 1); we have been trying to bring them up through the interaction with the social environment, especially with human caregivers.

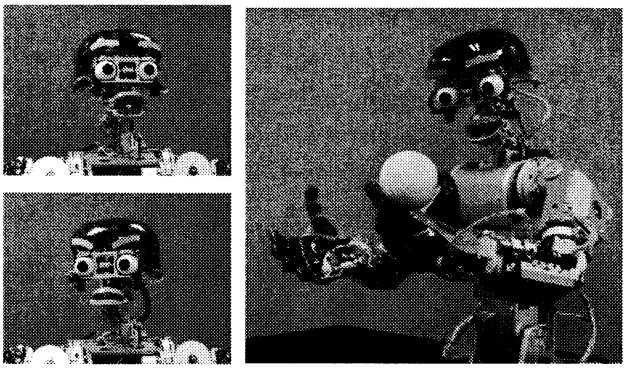


Figure 1. Infanoid, the epigenetic robot.

2 Empathy

Empathy, which is the ability to imagine oneself in the position of another person, gives us access to the person's mental states (e.g. emotion, desire, and belief), thus plays an indispensable role in our social interaction and communication [Dautenhahn 97]. Considering its role in the ontogeny of social capabilities, empathy gives us an opportunity to learn about how human mind works (i.e. folk psychology or a theory of mind). Not having enough access to others' mental states, autistic children have difficulty in developing their ability to understand others' mind, and so display certain disorders in pragmatic communication in their daily life [Hobson 93].

To access others' mental states, the robot has to be able to observe another person's behavior: how the person perceives and acts on the environment. From these observations the robot can vicariously experience the person's behavior and imagine what happens in the person's mind, which enables the robot to predict or control the person's behavior to some degree. This is the fundamental process of empathetic understanding of others' mental states. The following two sections describe "joint attention" and "mirror system" as possible mechanisms for observing others' perception and action.

Figure 2. Joint attention with a human caregiver.

3 Joint attention

Joint attention is the act of sharing each other's attentional focus. It plays an important role in understanding what the person is perceiving from the environment. Spotlighting the objects and events being attended to by the interactants, joint attention creates a shared context in front of them. The shared context is a subset of the environment, the constituents of which are mutually manifested among the interactants. The context plays a major role in reducing the computational cost of selecting and segmenting possible referents from the vast environment and in making their interaction coherent.

We have implemented on the robot a primordial capability of joint attention. The robot captures the

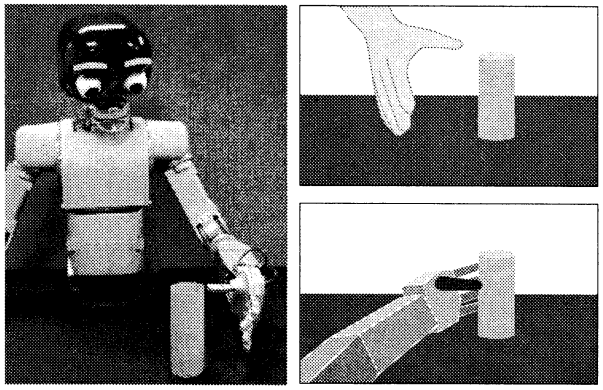


Figure 3. Self-other mapping mediated by objects.

direction of the caregiver's attention by reading the head direction from the video image taken by the CCD cameras; the robot searches in that direction and identifies an object or event of the careviver's attention (as shown in Figure 2). Occasionally the robot diverts its attention back to the caregiver to check if he or she is still attending to the object (i.e. referential-looking) and often looks into the facial expressions for his or her emotional attitude towards the object (i.e. social referencing).

4 Mirror system

To understand another person's action, the robot has to have a "mirror system" that maps the person's action (seen as a video image) onto the robot's action (as a motor program to execute). It is widely assumed that the mapping is a geometric transformation between bodies in the visual and motor spaces [Arbib 02].

We claim that the mapping is not a geometric transformation but a functional correspondence between the person's action on an object for producing an effect and the robot's own action on the same or similar object for a similar effect [Kozima *et al.* 02]. This functional correspondence is learnable because the person and the robot have functionally similar bodies, and so both tend to utilize an object's affordance in a similar way. Associating how another person utilized an object's affordance and how one utilized the object's affordance (as illustrated in Figure 3), one can learn the functional correspondence between the person's action and one's own action.

5 Conclusion

The prerequisites for empathetic understanding of others' mind are (1) joint attention capability to experience another person's perception and (2) mirror system to vicariously experience the person's action. These are also the prerequisites for imitative learning [Tomasello 99], which is the major driving force for human socialization. As illustrated in Figure 4, joint attention and mirror system enables us to vicariously experience another person's behavior and to estimate the mental state associated with the behavior, which will then be projected back onto the person.

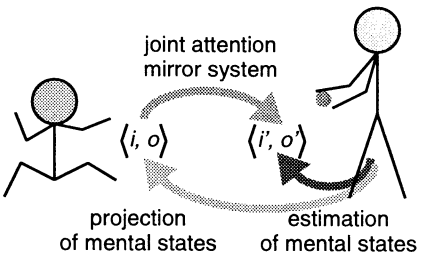


Figure 4. Understanding other's mental states.

We are currently implementing on Infanoid the learning model of the mirror system and planning to link it with the joint attention capability. We use the robot as a verifiable test-bed, on which we can modify both the prerequisite and the developmental process, and test the model in the real social environment.

References

[Arbib 02] Arbib, M.: The mirror system, imitation, and the evolution of language, In Nehaniv, C. & Dautenhahn, K. (eds) *Imitation in animals and artifacts*, MIT Press (2002)

[Dautenhahn 97] Dautenhahn, K.: I could be you: The phenomenological dimension of social understanding, *Cybernetics and Systems Journal*, Vol. 28, pp. 417–453 (1997)

[Hobson 93] Hobson, R. P.: *Autism and the development mind*, Psychology Press (1993).

[Kozima 02] Kozima, H.: Infanoid: A babybot that explores the social environment. In Dautenhahn, K. *et al.* (eds) *Socially intelligent agent: Creating relationships with computers and robots*, Kluwer Academic Publishers, pp. 157–164 (2002)

[Kozima et al. 02] Kozima, H., Nakagawa, C., and Yano, H.: Emergence of imitation mediated by objects. In *Proceedings of the 2nd international workshop on epigenetic robotics*, pp. 59-61 (2002)

[Tomasello 99] Tomasello, M.: *The cultural origins of human cognition*, Harvard University Press (1999)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Toward Machine Intuition: A Way-Finding Without Maps or Coordinates

Juan Liu
College of Information Science & Engineering, Central South University,
Changsha, Hunan, 410083, P. R. China, ljcic@263.net

Abstract

This paper concerns the problem of the construction and utilization of internal spatial representations for mobile robot navigation in a connectionist way and explores mechanisms underlying route-learning behavior. The proposed network is based on the belief that conversions between spatial and spatio-temporal patterns are involved in the navigation knowledge learning and retrieving process in which a primitive procedural/episodic memory is involved. This is like a story-based approach for representing and understanding the agent and the environment. Such a mechanism looks as if it is an intuitive system, not an analytic one. It is proposed that as an indispensable part of human intelligence, intuition should also be implemented in artificial systems. By combining logical and intuitive knowledge, we can get an integrated whole that is more profound and penetrating.

Key words: machine intuition, spatio-temporal patterns, environmental context, mobile robot, cognitive maps

1 Introduction

Those who build mobile robot navigation systems usually construct a spatial model of physical environments as a metric or topological maps and then cyclically find the robot's location in the map and execute a planned path leading to the goal [7]. Although the bird's eye mapping strategy is prevalent in a number of applications—in shipping, aviation, expedition and so on, it appears *not to be the only* mechanism underlying human being's way-finding ability. This is especially true in a familiar environment where people or animals shuttle between the same places everyday. We are not holding GPSs, compasses or range finders to localize ourselves. We seldom ask ourselves such questions as "what are my current coordinates? ", "what is the next step in the planned path?" We follow the right route, *not reactively*, but *intuitively* selecting the direction according to our intention and perception. How do humans quickly learn the spatial knowledge and recall it when necessary having neither a precise map nor Cartesian coordinates?

Many biologically inspired navigation strategies are based on models built to reflect the worldview of the agent. Some of the models (for example, a model of hippocampus [1] or place cells [6]) are based on the concept of "Cognitive Maps" introduced by Tolman [8] as a way to interpret findings in path selection behaviors of rats that used some form of internal spatial representation. Graph search algorithms or reinforcement learning are used to find its way. Those systems utilizing instant sensor readings are mostly confined to small open fields or carefully designed mazes where sensor information remains unambiguous. Nevertheless, the problem of learning and retrieving of internal spatial representation is far from being resolved.

This paper presents a mechanism supposedly underlying biological route-learning behavior. Grounded on the fact that learning, recognizing and recalling temporal patterns contribute greatly to human intelligence, we conceive that robots may also learn spatial knowledge from the regularity of temporal sequences of sensory and action flows. A connectionist model, Temporal Sequence Processing Network (TSPN), is proposed to memorize and correlate the spatial-temporal experiences of mobile robots, including its sensory inputs, past and current behaviors, and to effectively retrieve them when exposed to similar stimuli in later runs. Unlike the models using place cells, the presented approach assumes that the network itself is not a topological graph of the environment. In TSPN-based systems spatial information is implicitly coded in temporal characteristics of cells and connections that are constructed at run-time when the robot is exploring the environment. Without any a priori environmental model and global localization information, the robot learns goal-directed cognitive map from its own viewpoint. The navigation strategy is similar to that underlying our intuitive way-finding behaviors, which does not depend on maps and coordinates.

Section 2 describes TSPN architecture and its relevance to the navigation problem. In Section 3, its working mechanism and experimental results are provided. Its possible generalization and potential as a substrate of machine intuition and procedural/episodic memory are discussed in Section 4.

2 A connectionist model for navigation

Our strategy is to provide robot with as few "tips" as possible and put it in the way-finding situation. Without reference points and coordinates and being equipped with only proprioceptive sensors to learn egocentric

relationships, what else can the robot utilize to know more about the external world and navigate from one place to another? What can influence our way-finding action? The proposed list contains:

- Environmental cues (sensor inputs)
- Intention (e.g. the destination)
- Instinctive behavior (innate obstacle avoidance and other safeguard actions based on hardwired mechanisms)
- Spatial knowledge (learnt from self experience or other information including maps, maintained as long-term memory)
- Short-term memory of recent action and perception. (This factor is indispensable for thinking process, but is often ignored in many navigation systems and connectionist models.)

When people walk through a territory, the spatial structure of the world is transformed into spatio-temporal patterns that are perceived by our sensors and stored in short-term memory. For example, assuming the robot's translation velocity and rotation velocity are uniform during the learning phase, the duration of an action can reflect the distance or the change of its heading. So temporal sequences can be regarded as another kind of spatial representation. In order to maintain these patterns in long-term memory, another transformation is involved to convert the spatio-temporal pattern into a spatial one, i.e. to a neural network with cells and connections, as those in the grooves of a phonograph record. Most often they are dormant, but can be activated by internal desire and perception of external stimuli. The cells and strengths of various synapses predispose the system to produce various spatio-temporal patterns. A sequence of cell firing patterns represents a specific route and its environmental contexts. The construction and retrieval of an internal world representation is a "spatial world → spatio-temporal patterns → spatial representations → spatio-temporal patterns" procedure.

In the proposed architecture, no single neuron represents the world or a route, however any segment of memory is likely to involve a committee of neurons. "A single neuron, like any one key on the piano, is likely to play different roles in different melodies" [2]. Memories need internal intention or external stimuli to activate. They are human unique experiences that seem to be available to machines as well. When a network becomes a memory, the sense of the past history derived from it may contribute to the creation of the robot's self.

Neuron activations are assumed to be system outputs and acting decisions, or thoughts. Thoughts are combinations of sensations and memories that can also be considered as movements that have not yet happened. Their execution is influenced by innate safeguard behavior. A dangerous action will be inhibited.

In conclusion, an adequate connectionist model for "intuitive" navigation should have two basic features: (1) temporal characteristics and ability to deal with spatio-temporal patterns, (2) ability to integrate robot's sensation and action into an interrelated whole.

In the proposed model, TPSN, cells are differentiated into specialized groups to represent various sensory inputs and motor actions. Sensory cells are activated by range sensor readings or visual inputs, while the firings of motor cells and their combinations produce diverse tracks [5].

For the temporal demand, a decay mechanism is used as a bridge between spatial and temporal features. When a cell is excited at a given clock, it will not be reset at the next clock. The self-connection $tp_{ii} \in (0,1)$ causes the decay of activity of cell i at each time step by the same factor, which enables associations between cells reflecting time delay. If cell i is not activated or inhibited at time t,

$$y_i(t) = y_i(t-1) \cdot tp_{ii}. \quad (1)$$

To construct an interrelated network containing the robot's experiences, we introduce the concept of *environmental context* in TSPN. Meaningful sensor inputs and past actions during the task, i.e. cells with non-zero activities, constitute the context of current new motion. So in the network construction process, once a motion cell is activated, all other active cells in TSPN get connected to this cell and become its pre-synaptic cells. The connection weight tp_{ij} represents the time that passed between firing of cell j and activation of cell motion cell i. TSPN is a non-symmetrical recurrent network in which tp_{ji} and tp_{ij} may be not equal. The weight tp_{ij} is:

$$tp_{ij} = y_i^M(t) \cdot y_j^X(t) \quad (2)$$

where $j \neq i$, j is presynaptic to i and $X \in \{S, M\}$. S denotes sensor cells, while M denotes motion cells.

If the robot takes another action, a new motion cell is generated. Formation of TSPN is an incremental process, which enables learning throughout the entire lifetime.

3 Working mechanism and early experiments

After the exploration of a territory, the robot has gained some knowledge and experiences about the world, which is represented by a structure of a TSPN network, a cluster of cells and their connections. How can the static memory be evoked?

When a specific environmental context is presented again, related motion cells are fired in a precise temporal sequence retrieving the learnt motion chain. The activation is a cooperative effort of pre-synaptic cells. The weight of a connection determines when the effect of the pre-synaptic node firing will be maximal on the postsynaptic node. Once external stimuli of a cell cause its potential $V_i(t)$ to rise above a threshold θ, the cell can be activated. The activity of motion cell i is:

$$y_i^M(t) = x_i^M(t) + tp_{ii} y_i^M(t-1) + H[V_i(t) - \theta] \quad (3)$$

$$V_i(t) = \frac{1}{N_i} \sum_j C_{ij} \phi(y_j^X(t), tp_{ij}) \quad (4)$$

where $H(X) = \begin{cases} 1 & X \geq 0 \\ 0 & X < 0 \end{cases}$, $\Sigma_j C_{ij} = N_i$,

$$\phi(a,b) = \begin{cases} \mathrm{sgn}(ab) \min(\left|\frac{a}{b}\right|, \left|\frac{b}{a}\right|) & ab \neq 0 \\ |b| - 1 & ab = 0 \end{cases} \quad (5)$$

$\phi(a,b)$ is used to measure the proximity. N_i is the number of pre-synaptic connections. C_{ij} are bias factors.

In the following cycles, the newly activated cell will cooperate with other activated cells to fire the next cells. A wave of activity emerges and propagates stably, reproducing a learned motion sequence.

Like those in cellular automata, the state of a motion cell in TSPN is updated only according to its pre-synaptic cells, or its neighbor cells. In other words, cell updating is a parallel procedure. If the network is implemented as dedicated hardware, the expansion of the network scale will not dramatically deteriorate its performance.

In our preliminary experiment we used the AmigoBot robot manufactured by Activmedia Inc. It was equipped with a color camera, 8 sonar sensors and two motors with odometers. These sensors were also modeled in the simulator. Their readings were corrupted with 2% random noise. The vision system provided color category of an object and detected impassibleness via extracting edges in the visual field.

Owing to the decay mechanism, the excitation level of a sensor cell not only denotes the qualitative feature of current external state, but also the duration of that feature. In this way, the robot can incorporate present and past sensor inputs and comprehend the nature of its environment. The lower the activity, the fewer the fluctuations of data. Therefore a snapshot of activations contains past sensor information, which is a compact context for disambiguating similar instant sensory inputs so as to producing different movement decisions. Hence, it provides the system with much more intelligence than a reactive controller.

Initially the robot freely explored the environment using its innate controller (a reactive obstacle avoidance module). We placed some indications such as wooden blocks (in the physical world), dark lines (in the simulated world) in the places where the robot might be trapped. The robot could perceive them and would not go further. It memorized in TSPN what it had done and perceived, then utilized the knowledge to perform navigation tasks in an environment without cues. This ability is beyond the competence of the innate controller. The potential destinations had salient features such that the robot could notice them during its explorations. Some scenes during the experiment and corresponding states of the learnt network are shown in Fig. 2.

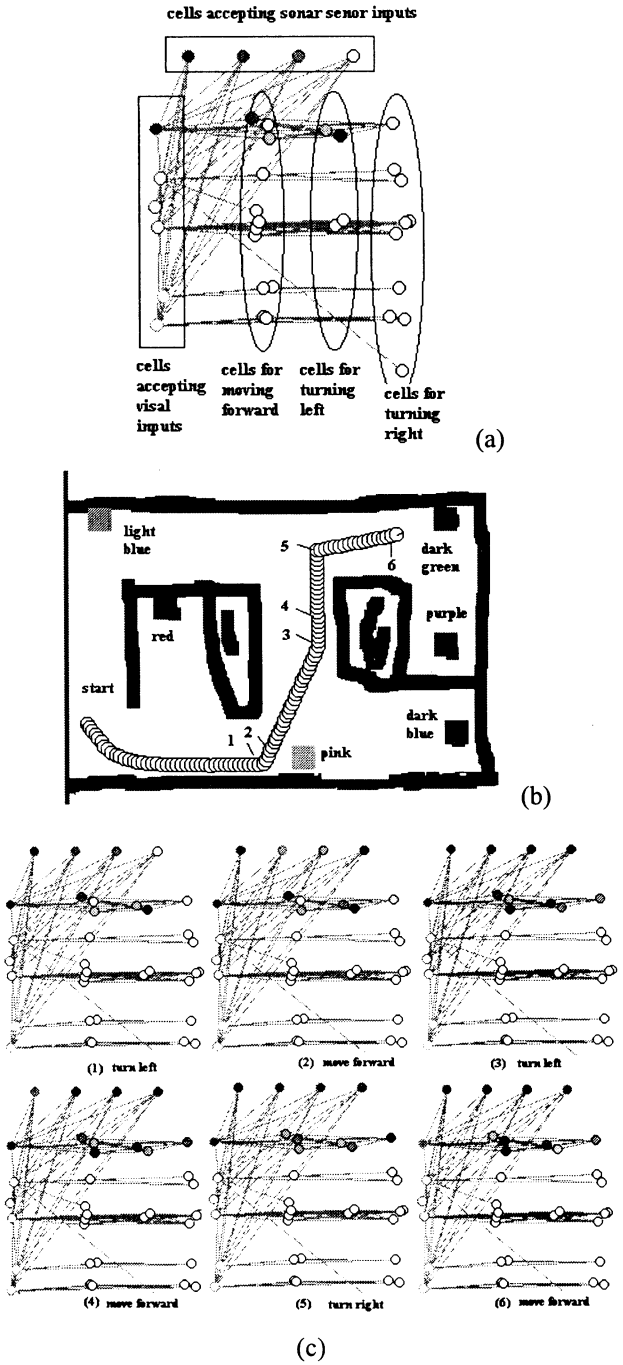


Fig.2 Experimental results: (a) the learnt TSPN network; (b) the area and the robot's trajectory guided by TSPN output; (c) the variation of cells' potential in different spots. The darker the cell, the higher the potential. The ensemble of cells represents robot's spatial knowledge of the world.

4 Machine intuition

Unlike most existing navigation systems, our system

is irrational. Its behaviors are not grounded on reason, however many can afterwards be broken down into their component elements and their origin thus brought into harmony with the laws of reason [5]. Action selection is an immediate decision based on sensation and memory, coming from its content without the use of reasoning process. It appears that the robot just knows what to do next. The navigation skill has become so much a part of the system that the robot needs to be no more aware of this skill than it is of its sensors and motors. Similar phenomenon happens when our intuition plays a more important role than reason in the decision making process. We usually do not produce deliberative rational thoughts when walking the correct route according to our intention and perception. Our idea of doing something presents itself whole and complete. We are not able to explain or discover how our behavior came into existence [4].

Since TSPN exhibits the distinct function of mediating perceptions in an unconscious way and producing results without conscious analytic decomposition and recombination, we argue that it may be a good starting point for a more ambitious goal—to demonstrate machine intuition. After decades of efforts, artificial intelligence is still far from its ultimate goal. Many researchers' bottom line is: "Computers as reasoning machines can't match human intuition and expertise that we believe is the core of human intelligence and skill. [3]" Does the unduly emphasis on deductive logic cause the stagnancy?

From the perspective of navigation, the TSPN-based method cannot surpass classical systems that use coordinates and maps in their precise localization, path planning and tracking performance. Its whole capacity is based on the procedural/episodic memory. But it seems to be a novel approach to an explanation of biological way-finding mechanisms, as well as a step toward artificial intuition. Furthermore, navigation is not its only faculty. With expansion of sensory inputs and motor capacities, the same mechanism can be applied to other tasks. Besides mediating sensation and behavior, it may also construct its internal representation of letters and words. In the language-learning context, it may conjoin the shape (visual input), pronunciation (auditory input) and meaning of a word with segments of episodic memory relative to the word and its synonyms. The semantic memory may gradually emerge from the abstraction of the procedural/episodic memory.

The discussed connectionist method is not proposed as a substitute of rational navigation strategy, but as a complementary way. Neither of the two approaches should be excluded from artificial agent building. Analysis and intuition work together in the human mind. Analytic thinking is necessary for beginners learning a new skill. It is also useful at the highest levels of expertise, where it can sharpen and clarify intuitive insights. However, intuition is coming out from a holistic perspective of the world and grasps the gestalt of a situation. Via bringing together these two partial, inaccurate ways of knowledge, we will get an integrated whole that is more profound and penetrating. [3]

5 Conclusion

We have described a connectionist model that learns a spatial representation of the world based on temporal memory of perceptions and actions of a mobile robot. It grows at run-time to merge past experiences to be retrieved during later runs to guide the robot to perform navigation tasks. The architecture seems to be a promising step towards machine intuition. It may be treated as a primitive form of procedural/episodic memory. This is like a story-based approach for making an artificial agent represent and understand the environment, its own relations with other agents and maintaining a model of perceived reality sprouting from pattern recognizer.

As in the case of consciousness, the question how intuition works still remains unanswered. We argue that research on irrational machine intelligence may help us find a short cut to synthesizing the humanly cognitive power.

References

[1] A. Arleo and W. Gerstner "Neuro-Mimetic Navigation Systems: A Computational Model of the Rat Hippocampus", available at *citerseer.nj.nec.com,* 1999

[2] William H. Calvin, How brains think? (Chinese version), Shanghai: Science and Technology Press, 1996.

[3] Hubert L. and Stuart E. Dreyfus, Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer , New York: The Free Press, 1986.

[4] C. G. Jung, Psychological Types. Princeton, N.J.: Princeton University Press, pp.453-454, 1971.

[5] Liu Juan, Cai Zixing, Tu Chunming. "Environmental Feature Extraction and Mergence: Make the Past Serve the Present", In *International Conference on Neural Information Processing*, Singapore November 18-22, 2002.

[6] H. A. Mallot, B. Bülthoff, B. Schölkopf and K. Yasuhara "View-based Cognitive Map Learning by an Autonomous Robot". In F.Fogelman-Soulié and P. Gallinari, editors, *Proc. of ICANN'95, vol II,* EC2, Nanterre, France, pp.381-386, 1995

[7] Sebatian Thrun, "Robotic Mapping: A Survey", in: Gerhard Lakemeyer, Bernhard Nebel (Eds.), Exploring Artificial Intelligence in the New Millennium. Morgan Kaufmann Publishers, pp. 1-35, 2002.

[8] E.C. Tolman, "Cognitive maps in rats and men", *The Psychological Review,* Vol. 55, pp: 189-208, 1948.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Digital Control of an Underwater Robot with Vertical Planar 2-Link Manipulator

S. Sagara
Department of Control Engineering, Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan
E-mail: sagara@cntl.kyutech.ac.jp

Abstract

Through experiments of an underwater robot with 2 dimensional and vertical planar 2-link manipulator, we have shown that a Resolved Acceleration Control(RAC) with the position and velocity feedback has a good control performance in spite of using an inaccurate hydrodynamic model. In this paper, for the underwater robot a digital RAC system is designed and the effectiveness of the control system is verified by computer simulation.

1 Introduction

Since underwater robots are necessary for ocean exploration and preservation, many studies have been done about Underwater Robotic Vehicle (URV) including Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) [1]. And recent years, dynamics and control of Underwater Vehicle-Manipulator Systems (UVMSs) have been studied because the manipulators are used for collecting resources, repairing cables, constructing structures, etc [2, 3, 4, 5]. However, the experimental studies of UVMS are only a few. Through simulations and experiments of a free floating underwater robot with 2 dimensional and horizontal planar 2-link manipulator, we have shown that a Resolved Acceleration Control (RAC) and a Resolved Motion Rate Control (RMRC) method with the manipulator's end-tip position and velocity feedback have good control performances [6, 7]. Furthermore, for an underwater robot vehicle with vertical planar 2-link manipulator shown in Fig. 1, we have derived the dynamic model and done experiment using RAC method [8].

This paper is concerned with a digital control of the underwater robot with vertical planar 2-link manipulator. In general, digital computers are utilized for URV's controllers and a few digital control methods for UVMS have been proposed [9]. In this paper,

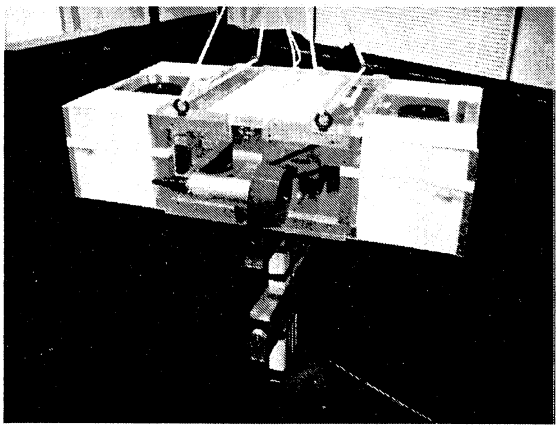


Fig. 1: Floating 2-link underwater robot

for the underwater robot vehicle with vertical planar 2-link manipulator described above, the digital control system is constructed. First, the kinematic equation of the robot with manipulator is summarized. Next, based on the continuous-time RAC method the digital version is designed. Furthermore, to verify the effectiveness of the control method, computer simulation is performed. The simulation result shows that the digital control method has a good control performance.

2 Modeling

The underwater robot model used in this paper is shown in Fig. 2. It has a robot base and 2-DOF manipulator which can move in a vertical plane. Three thrusters are mounted on the base to provide propulsion for position and attitude control of the base.

Symbols used in this paper are defined as follows:
[Symbols]
Σ_U : inertial coordinate frame
Σ_i : ith link coordinate frame ($i = 0, 1, 2$; link 0 means base)

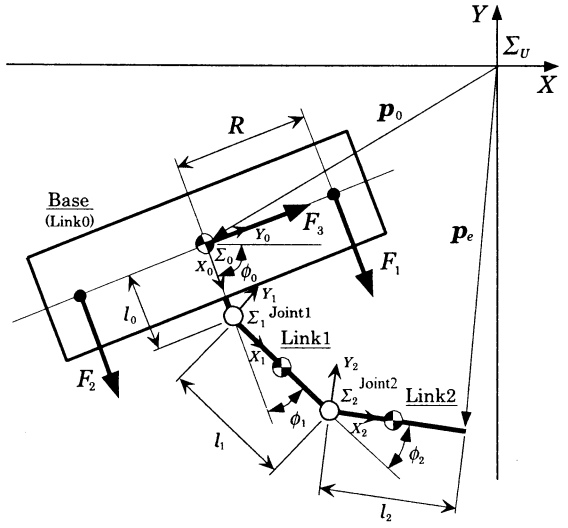


Fig. 2: 2-link underwater robot model

- ${}^U\boldsymbol{R}_i$: coordinate transformation matrix from Σ_i to Σ_U
- l_i : length of link i
- $\boldsymbol{v}_i$: velocity vector of link i with respect to Σ_i
- ϕ_i : relative joint angle
- θ_i : absolute joint angle $(= \sum_{j=0}^{i} \phi_j)$
- $\boldsymbol{p}_0$: position vector of origin of Σ_0 with respect to Σ_U
- $\boldsymbol{p}_e$: position vector of end-tip with respect to Σ_U
- $\boldsymbol{x}_0$: position and attitude vector of base with respect to Σ_U $(= [\boldsymbol{p}_0^T,\ \theta_0]^T)$
- $\boldsymbol{\phi}$: joint angle vector $(= [\phi_1,\ \phi_2]^T)$
- m_i : mass of link i (link 0 means the robot base)
- $\boldsymbol{M}_{a_i}$: added mass tensor of link i
- $\boldsymbol{I}_i$: inertia tensor of link i
- $\boldsymbol{I}_{a_i}$: added inertia tensor of link i
- $\hat{\boldsymbol{x}}_i$: position vector from joint i to joint $(i+1)$ with respect to Σ_i
- $\hat{\boldsymbol{a}}_i$: position vector from joint i to center of gravity of link i with respect to Σ_i
- $\hat{\boldsymbol{b}}_i$: position vector from joint i to center of buoyancy of link i with respect to Σ_i
- $\boldsymbol{E}$: unit matrix
- $\boldsymbol{g}$: gravitational acceleration vector
- F_i : thruster force $(i = 1,\ 2,\ 3)$
- R : length form origin of Σ_0 to thruster

2.1 Kinematics

First, a time derivative of the end-tip position vector $\boldsymbol{p}_e$ is

$$\dot{\boldsymbol{p}}_e = \boldsymbol{A}\dot{\boldsymbol{x}}_0 + \boldsymbol{B}\dot{\boldsymbol{\phi}} \tag{1}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$
$$a_{13} = -l_0 S_0 + b_{11}, \quad a_{23} = l_0 C_0 + b_{21},$$
$$b_{11} = -l_1 S_1 + b_{12}, \quad b_{12} = -l_2 S_2,$$
$$b_{21} = l_1 C_1 + b_{22}, \quad b_{22} = l_2 C_2,$$
$$S_i = \sin\theta_i, \quad C_i = \cos\theta_i \quad (i = 0, 1, 2).$$

Next, let $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$ be a linear and an angular momentum of the robot including hydrodynamic added mass $\boldsymbol{M}_{a_i}$ and added inertia $\boldsymbol{I}_{a_i}$, then

$$\boldsymbol{\eta} = [\eta_1,\ \eta_2,\ 0]^T = \sum_{i=0}^{2} {}^U\boldsymbol{R}_i(m_i\boldsymbol{E} + \boldsymbol{M}_{a_i})\tilde{\boldsymbol{a}}_i, \tag{2}$$

$$\boldsymbol{\mu} = [0,\ 0,\ \mu_3]^T = \sum_{i=0}^{2}(\boldsymbol{I}_i + \boldsymbol{I}_{a_i})\boldsymbol{\omega}_i + \hat{\boldsymbol{x}}_i \times \{{}^U\boldsymbol{R}_i(m_i\boldsymbol{E} + \boldsymbol{M}_{a_i})\tilde{\boldsymbol{a}}_i\} \tag{3}$$

where

$$\tilde{\boldsymbol{a}}_i = \boldsymbol{v}_i + \boldsymbol{\omega}_i \times \hat{\boldsymbol{a}}_i, \quad \boldsymbol{\omega}_i = [0,\ 0,\ \dot{\phi}_i]^T \ (i = 0, 1, 2).$$

From Eqs. (1), (2) and (3) the following equation can be obtained:

$$\boldsymbol{s} = [\eta_1,\ \eta_2,\ \mu_3]^T = \boldsymbol{C}\dot{\boldsymbol{x}}_0 + \boldsymbol{D}\dot{\boldsymbol{\phi}} \tag{4}$$

where $\boldsymbol{C} \in R^{3\times3}$ and $\boldsymbol{D} \in R^{3\times2}$ are matrices including the added mass $\boldsymbol{M}_{a_i}$ and the added inertia $\boldsymbol{I}_{a_i}$.

2.2 Hydrodynamic force and moment

Generally, the drag force and moment of the joint i can be represented as follows [10, 11]:

$$\boldsymbol{f}_{d_i} = \frac{\rho}{2} C_{D_i} D_i \int_0^{l_i} ||\boldsymbol{w}_i|| \boldsymbol{w}_i d\hat{\boldsymbol{x}}_i, \tag{5}$$

$$t_{d_i} = \frac{\rho}{2} C_{D_i} D_i \int_0^{l_i} \hat{\boldsymbol{x}}_i ||\boldsymbol{w}_i|| \boldsymbol{w}_i d\hat{\boldsymbol{x}}_i \tag{6}$$

where $\boldsymbol{w}_i = \boldsymbol{v}_i + \boldsymbol{\omega}_i \times \hat{\boldsymbol{x}}_i$, and ρ is the fluid density, C_{D_i} is the drag coefficient, D_i is the width of link i.

The gravitational and buoyant forces acting link i are described as follows:

$$\boldsymbol{f}_{g_i} = ({}^U\boldsymbol{R}_i)^T(\rho V_i - m_i)\boldsymbol{g}, \tag{7}$$

$$\boldsymbol{t}_{g_i} = ({}^U\boldsymbol{R}_i)^T(\hat{\boldsymbol{b}}_i \times \rho V_i \boldsymbol{g} - \hat{\boldsymbol{a}}_i \times m_i \boldsymbol{g}). \tag{8}$$

3 Digital RAC

Differentiating Eqs. (1) and (4) with respect to time, the following equation can be obtained:

$$\boldsymbol{W}(t)\boldsymbol{\alpha}(t) = \boldsymbol{\beta}(t) + \boldsymbol{f}(t) - \dot{\boldsymbol{W}}(t)\boldsymbol{v}(t) \qquad (9)$$

where

$$\boldsymbol{W}(t) = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{D} \\ \boldsymbol{A} & \boldsymbol{B} \end{bmatrix}, \quad \boldsymbol{H} = \boldsymbol{C} + \boldsymbol{E}, \quad \boldsymbol{f}(t) = \begin{bmatrix} \dot{\boldsymbol{s}} \\ \boldsymbol{0} \end{bmatrix},$$

$$\boldsymbol{\alpha}(t) = \begin{bmatrix} \ddot{\boldsymbol{x}}_0 \\ \ddot{\boldsymbol{\phi}} \end{bmatrix}, \quad \boldsymbol{\beta}(t) = \begin{bmatrix} \ddot{\boldsymbol{x}}_0 \\ \ddot{\boldsymbol{p}}_e \end{bmatrix}, \quad \boldsymbol{v}(t) = \begin{bmatrix} \dot{\boldsymbol{x}}_0 \\ \dot{\boldsymbol{\phi}} \end{bmatrix},$$

and $\dot{\boldsymbol{s}}$ is the external force including hydrodynamic force and thrust of the thruster which act on the robot.

Discretizing Eq. (9) by a sampling period T, and applying $\boldsymbol{\beta}(k)$ and $\dot{\boldsymbol{W}}(k)$ to the backward Euler approximation, we have

$$\boldsymbol{W}(k)\boldsymbol{\alpha}(k-1) = \frac{1}{T}[\boldsymbol{\nu}(k) - \boldsymbol{\nu}(k-1) + T\boldsymbol{f}(k) - \boldsymbol{W}(k) - \boldsymbol{W}(k-1)\boldsymbol{v}(k)] \qquad (10)$$

where $\boldsymbol{\nu}(k) = [\dot{\boldsymbol{x}}_0^T(k), \ \dot{\boldsymbol{p}}_e^T(k)]$ and the discrete time kT is abbreviated to k. Note that computational time delay is introduced to Eq. (10).

For Eq. (10), the desired acceleration is defined as follows:

$$\boldsymbol{\alpha}_d(k) = \frac{1}{T}\boldsymbol{W}^{-1}(k)[\boldsymbol{\nu}_d(k+1) - \boldsymbol{\nu}_d(k) + \boldsymbol{\Lambda}\boldsymbol{e}_\nu(k) + \boldsymbol{\Gamma}\boldsymbol{e}_p(k) + T\boldsymbol{f}(k)] \qquad (11)$$

where $\boldsymbol{e}_\nu(k) = \boldsymbol{\nu}_d(k) - \boldsymbol{\nu}(k)$ and $\boldsymbol{e}_p(k) = \boldsymbol{p}_d(k) - \boldsymbol{p}(k)$, $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_i\}$ $(i = 1, \cdots, 5)$, $\boldsymbol{\Gamma} = \mathrm{diag}\{\gamma_i\}$, $\boldsymbol{p}(k) = [\boldsymbol{x}_0^T, \ \boldsymbol{p}_e^T]^T$, $\boldsymbol{\nu}_d(k)$ and $\boldsymbol{p}_d(k)$ are the desired values of $\boldsymbol{\nu}(k)$ and $\boldsymbol{p}(k)$.

Assuming $\boldsymbol{W}(k) \approx \boldsymbol{W}(k-1)$ and $\boldsymbol{f}(k) \approx \boldsymbol{f}(k-1)$ for one sampling period, from Eqs. (10) and (11) we have

$$\boldsymbol{W}(k)\boldsymbol{e}_\alpha(k) = \frac{1}{T}\{[(q-1)\boldsymbol{E} + \boldsymbol{\Lambda}]\boldsymbol{e}_\mu(k) + \boldsymbol{\Gamma}\boldsymbol{e}_p(k)\} \qquad (12)$$

where $\boldsymbol{e}_\alpha(k) = \boldsymbol{\alpha}_d(k) - \boldsymbol{\alpha}(k)$ and q is the forward sift operator. Furthermore, applying $\boldsymbol{e}_\mu(k)$ to the backward Euler approximation, the following equation can be obtained:

$$\boldsymbol{W}(k)\boldsymbol{e}_\alpha(k) = \frac{1}{T^2}\{[(q-1)\boldsymbol{E} + \boldsymbol{\Lambda}](1-q^{-1}) + \boldsymbol{\Gamma}T\}\boldsymbol{e}_p(k) \qquad (13)$$

Now, γ_i $(i = 1, \cdots, 5)$ is selected as

$$\gamma_i = \frac{2 - \lambda_i - 2\sqrt{1-\lambda_i}}{T}.$$

Then Eq. (13) becomes

$$\boldsymbol{W}(k)\boldsymbol{e}_\alpha(k) = \frac{1}{T^2}q^{-1}(q\boldsymbol{E} - \tilde{\boldsymbol{\Lambda}})^2\boldsymbol{e}_p(k) \qquad (14)$$

where $\tilde{\boldsymbol{\Lambda}} = \mathrm{diag}\{\sqrt{1-\lambda_i}\}$. Therefore, if the feedback gain λ_i is selected to satisfy $0 < \lambda_i < 1$, $\boldsymbol{e}_\alpha(k) \to \boldsymbol{0}$ $(k \to \infty)$ and all elements of $\boldsymbol{W}(k)$ are bounded, the convergence of $\boldsymbol{e}_p(k)$ to zero as k tends to infinity can be ensured.

4 Simulation

In this section, to verify the effectiveness of the digital RAC method, the computer simulation are done.

Physical parameters of the underwater robot are shown in Table 1. And the dynamics of the robot and the thruster characteristic are shown in [8].

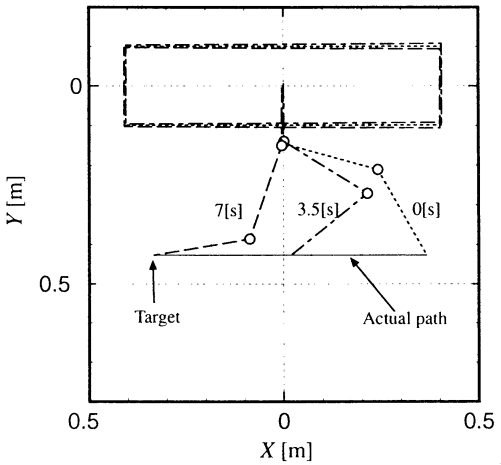
The simulation was carried out under the following condition. The desired end-tip position was set up along a straight path from the initial position to the target calculated from trapezoidal velocity pattern. On the other hand, the desired position and attitude of base were set up the initial values. The sampling period was $T = 0.02[s]$ and the feedback gains were $\lambda_1 = 0.6$, $\lambda_2 = 0.6$, $\lambda_3 = 0.4$, $\lambda_4 = 0.5$ and $\lambda_5 = 0.5$. The initial relative angles of the robot were $\phi_0 = -\pi/2[\mathrm{rad}]$, $\phi_1 = 5\pi/12[\mathrm{rad}]$ and $\phi_2 = -\pi/4[\mathrm{rad}]$.

Fig. 3 shows the simulation result. In this figure, (a) shows the motion of the robot, (b) is the desired velocity pattern of the end-tip, and (c) and (d) are the errors of the end-tip and base.

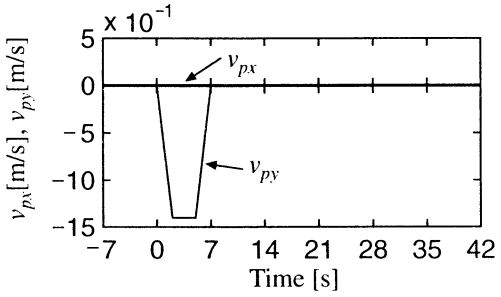
From Fig. 3 it can be seen that the end-tip and base follow the reference trajectories in spite of the influence of the hydrodynamic forces and the tracking errors are very small. Simulation result shows that the control performance can be improved by using the proposed method.

Table 1: Physical parameters of the underwater robot

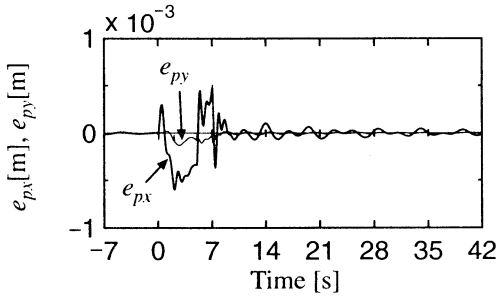
	Base	Link 1	Link 2
Mass [kg]	26.04	4.25	1.23
Moment of inertia [kg m^2]	1.33	0.19	0.012
Link length (x axis) [m]	0.2	0.25	0.25
Link length (y axis) [m]	0.81	0.04	0.04
Link width [m]	0.42	0.12	0.12
Added mass(x) [kg]	72.7	1.31	0.1
Added mass(y) [kg]	6.28	3.57	2.83
Added moment of inertia [kg m^2]	1.05	0.11	0.06



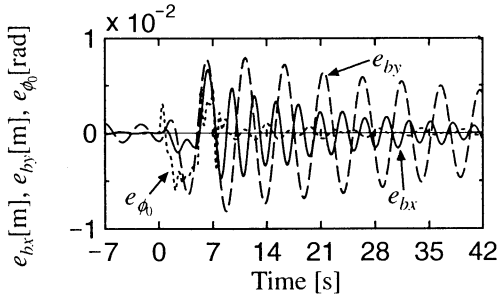
(a) motion



(b) desired velocity pattern of end-tip



(c) position error of end-tip



(d) position and attitude error of base

Fig. 3: Simulation result

5 Conclusion

In this paper, for a free-floating underwater robot with vertical planar 2-link manipulator a digital RAC method was proposed. Simulation result showed the effectiveness of the proposed method.

References

[1] J. Yuh ed., *Underwater Robotic Vehicles:Design and Control*, TSI Press, 1995.

[2] T. W. McLain *et al.*, "Experiments in the Coordinated Control of an Underwater Arm/Vehicle System", *Autonomous Robots 3*, Kluwer Academic Publishers, pp. 213 – 232, 1996.

[3] G. Antonelli and S. Chiaverini, "Task-Priority Redundancy Resolution for Underwater Vehicle-Manipulator Systems", *Proc. of 1998 IEEE ICRA*, pp. 768 – 773, 1998.

[4] G. Antonelli *et al.*, "Tracking Control for Underwater Vehicle-Manipulator Systems with Velocity Estimation", *IEEE J. Oceanic Eng.*, Vol. 25, No. 3, pp. 399 – 413, 2000.

[5] N. Sarkar and T. K. Podder, "Coordinated Montion Planning and Control of Autonomous Underwater Vehicle-Manipulator Systems Subject to Drag Optimization", *IEEE J. Oceanic Eng.*, Vol. 26, No. 2, pp. 228 – 239, 2001.

[6] S. Sagara *et al.*, "Experiments of a Floating Underwater Robot with 2 Link Manipulator", *Proc. of AROB 5th*, pp. 367 – 370, 2000.

[7] S. Sagara *et al.*, "Resolved Motion Rate Control of a Free-Floating Underwater Robot with Horizontal Planar 2-Link Manipulator", *Proc. of AROB 6th*, pp. 113-116, 2001.

[8] S. Yamada and S. Sagara, "Resolved Motion Rate Control of an Underwater Robot with Vertical Planar 2-Link Manipulator", *Proc. of AROB 7th*, pp. 230-233, 2002.

[9] H. Maheshi *et al.*, "A Coordinated Control of an Underwater Vehicle and Robotic Manipulator", *J. Robotic Systems*, Vol. 8, No. 3, pp. 339-370, 1991.

[10] B. Lévesque and M. J. Richard, "Dynamic Analysis of a Manipulator in a Fluid Environment", *Int. J. Robot. Res.*, Vol. 13, No. 3, pp. 221-231, 1994.

[11] S. McMillan *et al.*, "Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator", *IEEE Trans. Syst., Man, Cybern.*, Vol. 25, No. 8, pp. 1194-1206, 1995.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Efficiency of information spread on self-organized networks

Jun Matsukubo Yukio Hayashi
Knowledge Science
Japan Advenced Institute of Science and Technology, Hokuriku
Nomi, Ishikawa, Japan, 923-1292

Abstract

The almost all actual self-organized networks have the universal feature that the distribution in their vertex degrees follows a power-law. The networks with this feature are called scale-free networks. On one hand, like advertisement activity and collection of the Web pages by Web crawlers, it is desired for information spread on such networks to be able to cover the most possible vertices. However, the effect for costs hasn't become clear. Then, in this report, we experimented information spread on the scale-free-network generated artificially. On the simulation, the information-spread is carried out as breadth-first-search. The results show the following tendency regardless the vertex starting information-spread: hubs that have many out-going edges are discovered easier than others.

keywords: scale-free network, power-law, hub, information-spread, breadth-first search

1 Introduction

Most of actual networks have the properties of the self-organized networks as the costar relations between actors, the cite relation between papers, the neural networks of nematodes, airline maps, power grids, acquaintance relations, the link structures of routers and Web pages [1]~[5]. The feature of the self-organized networks is the autonomic developing.

The link structures of such networks are different from regular ones and random ones. There is a scale-free property in a self-organized network. Such a network with the property has the feature on the link structures that the distribution of the vertices for the edges obeys a power-law.

On the one hand, some self-organized networks is structured for information-spread. For example, ad-persons will be interested in cost performance of an advertisement on a network. On the Internet, suffering expectations are needed for the countermeasure to the infection of computer viruses. In addition, the search engines on the Web are correcting millions of the pages with Web crawlers.

Because scale-free networks have heterogeneous structures, information-spread is able to be also heterogeneous. Therefore the investigation of the efficiency of information-spread can be available for the solution of the some problems. However it has not been clear.

In this study we investigate the efficiency of information-spread on scale-free networks. The simulations show the following results: while enormous efforts are necessarily for information-spread over a whole network, hubs that are the vertices with many out-going links can be discovered easier than many of others.

2 Model for generateing scale-free networks

A model generating scale-free networks needs the mechanisms that vertices increase on a network and that those with many edges get furthermore ones. Kumar, et al [6] proposed the (α, β) model. The model can generate scale-free networks with arbitrary power coefficients of in-degrees and out-degrees decided by the parameters α and β, respectively.

3 Simulation

Here, we simulated information-spread on artificial scale-free networks. In our simulations, we used the value of the parameters α=0.52 and β=0.58 respectively for generating networks. The values correspond to the power-law coefficients for the link structure of Web pages.

Here we generated five networks with 10,000, 50,000, and 100,000 vertices respectively. We simulated 10 times information-spread on every network

as breadth-first-search. The graphs shown below are the averages of the simulation results on networks with 10,000, 50,000, and 100,000 vertices respectively.

Fig. 1 and Fig. 2 respectively show exapmles of in-degrees and out-degrees of the vertices on a scale-free network by bi-logarithmic plot. The number of the vertices on the example network is 100,000. Both in-degrees and out-degrees follow the power functions decided by the parameters of (α, β) model. Moreover, these figures show that degrees of vertices are extremely biased.

Fig. 3 is an example of the out-degrees of the visited vertices on the network with 100,000 vertices. It is shown that the vertices with many out-degrees are discovered in the left part of this figure more than the right part.

Fig. 4 shows the percentage of accumulated numbers of hubs to those of visited vertices, where the total of hubs are 0.5% of the total vertices on each network in decreasing-order of out-degrees. The horizontal axis is the percentage of visited vertices to the total. The vertical one is that of accumulated hubs to the total. When the percentage of visited vertices is about 20%, that of accumulated numbers of hubs is over 80%. Therefore, regardless of the vertices starting information-spread, hubs tend to be discovered easier than others.

Next, Fig. 5 shows the accumulated numbers of vertices newly obtained to those of visited vertices. The horizontal and vertical axes respectively show the percentage of visited vertices and the accumulated one of vertices newly obtained to the total. Regardless the total of vertices on a network, Fig. 5 shows that the percentage of the accumulated number of vertices newly obtained was 80% when that of visited vertices was by 20%. Afterward vertices newly obtained increased slowly. These indicate the following suggestion: although hubs connect many vertices densely, it is difficult to spread information to the vertices separated from hubs. Thus hubs are considered to be the portals on a scale-free network.

Then, Fig. 6 shows the accumulated numbers of duplicate vertices to those of visited vertices in bi-logarithmic plot. A horizontal axis and a vertical axis are the percentage of visited vertices and the accumulated one of the duplicate vertices to the total of vertices on a network, respectively. From Fig. 6, it turns out that the accumulated numbers of duplicate vertices to the number of the visited ones increases asymptotically to a power function. In the result, the coefficient of the power function was 0.73. Moreover, Fig. 6 shows that duplicate vertices increase faster than those newly obtained, since hubs connect many vertices densely.

In addition, Fig. 7 and Fig. 8 show the accumulated number of the duplicate vertices to that of those newly obtained. Fig. 7 shows that the accumulated number of the duplicate vertices increases as a power function by about 40% of the vertices newly obtained. Fig. 8 shows that it increases as an exponential function after above 40%. When the number of visited vertices turned into 3% of the total of vertexes on the whole network, the accumulated number of duplicate vertices was equal to that of those newly obtained. At this point, each accumulated number was about 40% of the total vertices on a network.

The above-mentioned properties are independently of the vertices from where information-spread starts, the structure of a network, and the total of vertices on a network. This may be the universal feature of scale-free networks.

4 Conclusion

In this report, we investigated the efficiency of information-spread in the simulations on scale-free networks generated artificially. The simulation results show that many efforts are needed for information-spread to many vertices on such a network since almost all vertices have few links from and to others. Therefore, the cost performance falls greatly. It corresponds to that there are many pages that are seldom to be visited on the Web. However, on a scale-free network, regardless of the vertices from where information spread starts, it is easy to discover hubs that are the vertices with many out-going links to others. Supposing this is realized also on an actual network, for example, in order to control that infection of a computer virus is expanded, the computer used as hubs needs to be protected from a virus. Conversely, since hubs can be found easily, once a virus occurs on a network, a virus will spread quickly. These properties are caused by that the link structure of a scale-free network follows a power-law distribution.

References

[1] Barabasi A, et al, "Emergence of Scaling in Random Networks," *http://www.nd.edu/ networks/slide/table.html.*

[2] Whitfield J, "The Small World Web," *http://www.nature.com/nsu/000824/000824-8.html.*

[3] Kleinberg J, "Navigation in a small world," *Nature*, 406, p.845, 2000.

[4] Boyle A, "Is the separation degree of Web is 19 clicks?(in Japanese)," *http://www.zdnet.co.jp/news/9909/10/www.html.*

[5] Tsujino T, "The complicated network exceeding anticipation(in Japanese)," *http://www5.ocn.ne.jp/ report /news/networktopology.htm.*

[6] Kumar R, et al, "Extracting large-scale knowledge base from the web," *IEEE Int. Conf. on VLDB*, Edinbargh, Scotland, 1999.

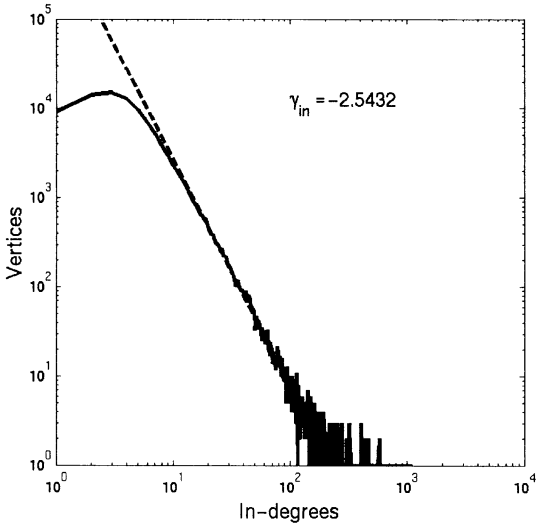


Figure 1: An example of the in-degrees on a scale-free network

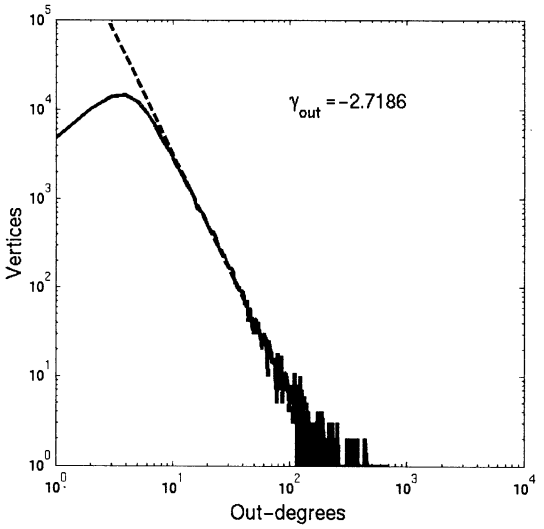


Figure 2: An example of the out-degrees on a scale-free network

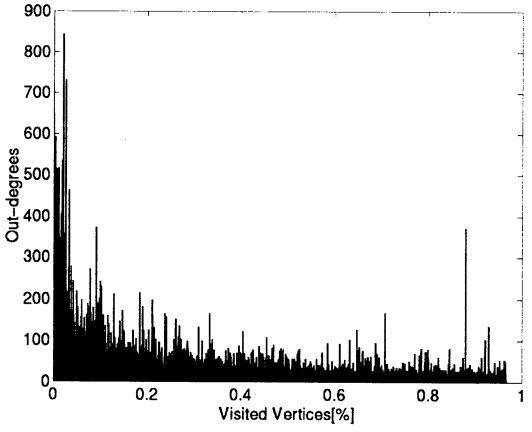


Figure 3: The Out-degrees of the visited vetices

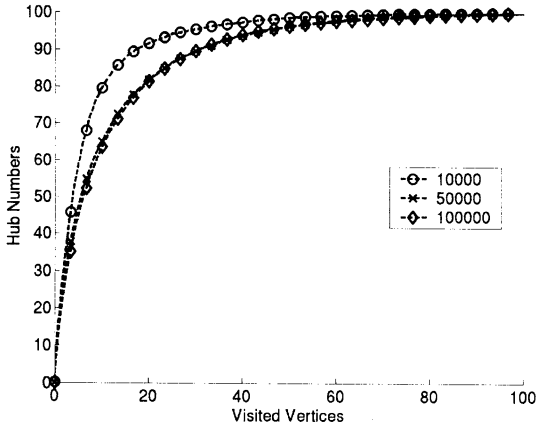


Figure 4: The accumulated number of Hubs to the visited vetrices

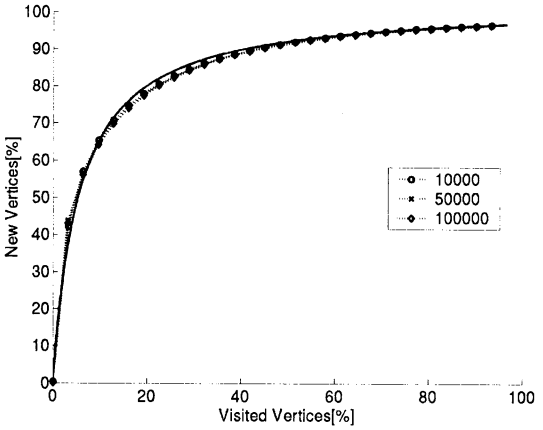


Figure 5: The accumulated number of the vertices newly obtained to that of visited vertices

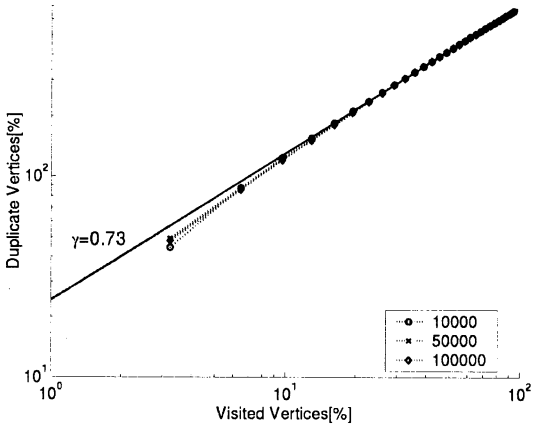


Figure 6: The accumulated number of the dupulicated vertices to that of visited vertices(bi-log)

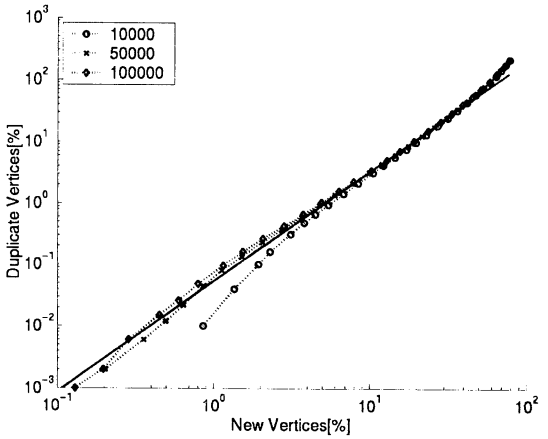


Figure 7: The accumulated number of the dupulicated vertices to that of those newly obtained(vertical axis: semi-log)

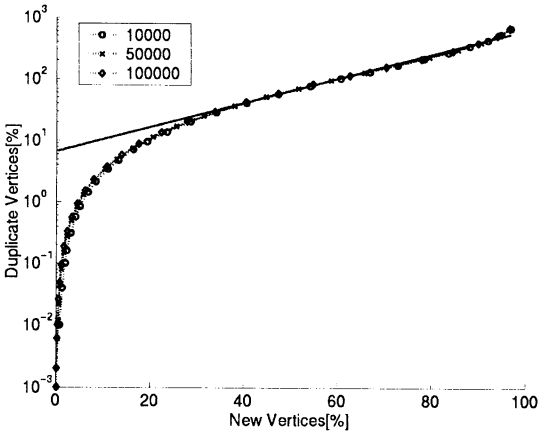


Figure 8: The accumulated number of the dupulicated vertices to that of those newly obtained(bi-log)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Robust Motion and Force Tracking Control of Robot Manipulators in Contact with Surface with Unknown Stiffness and Viscosity

Daiki Moriyama, Masahiro Oya and Makoto Wada,

Department of Control Engineering,
Kyushu Institute of Technology
Tobata, Kitakyushu, 804-8550, Japan
E-mail: a344222d@tobata.isc.kyutech.ac.jp

Toshinori Suehiro

Mechanics and Electronics Research Institute Fukuoka
Industrial Technology Center
3-6-1 Norimatsu, Yahatanishi, Kitakyushu,
807-0831, Japan

Abstract

In this paper, a new robust motion/force tracking controller is developed for two-link manipulators in contact with surface with unknown mass, stiffness, viscosity and friction. To develop a robust controller, the task space is partitioned into two subspaces. Then, using the analysis of the derivative of a positive definite function, robust force tracking controller and robust motion tracking controller are developed independently.

1. Introduction

The motion/force control will be required in many environments such as medical institutions, industrial world, space, home, etc. Therefore, many researchers focus on research of motion/force control schemes for robot manipulators in contact with constraint surface [1],[2]. In the early works, the accurate information of robot dynamics and constraint surfaces are required. In practical application, however, the accurate information cannot be obtained. To overcome the problem, in the case when the robot dynamics and the shape of surface are unknown, adaptive hybrid motion/force control schemes are proposed [3],[4]. The researches are studied based on assumption that the environment is rigid. In recent year, several researchers are studying based on the assumption that the environment has unknown flexibility [5],[6]. However, as for as authors know, there is no scheme that can ensure arbitrary tracking performance with respect to motion and force.

To aim for establishing a robust control scheme for the case when manipulators in contact with surface with unknown mass, stiffness, viscosity and friction, in this paper a new robust motion/force tracking controller is developed for two-link manipulators.

2. Dynamic Model of Constrained Robot Manipulator

Fig. 1 shows two-link manipulator in contact with constraint surface. The dynamic of constraint surface is

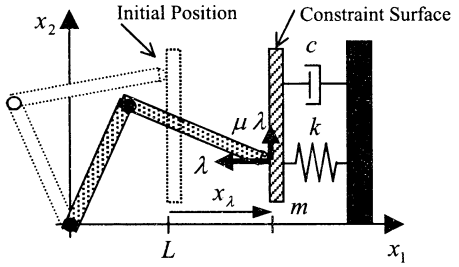


Fig. 1 Manipulator in contact with constraint surface

given by

$$\lambda = m\ddot{x}_\lambda + c\dot{x}_\lambda + kx_\lambda, \quad (1)$$

where λ is contact force, x_λ is displacement of constraint surface, m, k and c are the environment parameters mass , stiffness constant and viscosity constant. The environment parameters m, k and c are unknown but lower bound of m and upper bound of k, c are known. Constraint surface can be described by

$$c(\boldsymbol{x}) = x_1 - L = x_\lambda \ (\in R), \quad (2)$$

where $\boldsymbol{x} \in \boldsymbol{R}^2$ is the vector of the position of the end-effector. The dynamic equation of two-link manipulator is given by

$$\boldsymbol{H}_x(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{h}_x(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{g}_x(\boldsymbol{q}) + (\boldsymbol{C}+\boldsymbol{\eta})^T\lambda = \boldsymbol{u}, \quad (3)$$

$$\lambda = \{\boldsymbol{C}\boldsymbol{H}_x^{-1}(\boldsymbol{C}+\boldsymbol{\eta})^T\}^{-1}\{\boldsymbol{C}\boldsymbol{H}_x^{-1}(\boldsymbol{u}-\boldsymbol{h}_x\dot{\boldsymbol{x}}-\boldsymbol{g}_x)-\ddot{\boldsymbol{x}}\}, \quad (4)$$

where $\boldsymbol{q} \in \boldsymbol{R}^2$ is the joint angle vector, $\boldsymbol{H}_x \in \boldsymbol{R}^{2\times 2}$ is the positive definite inertia matrix of manipulator, $\boldsymbol{h}_x \in \boldsymbol{R}^{2\times 2}$ is centripetal and Coriolis terms, $\boldsymbol{g}_x \in \boldsymbol{R}^2$ is gravity terms, $\boldsymbol{u} \in \boldsymbol{R}^2$ is input vector. $\boldsymbol{C} \in \boldsymbol{R}^2$ is the compliance selection vector given by $\boldsymbol{C} = \partial c(\boldsymbol{x})/\partial \boldsymbol{x}$ and $\boldsymbol{\eta} \in \boldsymbol{R}^2$ is friction on constraint surface given by

$$\boldsymbol{\eta} = [0 \quad -\mu\,\mathrm{sgn}(\dot{x}_2)], \quad (5)$$

where μ is unknown friction constant between the constraint surface and end-effector.

It is well known that manipulator is characterized by following properties;

P1. The inertia and the centripetal and Coriolis matrix satisfy the following relationship.

$$\boldsymbol{x}^T(\dot{\boldsymbol{H}}_x - 2\boldsymbol{h}_x)\boldsymbol{x} = 0 \quad \forall \boldsymbol{x} \in \boldsymbol{R}^2. \quad (6)$$

P2. The matrix $\boldsymbol{H}_x$ is symmetric positive definite and

there exist bounded constant values $\sigma_i(i=1,2)$ such that

$$\sigma_1 \boldsymbol{x}^T\boldsymbol{x} \le \boldsymbol{x}^T\boldsymbol{H}_x\boldsymbol{x} \le \sigma_2\boldsymbol{x}^T\boldsymbol{x} \quad \forall \boldsymbol{x} \in \boldsymbol{R}^2 . \tag{7}$$

P3. There exist positive constants $\sigma_i(i=3,4)$ such that

$$\|\boldsymbol{h}_x\| \le \sigma_3\|\dot{\boldsymbol{q}}\|, \ \|\boldsymbol{g}_x\| \le \sigma_4 . \tag{8}$$

In **Fig. 1**, x_2 is selected as the motion controlled subspace. Let's define motion space as $p = d(\boldsymbol{x}) = x_2$. Then, we obtain

$$H_p\ddot{p} + h_p\dot{p} + n_1\lambda - (cn_2 - n_3)\dot{x}_\lambda - kn_2x_\lambda + g_p = u_p , \tag{9}$$

$$\begin{aligned}(n_4+1)\lambda = {} & u_\lambda - n_5u_p + n_6\dot{p} - (cn_7 - n_8)\dot{x}_\lambda \\ & - kn_7x_\lambda + n_9 ,\end{aligned} \tag{10}$$

where

$$\left.\begin{aligned} H_p &= \boldsymbol{F}^T\boldsymbol{H}_x\boldsymbol{F} & h_p &= \boldsymbol{F}^T\boldsymbol{h}_x\boldsymbol{F} \\ g_p &= \boldsymbol{F}^T\boldsymbol{g}_x & \psi &= (H_p^{-1}\boldsymbol{H}_x\boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{I}) \\ n_1 &= \boldsymbol{F}^T(\tfrac{1}{m}\boldsymbol{H}_x\boldsymbol{E} + \boldsymbol{\eta}^T) & n_2 &= \tfrac{1}{m}\boldsymbol{F}^T\boldsymbol{H}_x\boldsymbol{E} \\ n_3 &= \boldsymbol{F}^T\boldsymbol{h}_x\boldsymbol{E} & & \\ n_4 &= \tfrac{1}{m}\boldsymbol{E}^T\psi\boldsymbol{H}_x\boldsymbol{E} - H_p^{-1}\boldsymbol{E}^T\boldsymbol{H}_x\boldsymbol{F}\boldsymbol{F}^T\boldsymbol{\eta}^T & & \\ n_5 &= \boldsymbol{E}^T\boldsymbol{H}_x\boldsymbol{F}H_p^{-1} & n_6 &= \boldsymbol{E}^T\psi\boldsymbol{h}_x\boldsymbol{F} \\ n_7 &= \boldsymbol{E}^T\psi\boldsymbol{H}_x\boldsymbol{E} & n_8 &= \boldsymbol{E}^T\psi\boldsymbol{h}_x\boldsymbol{E} \\ n_9 &= \boldsymbol{E}^T\psi\boldsymbol{g}_x & & \\ u_p &= \boldsymbol{F}^T\boldsymbol{u} & u_\lambda &= \boldsymbol{E}^T\boldsymbol{u} \end{aligned}\right\} \tag{11}$$

$\boldsymbol{E}, \boldsymbol{F}$ are vector defined as

$$\begin{bmatrix}\boldsymbol{C}\\ \boldsymbol{D}\end{bmatrix}^{-1} = [\boldsymbol{E} \quad \boldsymbol{F}] \qquad (\boldsymbol{D} = \partial d(\boldsymbol{x})/\partial \boldsymbol{x} = [0 \quad 1]). \tag{12}$$

There exist bounded constant values $\sigma_i(i=6,\cdots,14)$ such that $\sigma_6 \le m$, $k \le \sigma_7$, $c \le \sigma_8$, $\mu \le \sigma_9$, $\|\boldsymbol{C}\| \le \sigma_{10}$, $\|\boldsymbol{D}\| \le \sigma_{11}$, $\|\boldsymbol{E}\| \le \sigma_{12}$, $\|\boldsymbol{F}\| \le \sigma_{13}$, $\|\boldsymbol{\eta}\| \le \sigma_{14}$.

The task space is completely partitioned into the motion control subspace and force control subspace. The robust motion/force controller is designed based on (9) and (10).

3. Robust Motion and Force Controller

To develop a tracking controller, the tracking errors are defined as

$$\left.\begin{aligned} \tilde{p} &= p - p_d \\ s &= \beta^{-1}\tilde{p} + \dot{\tilde{p}} \\ \tilde{\lambda} &= \lambda - \lambda_d \end{aligned}\right\}, \tag{13}$$

where p_d and λ_d are desired trajectories for motion and force, respectively. β is positive design parameter introduced to improve tracking performance of motion. Motion and force tracking error equations are given by

$$\begin{aligned} H_p\dot{s} = {} & \beta^{-1}u_p + H_p\ddot{\tilde{p}} - \beta^{-1}H_p\ddot{p}_d - h_ps + h_p\tilde{p} \\ & - \beta^{-1}h_p\dot{p}_d - \beta^{-1}n_1\lambda + \beta^{-1}(cn_2 - n_3)\dot{x}_\lambda \\ & + \beta^{-1}kn_2x_\lambda - \beta^{-1}g_p , \end{aligned} \tag{14}$$

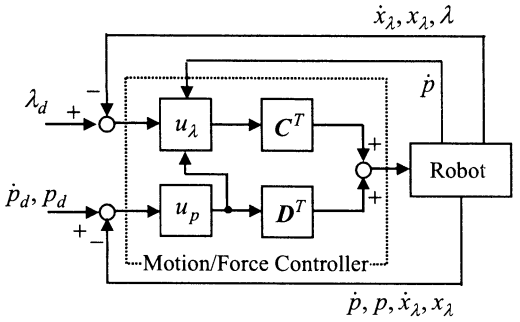


Fig. 2 Motion/Force controller

$$\begin{aligned} (n_4+1)\tilde{\lambda} = {} & u_\lambda - n_5u_p + n_6\dot{p} - (cn_7 - n_8)\dot{x}_\lambda - kn_7x_\lambda \\ & + n_9 + (n_4+1)\lambda_d . \end{aligned} \tag{15}$$

Analyzing the derivative of a positive definite function with respect to the error equations above, the following robust motion /force tracking controller is developed. The configuration of the control system is shown in **Fig. 2**. The stability of the proposed controller is shown in the next section.

$$\boldsymbol{u} = \boldsymbol{D}^Tu_p + \boldsymbol{C}^Tu_\lambda \tag{16}$$

$$\begin{aligned} u_\lambda = {} & -\gamma\{\rho_1|u_p| + (\rho_2 + \rho_5)|\dot{x}_\lambda||\dot{p}| + \rho_2|\dot{p}|^2 \\ & + \rho_5|\dot{x}_\lambda|^2 + \rho_3|\dot{x}_\lambda| + \rho_4|x_\lambda| + \rho_6\}\tilde{\lambda} - \rho_7\tilde{\lambda} \end{aligned} \tag{17}$$

$$\begin{aligned} u_p = {} & -(\beta^4|s|^2|\tilde{p}|^2 + \beta^4|\tilde{p}|^4 + \beta^4|\tilde{p}|^2 + \beta^4|s|^2 \\ & + \beta^2|\tilde{p}|^2|\dot{x}_\lambda|^2 + \beta^2|s|^2|\dot{x}_\lambda|^2 + |\dot{x}_\lambda|^2 + |x_\lambda|^2 \\ & + \beta^2)s \end{aligned} \tag{18}$$

Where, γ is positive design parameter introduced to improve tracking performance of force and $\rho_i(i=1,\cdots,7)$ are positive bounded constant values such that

$$\left.\begin{aligned} |n_5| &\le \rho_1 & |n_6| &\le \rho_2(|\dot{x}_\lambda| + |\dot{p}|) \\ |\hat{c}n_7| &\le \rho_3 & |\hat{k}n_7| &\le \rho_4 \\ |n_8| &\le \rho_5(|\dot{x}_\lambda| + |\dot{p}|) & |n_9| + |n_4+1||\lambda_d| &\le \rho_6 \\ |n_4| &\le \rho_7 & & \end{aligned}\right\} \tag{19}$$

4. Stability Analysis

In this section, at first, **i)** the boundedness of force tracking error is shown in the case of $c \ne 0$. Using the fact, next, **ii)** the boundedness of motion tracking error is proved. When the viscosity constant c of environment parameter is zero, x_λ becomes unstable. Finally, **iii)** a new definition of force tracking error is introduced to assume the stability x_λ.

i) Using the controller (16)-(18), force tracking error is given by

$$\begin{aligned} \tilde{\lambda} = T_1[& \gamma\{\rho_1|u_p| + (\rho_2 + \rho_5)|\dot{x}_\lambda||\dot{p}| + \rho_2|\dot{p}|^2 \\ & + \rho_5|\dot{x}_\lambda|^2 + \rho_3|\dot{x}_\lambda| + \rho_4|x_\lambda| + \rho_6\} + T_2]^{-1} , \end{aligned} \tag{20}$$

$$\left.\begin{aligned} T_1 &= -n_5 u_p + n_6 \dot{p} - (cn_7 - n_8)\dot{x}_\lambda - kn_7 x_\lambda \\ &\quad + n_9 + (n_4 + 1)\lambda_d \\ T_2 &= n_4 + 1 + \rho_7 \end{aligned}\right\}. \quad (21)$$

From (19) it can be seen that $|T_1|$ satisfies the relation

$$\begin{aligned} |T_1| &\le \rho_1 |u_p| + (\rho_2 + \rho_5)|\dot{x}_\lambda||\dot{p}| + \rho_2 |\dot{p}|^2 \\ &\quad + \rho_5 |\dot{x}_\lambda|^2 + \rho_3 |\dot{x}_\lambda| + \rho_4 |x_\lambda| + \rho_6 \ . \end{aligned} \quad (22)$$

From (20)-(22) and $T_2 \ge 0$ it is concluded that $|\tilde{\lambda}| \le 1/\gamma$. That is, force tracking error is bounded and, moreover, the norm of force tracking error can be arbitrarily reduced by increasing the value of design parameter γ.

ii) To analyze the stability of motion tracking error, a positive definite function $V_1 = sH_p s$ is introduced. It can be seen from the boundedness of force tracking error that there exists positive constant ε_1 such that

$$\begin{aligned} \dot{V}_1 &= 2sH_p\dot{s} + H_p s^2 \\ &\le 2s(\beta^{-1}u_p + H_p\ddot{\tilde{p}} - \beta^{-1}H_p\ddot{p}_d - h_p s + h_p\tilde{p} \\ &\quad - \beta^{-1}h_p\dot{p}_d - \beta^{-1}n_1\lambda + \beta^{-1}(cn_2 - n_3)\dot{x}_\lambda \\ &\quad + \beta^{-1}kn_2 x_\lambda - \beta^{-1}g_p) + H_p s^2 \\ &\le 2\beta^{-1}u_p s + 2\beta^{-1}(\beta^4|s|^2|\tilde{p}|^2 + \beta^4|\tilde{p}|^4 + \beta^4|\tilde{p}|^2 \\ &\quad + \beta^4|s|^2 + \beta^2|\tilde{p}|^2|\dot{x}_\lambda|^2 + \beta^2|s|^2|\dot{x}_\lambda|^2 + |\dot{x}_\lambda|^2 \\ &\quad + |x_\lambda|^2 + 0.5\beta^2)s + \varepsilon_1 \\ &= -\beta H_p^{-1}|s|^2 + \varepsilon_1 \ . \end{aligned} \quad (23)$$

It follows form (23) that $|s|$ is bounded and satisfies

$$|s|^2 \le \frac{\varepsilon_2}{\beta} \quad (\because s(0) = 0), \quad (24)$$

where ε_2 is positive constant value independent of the design parameter β. To analyze the motion tracking performance in detail, a positive function V_2 is defined as $V_2 = \tilde{p}^2$. There exists a positive constant ε_3 such that

$$\begin{aligned} \dot{V}_2 &= 2\tilde{p}\dot{\tilde{p}} = 2\tilde{p}(\beta s - \beta\tilde{p}) \le -2\beta|\tilde{p}|^2 + 2\beta|\tilde{p}||s| \\ &\le -\beta|\tilde{p}|^2 + \beta|s|^2 \le -\beta V_2 + \varepsilon_3 \ . \end{aligned} \quad (25)$$

From the relation above, we obtain

$$|\tilde{p}|^2 \le e^{-\beta t}|\tilde{p}(0)|^2 + \frac{\varepsilon_3}{\beta}. \quad (26)$$

It is concluded from (26) that motion tracking error is bounded and, moreover, the norm of motion tracking error can be arbitrarily reduced by increasing the value of design parameter β.

iii) We must consider the case when viscosity constant c of environment parameter is zero. In the closed loop system using the controller (16)-(17), it can be seen from (1) and (13) that x_λ can be rewritten as

$$x_\lambda(s) = \frac{1}{ms^2 + cs + k}(\tilde{\lambda} + \lambda_d). \quad (27)$$

It follows from (27) that when $c \to 0$, unstable vibration occur in the signal x_λ. To overcome the problem, we define the new definition of force tracking error as $\tilde{\lambda} = \lambda - \lambda_d + \alpha\dot{x}_\lambda$ where α is imaginary viscosity constant. Using same way of **i)**, it can be shown that the norm of force tracking error $\tilde{\lambda}$ can be arbitrarily reduced. Using the new force tracking error, it can be seen from (1) that x_λ can be rewritten as

$$x_\lambda(s) = \frac{1}{ms^2 + (c+\alpha)s + k}(\tilde{\lambda} + \lambda_d). \quad (28)$$

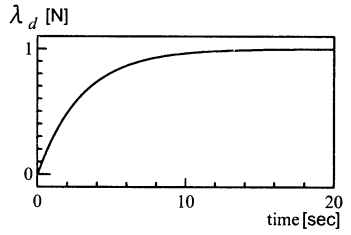
It is obvious from (28) that x_λ becomes stable and unstable vibration dose not occur even if $c \to 0$. Then, contact force λ between the environment and the end-effector can be rewritten as

$$\lambda(s) = \lambda_d + \tilde{\lambda} - \frac{\alpha s}{ms^2 + (c+\alpha)s + k}(\tilde{\lambda} + \lambda_d). \quad (29)$$

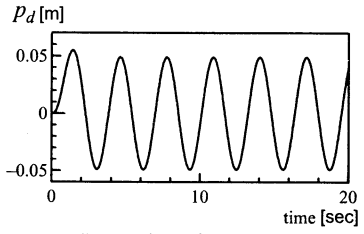
Even when $\tilde{\lambda} \to 0$, the signal $\lambda - \lambda_d$ does not become zero in general. However, it is assured that error signal $\lambda - \lambda_d$ converges to zero in the case of $\lim_{t\to\infty}\dot{\lambda}_d = 0$. Moreover, the convergent rate to a constant desired value can be improved. This fact is shown by using numerical simulations.

4. Simulation Results

In this simulation, the controller (16)-(18) is designed for a two-link robot manipulator shown in **Fig. 1**. The values of the manipulator and the environment are given by link length: $l_1 = 0.5$m , $l_2 = 0.5$m , mass of links: $m_1 = 1.5$kg , $m_2 = 1.5$kg , inertia of links: $I_1 = 0.5$kgm^2 , $I_2 = 0.5$kgm^2 ,

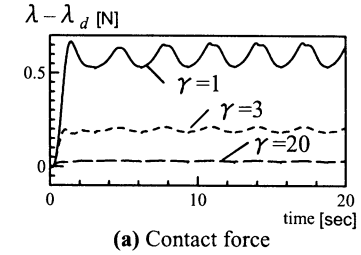


(a) Contact force

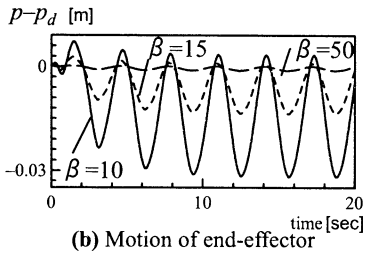


(b) Motion of end-effector

Fig. 3 Desired trajectory

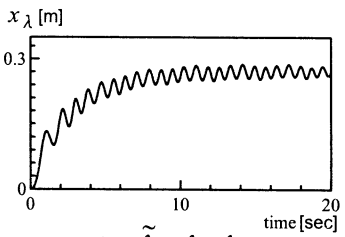


(a) Contact force

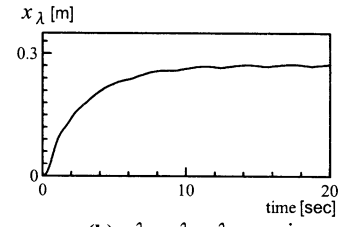


(b) Motion of end-effector

Fig. 4 Tracking error responses

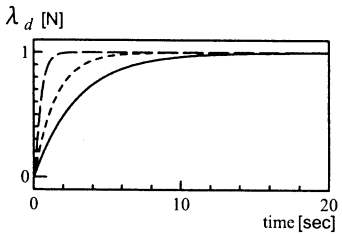


(a) $\tilde{\lambda} = \lambda - \lambda_d$

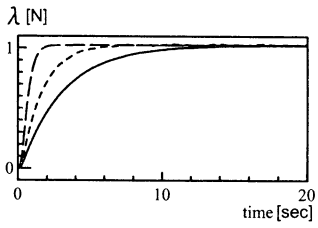


(b) $\lambda = \lambda - \lambda_d + \alpha \dot{x}_\lambda$

Fig. 5 Comparison in the case of $c = 0$



(a) Desired trajectory



(b) Contact force

Fig. 6 Force tracking error response when desired trajectory is changed

and $m = 0.5\text{kg}$, $c = 1\text{Ns/m}$, $k = 5\text{N/m}$, $\mu = 0.1$. It is assumed that these parameters are unknown but the upper bound ρ_i given by (19) are known. It is also assumed that the end-effector contacts with constraint surface before starting control.

The initial value of the manipulator is $q_0 = [120° \quad -120°]$. The desired trajectories of motion and force are shown in **Fig. 3**. **Fig. 4 (a)** shows force tracking error responses for $\gamma = 1, 3, 20$ in the case when the design parameter β is set to the fixed value $\beta = 50$. **Fig. 4 (b)** shows motion tracking error responses for $\beta = 10, 15, 50$ in the case when the design parameter γ is set to the fixed value $\gamma = 20$. As shown in **Fig. 4**, it is concluded that both the tracking performance can be easily improved by using design parameter β and γ, respectively.

In **Fig. 5**, x_λ for $\gamma = 3$, $\beta = 50$ are shown in the case of $c = 0$. **Fig. 5 (a), (b)** show x_λ in the closed loop system using tracking error $\tilde{\lambda} = \lambda - \lambda_d$ and $\tilde{\lambda} = \lambda - \lambda_d + \alpha \dot{x}_\lambda$, respectively. As shown in **Fig. 5**, using tracking error $\tilde{\lambda} = \lambda - \lambda_d + \alpha \dot{x}_\lambda$, vibration does not occur in x_λ. In **Fig. 6 (b)**, force responses are shown in the case when desired trajectory λ_d is changed as shown in **Fig. 6 (a)**. It can be seen that the convergent rate to a constant desired value can be improved arbitrarily by changing the convergent rate of desired trajectory.

5. Conclusion

In this paper, we proposed the robust motion/force tracking control schemes for two-link robot manipulator in contact with surface with unknown stiffness and viscosity. Using the way of partitioning into two subspaces, even when the environment has unknown flexibility, the proposed controller can control motion and force independently. Moreover, the proposed controller has robustness to uncertainties of the environment, and motion and force tracking errors can be improved arbitrarily by using independent design parameter β and γ, respectively.

References

[1]. Tsuneo Yoshikawa, "Force Control of Robot Manipulators", Proc. IEEE. Int. Conf. on Robotics and Automation, pp220-226, 2000

[2]. M.H. Raibert and J.J.Craig,"Hybrid Position/Force Control of Manipulator", Trans. ASME, J. DSMC, vol.102, June, pp.126-133, 1981

[3]. B.H.Park, Jin.S.Lee and S.Y.Park, "An Adaptive LearningControl Method for Constrained Motion of Uncertain Robotic System", Proc. IEEE. Int. Conf. on Robotics and Automation, pp531-536, 1997

[4]. Louis L.Whitcomb, Suguru Arimoto, Tomohide Naniwa and Fumio Ozaki, "Adaptive Model-Based Hybrid Control of Geometrically Constrained Robot Arms", IEEE. on Robotics and Automation, vol.13, no.1, February, 1997

[5]. Bin Yao and Masayoshi Tomizuka, "Adaptive Robust Motion and Force Tracking Control of Robot Manipulators in Contact With Compliant Surfaces With Unknown Stiffness", Trans. ASME, J. DSMC, vol.120, June, pp.232-240, 1998

[6]. H.G.Tanner and K.J. Kyriakopoulos, "Position and Force Control by Reaction Compensation", Proc. IEEE. Int. Conf. on Robotics and Automation, pp3926-3931, 2001

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

AN EFFECTIVE ADAPTIVE AUTOPILOT FOR SHIPS

Thi-Xuan DOAN, Van-Quang HOANG
Hung Long Co., Ltd, Hanoi, Vietnam
Thi-Thuy-Anh DUONG, Manh-Tan BUI
Hong Thang Co, Ltd.., Hanoi, Vietnam,
duongthithuyanh@yahoo.com

Thi-Kim-Thoa NGUYEN, Duc-Truong LUONG
Phan Anh Co., Ltd, Hanoi, Vietnam
Thi-Duc LE, Van-Loc DO, Thi-Hoa LE
Bac ninh Consultant and Investment Co, Ltd.., Bacninh
lebavung2002@yahoo.com

Abstract

Ship motion is a complex controlled process with several hydrodynamic parameters that vary in wide ranges with respect to ship load condition, speed and surrounding conditions (such as wind, current, tide, etc.). Therefore, to effectively control ships in a designed track is always an important task for ship masters. This paper presents an effective adaptive autopilot for ships that ensure the optimal accuracy, economy and stability characteristics. The PID control methodology is modified and parameters of a PID controller is designed to satisfy conditions for an optimal objective function that comprised by heading error, resistance and drift during changing course, and loss of surge velocity or fuel consumption. Designing of the controller for course changing process is based on the Model Reference Adaptive System (MRAS) control theory, while as designing of the automatic course keeping process is based on the Self Tuning Regulator (STR) control theory. Simulation (using MATLAB software) in various disturbance conditions shows that in comparison with conventional PID autopilots, the designed autopilot has several notable advantages: higher course turning speed, lower swing of ship bow even in strong waves and winds, high accuracy of course keeping, shorter time of rudder actions smaller times of changing rudder direction.
Key words: PID control, adaptive control, MRAS, STR, ship autopilot

1 Introduction

Autopilot is one of the most important equipments used in ships. Autopilots are not just used to lead the ship on a desired trajectory, but also to raise the safety level of the journey and control the ship economically. An optimal autopilot can shorten 3-5% length of the journey and therefore, reduce the fuel consumption, especially in bad whether conditions. A good autopilot can help to avoid undesired situations on maneuvering and remarkably reduce the numbers of ship operators. In the last decades, taking advantages of drastic development of micro-electronics and control theory several new and effective methods have been proposed and developed for designing Ship Autopilots [1], [2], [3]. Ship Autopilots designed based on the PD and PID controllers are simple, reliability and easy to construct, however their performance in various environmental conditions is not as good as desired. Therefore, ship Autopilots with PD or PID controllers are usually required aids from operators to adjust controllers' parameters corresponding to navigating conditions. This study concentrates on design of ship autopilots based on adaptive control theory and optimal control theory. Several maneuvers' simulations have been carried out using MATLAB software to verify the effectiveness of the method, and excellent results achieved show remarkable advantages of the ship autopilot in comparison with other conventional autopilots such as PD or PID ones.

The paper is organized as follows: The first session gives the nature of the problems considered, previous works, purpose and contribution of this study as well as related research. The second session presents the mathematical model used in this study to express ship steering motion. The model contains both linear and non-linear features of ship steering. Control algorithm used for the adaptive autopilot and designing issues are presented and discussed in details in the third session. To verify the effectiveness of the adaptive autopilot, MATLAB software was used to simulate ship motion during several different maneuvers. Session 4 gives simulation results, including comparison of the quality of a conventional PID autopilot and the adaptive autopilot. And finally, the fourth session draws some conclusions form this study and points out some possible direction for further study.

2 Mathematical model

The mathematical model used to express ship motions in this study is as follows [4], [5], [6], [7], [8], [9], [10]:

$$T_r(dr/dt) + H_r(r) = K_r\delta + K_r'\delta \quad (1)$$

$$T_u(du/dt) + H_u(u) = K_u r^2 \quad (2)$$

$$v + H_v(r) = 0 \quad (3)$$

$$r = d\Psi_L/dt \quad (4)$$

$$\Psi = \Psi_L + \Psi_H \quad (5)$$

where: T_r, T_u, K_r, K_r', K_u are ship hydrodynamic parameters depending on ship speed on the course, load condition and weather conditions such as waves, winds, tidal, water depth and so on;

δ is rudder angle;

K_δ' expresses deflection due to winds and tidal;

Ψ, K_r', K_r' are ship heading, ship heading in no noises conditions and ship heading in high noises

condition, respectively;

u, v, r are ship surge, sway and yaw velocities, respectively;

H_u, H_v, H_r are functions that express the nonlinearity of the system corresponding to u, v, r, respectively. In general, they have the following form:

$$H_a(a) = k_3 a^3 + k_2 a^2 + k_1 a + k_0 \quad (6)$$

k_0 shows the non-symmetry of the ship ($k_0 = 0$ means the ship is ideal symmetric), $k_1 = 1$ for stable ships and $k_1 = -1$ for non-stable ships;

Usually, in order to avoid the influences of ship speed and dimension, the above equations are expressed in a non-dimensional form by multiply the parameters, coefficients with following non-dimensional coefficients:

$$r^* = (L/u)r;\ v^* = v/u;\ u^* = (u/U_0) \quad (7)$$

with L and U_0 are ship length and speed, respectively.

Then, the above equations (1) - (5) can be re-written as follows:

$$T_r^*(L/u)(dr^*/dt) + H_r(r^*) = K_\delta^*\delta + K_r^{'*}\delta \quad (8)$$

$$T_u^*(L/u)(du^*/dt) + H_u(u^*) = K_u^* r^{*2} \quad (9)$$

$$v^* + H_v(r^*) = 0 \quad (10)$$

$$\Psi = \Psi_L + \Psi_H \quad (11)$$

3 Control algorithm

Usually a conventional PID autopilot is used, and the control signal δ has the following form:

$$\delta = K_p(\Psi - \Psi_r) = K_d(d\Psi/dt) + K_i\int_0^t(\Psi_r - \Psi)dt \quad (12)$$

where Ψ_r is reference (desired) heading, K_p, K_i, K_d are coefficients adjusted by ship operators according to ship speed, load and weather conditions and so on.

In good weather conditions, this kind of autopilots may work rather well (with not so high technical and economical characteristics). But in bad weather conditions (with presence of waves, winds, tidal, etc.), ship hydrodynamic parameters will vary in large scale, then this kind of autopilots shows tremendous disadvantage and in many cases, even integration and differentiation terms should be excluded. This will result in rocking of the ship and the rudder has to work with high frequency, that reduces durability of the steering systems. A method to overcome this disadvantage is to create non-sensitive zones, but it causes large deflection in ship course and hence requires large fuel consumption.

This paper presents another method to solve those problems, that is using optimal control algorithms to automatically the controller's parameters corresponding to the above-mentioned conditions (ship speed, load conditions, tidal, waves, winds and so on). The optimal control function is as follows:

$$J = \min\left\{(a/T)\int_0^T((\Psi_r - \Psi)^2 + l_1 r^2 + l_2\delta^2)d\tau\right\} \quad (13)$$

In this formula, the first term expresses heading error $e = \Psi - \Psi_r$, the second term expresses the influence of resistance and deflection during course turning process, and the third term expresses loss of ship speed and hence, the fuel consumption. In this study, the steering system is divided into 2 different processes with following characteristics:

Course changing process: In this process, economy plays a very modest role, but the most important is the accuracy of the heading after course changing process. Therefore, l_1, l_2 should have rather small values. Usually, 2 types of course changing are use, one is course changing with constant ship turning rate and the another is course changing with constant turning radius.

Here, the autopilot is designed based on application of Model Reference Adaptive System (MRAS) and the controller's parameters have the following forms [11], [12], [13], [14], [15]:

$$K_p = K_{p0} + \gamma_1\int_0^t(p_{12}e + p_{22}\dot{e})(\Psi'' - \hat{\Psi})d\tau \quad (14)$$

$$K_d = K_{d0} + \gamma_2\int_0^t(p_{12}e + p_{22}\dot{e})\dot{\hat{\Psi}}d\tau \quad (15)$$

$$K_i = K_{i0} + \gamma_3\int_0^t(p_{12}e + p_{22}\dot{e})d\tau \quad (16)$$

here $\hat{\Psi}$ is the estimated heading, γ_i (i = 1, 2, 3) are self tuning coefficients.

Automatic course keeping process: This process should be divided into 2 different cases: (1) Common case: when the ship running on the sea, then the economy plays an important role, therefore l_1, l_2 should have rather large values; (2) when the ship running in rivers, narrow channels or on confined water conditions, where the accuracy of course keeping is much more important than the economy features, therefore l_1, l_2 have rather values as in course changing process.

In this process, the autopilot is designed based on the Self tuning Regulator (STR) with following parameters of estimator:

$$(K_m^*/T_m^*) = (K_m^*/T_m^*)_0 - \gamma_1\int_0^t e(\delta - K_{im})d\tau \quad (17)$$

$$(1/T_m^*) = (1/T_m^*)_0 + \gamma_2\int_0^t e\dot{\Psi}_m d\tau \quad (18)$$

$$K_{im} = -\gamma_3\int_0^t ed\tau \quad (19)$$

Hence, the controller's parameters are as follows:

$$\delta_r = K_p(\Psi_r - \hat{\Psi}) - K_d\dot{\hat{\Psi}} + K_i \quad (20)$$

$$K_p = (z/2)(U_0/U);\ 0.5 < 2.5\xi < 5.0 \quad (21)$$

$$K_d = (L/U)(\sqrt{(1 + 2K_pK_m^*T_m^*)} - 1)/K_m^*)$$

$$U = \sqrt{u^2 + v^2};\ K_p < K_d < K_pL/U \quad (22)$$

$$K_i = K_{im}^{'} \quad (23)$$

here z is the optimal coefficient, $1 \le z \le 5$, z = 1 for accuracy optimal case and z = 1 for economy optimal case, subscript m denotes parameters of the estimator, K'_{im} is average value of K_{im}.

4 Simulation results and analysis

To verify the effectiveness of the new adaptive autopilot, the presented autopilot system has been simulated using MATLAB software. Several different ship maneuvers have been carried out, including course keeping, course changing processes and their combination. This session gives simulation results of a maneuver that the ship begins from zero heading angle changes its course to -23 deg., runs straight about 10 minutes, then change its course to 15 deg and finally keeps a straight course.

Figure 1 shows ship heading angles produced by the conventional PID autopilot and this adaptive autopilot during this maneuver. From this figure, the adaptive autopilot controls the ship rather well and the ship following the reference track much better than the PID autopilot case.

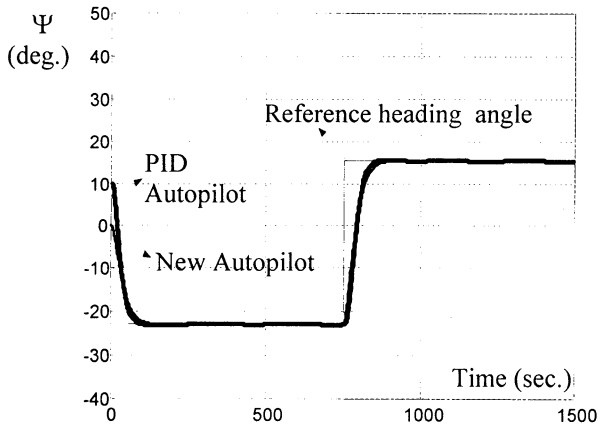


Fig. 1: Comparison between heading angle of PID autopilot and new autopilot in a maneuver

To consider the quality of the control processes, errors of ship heading angle for both PID autopilot and the adaptive autopilot are simulated and drawn in Figure 2. This figure shows clearly that the overshoot values of the new adaptive autopilot in course changing processes were reduced about 2 times compared to corresponding values of the PID autopilot.

Variation of parameters K_p, K_i, K_d during the above described maneuver are plotted in Figure 3. This figure shows that the autopilot adapted rather well with both the course changing and course keeping processes: in the course keeping process, the control parameters were kept almost at constant values, while as in the course changing process, the control parameters were changed accordingly. That proves the effectiveness of the adaptive autopilot in course keeping and course changing process as analyzed in session 3.

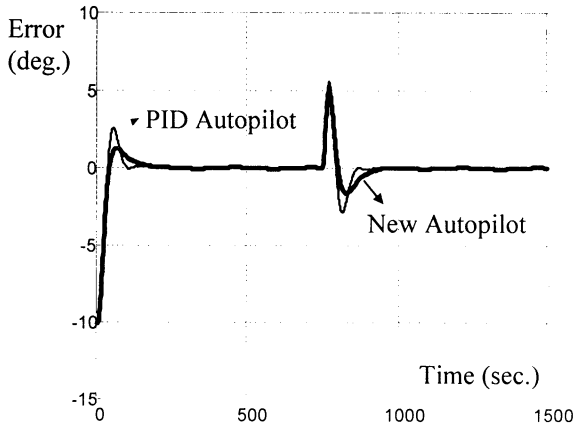


Fig. 2: Comparison between heading errors of PID autopilot and new autopilot in a maneuver

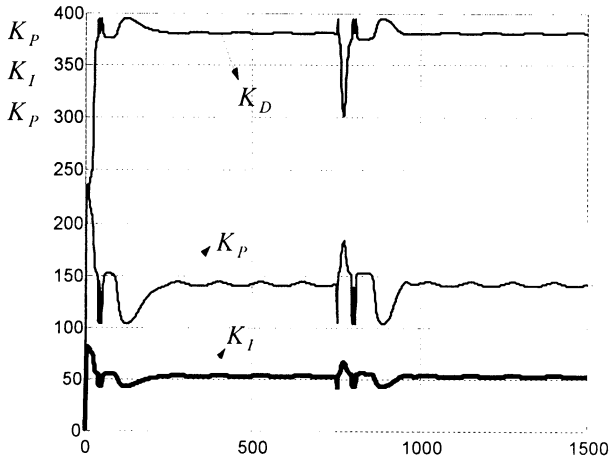


Fig. 3: Variation of control parameters of the new autopilot during the maneuver

5 Conclusions and Future Works

An effective adaptive autopilot for ships that ensures the optimal accuracy, economy and stability characteristics has been presented and its design issues have been discussed. The PID control methodology has been modified and designing of the controller for course changing process is based on the Model Reference Adaptive System (MRAS) control theory, while as designing of the automatic course keeping process is based on the Self Tuning Regulator (STR) control theory. Simulation (using MATLAB software) shows that in comparison with conventional PID autopilots, the designed autopilot has several notable advantages: higher course turning speed, lower swing of ship bow even in strong waves and winds, high accuracy of course keeping, shorter time of rudder actions smaller times of changing rudder direction.

The new ship autopilot is expected to have several important features, however, to be able to construct a real ship autopilot of this type, many further studied should be carried out. Among them some can be pointed out here. First is consideration of strong environment disturbances

such as tidal and currents. Second is choosing proper control parameters. Third is full consideration of an automatically design process. And finally, the autopilot should be put into real ships for practical use and correction.

Acknowledgements

This work was carried out with kind help from lectures and students of Vietnam Maritime University in HCMC. The authors would like to express sincere thanks for their support useful discursions.

References

[1] J. V. Amerongen and H. R. Nauta Lemke, "Criteria for optimum steering of ships", *Proceedings symposium on ship steering automatic control*, 1980.

[2] C. G. Kallstrom, "Identification and adaptive control applied to ship steering", PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Sweden, 1970.

[3] M. D Le, "Online Estimation of Ship Steering Dynamics and Its Application in Designing An Optimal Autopilot", *Proc. of IFAC Computer Aided Control System Design, CACSD2000*, Vol. 1, 7-12, 2000.

[4] Abkowits, M. A.: Measurement of Hydrodynamic Characteristics from Ship Maneuvering Trials by System Identification, *Transactions on SNAME*, Vol.88, No.1, 283-318 (1980).

[5] Astrom, K. J. and Kallstrom, C. G: Identification of Ship Steering Dynamics, *Automatic*a, Vol.12, No.1, 9-22 (1976).

[6] E. E. Allmendinger (ed.), Submersible Vehicle Systems Design, The Society of Naval Architects and Marine Engineers, 1990.

[7] M. A. Abkiwitz, "Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification", *Transactions on SNAME*, pp. 283-318, 1980.

[8] N. H. Norrbin, "On the added resistance due to steering on a straight course", *Proceedings of the 13th ITTC*, 1972.

[9] J. V. Amerongen and A. J. Udink Cate, "Model reference adaptive autopilot for ships", *Automatica*, Vol. 11, pp. 441-449, 1975.

[10] M. Bech, "The reversed spiral test as applied to large ships", *Shipping world*, 1968.

[11] V. N. Afanasiev, V. Konmagorovskii and V. R. Nosov, Mathematical theory of control systems design, Kluwer Academic Publishers group, 1995.

[12] K. J. Astrom, B. Wittenmark, Adaptive control, Academic Press, 1989.

[13] K. Agata, Modern Control Engineering, Kluwer Academic Publishers group, 1994.

[14] V. Komanovskii, V. R. Nosov, Stability of functional differential equations, Academic Press, 1986.

[15] V. Komanovskii, V. R. Nosov, Stability and periodic modes of control system with aftereffect, 1981.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Remote positioning and control architecture of mobile objects with wireless communication

Yong-Ho Kim
Korea Institute of Machinery & Materials(KIMM)
yhkim@icomm.re.kr
Changwon, South Korea

Dong-Hui Yu
Catholic University of Pusan
dhyu@cup.ac.kr
Pusan, South Korea

Young-Jin Lee
Pusan National University
yjlee4@hyowon.pusan.ac.kr
Pusan, South Korea

Abstract

This paper proposes an architecture design of remote positioning and control system for mobile objects. To control mobile objects needs mobile wireless communication infrastructure and positioning system such as GPS. In this paper, we use GPS as a self-positioning system and mobile wireless communication based on wireless data service of PCS network in Korea.

1. Introduction

There are lots of location based application services in IT(Information Technology) area. Location based service system basically consists of remote positioning system, wireless mobile communication system and monitoring and management system. It's expected that it has far-reaching effect to apply this technology of location-based system to automatic control and robot control area. In this paper, we introduce each technologies and consideration points of overall architecture and present an example design and implementation result.

2. Remote positioning and wireless communication

The system architecture consists of two main components, center system to control the remote mobile objects and mobile system carried on the mobile objects, which has the role of self-positioning and communication. Though the internal system flow and structures are very important, these systems, based on wireless mobile communication infrastructure, should satisfy the reliable communication in an air interface. The reliable wireless communication means that the system should be robust, transmission guarantee, data loss free, fast fault (error) recovery, and provide the transparent communication independent of wireless characteristics such as intermittent disconnection and high latency.

We define the session to be the unit of control and propose a session management scheme. Whenever the communication is required, the center system and mobile objects make a connection and assign a session id. Through the session id, the center system manages the activity state, priority, and error control. and transaction synchronization of multiple mobile objects effectively. During the session is activated; the session manager in the center station monitors the wireless communication states. If any problem occurs like intermittent disconnection or transmission errors, the session manager handles these conditions automatically.

2.1 Remote Positioning

GPS is funded by and controlled by the U. S. Department of Defense (DOD). While there are many thousands of civil users of GPS worldwide, the system was designed for and is operated by the U. S. military. GPS provides specially coded satellite signals that can be processed in a GPS receiver, enabling the receiver to compute position, velocity and time. Four GPS satellite signals are used to compute positions in three dimensions and the time offset in the receiver clock.

In order to provide accurate position acquisition, DGPS(Differential GPS) technology and other supplementary techniques can be used like direction sensors, gyro, INS and dead reckoning technologies for GPS satellites are not visible. The resolution of precision of GPS only is about 10 m and the resolution with DGPS and/or dead reckoning sensors is reported as about 2 m.

2.2 Wireless mobile communication

There are various wireless mobile communication technologies according to the

frequency, data bandwidth, service coverage, etc. They are wireless LAN, bluetooth, HomeRF and so on. These technologies should have another devices like access points and interface card. And another wireless mobile communication technology is to use data communication module of PCS(personal communication system). PCS is already deployed and easily accessible in Korea and construction of the network is simpler than other options.

According to the characteristics of applicatio n, wireless mobile communication technologies m ight be selected. In this paper, since we introduce some wide area application, we select data comm unication of PCS considering that PCS is easily accessible in Korea and construction of the net work is simpler than other options [6], [7], [8].

In today's market place, the most popular cellular telephone configuration is the handheld unit that also plugs into the vehicle kit - consumers want wireless communications wherever they go, and use their phones both in vehicles as car phones and outside vehicles as handheld portable. Integrating the GPS unit with the car kit was not an easy one, but starting from E-911, there is rapid evolving technology that GPS is integrated effectively into the portable handheld wireless phone [1].

There are two main advantages in using the existing cellular telephone networks: ubiquitous access to the general information, which is important to end users, and the transparency of mobility, which is important to software development. During a session, the data communication protocols do not see any mobility at all, neither handovers nor location updates. Cellular links have high latency with long variable delays, low and variable throughput, and they are prone to sudden disconnections [1]. The cellular telephone in KOREA offers the circuit switched data service like modem and packet switched data service that support IP stack.

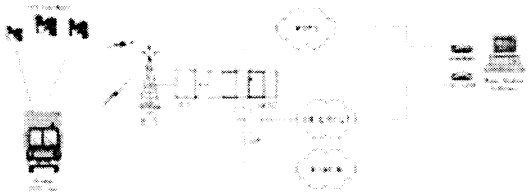


Fig. 1 Configuration of PCS data network.

PCS network supports two transmission modes, packet switched data service and circuit switched data service. Circuit mode operates like the wired modem and packet mode is for mobile Internet access. Packet mode shows faster call connection time than circuit mode but lower the reliability for transmission

2.3 Design Considerations

Reliable wireless communication system address mobile network characteristics and should adapt existing network technology. And it has to utilize the special requirements of application. As an example of remote positioning and control system, we introduce the requirements of the AVLS communication system like below.

- Definition of layered, scaleable and extensible architecture
- Optimisation for wireless narrowband bearers with potentially high latency, intermittent connectivity, limited link capacity, end error-prone links
- Optimisation for efficient uses of communication resources
- Reliable applications and communication
- Secure service
- Creation of flexible user interface with operator control

Since our focus is the reliable mobile wireless communication management in AVLS, we introduce the used bearer communication system and designs for reliable communication system.

3. Protocol architecture design

To reduce the layer interaction overhead and to provide the scalable architecture, we define 3 layers. Each layer has its own specific functions and offers a defined service to the layer above using the service provided by the layer below.

The functions of each layer are as follows.

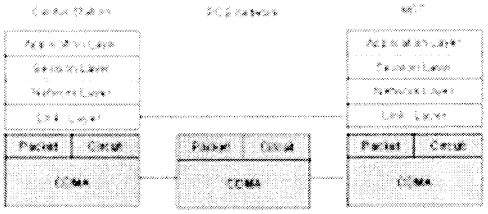


Fig. 2 Communication System based on CDMA

3.1 Session Layer Functions.

We defined the session as the manipulation unit of the communication between the control center and the MO(Mobile Object). Only one session for a MO can be existed for a certain time. Each session has a number of attributes. The most important attributes are those controlling the

exception handling, priority, and establishment of the wireless link on demand. Below, we briefly describe some of these attributes.

- Priority: higher priority is given to expedited messages.
- Link Reconnection Permission: This is the reuse rule of already opened connection.
- Link Reestablishment Permission: This is for the transparent transmission without operator's additional actions. The rule is whether or not the reestablishment of the link-level connection when the sudden disconnection or delay of connection occur.
- State timeout: The timer specifies how long the session can retain the current transmission state. Each state has the unique timer per messages. This makes the session is responsible for the transmission at best. The timer value should be evaluated from the processing time of layers below and the roundtrip time between the MO and the center.
- Acknowledgement scheme: According to the characteristics of the activity, the acknowledgement is needed or not.

The session is composed of the several activities. We categorized the activities like followings and designed with Object oriented approach.

- Unreliable one-way request activity
  - Periodic MO's position reporting
  - Lowest priority
- Reliable one-way request activity, and
  - Notification from MO
- Reliable tow-way requests and reply activity;
  - Commands from center to MO

The session layer enable activities to share and reuse the link connection because it takes so a long time to establish the connection in the circuit, and reduce the connection number of the circuit as possible.

The states of a session are followings.

- *Init*: Session manager wants to create a new session
- *Connecting*: A new session with MO is creating. The session start command is sent to the corresponding MO.
- *Idle*: After the creation of a session, the default state is idle.
- *Requesting*: Data is sent to MO, the state is changed from Idle to Requesting. After receiving the Transmission confirm primitive from MO, the state is changed to idle.
- *Waiting*: the port is occupied and another data is to be sent, the state is changed to waiting and data to be sent is queued.

3.2 Network Layer Functions

This layer also maintains the queues which are consists of down stream queue from the session layer and upstream queue from the transmission layer. Port manager in this layer manages the status of port. There are two port types, packet port for packet switched data service and circuit port for circuit switched data service. For an efficient data transmission, the network layer keeps the connection state. This is also applied to the simultaneous transmission and receipt for one connection. That is, the connection state is set to connected when the packet is transmitted and reset to idle when the acknowledgement of the corresponding packet and/or there is no data to transmit each direction.

3.3 Link Layer Functions

The main functions of link layer are connection establishment, data transmission, and connection reestablishment. As well as, the link layer manages packet port and circuit port. Packet port is the legacy LAN port that is for the packet switched data service from PCS network. This port can be opened when the client initiates the call establishment. Link manager listens to packet port continuously. As the connection request is received, link manager accepts the request and creates a thread responsible for this activity. When the disconnection request from client is received, link manager releases the thread. If a link is suddenly dropped during the activity, the link manager tries to re-establish the link to guarantee the reliable data transmission. If the connection can't establish with deterministic trial times, the link manager notifies the link error of the upper layers.

4. Implementation

In previous sections, we introduce options for remote positioning and wireless mobile communication techniques, design considerations of wireless communication, and design of an example application. This section explains the implementation of an example, Automatic vehicle location service.

Implementation environment of Center system are Windows NT 4.0, Informix DBMS for logging. Mapbasic and Mapinfo are used for

digital map management. MO is assumed a vehicle and remote positioning and communication program of MO is implemented on Windows 98 with Garmin GPS receiver and data kit of PCS. We use the AT command set for the circuit switched data service module and internet socket programming interfaces for packet switched data service module. Packet mode communication shows the traffic patterns like the Internet. For the circuit mode communication, we analyse some features from testing. The time for an entire activity consists of dialling, bearer connection establishment, wireless bandwidth negotiation, and data transmission. Most of elapsed times are related to the wireless bandwidth negotiation. It depends on the PCS network capacity, so it shows the various time delays.

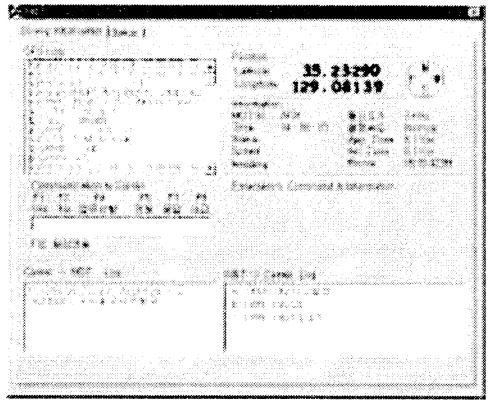


Fig. 3 MO interface

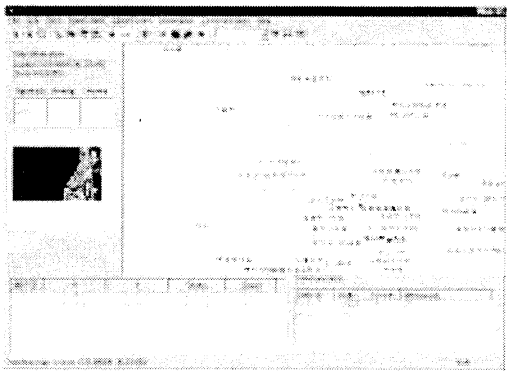


Fig. 4 Control Center interface

Summary

This paper intends to introduce some technologies that seem to be very useful to automatic control and robot control. Remote positioning and wireless communication technologies are well defined and still on going research area. Since the selection scope is very broad, according to the application service, we must select the certain technologies and should combine them. In this paper, we present a design of automatic vehicle location system to provide the wide area coverage and reliable dispatch service.

References

[1] Donghui. Yu(1999), Byungyeun Kim and Youngho Kim. Design of a Wireless Communication System for AVLS. In ICOIN-13, Vol 1. IEEE Computer society 4A-3.1-4A-3.6..

[2] J.G. Markoulidakis(1998), G.L. Lyberopoulos and M.E. Anagnostou. Traffic model for third generation cellular mobile telecommunication systems. In nternational journal of Wireless Information Networks, Vol. 4. Kluwer/Plenum, pp.389-400

[3] Gabriel Montenegro(1992), Masakazu Sengoku and Yoshio Yamaguchi. Time-dependent analysis of mobile communication traffic in a ring-shaped service area with nonuniform vehicle distribution. In IEEE transactions on vehicular technology, Vol. 41, No. 3. IEEE, pp. 243 – 254

[4] Henning Schulzrinne(1999) and Jonathan Rosenberg. The IETF internet Telephony Architecture and Protocols: IEEE network, May/June IEEE. 18 – 23

[5] htttp://www.webproforum.com/wap/

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

[7] Takeo Abe(1994), Masahiro Hayashi and Satoshi Nojo. A software Tool to Support the Reliability Design and Evaluation of Telecommunication Networks. In IEEE JSAC, Vol.12, No.2. IEEE, pp. 345 – 354

[8] Donghui Yu(2000), Youngho kim, sangjeong lee and et al. A Reliable session management for AVLS. 7th ITS congress, ITS America, ERTICO, VERTIS

[9] Sangheon Kim(2001). A Design of session protocol for improvement of reliability of AVLS wireless data communication. In Master's thesis. Pusan National University

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Development of a self-driven personal robot

Tohru Azuma, Yoshihiro Takenaga and Eiji Hayashi
Faculty of computer science and systems engineering, Kyusyu Institute of Technology
680-4, Kawatu, Iizuka-City, Fukuoka Prefecture, Japan

Abstract

In recent years, production technology has become a typical area of application for robots, and mechatronics has decreased various types of workloads for humans. Moreover, robotics technology has made remarkable progress in helping to shoulder the human burden of precise work. With regard to communication between humans and machines, users can accurately control and operate robots used in production technology by means of numerous operation commands and keys. However, at present, these operations are somewhat complicated for the general user. We have aimed to develop a personal robot that can move autonomously.

1. Introduction

In recent years, the rapid development of semiconductor technology has resulted in dramatic advances in computer processing speed and memory capacity. These advances have made it possible to control and process robots to have high degrees of freedom and multiple information inputs from sensors and CCD cameras.

Sony Corporation's AIBO robot is already on the market, as is Honda Corporation's ASIMO. The creation of a robot that uses such techniques constitutes a high level of progress achieved by means of development and marketing. The strong consumer demand for the AIBO has heightened concerns regarding the coexistence of robots in human society, thus necessitating this issue's further study. In Japanese society in the future, the proportion of nuclear families in relation to extended families will increase, and at the same time the elderly will account for the largest share of the population. Together, these trends mean that more elderly people will be living alone, without the aid of their children or grandchildren. So even while it becomes more important to help the elderly in their home lives, there will also be a shortage of people available to provide such help. Therefore, robots are needed to provide such help.

The aim of this study is to develop a personal, robot. The goals of this robot are to perform OR- complete simple tasks that are very easily performed by humans; for example, bringing a cup and a newspaper, or turning a television or other device on and off. In this paper we describe the mechanism, control, and information processing of such a robot.

2. Composition of personal robot

The robot that is the subject of this study has already been used in factories. This type of robot is physically efficient at high output, and works at high speed and with great precision. However, it is too difficult for humans to operate without advanced skills and knowledge.
Communication between the human and the robot is important. Because the personal robot "lives" in the house with humans, situations in the living area and the tasks that the humans require will change.

In the future, the personal robot will have to operate autonomously and to meet the demands humans place upon it, not only in the home but in various environments including offices and sickrooms.

Robots of the present are unable to meet these kinds of demands. We have suggested that, for a robot to operate autonomously, it should be set up in a one-room environment, such as those described above referred to as "finite spaces". In such a space, the robot is capable of moving autonomously, though at the user's direction, to a desired location. This study undertakes the development of a system for basic functions, such as spatial recognition to understand the environment, movement processing based on the finite space map and revision of a path using a visual system. The system and the processing composition are shown in Figure 1. The spatial recognition faculty of the ① is composed of a visual system that lets the robot see or understand the environment, as well as a sensor combination system. The operation determination faculties of the ② are the systems that determine the robot's action, direction and speed via feedback input from the space recognition faculty and feedforward input to refer to the finite spatial map. The design of the robot is shown in Figure 2.

3. Map of finite space

The environment assumed in this study is a finite space such as a family room, an office or a hospital room. The finite space shows the same floors in the building, "shows the same floors" is unclear. It begs the question, and big bumps such as stairs aren't considered. The finite space is the limit of the radius of action of the robot. When the robot moves across an equal finite space many times, it would inefficient for the robot to have to vary its processing each time it sees an obstacle or must otherwise change its route.

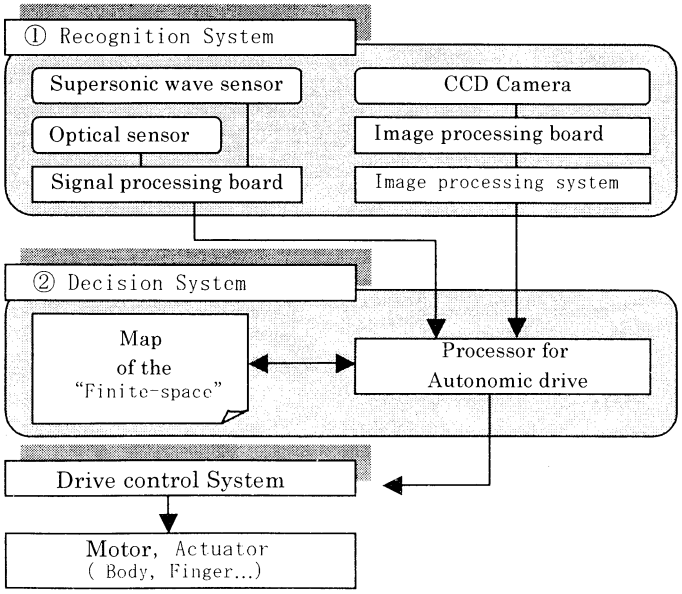


Fig. 1 Processing system for the personal robot.

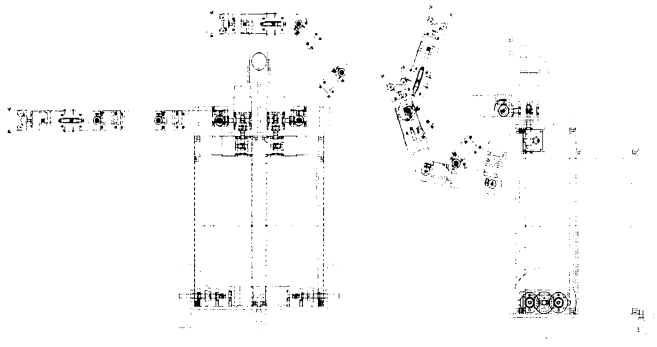


Fig. 2 Robot design

So, the system maintains a map of the finite space. When the robot already knows its finite space, it can control its own movement more efficiently, simply by referring to the map. The finite space map has two-dimensional layout information, including the size of each parameter of the map, the initial position coordinates of the robot, the coordinates of the movement needed to get to the goal position, the number of placed objects and the positions and sizes of those objects. A concept chart of the finite space map is shown in Figure 3.

4. Processing for self-control drive

4.1 Basic processing system

The basic movement operation of the robot is to recognize a placed object, then to reach a destination while avoiding obstacles along the way. At that time, the robot uses the distance sensor to learn the distance, and uses the supersonic wave sensor to detect the obstacle. While moving, the robot coordinates its movement with reference to the map. In this processing, the robot changes over "changes over" is unclear. by reacting to the detection of obstacles. When a robot actually acts, I don't understand "really acts". Both processing reactions proceed by way of basic action. The flow of basic processing is shown in Figure 4, and each form of processing is explained in the following paragraph.

4.2 Basic action

The robot reaches a goal position by combining such operations as going straight and turning. The next sentence is unclear. This action contains the action of going straight by reacting to data from the sensor.

4.3 Reacting to obstacles

The robot can react to obstacles in either of two ways. In one form of reaction, the robot judges which way it must move (right or left) in order to avoid the obstacle. In the other form of action, suppose the robot has, in pursuing the shortest route to its goal, failed to avoid the obstacle. It must then decide how to evade the obstacle. In this case, the robot first uses the distance sensor to measure the distance to the goal and then it gets the coordinates of each of the room's four corners. Then the robot measures the distance between the walls and the obstacle. The next sentence is unclear. It then decides whether or not it can avoid the obstacle by way of the middle goal, by keeping a safe distance away from the corner of the obstacle. In carrying out such an evasion, if the robot reaches a dead end it will then return to the point of its most recent turn, from which it will attempt to evade the obstacle by moving to the opposite side.

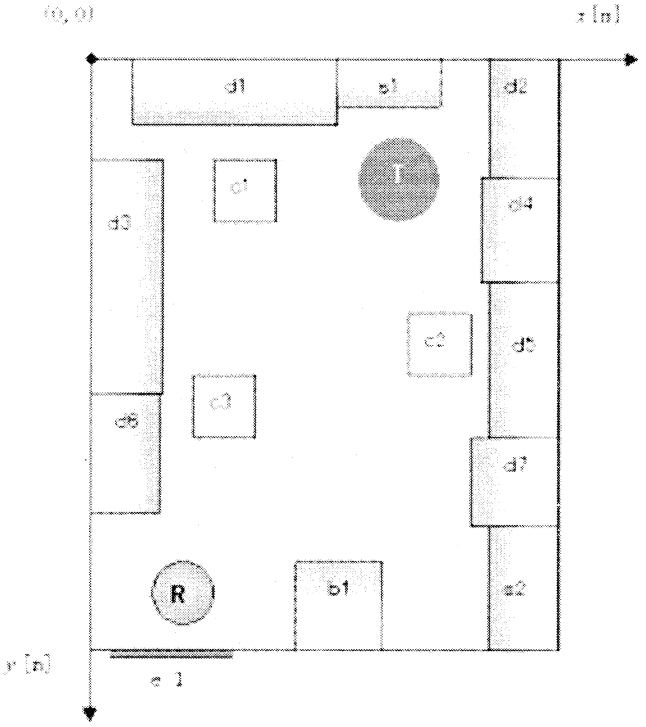
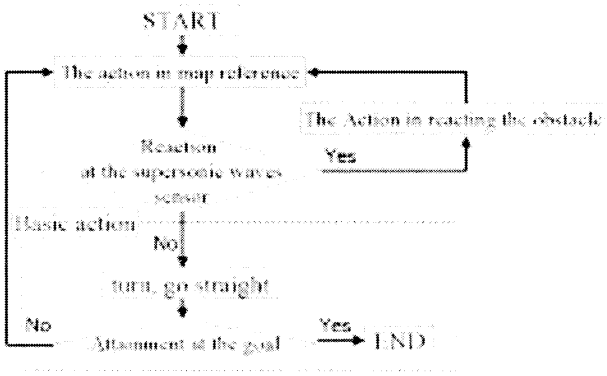


Fig. 3 Map of a finite space



4.4 Action in map reference

In map reference, the robot calculates the shortest route just after it has begun to act. At this time, if the robot detects an obstacle, then it begins instead to decide how to reach the goal while avoiding the obstacle. It then calculates its path as follows. First, the robot calculates the angle from itself to each of the four corners of the room, and calculates the maximum and minimum angles. Next, the robot compares an angle to the maximum, to the minimum and to the goal. Then the robot moves toward the corner whose difference is small. At that time, the robot sets up a middle goal as good as the Action in reacting to the obstacle, then the robot evades by pursuing that path.

5. Processing for spatial recognition

5.1 Ensuring straight movement

When the robot moves, the positions in the trial map need to be synchronized with its real position. Therefore, we must consider a case in which this synchronicity shifts because of a subtle unevenness in the traveling surface and the resulting slight difference in the rotation of the right and left driving wheels. In processing movement while referring to the map, then, the robot must cope with gaps that accumulate from this difference. In this case, the robot's processing revises errors in the movement route with visual systems that recognize spatial divisions. The robot uses information known as feature point data, which it gains from the optical flow of a picture series.

5.2 Revision of a movement route through use of the visual system

While the robot moves, it gets a picture whenever it moves a certain distance, generating a feature point flow by processing the data obtained from the pictures. To derive feature points, the robot does edge processing of the pictures, changes into two values and then performs a Hough transform. In this way, the robot sets up a random feature point base on the points of intersection of the straight element of an extracted picture. Then, data to revise the robot's path are calculated from the predicted values of the feature point flow of initial data from the time series picture and the measured values actually obtained. A prediction track of a feature point flow and a measurement plot to go straight and the root error occurring are shown in Figure 5.

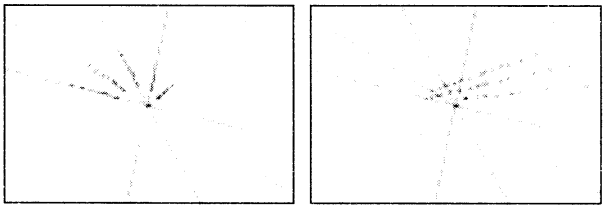


Fig. 5 Characteristic point flow

6. Conclusions

In this study, an evaluation of the self-control drive processing of a personal robot was completed, and the following results were obtained. The robot was able to avoid obstacles in two dimensions, and a processing system that allows the robot to reach its goals was developed.

Also, if there are no obstacles in simulation, then the robot calculates its path and moves to the goal. If there is an obstacle, on the other hand, then the robot detects and reacts to the obstacle, either avoiding it altogether or figuring out a way to get around the obstacle. In visual processing for spatial recognition, gentian of the feature point flow, using a visual system and revision processing of a movement route by comparing predicted values against measured values, was developed. In the future, we will refine the robot's understanding of an object through the use of its visual system, and mapping and processing systems are planned to be developed into three levels.

References

[1]Funakubo, Process and recognition of the vision paturn, Keigakusyuppan, 1990.

[2]Shinoda, Self-driven robot on the grand, Journal of the Robotics Society of Japan, Vol. 18, No.7, pp.928-932. 2000

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Path Planning for the Autonomous Mobile Robot Under the Constraints of the Driving Condition with Unknown Obstacles

Young-Jin Lee[1], Young-Jin Yoon[2], and Man-Hyung Lee[3]

[1]Inst. of Computer and Information-Telecomm., Pusan National Univ., Pusan, Korea, email:yjlee4@pnu.edu
[2]Dep't of Mechanical and Intelligent Systems Eng., Pusan National Univ., Pusan, Korea, email:yoonyj@pnu.edu
[3]School of Mechanical Eng., Pusan National Univ., Pusan, Korea, email:mahlee@pnu.edu

Abstract **Path planning method for the autonomous mobile robot is considered. For the practical applications, the simplified local potential field methods are applied under the constraints of the driving condition. To improve the performance, the fuzzy-approximated linear function method is also used.**

I. INTRODUCTION

Path planning for the autonomous mobile robot that is under constraints of the driving is introduced. When we define the goal of the the path planning as 'approach to the target point without any collision with obstacles,' the most common method to accomplish the goal is applying global artificial potential field method shown in [1]

As the global artificial potential field method depend on the informations of the driving environment such as the location of the fixed obstacles to evaluate the gradients between the target point and the obstacles, it is needed for the mobile robot to know the driving conditions before it starts operation. To solve this demerit, one can apply the local artificial potential field method that makes the informations of the driving environment by the internal sensors of the mobile robot. While the algorithms can avoid the unknown obstacles because the environmental informations are gathered during the driving of the robot, it is also true that the navigation time can be longer that the global version of the algorithms.

For the practical and actual applications like the goods-carrying mobile robot in very large scale warehouse or the smal size container transporter robot in the harbor, it is assumed that the mobile robot in this paper is 'blind' or it does not know the exact position of the obstacles when it starts driving; rather, the mobile robot avoids the obstacles by the informations of the internal sensors equipped on the robot that measures the distance from the obstacle to the robot. During the carrying or transporting operation, it may possible for the mobile robot to face with unexpected goods or containers that are in the driving course of the robot. In this case, one can use the local potential field method to avoid the situation.

It is noted that following conditions should be satisfied when the mobile robot is applied in the industrial tasks:

Fisrt, there can be an area that the mobile robot should not pass for the reason of the worker's safety or the spatial constraints due to the allocation of the machinery or the structures. The mobile robot never go through this 'tabooed' area.

Second, it is needed for the mobile robot to conduct effective avoidance maneuvering when unknown obstacles are detected during its driving.

To meet the first condition, we quote the the principals of the driving law of the line tracer. That is, if the mobile robot can track the driving guide line that can offer sufficient informations for the avoiding tabooed area, the problem can be solved. Second condition is considered under the sensor informations and pre-defined driving rules of the mobile robot. It is emphasized that we do not describe all conditions of the encounting the obstacles; just simple and a few prime rules are enough to drive. This concept of describing not all rules but just essential rules is the basic principal of the artificial life[2]. For successful maneuvering the simplified local potential field method with prime driving rules are used.

While applying the simplified local potential field method, if the avoiding action that evaluate the turning angle is determmined by the distance from the obstacle, then it can be regarded as SISO system. Also, it is noted that we can use the fuzzy logic bacause the distance information can be written as the linguistic expressions such as 'far', 'near', 'almost contact', etc.

Using the property of the universial approximator of the fuzzy logic[3], we can get the linear first order function that roles same rule of the fuzzy logic. This 'fuzzy-approximated lienar function' is useful because it skips most of the traditional fuzzy operatioins like fuzzification, inferences, and defuzzification while it shows almost same effect of the fuzzy logic.

As a preliminary stage of developing the actual robot, this paper considers the path planning for the line-tracable autonomous mobile robot, and verifies the suggested methods by the simulations.

This paper consists of five sections. In section II, the configurations of the mobile robot and the coordinate systems, and the simple path planning laws are introdueced with the results of the simulation with the simplified local potential method. The enhancement of the planning performance with the fuzzy-approximated linear function is shown in section III by the results of the simulation. Final conclusions and further studies are in section IV.

II. PATH PLANNIG WITH SIMPLE DRIVING LAWS

A. Configuration of the autonomous mobile robot and coordinate systems

Fig. 1 shows the brief informations about the configuration

of the mobile robot and the coordinate systems used in this paper. X-Y coordinate system is a world coordinate which expresses the target point and has the origin point or the start point of the robot. U-V coordinate system is a relative coordinate system that is equipped at the center-of-gravity point of the mobile robot.

The axis U of U-V coordinate system is always identical direction with the advancing direction of the mobile robot, and axis V is orthogonal to the axis U. As U-V coordinate system is set as the relative coordinate system of the mobile robot to the target point, the C.O.G point of the robot on the U-V coordinate system is (0, 0). The ideal case is that the target point is located on (m, 0) with U-V coordinate system, and the mobile robot go forward to decrease the distance m.

The configuration of the mobile robot is assumed like Table 1. For the simplicity, we just consider two motion of the mobile robot: go advance or rotate. The reason of just considering two motion is to make the driving rules as simple as possible.

B. Simple driving laws for the avoidance of obstacles and following the driving-guide line

The mobile robot gets the informations about the obstacles by the five sensors notated as 'sensor 0'~'sensor 4' in Fig. 1.

Table 1. Assumed configurations of the robot

Actuators
two wheeled mobile robot. the wheel is rotated by the motor which can be controlled with the unit of the degree. additional mechanical part like ball-caster can be used for the driving balance. (recommended: stepping motor)
Sensors for detecting obstacles
five sensors are equipped to detect the obstacle and measure the distance from the robot to the distance. sensing range or coverage is complied with the specifications of the sensor. (recommended: infrared sensor or ultrasonic sensor)
Motions
allowed motions of the robot are 'go forward' and 'rotate.' for the simplicity of the driving rules, another motions like 'go backward' is not considered in the paper.
Sensors for line tracing
eight sensors are equipped bottom side of the robot to trace the driving guide line. the gap between sensors should be same and the width of the gap is equal to the width of the driving guide line. if the robot is located on the line exactly, the sensored value is binary 00011000 or 0x18. (recommended: infrared sensor)

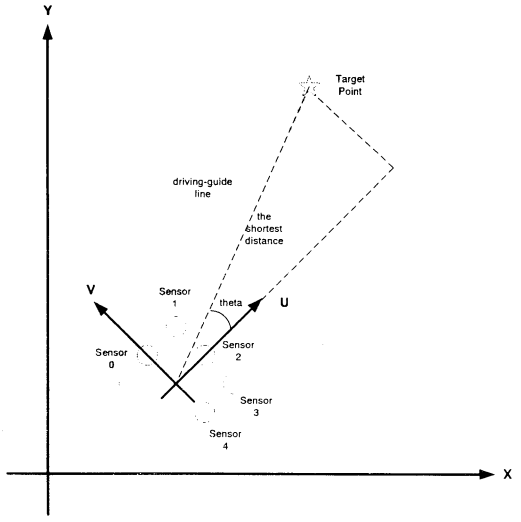


Fig. 1 . the brief configuration of the robot and the coordinate systems

Internally, sensored informations are replaced with 5 digit binary data. That is, sensor 0 is MSB and sensor 4 is LSB.

For example, if sensor 0 and sensor 1 detect the obstacle in the left of the robot, the detecting information is binary 11000 or 0x18.

Although five digit of binary can express 2^5=32 cases, we just describe common and major case of the obstacle detection. The considered case of the detection follows Table 2. If the sensored value are not listed in the Table 2, the most similar value is selected. That is, if the value is binary 11110, then the value is replaced with the binary 11100.

The tracking of the driving guide line is operated under the similar configuration of the obstacle detection. If the driving guide line is on the axis U of U-V coordiate system, and the mobile robot need not turn to calibrate deviated angle, then the data made by the eight sensors is binary 00011000 or 0x18. In this paper, angle estimation for compensation of the deliated angle is proportional to the sensored value. That is, 8 bit sensored data is separated by bit 7~bit 4(assigned for the detection of the left side of the line) and bit 3 ~ bit 0(assigned for the rigt line), and last four bits need bit-inversion to relate with the input of the motor directly.

Table 2. Sensor informations under considerations

No.	*Sensored value*	*Description*
1	11000	Left-near
2	01100	Left-front
3	00110	Right-eront
4	00011	Right-near
5	10000	Left obstacle following
6	00001	Right obstacle following
7	01000	Left-just
8	00100	Front-just
9	00010	Right-just
10	00000	No-obstacle
11	01110	Front-lumped
12	11100	Left-lumped
13	00111	Right-lumped

Then the input voltage for each right or left motor of the

robot is set as separated four bit value. For example, if the C.O.G of the robot is on the left edge of the driving guide line and the sensored value is binary 00001100, then the input for the right motor is 0x00(bit 7 ~ bit 4 of the sensored data, that is 0000) and left motor is 0x03(inversed value of bit 3 ~ bit 0 of the sensored data, that is 0011).

The driving of the mobile robot consist of two motion; going forward and rotating. In this paper, the going velocity of the robot is assumed to be a constant; that is, distance by going forward of one step is fixed. When the obstacle is detected, the major rule of the robot is rotating until no obstacle is detected on the axis U of the U-V coordinate system. As a result, the rotating and evaluating proper rotating angle is important in this paper. Therefore, dominant motion for the driving or avoiding the obstacles is rotating.

The logic for the tracking the driving guide line when the obstacle is detected is as follows. Basically, the major role of the mobile robot is following the driving guide line, and no avoidance maneuvering is needed when the robot does not face the obstacle. If the obstalce is detected, the avoiding maneuvering is higher priority than any other operations. Then tracking and following the driving guide line is continued when the robot finds the line during the avoidance maneuvering. That is, if the obstacle is located on the driving guide line, the robot can avoid the obstacle and find the line by following the surface of the obstacle that is accomplished by the sensor data of sensor 0(for the following of the left obstacle) or sensor 4(for right obstacle).

The evaluation of the turning angle for the rotating is simplified like following rule 1 ~ rule 3.

rule 1. If the obstacle is detected on the left or right, the turning angle is LOW.

rule 2. If the obstacle is detected in front of the robot, the turning angle is HIGH.

rule 3. If no sensor signal is found during tracking the surface of the ostacle, that is, if the robot is reached at the edge of the obstacle, then tunring angle is MID.

C. *Simplified artificial local potential field method*

Unlike artificial local potential fiedl method, the simplified version used in this paper has following difference: the simplified method only evaluates if the obstacle is detected or not. The original version gets the repulsive potential continously, but the simplified version gets just 1 or 0. It is expressed in Eq. (1)

$$U_{REP} = \begin{cases} 1 & \text{if } L \le L_D \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where L is distance from the robot to the obstacle, and L_D is the range of the sensor or the maximum sensible distance between the robot and the obstacle. The idea of the Eq. (1) is heaviliy depend on the Bang-Bang control technique or On-Off control method. It is also noted that the concept of the potential field is modified as the robot-obstacle distance in this paper.

D. *Simulations*

The proposed method shown in II-A ~ II-C is verified with the robot simulator program. According to Fig. 2 ~ Fig. 5, it is known that the simple path planning method meets for given test cases.

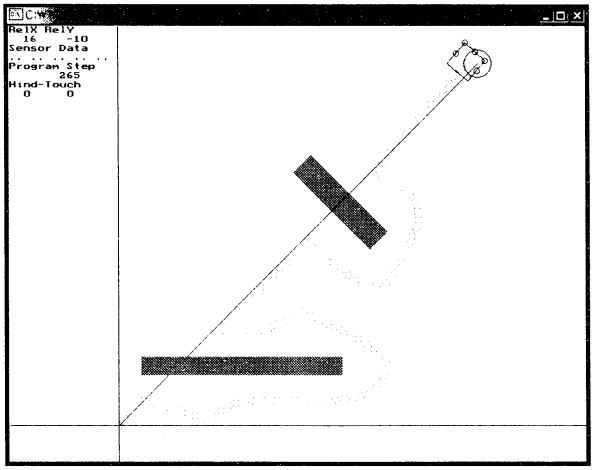


Fig. 2 . CASE-I : initial driving

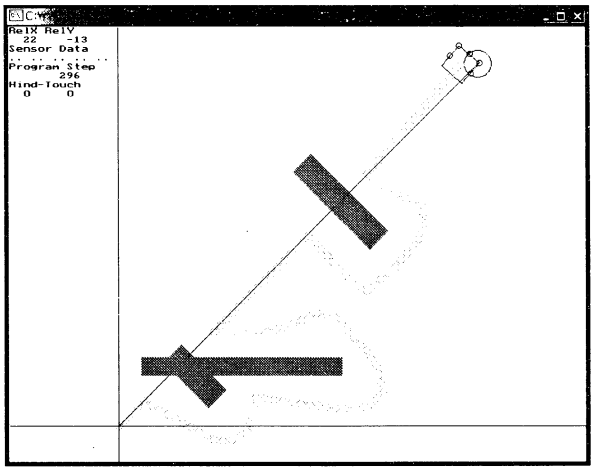


Fig. 3 . CASE-II : obstacle added at the beginning stage

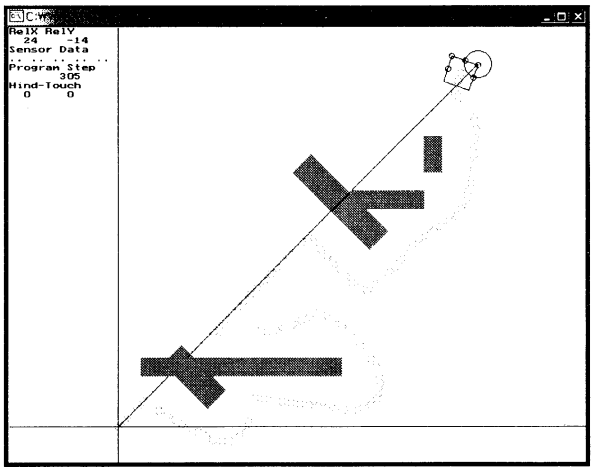


Fig. 4 . CASE-III : obstacle added at the ending stage

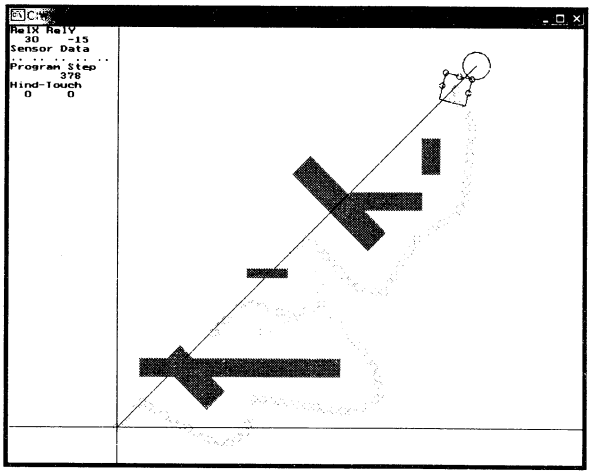


Fig. 5 . CASE-IV : obstacle added at the emiddle stage

III. PATH PLANNING WITH FUZZY LOGIC

A. *Modification of the evaluating turning angle*

As mentioned in the Section III, we can accomplish the purpose of the tracking the driving guide line and the avoiding the obstacles by simple driving rules, not so complex. The evaluating of the turning angle is merely depend on the existence of the obstacle and the direction where the obstalce is located. If the consideration about the distance from the mobile robot and the obstacle is included for evaluating the angle, we could expect enhanced performances such as the increasement of the accuracy, reduced total ellapsing time to go to the target point, etc. Brief description on evaluating the angle is as follows: if the distance is far then the angle may small; or if the obstacle is near, the angle should be large to perform the rapid maeuvering.

B. *Applying fuzzy logic for SISO system*

We relate the evaluation of the turning angle only with the distance to the obstacle. This can be regarded as the SISO system, which takes input as the distance and output as the turning angle. Also, the experiences that is inevitably occur during the consideration with trial-and-error way for the various cases of the distance and the angle, about the system input and output may be helpful for evaluating of the angle.

If we describe the experiences with the fashion of the fuzzy logic, we get following Table 3 for the fuzzy rule base with ten fuzzy variables notated in the table.

We configure all fuzzy membership functions to have the same support, like Fig.6 and Fig. 7.

Table 3. Fuzzy rule base

Distance	*Turning angle*
So highly near	A little
Highly near	Somewhat
Somewhat near	Much
Almost near	So much
near	Very much

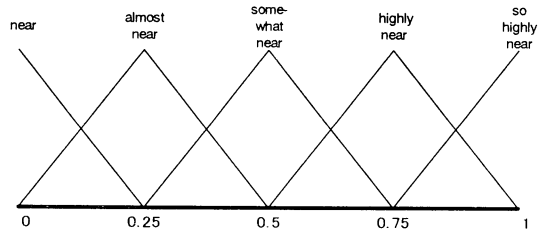


Fig. 6. Fuzzy membership function for the input of distance

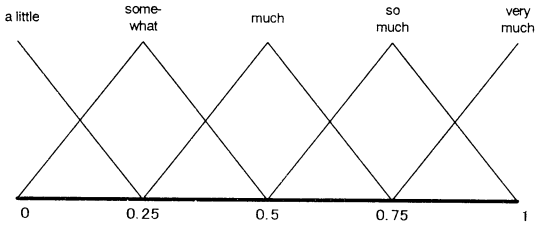


Fig. 7. Fuzzy membership function for the input of turning angle

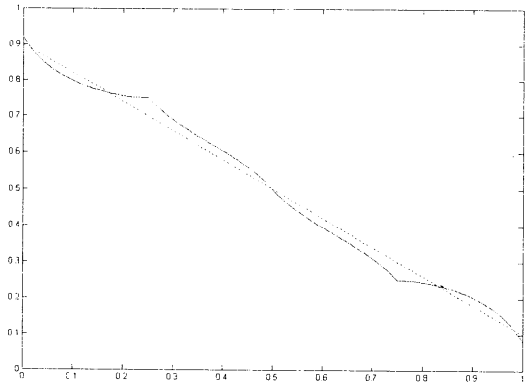


Fig. 8. Fuzzy phase(solid line) and fuzzy-approximated linear function (dashed line)

Fig. 8 shows the fuzzy phase or the relationship of the input-output fuzzy variables. It is noted that within error range, we can approximate the fuzzy pahse as the linear first order function like $y=ax+b$. It is also noted that this approximation procedure is based on one of properties of the fuzzy logic, 'universal approximator.'

The benefit of the fuzzy-approximated linear function is that we can skip most of fuzzy logic operations such as fuzzification, inference, and defuzzification. The demerit is that the approximated can not simulate the fuzzy logic completely, which is solved with loosen error margin or applying another type of function such as tangent function or inverse exponetial function.

C. *Simulations*

Like same procedures for the simulations on the section II, same simulation test case with fuzzy-approximated linear function is performed. The simulation results, Fig. 9 ~ Fig. 12, shows almost same results compared to the Fig. 2 ~ Fig. 5. Table 4 shows the performance enhancements on ellapsing time.

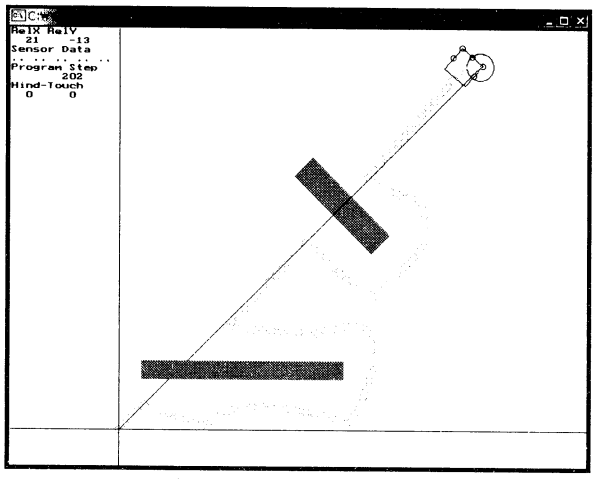


Fig. 9 . CASE-I : initial driving

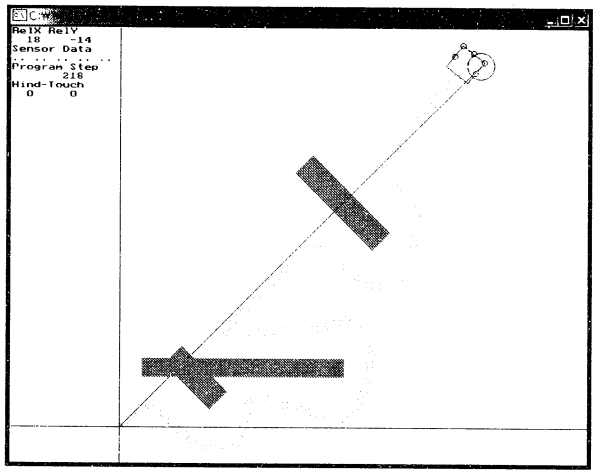


Fig. 10. CASE-II : obstacle added at the beginning stage

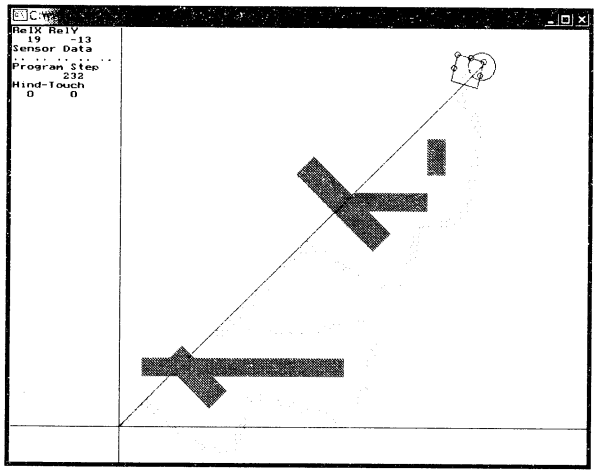


Fig. 11. CASE-III : obstacle added at the ending stage

The table contains 'going advance' step and 'rotating' step for each methods suggested in the section II and section III. As the step is linearly proportional to the time, we can guess that total ellapsing time for each cases. Compared to the method of section II, the method in section III has a little improvement about 1.72% ~ 7.87% in 'go advance' count, and relatively much enhancement about 52.42% ~ 83.71% in 'rotating' count.

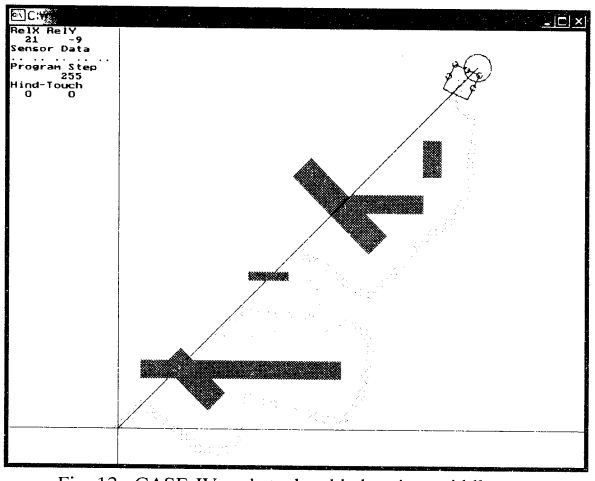


Fig. 12 . CASE-IV : obstacle added at the emiddle stage

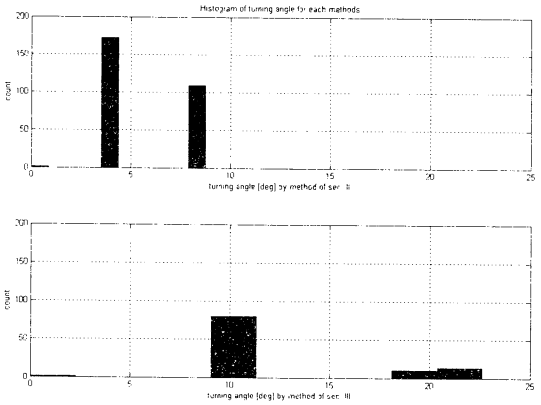


Fig. 13. Histogram of turning angle for each methods

Table 4. Performance enhancements

	Case I	Case II	Case III	Case IV
Go Adv. step by Sec. II	113	120	116	127
Go Adv. step by Sec. III	108	115	114	117
Enhancement	4.42%	4.17%	1.72%	7.87%
Rot. step by Sec. II	517	688.75	754.75	1993
Rot. step by Sec. III	246	248	243.5	324.75
Enhancement	52.42%	63.99%	67.74%	83.71%

The reason of the enhancement is that various rotating angles are evaluated by the method of section III. Fig.13 shows the histogram of the rotating angle for the CASE-IV with each methods.

IV. Conclusions and Further Studies

We considered the path planning of the autonomous mobile robot by simplified artificial potential field method with the minimum number of driving rules, also by enhanced method using fuzzy-approximated linear function. To accomplish the improvement of the simplified method which meets the purpose of the study, we applied the

fuzzy-approximated linear function to the relationship of the distance from the robot and the obstacle and the turning angle, and verified the performance enhancements.

Although the proposed algorithms are simple and easy to implement, unresolvable local minima phenomenon can be occurred. When a geometric configurations of the robot and the obstacles are formed and no suitable driving rule is found, the robot can not go and sometimes oscillate to escape the trapped situation, but finally fail to overcome. If the blind robot has to escape the local minimum, one solution is gathering more informations about its working environment by extending the sensing range. On the other hand, one can reduce the sensing coverage in despite of the possibility of the collision.

Most effective is apoting 'go backward' and extending the driving rules, but this case may increase the complexity on describing the robot rule. The problem and applying practical solution are remaind as further studies.

V. REFERENCES

[1] O. Khatib, "Realtime obstacle avoidance for manipulators and mobile robots," *Int'l Journal of Robotics Research*, 1986

[2] S. Levy, *ARTIFICIAL LIFE*, Pantheon Books, 1992.

[3] L. X. Wang, Adaptive fuzzy systems and control, PTR Prentice Hall, 1994

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Run Control of the Mobile Recognition Vehicle by Information of Internal Sensor and Vision

Masanori SUGISAKA and Shohei KURIYAMA
Department of Electric and Electronics Engineering, Faculty of Engineering, Oita University
700 Dannoharu, Oita 870-1192, JAPAN
Tel : +81-97-554-7831 Fax : +81-97-554-7818
E-mail : msugi@cc.oita-u.ac.jp s0932037@mail.cc.oita-u.ac.jp

Abstract

For the robot to do the autonomous run, it's absolutely essential that the robot recognize environment and position. In this paper, we present the run control of the robot using information of internal sensor and vision. The robot can recognize position by the using the dead reckoning, but in that case substantial of error occurs. And so, we think that the robot is able to run by compensating the error using information from vision in the way hereinafter prescribed.

KeyWords: mobile recognition vehicle, dead reckoning, tracking control, hough transform, internal sensor

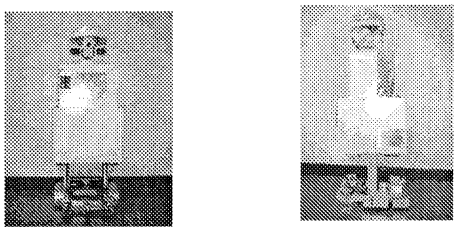
1. Introduction

Recently, various robots are created and deal with our life deeply. These robots develop not only use area of industry but also use area of non-manufacturing. In this circumstance, it is so much more increase that public requires robot's function. Our laboratory created, mobile recognized vehicle and research to development for robot can correspond public requirement. We notice the area of nursing and caring mainly. Our country become arrival of a graying society at present, therefore, we need to secure manpower relation to the area of nursing and caring. But as it is, manpower that relation to these areas is shortage owing to low birth late and diverse reasons. And so we think available for solving this problem if we can create the robot effective with the area of nursing and caring. In order to correspond to the above demand, we think that it is necessary for the robot to have function of autonomous run. If the robot can be autonomous run, it is increase robot's availability because the efficiency of the robot works and the profitability are better. To recognize own position, to recognize environment (ex. use CCD camera or senor), to decision situation, we think that it is necessary for the robot to put in for itself above three points. Dead reckoning is technique that the robot can recognize own position by internal sensor (ex. Encoder etc.) To use this technique is very convenient means. But the technique have problem that the error which occur due to slip of tire, floor levelness increase cumulatively. We made an attempt to compensate for the error which was based on dead reckoning principle.

2. Mobile recognition vehicle system

In this chapter, we explain the mobile recognition vehicle that is prepared in collaboration with DENKEN engineering. This vehicle is 130cm high, 45cm wide, 50cm depth. It has two drive wheels from side to side, two castors backward and forward. But we leave these castors out of consideration in modeling this vehicle because these castors have little impact on run control of the vehicle. This vehicle can drive by DC motor on right and left. It has two CCD cameras in the head. In this research, we use left CCD camera. The error from center position to camera is 8 cm. Here, we convent the error into image, the error is 30 pixels from image center in minimum distance if we use image size of 160×120 pixels.

The vehicle's encoder are 80 pulses per resolution and sampling time is 100ms.



(a) Front (b) Side

Fig.1 mobile recognition vehicle system

3. Self position estimation

3.1 Vehicle model

In this chapter, we present our vehicle's model. We define center of right-and-left driving wheel as center of vehicle body. We account this position (x, y) and angle from vehicle to X axis is θ. The vehicle condition is represented as three generalized coordinates which are two positions and one stance. The angle of rotation of the vehicle is not comprised in generalized coordinates because we consider that the angle of rotation of the vehicle is manipulated variable. Here,

$$v=\frac{1}{2}(v_R+v_L) \cdots (1), \omega=\frac{1}{2d}(v_R-v_L) \cdots (2)$$

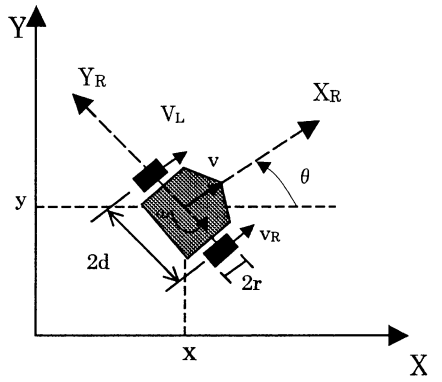
v is translation speed of the center of vehicle, ω is angular late. The relation between derivative times of generalized coordinates and translation speed · angular late are calculated by following equation(3).

$$\dot{x}=v\cos\theta \quad \dot{y}=v\sin\theta \quad \dot{\theta}=\omega \qquad \cdots (3)$$

$$\dot{x}\sin\theta-\dot{y}\cos\theta=0 \qquad \cdots (4)$$

This model is non- holonomic by formula (4).
On the assumption that the actuators are speed controlled, control input define as (v, ω) or ($\dot{\phi}_R$, $\dot{\phi}_L$).

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix}=\begin{pmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}v\\ \omega\end{pmatrix}=\frac{r}{2}\begin{pmatrix}\cos\theta & \cos\theta\\ \sin\theta & \sin\theta\\ \frac{1}{d} & -\frac{1}{d}\end{pmatrix}\begin{pmatrix}\dot{\phi}_R\\ \dot{\phi}_L\end{pmatrix} \cdots (5)$$



d: Distance from center of vehicle to drive wheel
r: Wheel Radius
V_L, V_R: touchdown speed of drive wheel
ϕ_R, ϕ_L: the angle of rotation of drive wheel

Fig.2 mobile recognition vehicle model

The formula (5) represents relation between control input and generalized coordinates. The minimum rotation radius ρ is calculated using the following equation (6).

$$\rho=\frac{v}{\omega}=d\frac{v_R+v_L}{v_R-v_L} \qquad \cdots (6)$$

This model has the advantage that vehicle is able to make small adjusting, therefore, it is easily to plan track and control using track. But it has bad point to the extent of direct advance.

3.2 Track control

In this chapter, we think the technique for controlling drive wheel. In this research, we use the deviation model which is suggested by Kanayama. The following equation represents above deviation model.

$$e_x=(x_r-x)\cos\theta+(y_r-y)\sin\theta \qquad \cdots (7)$$

$$e_y=-(x_r-x)\sin\theta+(y_r-y)\cos\theta \qquad \cdots (8)$$

$$e_\theta = \theta_r - \theta \quad \cdots (9)$$

e_x, e_y, e_θ: Deviation between target track and real robot position

x_r, y_r: Follow the track at time distance

Translation speed and angular late are modified using equations (10) and (11).

$$v = v_r \cos e_\theta + K_x e_x \quad \cdots (10)$$

$$\omega = \omega_r + v_r (K_y e_y + K_\theta \sin e_\theta) \quad \cdots (11)$$

K_x, K_y, K_θ: feedback gain

3.3 Self position estimation by the vehicle

Our vehicle is not able to recognize own position in specified time by rotation angle of drive wheel momentary. Now therefore, it is necessary to calculate of the sensor counts in chronological order. In this research, we use dead reckoning as technique for recognition own position in its own. The equation (12) ~ (14) represent relational expression of dead reckoning.

$$x_2 = x_1 + \frac{r}{2}(\Delta\phi_R + \Delta\phi_L)\frac{\sin\theta_2 - \sin\theta_1}{\theta_2 - \theta_1} \quad \cdots (12)$$

$$y_2 = y_1 - \frac{r}{2}(\Delta\phi_R + \Delta\phi_L)\frac{\cos\theta_2 - \cos\theta_1}{\theta_2 - \theta_1} \quad \cdots (13)$$

$$\theta_2 = \theta_1 + \frac{r}{2d}(\Delta\phi_R - \Delta\phi_L) \quad \cdots (14)$$

Dead reckoning is very easy for estimating own position, but the error steam from various factors. Therefore it is necessary to compensate the error.

4. Image processing

In this chapter, we used image processing methodology. In this go round, we used image information by camera to compensate the error come of dead reckoning. We set boundary line between the wall and floor up as landmark for compensating the error. At the start, the vehicle captured color image using CCD camera. Next, converting into gray scale image, the image was processed by binarization and edge detection. At last, we attempted to extract boundary line using θ-ρ Hough Transform.

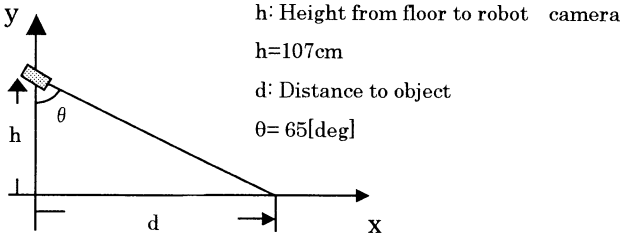


Fig.3 Distance from the robot to an object on the flat floor

5. Experimental method

In this chapter, we represent experimental method in this research. We placed a restraint on the area of move for vehicle. First, the vehicle can move only flat area indoors, because we envisioned the vehicle for using in hospital. Second, target track was showed in advance. Next, initial position and stance was provided in advance. We conducted the following experiments after being based on the above condition this time. First to experiment, tracking in terms of straight line. Next, the run experiment to the path specified beforehand. The floor map is shown in Fig.4.

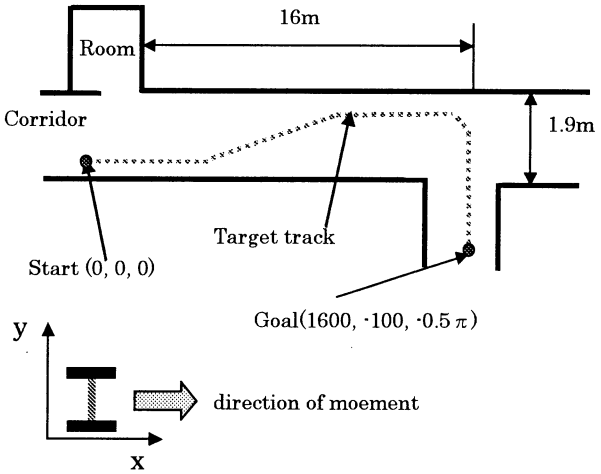


Fig.4 Move area of movile vehicle

Finally, we attempted compensation error which is based on the dead reckoning using vision information.

6. Experimental result

(1) Tracking in terms of straight line (wheels are floated)

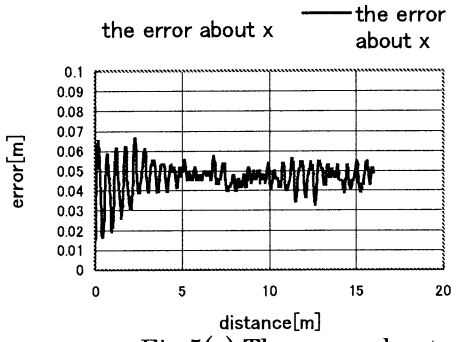


Fig.5(a) The error about x

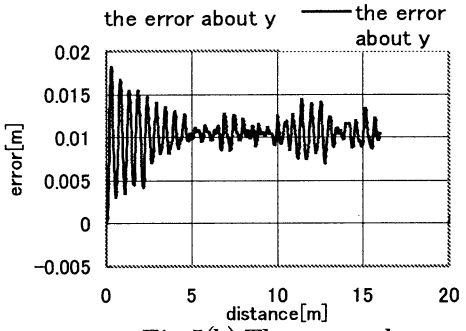


Fig.5(b) The error about y

(2) The run experiment to the path specified beforehand (wheels are floated)

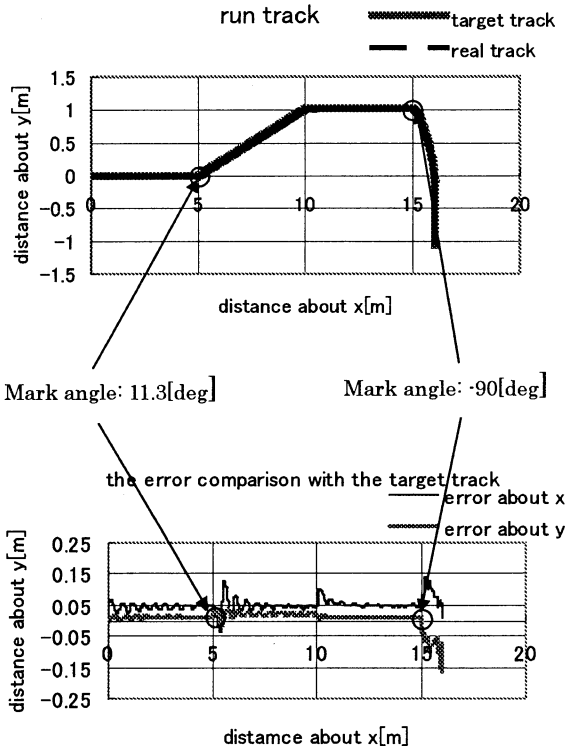
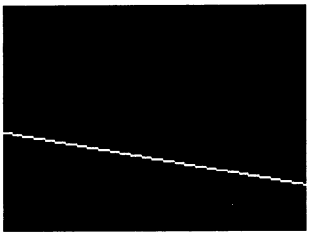


Fig.6 Run track and error

(3) To compensate for the error using vision information

In this research, we configured boundary line between floor and wall up as landmark. The vehicle is made to recognize by assigning a position beforehand.

$$d = h\tan(\theta_m + \frac{\theta_v y}{v}) \quad \cdots(15)$$



d: the distance from camera to object along the floor
h; vertical image resolution
v: vertical resolution
y; the pixel offset of he object in the image, from the image center

Fig.7 Result of image processing

7. Conclusion

In this paper, we described the method relevant to autonomous run by our mobile recognition vehicle. In this state where wheels were floated, the vehicle moved along target orbit scarcely accurate. However, the method has problems. The error occurred when target angle was large. It is necessary to consider the provision in the near future. And so, in terms of compensation the error using vision information, we only suggested the method, the vehicle couldn't move in this method. We think to clear up the questionable points on the basis verifying immediately to this method.

Reference

[1]F.Miyazaki,Y.Masutani,A.Nishikawa, "Robotics Manual", P87-107, Kyoritu Publishing (2000)

[2]T.Mita, "Introduction to Nonlinear Control Theory-Skill Control of underactuated Robots-", P145-P146, Shokodo (2000)

[3] T.Maeyama, M.Sugisaka, "Obstacle Avoidance in the Mobile Robot Using the CCD Camera", PROC. of AROB 7th, P238-p240, (2002)

[4]Cathy-Ann Radix,M.Sugisaka, "Anatomy & Control of a Research Robot", PROC. of SICE-Kyushu (2000)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Planning mobile robot with single visual aid

Xin Wang* Masanori Sugisaka**

*Department of Electronic, Oita Institute of Technology, Nakatsu 871-0006, Japan
**Department of Electrical and Electronic Engineering, Oita University, Oita 870-1192 Japan
(E-mail: wang@oita-it.ac.jp*; msugi@cc.oita-u.ac.jp**)

Keywords: Planning, Recognition, Neural Network

Abstract

An approach for planning a mobile robot is presented in this paper. This intelligent system pay attention to planning a mobile robot in the rigid marked object/obstacle environment. The robot is equipped with a single CCD camera aided sensor. Moment invariants computation is used to analyze shape attributes. A hardware artificial neural network, Neurocomputer, play the role of classifier. Fuzzy logic control is for system optimal control.

1 Introduction

The research of navigation of robot has lasted for several decades. This field is multidisciplinary and combines techniques from control theory, computer science, psychology, physiology and operations research. To realize more flexible control systems it is necessary to incorporate other elements, such as logic, reasoning and heuristics into the more algorithmic techniques provided by conventional control theory, and such systems. J. Laumond etc.[1] applied three-step algorithm to a mobile robot and solved the problem that how to compute optimal feasible paths in a configuration space (the space in which the motion of rigid bodies amid obstacles in the 3-D dimensional Euclidean space is translated into the motion of point in some space) equipped with the singular metric induced by the nonholonomic constraints. Y. M. Enab[2] built a controller for the unknown nonlinear time variant dynamic process based on modeling the behavior of its human operator. D. Gorinevsky etc.[3] designed a controller for an automated support system and performed desired features of such a system and parking controller. This research pay attention to planning problem with a single CCD camera aided sensor. The robot in our research is named Neural Mobile Robot (denoted by NMR).

The goal of this paper is to present an ideas from the practical point of view and feasible application. We discuss this problem in such environment: 1) the obstacle should not conceal the object at the any time; 2) either an object or an obstacle is not more than one at the same time.

The outline of the paper is as follows. In section 2, we describe the dynamic model of this kind of control problem. Section 3 presents the principle of the algorithm and the neural classifier. Sections 4 introduces a feasible image processing method, which is especially suit those simple visual sensors, for shape recognition by the moment invariants analysis. Finally, some experimental discussion and conclusions are given. Parts of this work have been presented in [4].

2 Dynamic model

A modeling mobile robot is illustrated in Figure 1. It is a simplified model of the constraints on the movement of a real car and takes into account of the nonholonomic constraint

$$\dot{x}_2 cos\theta - \dot{x}_1 sin\theta = 0 \tag{1}$$

that specifies the tangent direction along any feasible path for the mobile robot and a bound on the curvature of the path. Assume that the distance between both rear and front axles is d. From the driver's viewpoint, a car has two degrees of freedom: the accelerator and the steering wheels. We consider the midpoint s of the rear wheels as the reference point, denote by $\mathbf{v}$ the speed of the driving wheels, and by ϕ the angle between the front wheels and the main direction of the NMR. Consequently, the motion of the NMR can be described by the following model.

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos\theta \\ sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1/d \end{pmatrix} v \cdot tan\phi \tag{2}$$

(2) is subject to $\mid v \mid \leq v_{max}$, $\mid \phi \mid \leq \phi_{max}$. v is the magnitude of the speed $\mathbf{v}$. The body center coordinates x_1 and x_2 and the position θ constitute the system state vector $q = \{x_1, x_2, \theta\}$, the driving speed v and the steering angle ϕ are regarded as two control inputs. This system can be viewed as a nonlinear system where controls are v and ϕ. The constraint on the turning radius is expressed by $\mid \phi \mid \leq \phi_{max}$, ϕ_{max} is strictly positive real. Let us consider a reference trajectory of the robot motion:

$$q^r(t) = col\{x_1^r(t), x_2^r(t), \theta^r(t)\} \in D \tag{3}$$

here D is a domain of possible configurations of the NMR. Assume that the robot state vector $q^s(t)$=col$\{x_1^s(t)$,$x_2^s(t)$, $\theta^s(t)\}$ is not belong to the reference trajectory (3). Then the planning problem is equivalent to drive the NMR from an arbitrary state $q^s(t)$ into a desired state(location) $q^r(t)$ with the control inputs v and ϕ. We define the reference state vector as $R = (x_1^r, x_2^r, \theta^r)^T$ and the state vector as $T = (x_1^s, x_2^s, \theta^s)^T$. We want to realize the constraints as follows.

$$J = \|R - T\|_{[u(t),\phi(t)]} \mapsto minimum \tag{4}$$

here J is cost function of the system. The dynamic optimal model of the mobile robot are (2) and (4). Though

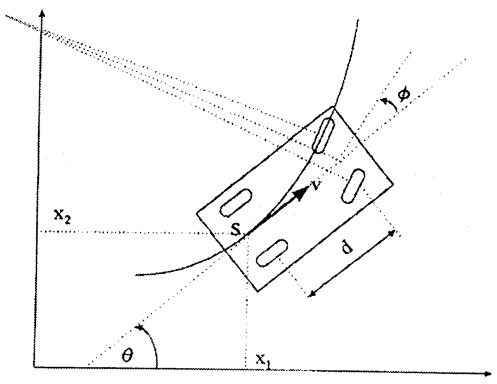


Figure 1: Dynamic model of the NMR

there are virious mathematical solutions about this planning problem [1][2][5]. We select human knowledge approach to make this problem easy and feasible. For the clarity of exposition, we concentrate forward running and assume $v>0$ during a whole control process.

3 Algorithm and neural classifier

Our system consists of three steps. (1) Moment invariants computation, (2) neural network recognition, (3) fuzzy control. This intelligent system drives the NMR from arbitrary states $q^s(t)$ to the desired state $q^r(t)$. The architecture is shown in Figure 2. Details about (3) have been discussed in [4]. The item (1) and (2) consist of following four steps.

1. Learning both of the shape arttributes of obstacles and objects;
2. Searching: the camera search spatial objects, if necessary rotating to expend its viewing range;
3. Recognizing environment: to decide whether there is an obstacle between the object and the mobile robot;
4. Planning $q^s(t)$ for $J|_{q^s(t)} < J_{max}$. J_{max} is the maximum J in 4.

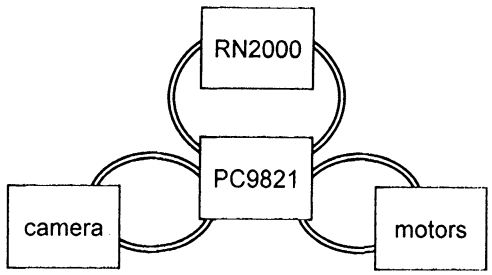


Figure 2: The control architecture, RN=RICOH

The train of the neural network or neural controller is based on human expert experience. The back propagation network (BPN) model changes the input-output problem of a set of sample into a nonlinear optimal map. The learning and remmbering are realized by weight update with recurrent computation in the inner neural network. Let N be samples $\{x_k,y_k\}$ (k=1, 2, $\cdots$, N). The BPN with m input neurons and n output neurons is the map $\Re$ from $\Psi = \{x_1, x_2, \cdots, x_m\}$ to $\Omega = \{\bar{y}_1, \bar{y}_2, \cdots \bar{y}_n\}$. Here, $\bar{y}_k(k = 1, 2, \cdots, n)$ is the real value of the output of the BNP. If $\Omega=G(\Psi)$, then $\Re$ is a mathematical approach of the map G.

$$\bar{y}_i=\Re(x_i) \qquad i=1,\ 2,\ \cdots,\ m;$$

Without lose of the generality, single output neuron y is discussed. The output of a neuron is denoted by O_i. To a certain input x_k and an output y_k, the output of a neuron i to a neuron j is O_{ik}, the activation functions $\{f(x_k)\}$ of all neurons in a BPN are sigmoid. Thus the input of neuron j is

$$net_{jk} = \sum_i \omega_{ij} \cdots O_{ik}$$

The cost function E is defined as

$$E = \frac{1}{2}\sum_{k=1}^{N}(y_k - \bar{y}_k)^2, \qquad \delta_{jk} = \frac{\partial E_k}{\partial net_{jk}}$$

1. When j is an output neuron, $O_{jk}=\bar{y}_k$.

$$\delta_{jk} = \frac{\partial E_k}{\partial \bar{y}_k} \cdot \frac{\partial \bar{y}_k}{\partial net_{jk}} = -(y_k - \bar{y}_k) \cdot f'(net_{jk}) \qquad (5)$$

2. When j is not an output neuron,

$$\delta_{jk} = f'(net_{jk}) \sum_m \delta_{mk} \cdot \omega_{mj} \qquad (6)$$

$$\frac{\partial E_k}{\partial \omega_{ij}} = \delta_{jk} \cdot O_{ik} \qquad (7)$$

The BPN algorithm of a neural network, which total layers are M and the Mth layer consist of the output neurons only and the first layer is the input layer, is described as follows:

1. The value of weights is initialized,
2. The following processes are repeated until the BPN convergence,
   (1). from $k=1$ to N
      a. forwards calculating O_{ik}, net_{jk}, y_k and,
      b. backwards calculating from the Mth layer to the 2nd layer;
   (2). To the neurons in a same layer, $\forall j \in M$, δ_{jk} is calculated by (5)-(7).
3. The value of weight is adjusted

$$W_{ij(new)} = W_{ij(old)} - \mu \frac{\partial E}{\partial W_{ij}} \qquad \mu > 0 \qquad (8)$$

here $W_{ij(new)}$ is the value of a weight from neuron i to neuron j after adjustment, $W_{ij(old)}$ is a one prior to adjustment, μ is leaning-rate coefficient, and

$$\frac{\partial E}{\partial W_{ij}} = \sum_{k=1}^{N} \frac{\partial E_k}{\partial W_{ij}}. \qquad (9)$$

The neural classifier is realized by a neurocomputer

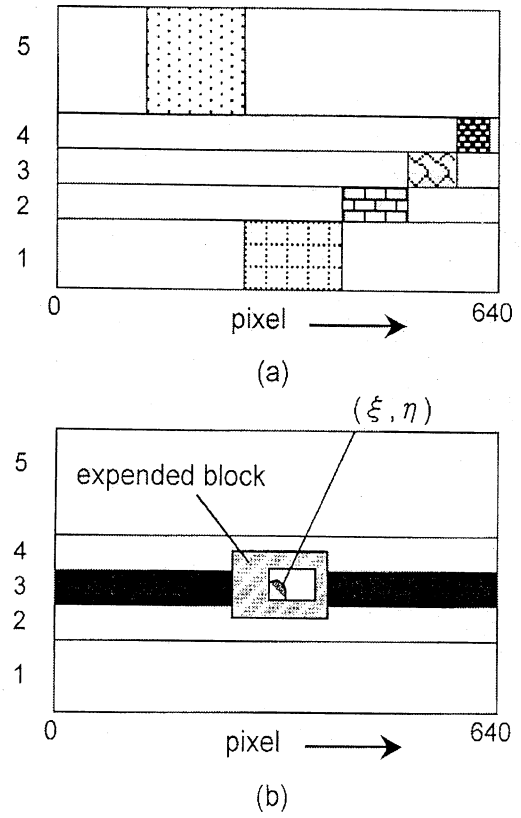


Figure 3: The five parts in vertical direction is marked by the numbers (down-up). (a) each sub-windows in every parts is shown by different textures, (b) for example, an object is caught in the black area, part 3.

called RICOH2000, which is based on the back propagation algorithm. The RICOH2000 consist of seven RN-200 digital neural network VLSI chips and sixteen neurons in each layer, and whole 256 synapses are integrated in a $13.73\times13.73mm^2$ VLSI chip. This chip can perform 5.12 giga pulse operations per second. It corresponds to effective neural computing rate of 40M CPS[4]. Let $\Psi=\{x_1(t),\cdots,x_m(t)\}\in R^m$ be a vector containing m measurements representing the process states at time index t, $\Omega=\{y_1(t),\ldots,y_n(t)\}\in R^n$ be the outputs which response the inputs.

The 16×16 two-value compressed image were used to form the moment invariants vector $\Psi = \{x_i(t)\}$ (i=1, ..., 7) as input vector of the neural network. The moment invariants computation has been applied into shapes recognition[6] and the same way is used in this research. The RICOH2000 execute the map $\Re$: $\Psi\mapsto\Omega$ ($m = 7$ and $n = 1$) to classify different objects that have their unique moment invariants vectors. A fuzzy controller is used in the system, but not our point in this paper.

4 Simplified image recognition

The NMR's single visual sensor, a CCD camera, produces well-known dib color images, which has 256 levels in each basic color. There are 640×400 pixels in the available viewing range of the camera. Five parts including 48 blocks in a 640×400 pixels (see Table 1 and Table 2), which is based on real distance and pixels distance,

Table 1: Distribution of block units

part	block width	block height	Δx_b	Δy_b
1	120	100	4	3
2	80	50	3	2
3	64	50	2	2
4	32	50	1	1
5	120	150	4	3

Table 2: Height of block units

part	block start y_1	block end y_2	$\Delta y_p=y_1-y_2$
1	300	400	100
2	250	300	50
3	200	250	50
4	150	200	50
5	1	150	150

are established (see Figure 3a). In these tables, Δx_b is searching pixel interval in horizontal x direction, Δy_b is searching pixel interval in verticai y direction. The objects are recognized according to their plane moment invariants and RGB attributes ($r_l \le r/s \le r_h, g_l \le g/s \le g_h, b_l \le b/s \le b_h, s = r+g+b$). r, g, b are image units and p_l and p_h ($p = r, g, b$) are their constraints. The parameters of one of six patterns in experimental circumstance are listed in Table 3. If a block contains part of an object, its center of gravity (ξ, η) of the object will be calculated in this block, and an expending block which center is (ξ, η) will be formed (see Figure 3b). In this expending block, compressed image for recognition using moment invariants computation are finished.

Table 3: Parameters of pixel units

color	r_l/s	r_h/s	g_l/s	g_h/s	b_l/s	b_h/s
red	0.41	0.56	0.20	0.30	0.22	0.31
white	0.28	0.32	0.33	0.40	0.28	0.34

The CCD camera is in the middle of the head and has total 60° viewing range[7]. During a process, firstly, the NMR search the marked object from part 1 to part 5 block by block, down-up and left-right in each part. Secondly, if definite color is found, the searching action will be paused to form a expending block which center is (ξ, η) (see Figure 3b). Finally, compressed image is obtained from the expending block which has 16×16 pixels. The moment invariants which are the inputs of the classifier ie. RICOH2000 were computed on the basis of the white/black value 16×16 pixel units. If the shape is identified as the designated object by the classifier, the mobile robot will use the image information to plan a near optimal path by fuzzy controller to track the object, else the NMR will continue to search. [8] shows our design on a fuzzy controller. The available viewing range of the CCD camera was expended to 140° with aid of its rotation in the horizontal and vertical directions. The image units (r_r, g_r, b_r) corresponding to 1.3m ahead of the NMR are limited. The pixel distance from (320, 400) in part 1(see Table 2) to the center of gravity in a expended block is calculated. The NMR will utilize a

pixel guide map[4] to get the real physical distance and plan a way to avoid an obstacle in order to arrive its object.

Remark: The searching blocks in part 5 are larger than others because those blocks are designed by considering the relationship between real physical distance and pixels distance. If an object is located in part 5, most possibility is that an object with some height is over/on the road. In the present stage, we also assumed any obstacle did not conceal the marked object.

5 Experiments and discussions

The NMR consists of a microcomputer NEC PC9821Ap, some I/O boards including a video interface SUPER CVI. The two rear wheels can rotate forwards and backwards about the axle shaft and are driven by a 24V DC motor using cross helical gears. Its 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 CCD camera in the horizontal and vertical directions. Three encoders detect the location of the front wheels and camera. Based on the machanical features, v, ϕ in (2) is subject to $\phi_{max} = \pi/5$, the velocity v_{max}=2.0m/s.

Figure 4a shows a, b, c, d, and e, five tracking trajectories without obstacle. The initialized statements are $q^s(0)$=(0.50, 2.50, -30°), (0.50, 2.00, 0°), (0.50, 1.50, 0°), (0.50, 1.25, 30°), (0.50, 0.50, 45°) independently. The aimed positions are the same $q^\tau(0)$=(4.00, 1.50, 0°). c is the simplest case. In the cases of a and e, the NMR adjusted its control inputs v and ϕ to decrease the position errors, and plan a near optimal path. Figure 4b give a trajectory including an obstacle (the mark with a cross inside). The initialized statement is $q^s(0)$=(0.50, 1.00, 0°), there is an obstacle located in (3.50, 1.00) and an object in $q^\tau(0)$=(6.00, 1.75, 0°). The dash line g is human plan and the dark line f is the NMR's result. The initialized statements represented the common case of tracking a designated object. The smooth motion was accomplished by installed NEC PC9801Ap series interruption control. The interruption inputs, IT0 and IT1 in I/O i5822, were used because of the structure of the mobile robot. The sampling time was 20ms. One of six patterns is used in this experiment.

In all experimetal environment, the marked object is necessary, and if it disappear from the available visual range, this robot will stop moving to rotate the camera to search disappeared target before making a lost/finished selection in two go/stop states.

6 Conclusions

An intelligent system is developed by application of moment invariants, neural network, and fuzzy control algorithms. The mobile robot is capable of tracking/avoiding marked obstacles and object. Moment invariants computation was used to recognized the shape of objects. This image processing strategy can also be adaptable to other cases of rigid object, for example navigating transport in plants, parking problem by the aid of landmarks, navigating cultivating machine in land etc. We are also interested in doing further works on developing techniques for high speed cars, for instance an electrical car which maximum speed can up to 60 km/hour.

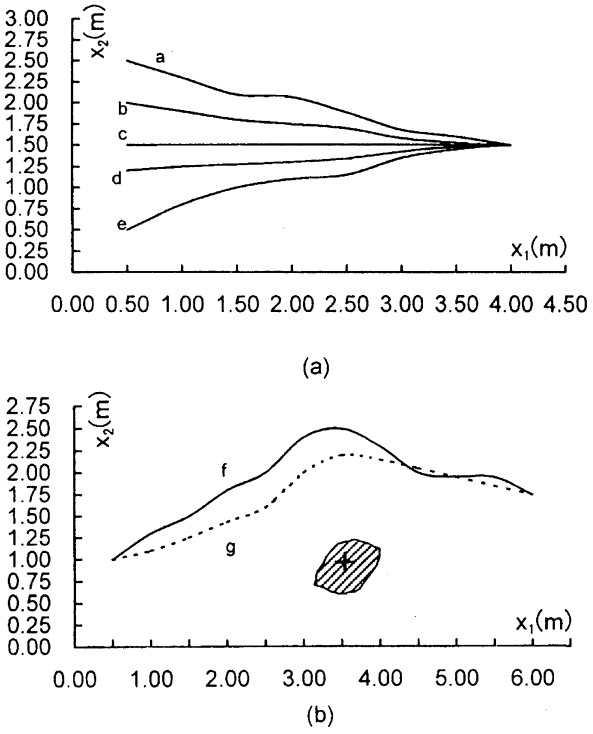


Figure 4: Some of experimental results

References

[1] Laumand, J. P, A motion planner for nonholomic mobile robots, *IEEE Trans. on Robotics and Automation*, 10(5), 577-593, (1994).

[2] Enab, Y. M., Intelligent controller design for the ship steering problem, *IEE Proc. Control Theory Appl.*, 143(1), 17-24, (1996).

[3] Gorinevsky, D., Kapitanovsky, A., and Goldengerg, A., Neural network architecture for trajectory generation and control of automated car parking, *IEEE Trans. on Control System Tech.*, 4(1), 50-56, (1996).

[4] Sugisaka M., Wang X, Lee J.J., Intelligent control strategy for smooth running a mobile vehicle with neurocomputer, *Proceedings of the Twelfth International Conference of System Engineering*, 9-11, (1997).

[5] Fleury, S., Primitives for smoothing mobile robot trajectories, *IEEE Trans. on Robotics and Automation*, 11(3), 441-448, (1995).

[6] Sugisaka, M., Teshnehlab, M., Fast pattern recognition by using moment invariants computation via artificial neural networks, *Control-Theory and Advanced Technology*, 9(4), 877-885, (1993).

[7] Sugisaka, M., Wang, X., Artificial brain for a mobile vehicle, *Proc. of Int. Symposium on Artificial Life and Robotics*, 234-237, (1998).

[8] Sugisaka, M., Wang, X., A complex control method for an intelligent mobile vehicle, *Proc. of the 1996 IEEE Intelligent Vehicles*, 53-57, (1996).

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Swarm Behavior of Multi-Agent System Based on Artificial Immune System

Kwee-Bo Sim and Dong-Wook Lee

School of Electrical and Electronic Engineering, Chung-Ang University, Korea
221, Huksuk-Dong Dongjak-Ku, Seoul 156-756, Korea
kbsim@cau.ac.kr, dwlee@ms.cau.ac.kr

Abstract

In this paper, we proposed a decision-making method of emergent and swarm behavior strategies based on immune system in multi-agent system (MAS). Immune system is living body's self-protection and self-maintenance system. These features can be applied to decision making of optimal swarm behavior in dynamically changing environment. For applying immune system to MAS, an agent is regarded as a B-cell, an environmental condition as an antigen, a behavior strategy as an antibody and control parameter as a T-cell respectively. When the environmental condition (antigen) changes, an agent selects an appropriate behavior strategy (antibody). And its behavior strategy is stimulated and suppressed by other agent using communication (immune network). Finally much stimulated strategy is adopted as a swarm behavior strategy. This control scheme is based on clonal selection and immune network hypothesis, and it is used for decision making of optimal swarm strategy. Adaptation ability of robot is enhanced by adding T-cell model as a control parameter in dynamic environments.

Keywords: Artificial Immune System, Multi-Agent System, Swarm Behavior, T-cell model

1. Introduction

The most significant features of multi-agent systems (MAS) are that each agent perceives its environments such as object and the other agent's behavior etc., and they determine their behaviors independently, and cooperate with the other agents in order to perform the given tasks very well [1~4]. MAS has no function to integrate the whole system. But the agent, which is a component of the system, individually understands the objective of the system, environment, behavior of other agents, etc., and decides its behavior autonomously to cooperate with other agent and to establish and maintain order of the whole system.

On the other hand, immune system has various functions that are ability to recognize foreign pathogens, ability to process information, ability to learn and memorize, ability to discriminate between self and non-self, and ability to keep up harmony of the whole system. Immune system is also distributed autonomous system that protects and maintains living body [5~6]. The components of immune system do not follow commands of the brain but cope with environment autonomously. In this point of view, we analogize MAS from immune system. So this analogy can be used for mechanism that decides group behavior strategy of MAS. It is thought that these functions of immune system can be applied to various engineering fields [7~12].

In this paper, immune system is applied to making action strategy in collective autonomous mobile agents with dynamic environmental changing.

2. Immune System Modeling

The protection system that eliminates foreign substances that invade living body is called immune system [8]. The basic component of the immune system is lymphocyte that occurs as two major types, B-cell (B lymphocyte) and T-cell (T lymphocyte).

Farmer et al. [13] proposed immune network equation based on Jerne's idiotopic network hypothesis [14]. In this paper, to improve the adaptation ability of the system, we propose the modified immune network equations that is added helper and suppressor T-cell model. Equations (1) to (3) show the modified immune network equations which are modelled based on relationship of antigens, B-cells (or antibodies), and T-cells of immune system.

$$S_i(t) = S_i(t-1) + (\alpha \frac{\sum_{j=1}^{N}(m_{ij}s_j(t))}{N} - \alpha \frac{\sum_{k=1}^{N} m_{ki}s_k(t)}{N} + \beta g_i - c_i(t-1) - k_i)s_i(t) \quad (1)$$

$$s_i(t) = \frac{1}{1+\exp(0.5 - S_i(t))} \quad (2)$$

$$c_i(t) = \eta(1 - g_i(t))S_i(t) \quad (3)$$

where $i, j = 0,\ldots,N\text{-}1$, N is a number of antibody types, $S_i(t)$ is stimulus value of antibody i, $s_i(t)$ is concentration of antibody i, $s_j(t)$ is not concentration of self-antibody but one of other agent's antibody obtained by communication, $c_i(t)$ is concentration of T-cell which control concentration of antibody, m_{ij} is mutual stimulus coefficient of antibody i and j, g_i is affinity of antibody i and antigen, are constants.

According to the equation (3), when the stimulus value of antigen ($g_i(t)$) is going on large and the stimulus value of antibody ($S_i(t)$) is going on small, the concentration of T-cell ($c_i(t)$) is gradually going on small. Therefore, in this case c_i(t) take a role of helper T-cell that stimulates B-cell. On the contrary, the stimulus value of antigen is small and the stimulus value of antibody is large, the $c_i(t)$ is large. So, it takes part in suppressor T-cell. In biological immune system the helper T-cell activates B-cell when the antigen invades it, and the suppressor T-cell prevents the activation of B-cell when the antigen was eliminated. Consequently, the performance of the system can be improved by adding T-cell in immune network

model. The main reason why the T-cell model was added is that the system adapts the environment quickly by recovery of the concentration of antibody to the initial state when the antigens are removed by antibodies. This is more similar to the biological immune system.

Table 1. The role of T-cell in various states

$g_i(t)$	$S_i(t)$	c_i(t)	state	role of T-cell
large	small	very small	intrusion of antigen	helper T-cell
large	large	small	eliminating	-
small	large	large	eliminated	suppressor T-cell
small	small	small	stable	-

3. Swarm Behavior Algorithm Based on Artificial Immune System

Table 2 shows relationship between MAS and immune system.

Table 2. Relationship between MAS and immune system

MAS	Immune System
Agent's environment	Antigen
Action strategy	Antibody
Agent	B-cell
Control parameter	T-cell
Adequacy	Stimulus
Inadequacy	Suppression
Excellent agent	Plasma cell
Inferior agent	Inactivated cell

In this paper, we developed the artificial immune network for selecting strategy to execute group behaviors in MAS. In general, the process of MAS to execute the given task is as follows.

[Step 1] Selection of the group behavior strategies
[Step 2] Execution of the selected strategy
[Step 3] Sensing of the changed environment
[Step 4] Arbitration of behavior strategies
[Step 5] Return to the step 1.

In order to achieve this process, we modeled the group behavior strategies and their relation in the next section 3.1, and proposed the algorithm for behavior arbitration of group behavior in the section 3.2.

3.1 Modelling of Group Behavior

Objective of the system and the proposed algorithm

The objective of the system is for agents to find and carry out given tasks; the tasks are spread out in the environment broadly. By proposed immune algorithm, strong strategy is selected as a swarm strategy. Namely, almost all agents select strong strategy. After that, group behavior can be realized. If environmental changes should occur, all agents can adapt itself to new situation rapidly. This algorithm is modeled based on clonal selection and immune network hypothesis.

Definition of the antigen

According as the given tasks is broadly distributed in the workspace, we classified the density of the tasks into four levels, that is, high, medium, low, and nothing. For each of these densities, an agent faces with several strategies that are *[1] Aggregation*, *[2] Random search*, *[3] Dispersion*, and *[4] Homing*. Accordingly, each environment of four levels is regarded as antigens and each of these strategies is regarded as antibodies.

Definition of the antibody

In order to find and execute the tasks, for each environment that is defined above, four strategies are introduced. Each strategy (antibody) and its definition are as follows.

Aggregation (Ab0): the ability of a group of agents to gather in order to establish and maintain some maximum inter-agent distance.
Random Search (Ab1: basic strategy): the ability to find task by moving random direction.
Dispersion (Ab2): the ability of a group of agent to spread out in order to establish and maintain some minimum inter-agent distance.
Homing (Ab3): the ability to find and go to a particular region or location.

Stimulus value of antigen to antibody is defined as figure 1, according to the percentage of task detection during past given times (T_c). When density of task is high, stimulus value for aggregation (*Ab0*) is also high. Likewise, other functions are defined. Figure 1 represents the definition of every function (g_0~g_3) that has overlapping region. Particularly, when the density of the task is zero, g_3 sets as 0.5 and g_2 sets as 0.5. Because of the strategy of g_3 is 'Homing'. Table 3 represents the mutual stimulus coefficient that is the affinity of antibodies in equation (3). Figure 2 shows the conceptual diagram of immune network of equation (3) to (6). This figure represents the interaction between antigens and antibodies (B-cells), antigens and T-cells, and antibodies and antibodies.

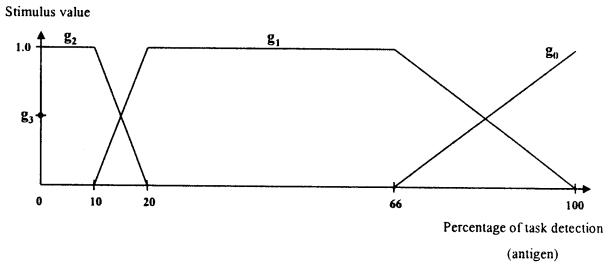


Fig. 1. Stimulus function of antigen to antibody (g_i)

Table 3. Mutual stimulus coefficient (m_{ij})

Agent j / Agent i	Aggregation (Ab_0)	Search (Ab_1)	Dispersion (Ab_2)	Homing (Ab_3)
Aggregation	1	-0.4	-0.2	-0.4
Search	-0.4	1	-0.4	-0.2
Dispersion	-0.2	-0.4	1	-0.4
Homing	-0.4	-0.2	-0.4	1

3.2 Decision making for Swarm Behavior

When an agent carries out the given task in MAS, the agent must decide its behavior by local information. At this time, the agent is not able to know all the information of the whole system. Thus, it is difficult to realize group behavior such as movement of group or arrangement of system. The idea of the immune response is applicable to arbitration of group strategy in MAS.

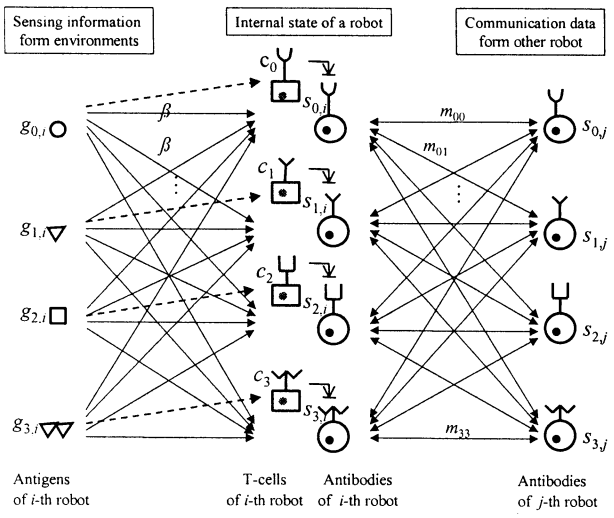


Fig. 2. Proposed immune network include T-cell and B-cell model

Once an agent decides its behavior by perception of its environment, its behavior strategies is stimulated and suppressed by relationship of other agent that encounters. Naturally, this process is accomplished by local communication of autonomous mobile agent. When an agent encounters other agent, if other agent's strategy has the same or similar strategy, this strategy is stimulated, if not, this strategy is suppressed. At this time, if an agent is stimulated very much, its behavior is regarded as adequate one in the system. Therefore, this agent transmits its strategy to other agents. In this way, swarm strategy is decided.

The detailed algorithm is as follows.

[Step 1] Initialize stimulus value and concentration of antibody for all action strategies.

$$t = 0$$

$$S_i(0) = 0.5,\ s_i(0) = 0.5 \text{ for } i = 0, \ldots, N\text{-}1$$

where N is the number of action strategies.

[Step 2] Select and execute strategy (antibody) that has bigger concentration of antibody (s_i) than others (In start, basic strategy (Ab_1) is selected).

[Step 3] When an agent encounters other agent, they stimulate and suppress each other using local communication. Stimulus value of B-cell (S_i), concentration of antibody (s_i), and concentration of T-cell are calculated by equation (6), (4), and (5) respectively.

The stimulus term and the suppression term are put together as the second term of equation (6), because m_{ij} is plus (stimulus) or minus (suppression) value.

$$S_i(t) = S_i(t-1) + (\alpha \frac{\sum_{j=1}^{N} (m_{ij} s_j(t))}{N} + \beta g_i - c_i(t-1) - k_i) s_i(t) \quad (6)$$

where $i, j = 0,\ldots,N\text{-}1$, s_j is concentration of other agent's antibody, m_{ij} is mutual stimulus coefficient of antibody i and j(table 2), α, β are parameters of response rate of other agents and environment (antigen).

[Step 4] If an agent has a strategy that was stimulated over upper threshold ($\bar{\tau}$), then it becomes excellent agent.

→ it can transmit strategy to inferior agent when it encounters inferior agent.

If an agent has all strategies that were stimulated under lower threshold ($\underline{\tau}$), then it becomes inferior agent.

→ it receives good strategy from excellent agent when it encounters excellent agent.

$$\bar{\tau} \text{ (upper threshold)} = 0.622 (= \frac{1}{1+e^{-0.5}}) \quad (7)$$

$$\underline{\tau} \text{ (lower threshold)} = 0.378 (= \frac{1}{1+e^{0.5}}) \quad (8)$$

[Step 5] If an inferior agent encounters excellent agent, it receives all strategies and renews concentration of each strategy.

[Step 6] $t = t + 1$, return to the Step 2.

Above swarm immune algorithm was modeled on the three parts of immune system that are clonal selection, immune network, and the function of T-cell.

(1) Clonal selection: excellent agent transmits its strategy to others.

(2) Immune network of antibodies (B-cell): agents stimulate and suppress others by comparison of action strategy using communication.

(3) Helper T-cell and suppressor T-cell: Immune network makes the system identically. Nevertheless, when an environment changes, an agent adapts the environment quickly by recovery of the concentration of antibody to the initial state. This is the function of T-cell.

4. Simulations

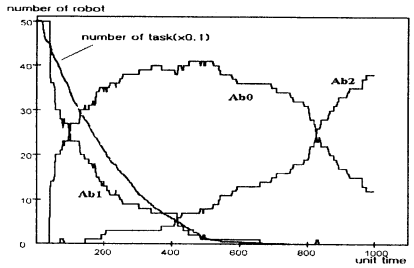
The simulation conditions, for verifying the effectiveness of the proposed swarm immune algorithm, set as follows.

The fifty agents are spread out in the workspace of 10[m]×10[m] wide. The size of agent is 50[mm] in diameter. The objective of system is that agents find and carry out the given tasks. We set a communication radius set as 500[mm] in simulation. Also we set the antibody evaluation time (T_c) as 40 unit times to calculate the percentage of task. During 1 unit time, an agent can change its direction or move 25[*mm*] forward. An agent takes 15 unit times to carry out a task. The number of task is 500. Another simulation parameter values set as α =0.3, β =0.05, k=0.002.

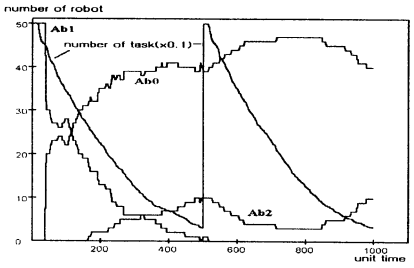
To evaluate the proposed swarm immune algorithm includes B-cell and T-cell modeling and clonal selection, we took the two cases of simulations for verify the effect of T-cell model. **Case 1** is that only using modeling of B-cell and clonal selection. That is equation (6) is not used. **Case 2** is that using modeling of B-cell, T-cell and clonal selection.

Figure 3 shows the result of the case 1. In figure 5(a), during from 100 to 200 simulation times, many agents chose the strategy of aggregation, because of the density of task around agent is high. As the simulation time increases, most of agents' strategies are continuous on the aggregation, though the density of task has decreased gradually. In this case, agent's strategy doesn't follow the change of the environments, and the case of figure 4(b) shows the similar result of it.

This result shows that B-cell modeling is effective only static environment, because of the lack of adaptation ability. In the biological immune system, the helper T-cells activate B-cells when the antigens invade it, and the suppressor T-cells prevent the activation of B-cells when the antigens were eliminated. Therefore, it is expected that the adaptation ability is enhanced by adding T-cell model.

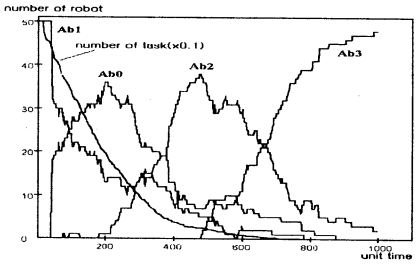


(a) The case that 500 tasks are supplied at 0 unit time

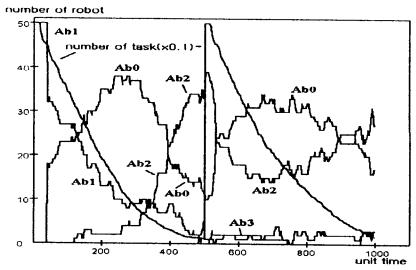


(b) The case that 500 tasks are supplied at 0 and 500 unit time

Fig. 3. Result of using modeling of B-cell and clonal selection



(a) The case that 500 tasks are supplied at 0 unit time



(b) The case that 500 tasks are supplied at 0 and 500 unit time

Fig. 4. Result of using modeling of B-cell, T-cell, and clonal selection

Figure 4 shows the result of case 2. In figure 4(a), during from 100 to 200 simulation times, many agents chose the strategy of aggregation like case 1. However, as the time goes on, the strategy of agent changes to dispersion mode and homing mode. The simulation time reached about 700, agents' strategies change to the strategy of homing. This is the reason why the most tasks are removed. It is similar to the natural immune system that includes T-cells. The adapting ability of proposed immune system improves by using T-cell modeling. Fig. 5(b) is that the tasks are supplied at 500-unit time; in this case, the agents adapt the environment well.

Immune network of B-cell model makes the system identically. Nevertheless, when an environment changes, an agent adapts the environment quickly by recovery of the concentration of antibody to the initial state. Therefore, as the number of task is decrease, the group behavior is emerged following order; (1) random search (2) aggregation (3) dispersion (4) homing.

5. Conclusion

This paper proposed the algorithm based on immune system to achieve the goal of MAS. We found the analogy between MAS and immune system, and applied to making swarm strategy in MAS. We made a condition that swarm behavior emerged; it could be achieved only when successful strategies are selected by all agents in the system. This is based on clonal selection that a successful clone is selected and proliferated, and immune network hypothesis that is modeled after mutual action of antibodies. In order to improve the adapting ability of agent in changing environment, we proposed the T-cell model of the immune network equation.

Acknowledgement

This work was supported by a grant No. 2000-2-30300-003-3 from the Basic Research program of Korean Science & Engineering Foundation (KOSEF)

References

[1] A. Asama et al. eds, *Distributed Autonomous Robotic Systems I, II*, Springer-Verlag, 1994, 1996.

[2] D. W. Lee, K. B. Sim, "Behavior Learning and Evolution of Collective Autonomous Mobile Robots using Distributed Genetic Algorithms," *Proc. of 2nd Asian Control Conference*, vol.2, pp. 675-678, 1997. 7.

[3] D. W. Lee, K. B. Sim, "Development of Communication System for Cooperative Behavior in Collective Autonomous Mobile Robots," *Proc. of 2nd Asian Control Conference*, vol. 2, pp. 615-618, 1997. 7.

[4] D. W. Lee, K. B. Sim, "Artificial Immune Network-based Cooperative Control in Collective Autonomous Mobile Robots," *Proc. of the 6th IEEE Int. Workshop on RO-MAN*, pp. 58-63, 1997. 9.

[5] I. Roitt, J. Brostoff, D. Male, *Immunology, 4th edition*, Mosby, 1996.

[6] R.A. Wallace, G.P. Sanders, R. J. Ferl, *BIOLOGY: The Science of Life, 3rd eds.*, HarperCollins Publishers Inc., 1991.

[7] Y. Ishida, N. Adachi, "An Immune Algorithm for Multiagent: Application to Adaptive Noise Neutralization," *Proc. of IROS 96*, pp. 1739- 1746, 1996.

[8] S. Forrest, B. Javornik, R. E. Smith, A. S. Perelson, "Using Genetic Algorithms to Explore Pattern Recognition in the Immune System," *Evolutionary Computation*, vol. 1, no. 3, pp. 191-211, 1993.

[9] A. Ishiguro, Y. Watanabe, Y. Uchikawa, "An Immunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots," *Proc. of IROS 95*, pp. 495-500, 1995.

[10] A. Ishiguro, Y. Shirai, T. Kendo, Y. Uchikawa, "Immunoid: An Architecture for Behavior Arbitration Based on the Immune Networks," *Proc. of IROS 96*, pp. 1730-1738, 1996.

[11] H. Bersini, F.J. Varela, "The Immune Recruitment Mechanism: A Selective Evolutionary Strategy," *Proc. of 4th Int. Conf. on Genetic Algorithms*, pp. 520-526, 1991.

[12] P. D'haeseleer, S. Forrest, P. Helman, "An Immunological Approach to Change Detection: Algorithms, Analysis and Implications," *Proc. of IEEE Symp. on Security and Privacy*, 1996.

[13] J.D. Farmer, N.H. Packard, and A.S. Perelson, "The immune system, adaptation, and machine learning," *Physica 22-D*, pp. 184-204, 1986.

[14] N.K. Jerne, "Idiotopic Network and Other Preconceived Ideas," *Immunological Rev.*, vol. 79, pp. 5-24, 1984.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Bayesian Clustering for Determination of Dynamic Web Preference

Dae Su Kim*, JunHyeog Choi**, Mun-Sung Han***

* Dept. of Computer Science, HanShin Univ., KOREA
**Dept. of Computer Science, Kimpo College, KOREA
*** Computer Software Technology Lab, ETRI, KOREA

Abstract

Collaborative filtering method for personalization can suggest new items and information that a user hasn't expected. But there are some problems. Not only the steps for calculating similarly value between each user is complex, but also it doesn't reflect user's interest dynamically when a user input a query.

In the paper, classifying users by their interest makes calculating similarly simple. We propose the algorithm for readjusting user's interest dynamically using the profile and Bayesian learning. When a user input a keyword searching for a item, his new interest is readjusted. And the user's profile that consists of used keywords and the presence frequency of keywords is designed and used to reflect the recent interest of users.

Our methods of adjusting user's interest using the profile and Bayesian learning can improve the real satisfaction of users through the experiments with data set, collected in a University's library. It recommends items which the user would be interested in.

Key words: Bayesian learning, cluster analysis, book recommendation system

1. Introduction

In most web sites, they recommend users preference products or other information statically based on user's past profile information, product retrieval, and product purchase information. This kind of recommending system using collaborative filtering method have the advantage to provide related links dynamically by using other user's evaluation information[3, 4]. But this method has the disadvantage of process complexity and inefficiency of calculating item similarity because we need to apply to the all the users. Also static user profile reduces the efficiency as time goes on and have the high probability of providing erroneous information to the user[4, 5].

Therefore preference information based on profile information needs to be updated dynamically. To solve this problem, in this paper, we utilize clustering method to cluster similar preference users. This method simplifies the calculating process by calculating representative user instead of all the users. Also we designed this system utilizing Bayesian learning to update. When the key is given, the corresponding keyword preference information is to be updated dynamically.

2. Preference update utilizing cluster analysis and Bayesian learning

Previous methods does not reflect present user keyword retrieval pattern and it uses calculated information based on past data set[6]. This method may have high probability of erroneous results by not reflecting new item preference. We designed [algorithm 1] to overcome this problem.

[Algorithm 1] User preference update algorithm based on Bayesian learning

[step 1] : Processes clustering using user's profile.
[step 2] : Calculates keyword preference for each cluster using keyword(book name, classification number)

$$P(C_j) = \frac{\sum_{i=1}^{n} f_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{k} f_{i,j}}$$

where i=1,2,...,n and j=1,2,....k and $f_{i,j}$: frequency of borrow

[step 3]: Calculates preference degree for each keyword for all user cluster

$$P(C_j \mid U_i) = \frac{P(C_j, U_i)}{P(U_i)}, \text{ where}$$

$$P(C_j, U_i) = \frac{f_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{k}} \quad \text{and}$$

$$P(U_i) = \frac{\sum_{j=1}^{k} f_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{k} f_{i,j}} \cong P(C_{j,i})$$

$C_{i,j}$ can be calculated as follows

$$P(C_{j,i}) = \frac{f_{i,j}}{\sum_{i=1}^{n} f_{i,j}}$$

[step 4] : Updates keyword preference degree by Bayesian learning for the specific user's keyword input

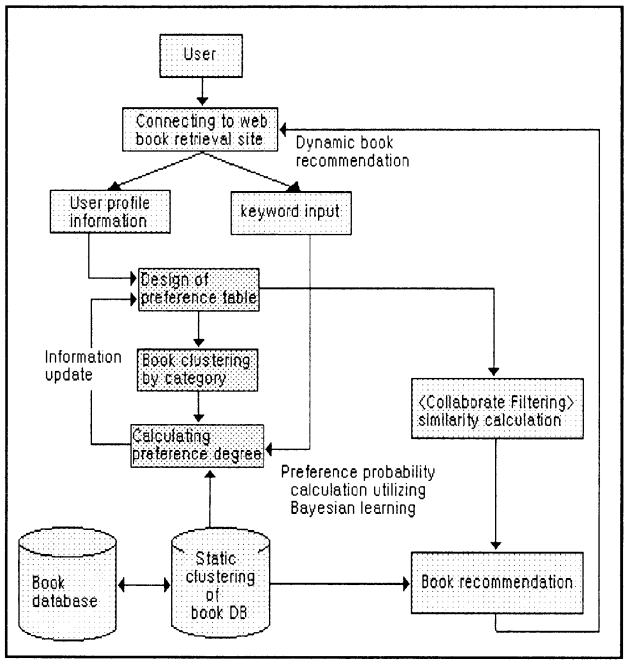
$$P(C_{j,i} \mid X_i) = \frac{P(C_{j,i}, X_i)}{P(X_i)} = \frac{P(C_{j,i})P(X_i \mid C_{j,i})}{P(X_i)}$$

where $P(X_i) = \sum_{j=1}^{k} P(C_{j,i})P(X_i \mid C_{j,i})$

[step 5] : Recommends book dynamically by collaborate filtering between a keyword input user and a user who get the similar keyword preference degree in the cluster.

3. Design of dynamic web book preference recommending system

Design of dynamic web book preference recommending system is illustrated in <Figure 1>. It includes collaborative filtering module by Bayesian learning based on user profile. Here book database is classified by category.



<Figure 1> System Configuration

3.1 Book classification table and matrix

This dynamic web book recommending system restricts on the computer related books. The book classification method uses Dewey's 10 digit classification system, and each books recorded in book classification table has a unique representative classification number.

The user profile includes book classification number, frequency for borrowing, keywords used for retrieval and those are utilized as data for the preference information. <Figure 2> illustrates an example of book classification table.

005 Computer programming, program, data
.1 Programming
.11 Special programming techniques
.111 Modular programming
.112 Structured programming
.113 Functional programming
.114 Object-oriented programming
.115 Visual programming
.12 Software systems analysis and design
.13 Programming languages
.131 Symbolic(Mathematical) logic
.132 Specific programming languages
.14 Verification, testing, measurement, debugging
.15 Preparation of program documentation
.2 Programming for specific types of computers, for specific operating systems
.21 Programming for digital supercomputers
.22 Programming for digital mainframe computers

<abbreviation>

<Figure 2> Book classification table

The characteristics of cluster can be determined by utilizing keyword preference in each cluster. Formula (1) is to determine preference degree with in a cluster for the specific keyword., and the total number is number of cluster * total number of keywords.

$$P(\text{preference degree of a specific keyword}) = \frac{\text{the frequency of borrowing book of a specific keyword}}{\text{the number of borrowing book} \times \text{the number of keywords in that cluster}} \quad (1)$$

Also the keyword preference degree for user in a cluster can be represented as formula (2).

$$P(\text{preference degree of keyword} \mid \text{specific user}) = \frac{\text{the number of borrowing book of a specific keyword for the specific user}}{\text{total frequency of borrowing book for a specific user}} \quad (2)$$

3.2 Update of user preference by Bayesian learning

When a specific keyword is given, Formula (3) is calculates the preference probability of the corresponding keyword.

$$P(\text{preference degree of keyword} \mid \text{input keyword}) = \frac{P(\text{keyword}_i\text{preference degree for a user}) \times P(\text{the ratio of input keyword})}{\sum_{keyword=1} P(\text{keyword}_i\text{preference degree for a user}) \times P(\text{the ratio of input keyword})} \quad (3)$$

According to the Bayesian formula, formula (3) can be represented as formula (4)

$$P(C_{j,k} \mid X_i) = \frac{P(C_{j,k}, X_i)}{P(X_i)} = \frac{P(C_{j,k})P(X_i \mid C_{j,k})}{P(X_i)}$$

where $$P(X_i) = \sum_{j=1}^{k} P(C_{j,k})P(X_i \mid C_{j,k}) \quad (4)$$

Bayesian learning calculates the probability of user's preference degree reflecting user's posterior data changed by user's action. When a user(U3) input a keyword(X=C4), the probability of preference degree for that keyword C1 updated by Bayesian learning can be calculated by formula (5).

$$P(C_1 \mid X = C_4) = \frac{P(C_1)P(X = C_4 \mid P(C_1))}{\sum_{j=1}^{4} P(C_j)P(X = C_4 \mid C_j)} \quad (5)$$

The user's probability of preference degree for the total keywords can be calculated like formula (5).

4. Experiments and evaluation

For the experiments, we selected 941 number of data for the 98 students who borrowed 10 or more books. The total number of borrowing book for the computer related books was 4,644 during march 1 to June 10 in year 2002 in Inha University. <Table 1> shows part of 941 number of user book borrowing.

[Table 1] Part of 941 number of user book borrowing.

Student ID	Name	Book name	Author	Borrowing data
22011678	Yoon Lee	C programming 700	Byungkhun Jung	20001080 2
22001282	Sik Park	Photoshop	David	20010820
21991333	Doyoung Yoo	Java for engineers& scientist	Champman, Stephen J	20010821
11940871	Soohak Kim	Programming Languages	Sevester	20011005
11980079	Yonghoon Kim	SQL server bible ver. 7	Byunghee Kwon	20011005
11990070	Eunsuk Kim	Structured C programming languages	Jongkyo Kim	20011005
22011360	Jungyoon Kim	Window98 Bible and Secret	Simpson, Allen	20011005
12001025	Jineun Kim	Introduction to UNIX System	Dongho Lee	20011005
. .	. .	. . .	. .	. .

The algorithm for the web book recommendation was implemented as Visual C++ 6.0 version in Pentium III 450 MHz, 256MB RAM environment

After the implementation of user clustering utilizing cosine based similarity measure, the book borrowing situation of users for each cluster is depicted as <Table 2>.

[Table 2] The book borrowing situations of users for each cluster

<cluster 1>				
Student ID	Name	Book name	Author	Borrowing data
12010008	Minhung Kang	C Language	David	20011030
12010008	Minhung Kang	Borland C++ Labray	Sungkook Kang	20011011
11951698	Minsoo kim	UNIX programming for the novice	Matthew, Nail	20011102
11980523	Hyungsuk Kim	Turbo-C language	Kwanghee Kim	20010924
...	...	...		
<cluster 17>				
Student ID	Name	Book name	Author	Borrowing data
11940362	Hyuncheol Kim	Operating System	Jaejoon Oh	20010927
11981590	Sohyun Park	Design and Implemantation of UNIX/LINUX Kernel	Hyungbong Lee	20010920
...	...	...	...	...

When a specific user logged on a system and input a specific keyword 'photoshop', the comparison of before/after update of the probability of user preference degree based on our algorithm that is 3 highest probability is illustrated in <Table 3>.

<Table 3> The comparison table of before/after update of the probability of user preference degree.

Book classification no.	.	1.64249	.	1.64255	.	5.7565	

probability of user preference degree. before update	..	0.02123	..	0.0745	...	0.0532	
probability of user preference degree. after update	..	0.02110	..	0.0842	...	0.0526	

Our system was compared with a Pearson's correlation coefficient algorithm that is a well known memory-based web book recommendation system[1].

Among the 98 sample students, 65 student who showed feedback for the suggested method are selected. They evaluated the performance of the system based on satisfaction information.

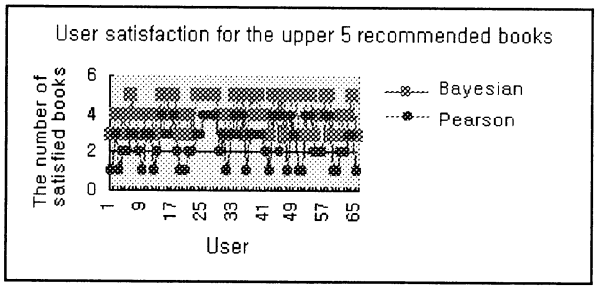
The comparison of the two methods is illustrated in <Table 4>. When a system recommends the most necessary book for the user, the ratio of user satisfaction of the recommended book is calculated.

<Table 4> User satisfaction degree evaluation

	Bayesian learning preference update algorithm	Pearson's correlation coefficient algorithm
Degree of satisfaction	86.15%	61.54%

Among the 65 students, 56 students(86.15%) showed satisfaction for our method, and 40 students(61.54%) showed satisfaction for the Pearson's correlation coefficient algorithm. This result shows that the proposed Bayesian method is more satisfactory than the Pearson's correlation coefficient algorithm

Next, we experimented two algorithms such that each algorithm recommended 5 upper satisfactory looking books. The number of books that each user was satisfied among 5 book are depicted in <Figure 3>. We can notice that most of the students are more satisfied in our method than that of Pearson's.



<Figure 3> User satisfaction for the upper 5 recommended books

5. Conclusion

In this paper, we suggested a dynamic update algorithm by improved collaborative filtering method utilizing clustering technique and Bayesian learning. We formed a user cluster that showed high similarity and regarded it as a representative user that showed cluster characteristics.

User need to calculate similarity degree using only representative user instead of the total users, therefore it simplified the processing procedure. Also prior and posterior probability is dynamically reflected utilizing Bayesian learning and it provided a more satisfactory results.

Further research is now on progress considering the user's relation between input keywords and applying neural networks and genetic algorithm for the efficient classification algorithm

6. references

[1] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

[2] Bamshad Mobasher, Hoghua Dai, Tao Luo - Yuqing Sun, Jiang Zhu, "Integrating Web Usage and Content Mining for More Effective Personalization," EC-Web 2000.

[3] M. Pazzani, D. Billsus, Leraning and Revising User Profiles:The Identification of Interesting Web sites, Machine Learning 27, Kluwer Academic Publishers, pp. 313-331, 1997.

[4] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," Accepted for publication at the WWW10 Conference. May, 2001.

[5] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu "Mining Access Patterns Efficiently from Web Log", Proc. 2000 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00), Kyoto, Japan, April 2000.

[6] A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng, ``Constraint-Based Clustering in Large Databases '', Proc. 2001 Int. Conf. on Database Theory (ICDT'01), London, U.K., Jan. 2001.

[7] Jitian Xiao, Yanchun Zhang, "Clustering of web users using session-based similarity measures", Computer Networks and Mobile Computing International Conference on, 2001

[8] H. Yin and N.M. Allinson, " A Bayesian self-organizing map for Gaussian mixture," IEE Proc.- Vision, Image and Signal Processing, Vol. 148, No. 4, pp. 234-240, 2001.

[9] Feenan, J., Fry, P., Ming Lei, "Clustering Web Accelerators", Advanced Issues of E-Commerce and Web-Based Information Systems, 2002. (WECWIS 2002). Proceedings. Fourth IEEE International Workshop on, 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

The Research about Growth and Behavior of a Virtual Life by using Genetic Algorithm

Min-Su Kwon, Do-Wan Kim, Ja-Yong Lee and Hoon Kang

School of Electrical and Electronic Engineering, Chung-Ang University, Korea
221, Huksuk-Dong Dongjak-Ku, Seoul 156-756, Korea
impactblue@sirius.cie.cau.ac.kr angeldarker@sirius.cie.cau.ac.kr jalnans@sirius.cie.cau.ac.kr
hkang@cau.ac.kr

Abstract

In this paper, we modeled a virtual life (VL) that react to the user's action according to its own behavioral characteristics and grows itself. We established some conditions with which such a VL is designed. Genetic Algorithm is used for the growth process that changes the VL's properties. In this process, the parameter values of the VL's properties are encoded as one chromosome, and the GA operations change this chromosome. The VL's reaction to the user's action is determined by these properties as well as the general expectation of each reaction. These properties are evaluated through 5 fitness measures so as to deal with multi-objective criteria. Here, we present the simulation of the growth process, and show some experimental results.

Keywords: Virtual Life, Genetic Algorithm, Growth Process, Artificial L
ife, Behaviors and Reaction

1. Introduction

In these days, due to the vast internet infra and the powerful computing speed, we easily experience the online virtual environments, such as online games, avatar chatting and 3D shopping malls from which many peoples and companies induce some value-added profits.

The aims of our study are to make the virtual life (VL) that sympathizes with human and behaves itself like a real creature in the virtual environment. In this paper, we design a virtual life that react to user's action according to it's own properties and grows itself as an artificial life form. We choose a puppy-like artificial creature, a most intimate friend with the human, for a candidate of a VL. A user grows one's own VL in the virtual environment. We define some conditions, with which such a VL must possess, and it is simulated according to those conditions.

First, the VL should act differently according to its own properties. Second, the VL's properties are to be changed interactively with a user. Such evolving properties grow the VL. Finally, the VL's properties, converging to one direction, must not be changed easily as a user provides other action for it and leads the properties into another direction.

Among computational techniques of artificial intelligence, we use Genetic Algorithm (GA) for the growth process of the Virtual Life due to its simplicity and efficiency.

GA was proposed by John Holland in early 1970s. It is a search algorithm based on the mechanics of natural selection and evolving genetics. GA has been found to be one of the most flexible, efficient and robust among all search algorithms known to artificial intelligence. Because of these properties, the method is now widely used to solve a broad range of different optimization problems.

This work was supported by grant no. R01-2000-000-00277-0 from the Basic Research Program of the Korean Science & Engineering Foundation (KOSEF).

2. The Growth Process of Virtual Life

In this section, we will explain the growth process of Virtual Life using GA. Virtual Life grows under the influence of user's action after it creates first. Virtual Life's growth was composed of variation of personality and body two special properties.

2.1 Chromosomes and Population

The VL's properties are composed of two behavioral properties and three physical properties. These five properties are coded as the chromosomes, whose parameters are shown in Table 1.

Table 1. Chromosomes of Virtual Life

Phenotype (Properties)			Genotype	
Tendency	Docility	-63 ~ +64	7bit	32 bit binary string
	Activity	-63 ~ +64	7bit	
Body	Fatness	-63 ~ +64	7bit	
	Physical Strength	-63 ~ +64	7bit	
	Stature	-7 ~ +8	4bit	

Each property acts as the following roles. The docility is the most important property that determines the VL's behavior.

the VL behaves obediently if the obedience value is high while it shows a violent action if the obedience value is low.

The activity is the parameter of positiveness in reaction. The VL behaves a positive action if its value is higher or equal to 0 while it shows a negative action if its value is lower than 0.

The fatness is relatively less effective than the above 2 behavioral parameters but it has an effect on the external form as the VL becomes embodiment in graphics. When the value is 0, it has a general form; if the value is high than 0, it is in a fat form; and if the value is low than 0, it is in a skinny form.

The physical strength displays the condition of the VL's health. If the physical strength is lower than a regular value, the VL becomes dead in the end.

Finally, the stature is a parameter, which has no influence on VL's action, but is used graphically.

These five properties represent the VL's status that changes according to the user's action. The VL's properties are stored in a variable of 4 bytes (32 bits) in this composition. With GA, a VL has a total 100 chromosomes. Initially, 100 chromosomes are generated by the uniform random numbers. Therefore, a VL born at the first time shows a very unstable reaction and it converges to a direction that corresponds with the user's action.

2.2 Fitness evaluation based on user's action

Each chromosome gets the fitness value from the user's action, and the effect is the change of each chromosome as shown in Table 2.

Table 2. The effect that user's action gives in VL's property.

User's action \ VL's Property		Docility	Activity	Fatness	Physical Strength	Stature
Embracing		5	3	0.3	3	0
Stroking		4	2	0.1	2	1
Neglecting		-0.5	-0.1	0.3	-0.7	0
Shouting		-3	-0.4	-1	-0.4	-1
Threatening		-5	-3	-5	-5	-1
Hitting		-10	-1	-10	-7	-1
Feeding	few	-5	-7	-5	-10	-5
	a few	0	-1	-3	-3	2
	middle	1	0.5	0	2	5
	many	0	1	4	1	4
	very many	-2	-7	9	-10	3
Exercising	a few	0	-5	7	-3	0
	middle	1	2	-3	6	5
	many	-3	-2	-5	-3	-2

These numerical values are decided during an experiment of the user's action so that the chromosome may converge to a suitable direction. Actions such as embracing, stroking in Table 2 let the population evolve generation by generation through GA operations as the user's actions add. Actions such as feeding, exercise are counted for a given number (for example, 10), and then the operations are performed. The fitness of each chromosome is calculated by the next formula..

$$F_i = (f_i - f_{\min}) + (f_{\max} - f_{\min})/(K-1) \qquad (1)$$

$$f_i = \sum_{all\ j} (V_j \times T_j)$$

f_i : the fitness value of i th chromosome
f_{min} : the minimum fitness value among chromosomes
f_{max} : the maximum fitness value among polulation
F_i : the last fitness value of i th chromosome
K : the selection pressure
V_j : j th property value of chromosome
T_j : the proportional factor of Table 2's j th property

The greatest value of the whole individuals becomes K times of the smallest value of the whole individuals in this formula and we can keep balanced between the variance of the population and the convergence speed by controlling the selection pressure K. In the case of feeding, when the user do not give food continually, it causes a fatal effect, but the feeding is shown to be less effective than other actions because it runs the operations only once for the specified period of time. When the user does not give food for the certain period, we can regulate the reflection ratio by lowering the selection pressure of other actions and by increasing the selection pressure of the feeding, in order to solve the problem.

2.3 Crossovers and Mutations

The Simple GA operations such as crossovers and mutations are used here. We select the chromosomes of 50 pairs through the roulette selection according to the fitness value calculated in Section 2.2 and the chromosomes of the selected 50 pairs create a 100 offspring generation to replace the parent generation through the crossovers and mutations.

The fitness value changes each time as the user's action differs. Therefore, we do not use the method that some parent generation's individual is passed to the offspring generation with the elite conservation. The number of the crossover sites is set between 1 and 5 at random on each pairs. The mutation is performed on a bit-by-bit basis with probability and the mutation rate is 0.3 ~ 1.5%.

3. Reaction of Virtual Life

The VL's reaction to user's action depends on the expectation value of an anticipated reaction (Table 4) and the effect that gives VL's property calculated in Section 2.3 (Table 5).

Table 3. The expectation value of an anticipated reaction

VL \ User	Embracing	Stroking	Neglecting	Shouting	Threatening	Hitting
Jumping	9	8	5	3	4	5
Seducing	10	10	6	2	2	1
Sitting	9	9	9	6	4	3
Stretching	7	8	9	6	4	2
Neglecting	5	7	10	7	8	3
Warning	5	6	9	10	9	9
Barking	3	5	8	10	10	9
Escaping	2	5	5	8	10	10
Biting	1	3	2	9	9	10

Table 4. The effect that gives VL's property

Property / Reaction	Docility	Activity	Fatness	Physical Strength
Jumping	5	4	-3	4
Seducing	7	3	-1	5
Sitting	3	-2	2	0
Stretching	1	-3	3	0
Neglecting	-2	-3	2	0
Warning	-4	-1	1	-1
Barking	-6	2	0	3
Escaping	-7	4	-2	4
Biting	8	2	-3	5

The values of Table 3~4 are determined similarly by an experiment as before. The expectation values of general reaction to the user's action in Table 3 play the role of not producing biased action to appear too much by such VL's property. The effect that the VL's property in Table 4 gives in reaction is established as explain in Section 2.1.

We select one chromosome by roulette selection to decide VL's reaction and calculate the expectation value that will be selected each reaction about all kind of reaction by next formula.

$$E_i = G_i + \sum_{all\ j} A_{ij} \times V_j \qquad (2)$$

E_i : the expectation value that i th reaction will be selected
G_i : the general expectation value about i th reaction
A_{ij} : the proportional factor that j th property is reflected in i th property
V_j : the j th property value

Then, the VL's reaction is decided by the roulette selection according to the expectation values,

4. Experimental Results

The block diagram of the simulation system is shown in Fig.1.

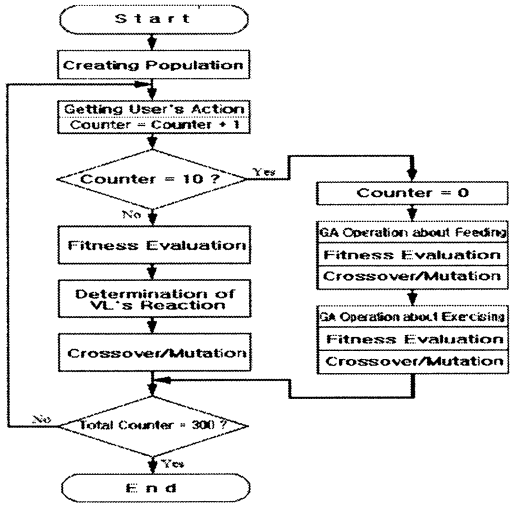


Fig 1. Block diagram of Simulation System

The input of this simulation system is user's action. The user is assumed to act and correspond to the VL, 10 iterations in a day and total 300 iterations during 30 days in this simulation. Every time the user corresponds to the VL, the system executes the GA operation, but user's action is feeding or exercising. If input is feeding or exercising, the system counts to 10 and execute one GA operation for these feeding or exercising inputs.

In the experiments, the user's action is classified into 2 categories (Table 5.). One trains the VL to be gentle and the other does it to be rebellious. In the similar context, the reaction of the VL is classified into 2 categories (Table 6.). One is a gentle reaction, and the other is a violent reaction.

Table 5. Categories of the user's action

Symbol	Action	Category
a	embracing	Actions that make the VL docile tendency
b	stroking	
c	neglecting	Actions that make the VL violent tendency
d	shouting	
e	threatening	
f	hitting	
g	fedding	Actions that give critical effect to VL
h	exercise	

Table 6. Categories of the VL's reaction

Symbol	Action	Category
A	jumping	Reactions of docile tendency
B	seducing	
C	sitting	
D	stretching	Reactions of violent tendency
E	neglecting	
F	warning	
G	barking	
H	escaping	
I	biting	

We observe that the VL's properties, reactions, and the ratio of 2 behaviors of the user's action during 300 generations are changed. However, the ratio of feeding and exercising tends to be fixed about 2 times and 1 time, respectively, in a day, since they give a critical effect to the survival of the VL. Therefore, we used 210 inputs and the associated outputs, to 300 inputs in order to analyze the simulation results. Next, we present the simulation results.

In the Simulation, we assumed that one breeds a pet in the general manner of the real world. We choose most of the simulation inputs for the user's actions as embracing and stroking, so that the VL may evolve to be docile. And we choose a few inputs as violent action, in order to observe the change of the VL's reactions and properties, as the user gives some violent action to it, while it grows into a docile one. 70% of the inputs are the user's actions that make the VL docile and the remaining 30% are the actions that make the VL violent.

In this case, the VL's property values are changed as shown in Fig. 2. The graph shows the change of the mean value of each property value by increasing the user's action.

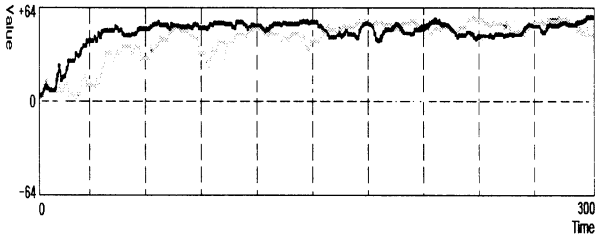


Fig. 2. Change of VL's property mean values on time domain.

The thickest line in Fig. 2 represents the change of the mean value of docility. This line confirms that the VL's behavior converges into docile tendency as the user's actions are on the increase. Fig. 3 displays a distribution of VL's reaction about increase of user's action.

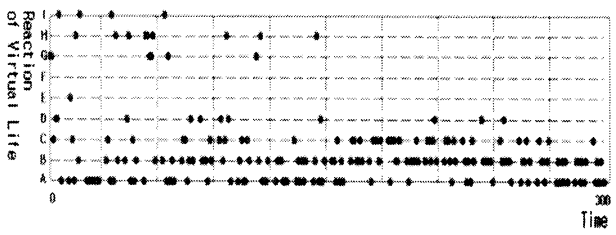


Fig.3. A distribution of VL's reaction in time domain.

The VL's reactions are uniformly distributed over 'A' to 'I' before the properties converge to a docile tendency. The VL's reactions are inclined toward a docile tendency after the properties converge to a docile tendency as shown in Fig. 2~3. As a result, we can see that the VL's reaction is influenced by the VL's properties, which converges to a particular tendency. we can confirm this result more definitely in Fig. 4~5.

Fig. 4 shows the histograms of the user's action and the VL's reactions for all the users, while Fig. 5 shows the histograms of the 100 repeated experiments under the same conditions.

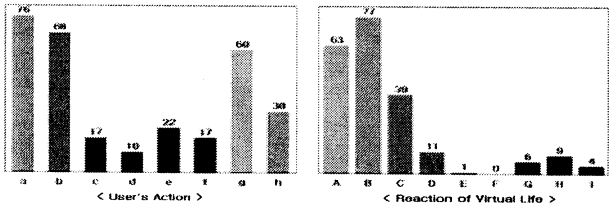


Fig. 4. Histograms of the user's action and the VL's reaction.

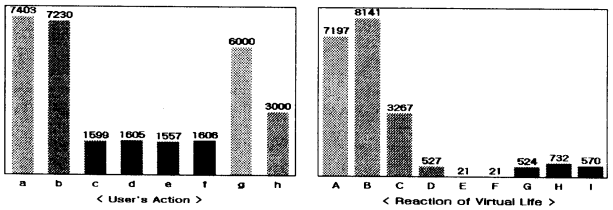


Fig. 5. Histograms of the user's action and the VL's reaction for the 100 repeated experiments.

The ratio of the VL's reaction of violent tendency in the total reaction is less than half of the user's action that makes the VL violent as shown in Table. 7.

Table. 7. Ratios of results in Fig. 5 and Fig. 6

		Fig. 5	Fig. 6
user's action	to make the VL docile	68.6%	69.7%
	to make the VL violent	31.4%	30.3%
VL's reaction	a docile tendency	85.3%	88.6%
	a violent tendency	14.7%	11.4%

As demonstrated in the Simulation, the growth process of the VL is influenced by the user's action, which results in a higher ratio in the total actions. Moreover, when the VL's properties converge to a particular tendency, they give a dominant affect to the VL's reaction. Finally, when the VL's properties have converged to a particular tendency, they don't change easily as the user gives some actions leading them to another direction. It means that 'what is a VL' have a tendency to maintain the present status, like the homeostasis.

5. Conclusions and Future Works

Important and yet difficult questions are how to catch the rules of the real life's behaviors and how to model the Virtual Life efficiently on these rules. In our opinion, the most important matters to make the Virtual Life that interacts with the users are the methods to determine the reaction and to change the properties of the Virtual Life.

In this paper, we modeled, through Genetic Algorithms, the Virtual Life that grows itself and reacts to the user's action according to its own properties. In order to embody this VL in the online virtual space, much consideration is needed, for example, the interaction with the virtual environment and other VLs.

We are encouraged to find the method that changes dynamically and adaptively some parameters or numerical values that are determined intuitively through experiments.

References

[1] Wen Tang and Tao Ruan Wan, "Inteligent Self-Learning Characters for Computer Games", *The 20th Eurographics UK Conference (EGUK'02)*, Leicester, Leicestershire, UK , pp. 51-58, 2002.

[2] D.E.Goldber, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Massachusetts, Addison-Wesley, 1989.

[3] P.Bentley, *Evolutionary Design by Computers*, California, Morgan Kaufmann, 1999.

[4] R.L.Haupt and S.E.Haupt, *Practical Genetic Algorithms*, New York, Wiley-Interscience, 1999.

[5] Willam M.Spears, *Evolutionary Algorithms*, Germany, Springer Verlag, 2000.

[6] David B.Fogel, *Evolutionary computation*, New York, IEEE press, 1999.

[7] M. Mitchell, *An Introduction of Genetic Algorithms*, Massachusetts, MIT press, 1997.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Structure Identification of Neuro-Fuzzy Models using Genetic Algorithms

Bo-Hyeun Wang and Hyun-Joon Cho*

Department of Electronic Engineering, Kangnung National University,
*Department of Electrical Engineering, Purdue University, West Lafayette, IN, USA, 47907

123 Chibyun-dong, Kangnung, Gangwon, Korea
E-mail: bhw@kangnung.ac.kr and hyunjoon@purdue.edu

Abstract – This paper proposes an input space partitioning method for structure identification of neuro-fuzzy models. The goal of the proposed method is to improve both accuracy and efficiency of the models. To achieve this goal, the proposed method clusters the output training data and then projects the resultant output clusters into the input space. The method next refines the arbitrary shapes of the projected clusters to hyper-rectangular partitions so that the neuro-fuzzy models can implement them in their architecture. For this, we formulate the shape refinement as an optimization problem and apply genetic algorithms to solve it. Simulation results reveal that the proposed method can achieve a remarkable improvement compared with earlier approaches.

Keywords: Neuro-fuzzy systems, structure identification, genetic algorithms, modeling.

1. Introduction

Neuro-fuzzy modeling is a problem to identify a fuzzy model of a system on the basis of input-output data by using the neuro-fuzzy systems [3]. Identification of the neuro-fuzzy models consists of two subproblems. First is structure identification that selects the most relevant input variables and partitions the associated input space. Although both input variables selection and input space partitioning are essential problems, this paper only deals with the latter one. Second is parameter identification that optimizes the adjustable parameters of the neuro-fuzzy models.

Most practical modeling problems have to concern with model's efficiency and model's accuracy. A common neuro-fuzzy modeling practice, however, uses numerical data for parameter identification, while employing an arbitrary number of fuzzy rules and setting the initial parameters either by intuition or by trial and error. Without a model efficiency requirement, it may be a feasible approach. However, when we must deal with both efficiency and accuracy at the same time, we inevitably face with the problem of the minimum structure that is indeed very difficult.

In this paper, we propose an efficient and systematic partitioning method for structure identification of neuro-fuzzy models using genetic algorithms. The objective of the proposed method is to build a consistent neuro-fuzzy model that satisfies the predetermined accuracy requirement, while using the least possible number of fuzzy rules. To achieve this goal, the proposed method performs the clustering task on the output training data and projects the resultant clusters into the input space. The cell map that can represent an approximate cluster structure is also introduced to simplify the shape refinement procedure. In order to modify the arbitrary shapes of the partitions we formulate the problem of shape refinement as an optimization problem. A cost function for optimization is developed according to a partitioning strategy.

2. Identification of Neuro-Fuzzy Models

2.1 Neuro-Fuzzy Systems

Suppose that we have two fuzzy rules where each fuzzy rule has two inputs and a single output:

Rule: If (x_1 is A_1) and (x_2 is A_2), then y is q, (1)

The architecture of the corresponding neuro-fuzzy system is shown in Fig. 1. The input term node denoted as A_j^i accepts x_j as the input and produces the degree of matching between x_j and its corresponding membership function. If we use the Gaussian membership functions for A_j^i, the output of the input term node is given by

$$\mu_{A_j^i}(x_j) = \exp\left[(x_j - c_j^i)^2 / (\sigma_j^i)^2\right], \quad (2)$$

where c_j^i and σ_j^i are called the premise parameters. The ith node in the second layer produces the output that represents the firing strength of the ith rule:

$$R_i(x) = \prod_{j=1}^{n} \mu_{A_j^i}(x_j) \quad (3)$$

The output term node in the third layer computes the value of the normalized firing strengths and provides the computed value to the last layer that acts as the defuzzifier. If we use the center of gravity defuzzification from local centroids, the output of this layer can be written as

$$y = \sum_{i=1}^{p} q^i R_i \Big/ \sum_{j=1}^{n} R_j \quad (4)$$

where p denotes the number of fuzzy rules and n represents the number of input variables.

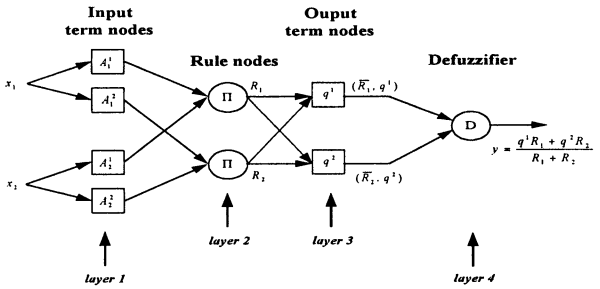


Fig. 1. Neuro-fuzzy system.

2.2 Structure Identification of Neuro-Fuzzy Models

Structure identification of the neuro-fuzzy models deals with two problems: input variables selection and input space partitioning. The problem of input variables selection is to select a set of input variables that affect the output of a system. In most cases, a finite number of possible candidates are assumed to be given. The problem of input space partitioning, on the other hand, is to find a set of initial fuzzy partitions. Although both problems are equally important, we mainly deal with the problem of input space partitioning in this paper.

The structure identification in general and the input space partitioning in particular are imperative for improving learning/operation efficiency of the neuro-fuzzy models. Without a proper solution to input space partitioning, we have to make an educational guess on the number of fuzzy partitions and initialize the associated parameters either intuitively or randomly. However, in this way, some neuro-fuzzy models do not exhibit an acceptable accuracy. Furthermore, among the models which have acceptable accuracies, most of them have many inconsistent rules. The inconsistent rules prevent us from interpreting the extracted knowledge after the modeling process. This leads to the following question: "How can we determine the minimum number of fuzzy rules so that the identified model satisfies a given accuracy requirement, while its rulebase is consistent enough?" Developing an efficient method to deal with this problem is the ultimate goal of the structure identification of the neuro-fuzzy modeling.

A number of attempts have been made to exploit the flexibility of either the tree partition or the scatter partition. Sun [4] proposed fuzzy *k-d* trees which partition the input space from a series of guillotine cuts. Sugeno and Yasukawa [3] developed a scatter partitioning method that used the fuzzy *c*-means algorithm. The input space partitioning based on clustering techniques has a couple of problems. The first is that the resultant partitions or clusters are not necessarily hyper-rectangles. Therefore, shape refinement is required for the neuro-fuzzy model to accommodate them. The second is that the number of clusters has to be predetermined in the clustering algorithm.

3. The Proposed Method

Fig. 2 illustrates the flow chart of the proposed input space partitioning method that consists of three steps: initial clustering, cell map construction, and shape refinement. As shown in Fig 2., the method starts with the minimum number of clusters, i.e., c=2 and then increments it until the identified model satisfies the prespecified conditions.

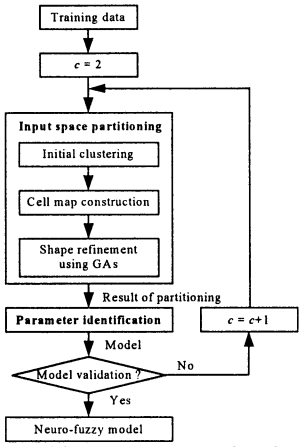


Fig. 2. Proposed input space partionting method.

3.1 Initial Clustering and Projection

The initial clustering clusters the output data and projects the resultant clusters into the input space. For the initial clustering, we adopt the method presented in [3] which uses the fuzzy *c*-means (FCM) algorithm [1]. As a result of the clustering, every output data y^i is associated with the grade of membership belonging to the fuzzy clusters $\tilde{O}_j$'s where i=1,2,...,N and j=1,2,...,c.

$$\left[y^i; \mu_{\tilde{O}_1}(y^i), \mu_{\tilde{O}_2}(y^i), \ldots, \mu_{\tilde{O}_c}(y^i)\right], \tag{5}$$

where $\mu_{\tilde{O}_j}(y^i)$ is the grade of the ith data belonging to the jth cluster, N is the number of data to be clustered, and c is the number of clusters.

Once we complete the clustering, we project the fuzzy clusters in the output space into the input space. For this, we first transform the fuzzy clusters into the crisp clusters. The crisp cluster to which y^i belongs is determined by taking the fuzzy cluster whose grade of membership of y^i is the maximum. A projected input cluster P_j is found by identifying a group of the input data which are associated with all the output data belonging to a crisp cluster O_j. This leads us to c groups of the input data.

3.2 Cell Map Construction

In order to perform the shape refinement, we construct a cell map from the groups of data in the input space. We decompose the space of interests $\boldsymbol{X} = X_1 \times \cdots \times X_n$ into a finite collection of rectangular cells where X_i is defined by the interval $[x_{i\min}, x_{i\max}]$. The size of a cell in the x_i-direction is given by

$$s_i = \frac{x_{i\max} - x_{i\min}}{r_i - 1}, \tag{6}$$

where r_i is the number of intervals in the x_i-direction.

To label a cell, we inspect all training data that belong to the cell. If a cell contains at least one data which belongs to the input cluster P_j, then it is called P_j-labeled cell. When we label the cells, we may have three types of cells: empty cells, homogeneous cells, and nonhomogeneous cells. The empty cell does not contain any data at all and thus has no label. The homogeneous cell has only one label, whereas the nonhomogeneous cell has more than one label.

3.3 Shape Refinement using GAs

We view the shape refinement as an optimization problem. In order to build a cost function for the shape refinement, we first set up a partitioning strategy as follows:

Partitioning Strategy: Minimize the overlaps between the hyper-rectangular partitions.

Too much overlap between two hyper-rectangles implies that they are either inconsistent or redundant with each other. The partitioning strategy leads us to the following simple cost function:

$$g(H_1, H_2, \ldots, H_p) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} V(H_i \cap H_j) \qquad (7)$$

where p is the number of hyper-rectangles, H_i is the ith hyper-rectangle, and $V(H_i \cap H_j)$ is the volume of $H_i \cap H_j$.

In order to refine the shapes of the input clusters, the resulting hyper-rectangles have to satisfy the following constraints:

Constraint 1 (VPC): The P_j-labeled hyper-rectangles must include all P_j-labeled cells.

Constraint 2 (NIC): A hyper-rectangle must include at least one nonempty cell.

The problem of shape refinement is to find a set of p hyper-rectangles which satisfy both VPC and NIC so that they minimize the cost function given in (7). The penalty method associates a cost with the constraint violations:

$$f(H_1, \ldots, H_p) = g(H_1, \ldots, H_p) + \gamma \Psi(H_1, \ldots, H_p) \quad (8)$$

where γ is a penalty coefficient and

$$\Psi(H_1, \ldots, H_p) = \begin{cases} 0, \text{ if } \{H_1, \ldots, H_p\} \text{satisfies VPC,} \\ 1, \text{ otherwise} \end{cases} \quad (9)$$

It should be noted that NIC is not considered in the cost function with constraint violations in (8). We will take NIC into account when we initialize strings of populations during GA search.

We apply GAs to this optimization problem. To do so, the hyper-rectangles that are the variables of the problem have to be represented as a string. Suppose that we have p hyper-rectangles in the n dimensional input space. Since a hyper-rectangle can be uniquely defined by two pointers on each of the input variables, p such vectors are concatenated to form a string of $2np$ length.

Now, we apply GA search in order to solve the problem. The detailed procedure of shape refinement that includes the GA cycle is shown in Fig. 3. The GA cycle involves initialization of a population, evaluation of the strings in a population, and genetic operations on a population.

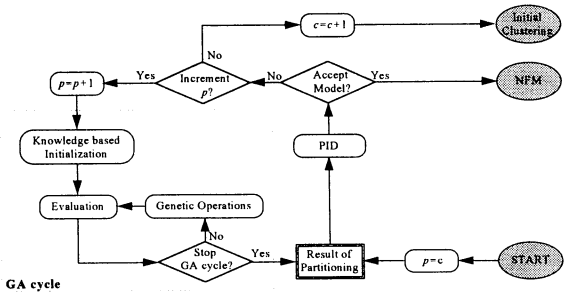


Fig. 3. The GA cycle

As shown in Fig. 2, the whole procedure of the input space partitioning starts with the smallest c. With a given c, we set p to c. When we obtain the results of partitioning for $p=c$, we perform the parameter identification, while the parameters of the neuro-fuzzy model are initialized by using the results of partitioning. After the parameter identification, we validate the identified neuro-fuzzy model. If we can accept the model, then we stop the procedure. Otherwise, we increment p to $p+1$ and go to the GA cycle again.

The first step of the GA cycle is to initialize a population. Instead of random initialization of a population, we form an initial population of strings by using the result of partitioning obtained by the previous GA cycle. It should be noted that during the initialization of a population, we generate the strings so that none of them violates NIC by simply discarding a string when it does not satisfy NIC.

In the second step of the GA cycle, we evaluate the strings in the population. In order to evaluate the strings, we use the cost function given in (8). The last step of the GA cycle involves the genetic operations such as reproduction of the population, crossover, and mutation.

We terminate the execution of the GA cycle after the number of executions of the GA cycle reaches a prespecified value. After termination, we select the string that has the minimum cost function value.

Using the string chosen, we initialize the premise parameters of the neuro-fuzzy model. If we assume that the Gaussian membership functions are employed, the centers of the Gaussian membership functions for the ith hyper-rectangle (fuzzy rule) are simply given by

$$c_j^i = x_{j\min}^i + \frac{x_{j\max}^i - x_{j\min}^i}{2}, \text{ for all } i \text{ and } j, \qquad (10)$$

where $x_{j\min}^i$ and $x_{j\max}^i$ are the values of the left edge and the right edge in the x_j-direction, respectively. The width of the membership function is given by

$$\sigma_j^i = \frac{x_{j\max}^i - c_j^i}{[\ln(1/\alpha)]^{1/2}}, \text{ for } i = 1, \ldots, p \text{ and } j = 1, \ldots, n \quad (11)$$

where α is the value of the α-cut fuzzy set which is used for constructing the hyper-rectangle. The consequent parameters associated with the hyper-rectangles are also

initialized by using the centers of the output fuzzy clusters:

$$q^i = v_j, \text{ if } H_i \text{ is the } P_j \text{ labeled hyper - rectangle} \quad (12)$$

We perform the parameter identification with the initial parameters determined as above. If the identified model is accurate enough, we stop the modeling process. Otherwise, we increment p and repeat the GA cycle

4. An Example

We apply the proposed input space partitioning method to a nonlinear dynamical process modeling using the gas furnace data of Box and Jenkins [3]. The data set consists of 296 pairs of input and output measurements. The input $u(k)$ is the gas flow rate into the furnace and the output $y(k)$ is the concentration of CO_2 in the outlet gas. The sampling interval is 9 seconds.

Since we are interested in the problem of input space partitioning in this paper, we assume that $u(k\text{-}4)$ and $y(k\text{-}1)$ are the input variables and the structure of the process is as follows:

$$y(k) = F\big(u(k-4), y(k-1)\big) \quad (13)$$

In this structure, we could generate 292 training samples and performed the fuzzy clustering on the output training data, while setting m to 2 in the FCM algorithm. After clustering, we projected the output clusters into the input space and constructed the cell map.

To run each GA cycle, we set the number of strings in a population to 200. For all the GA cycles, the steady state reproduction without duplicates with 1.0 as the crossover rate and 0.1 as the mutation rate was used, since all the initial populations for each c and p satisfied the property discussed in the knowledge based initialization scheme. We terminated the GA cycle after 20,000 generations. Each GA cycle for a particular c and p took less than five minutes on a Sun Sparc 20 workstation. For illustration purpose, the results of partitioning for c=4 and p= 8 is shown in Fig. 4.

After a GA cycle, we initialized the parameters of the neuro-fuzzy model using the result of partitioning. With the initial parameters, we identified the parameters of the model. The mean square error (MSE) was used for measuring the accuracy of the model. The identification results are listed in Table 1. If the accuracy requirement is that MSE is better than 0.15, then the modeling procedure stops at c=2 and p=3. If it is 0.11, then the neuro-fuzzy model of only 8 fuzzy rules is enough (See c=4 and p=8).

Table 1. Accuracies of the identified models for c's and p's.

	p=2	p=3	P=4	p=5	p=6	p=7	p=8
c=2	0.16	0.13	–	–	–	–	–
c=3	–	0.14	0.13	0.12	0.12	–	–
c=4	–	–	0.13	0.12	0.12	0.11	–
c=5	–	–	–	0.12	0.11	0.11	0.11

In Table 2, we compare the neuro-fuzzy models identified by our method with other fuzzy models identified from the same data. It can be seen from this table that a considerable improvement in terms of the model efficiency and the model accuracy can be achieved by using the proposed method.

Table 2. Comparison of our model with other fuzzy models.

Model names	No. of rules	MSE
Tong's model	19	0.46
Xu's model	25	0.32
Wang's model	5	0.15
Kim's model	11	0.10
Our model (c=4, p=8)	8	0.10
Our model (c=5,p=12)	12	0.09

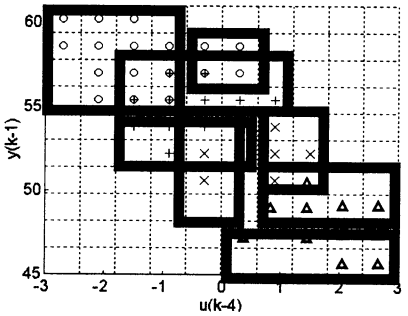


Fig. 4. Result of input space partition.

5. Conclusion

In this paper, we proposed an input space partitioning method for structure identification of neuro-fuzzy modeling using genetic algorithms, which guided us to build a better model in terms of the model's efficiency and the model's accuracy. Extensive simulations in Section 4 revealed that the proposed method could achieve a considerable improvement compared with other modeling approaches. If a number of fuzzy rules were given, the proposed method provided us with a set of initial parameters of a model which eventually converged to a more consistent model of a better accuracy than a model identified without a similar such structure determination. Furthermore, if an accuracy requirement was given, the method could suggest a model of a least number of rules which were less inconsistent with each other in order to meet the given requirement. As a consequence, we can decrease the number of fuzzy rules and thus increase the learning/operational efficiency through our proposed approach.

Acknowledgements

This work was supported by grant no. R01-2000-000-00277-0 from the Basic Research Program of the Korean Science & Engineering Foundation (KOSEF).

References

[1] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York: Plenum Press, 1981.

[2] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control, San Francisco: Holden Day, 1976.

[3] M. Sugeno and T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems 1 (1993) 7-31.

[4] C. T. Sun, Rule-based structure identification in an adaptive-network-based fuzzy inference system, IEEE Trans. Fuzzy Systems, 2 (1994) 7-31.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Complex behavior of a simple partial discharge model

Hideyuki Suzuki[1], Kazuyuki Aihara[2] and Shunji Ito[3]

1. Department of Mathematical Informatics, Graduate School of Information Science and Technology,
The University of Tokyo, 7–3–1 Hongo, Bunkyo-ku, Tokyo 113–8656, Japan
hideyuki@sat.t.u-tokyo.ac.jp
2. Department of Complexity Science and Engineering, Graduate School of Frontier Sciences,
The University of Tokyo, 7–3–1 Hongo, Bunkyo-ku, Tokyo 113–8656, Japan
aihara@sat.t.u-tokyo.ac.jp
3. Department of Information and Systems Engineering, Faculty of Engineering,
Kanazawa University, 2–40–20, Kodatsuno, Kanazawa, Ishikawa 920–8667, Japan
ito@t.kanazawa-u.ac.jp

Abstract

The present paper examines complex behavior of a simple partial discharge model, which is called the equivalent circuit model of partial discharges. Although it is an old model, its behavior has not been well investigated for a long time. Here we show that the average discharge rate as a function of the applied voltage has a characteristic like a devil's staircase.

1 Introduction

An electrical discharge is called a partial discharge, if it is localized or confined within only part of the insulation between two separated conductors. A partial discharge occurs when the local electrical field within the part of the insulation exceeds a critical value, while it is surrounded by insulation that is strong enough to avoid a complete breakdown.

In the real world, partial discharges are caused by existence of local defects in insulation, for instance. Since partial discharges in voids of insulation destruct the insulation gradually, they are considered to be harmful especially in high voltage electrical systems. Consequently, understanding of partial discharges is important.

Behavior of partial discharges is very complicated, and the cause of the complexity is mainly considered to be various probabilistic factors. However, deterministic characteristics of the phenomena cannot be ignored, as shown in the several studies on the deterministic aspects of partial discharges [1–9].

For understanding of the deterministic aspects, investigation of deterministic partial discharge models is necessary. Nevertheless, there has been no detailed mathematical study on deterministic models.

In the present paper, as a first step, we will investigate the simplest and deterministic model of partial discharges, which is called the equivalent circuit model of partial discharges [10, 11].

2 Equivalent circuit model

The equivalent circuit model [10, 11] is the simplest model of partial discharges, which was proposed more than fifty years ago. However, its mathematical characteristics has not been well investigated.

As shown in Figure 1, the equivalent circuit consists of three capacitors and a discharge gap. C_g represents the capacitance of the region where partial discharges occur, and C_b and C_a represents the capacitances of the insulation in parallel and in series with C_g.

The gap G is the element such that, when the potential difference between the gap reaches a certain *discharge voltage* h_+, a discharge occurs and the difference between the gap is reduced from h_+ to a certain *residual voltage* b_+ by the compensation caused by the discharge. Discharges occur also in the opposite direction; when the potential difference between the gap reaches h_-, the difference is reduced to b_- by a discharge in the $-$ direction.

Applied voltage u is defined as the potential difference between the gap, provided that no discharges are assumed to occur between the gap. When sinusoidal AC voltage is applied, $u(t)$ is also sinusoidal. We assume $u(t)$ is sinusoidal and the amplitude of $u(t)$ is V.

The actual potential difference between the gap $v(t)$

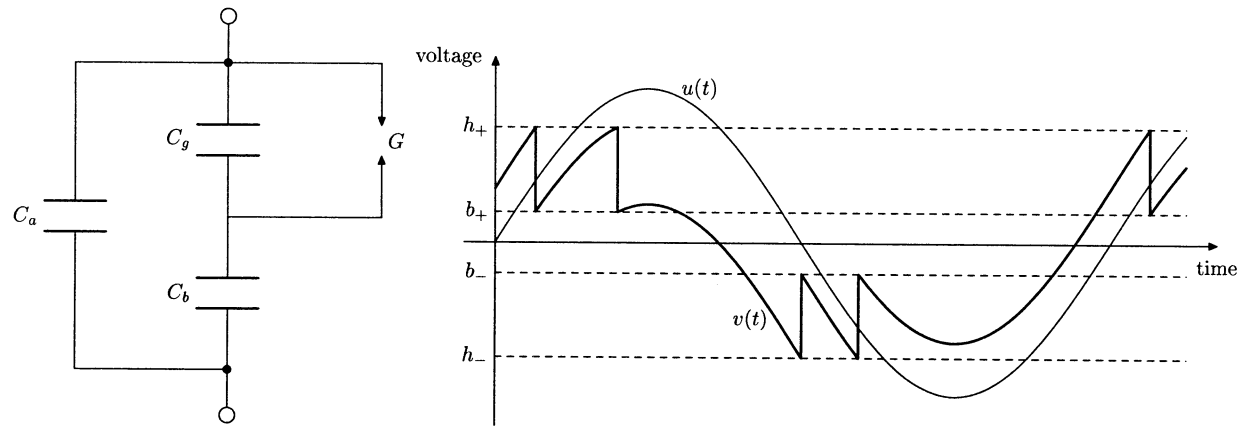


Figure 1: Equivalent circuit model

is canceled by partial discharges, when $v(t)$ reaches the discharge voltage $h_\pm$. $v(t)$ varies in the way as shown in Figure 1. Note that the difference between $v(t)$ and $u(t)$ is constant between discharges.

As a model of real phenomena, the parameters $h_\pm$ and $b_\pm$ cannot be considered as constants. In fact, fluctuations of these parameters play an important role in the behavior of partial discharges.

However, in the present paper, we assume that parameters $h_\pm$ and $b_\pm$ are constants, for investigation of fundamental behavior of the equivalent circuit model.

Under the assumption, the behavior of the model is completely deterministic, and only depends on the initial state of the model. We will investigate the details of this dynamical system in the following sections.

3 Dynamical system of equivalent circuit model

As we have assumed, let $u(t) = V \sin \omega t$. Note that $+$ and $-$ discharges occur in the region such that $du/dt \geq 0$ and $du/dt \leq 0$, respectively.

Assume that a $+$ discharge occurs at the time t_0. Then,

- if $u(t_0) - b_+ \leq V - h_+$, the next discharge occurs in the $+$ direction at the time t_1 such that $h_+ = b_+ + u(t_1) - u(t_0)$, and,
- if $u(t_0) - b_+ > V - h_+$, the next discharge occurs in the $-$ direction at the time t_1 such that $h_- = b_+ + u(t_1) - u(t_0)$.

In this way, we can determine the value of $u(t_1)$ from $u(t_0)$. Similarly, the value of $u(t_1)$ can be determined from $u(t_0)$, if the first discharge is in the $-$ direction.

This map can be expressed in the following way. Let $(v, +)$ and $(v, -)$ denote the $+$ and $-$ discharges, respectively, at the time t such that $u(t) = v$. Then the next discharge is $g(v, +)$ and $g(v, -)$, respectively, where g is the map on $[-V, V] \times \{+, -\}$ defined by

$$g(v,+) = \begin{cases} (v + h_+ - b_+, +) & \text{if } v + h_+ - b_+ \leq V, \\ (v + h_- - b_+, -) & \text{if } v + h_+ - b_+ > V, \end{cases}$$

$$g(v,-) = \begin{cases} (v + h_- - b_-, -) & \text{if } v + h_- - b_- \geq -V, \\ (v + h_+ - b_-, +) & \text{if } v + h_- - b_- < -V. \end{cases}$$

Since the coefficient of v is 1 in all cases, behavior of the system is basically neutral in the sense that fluctuations in the system basically remains. However, dependencies on initial values may be sensitive because of the existence of the discontinuity in the map.

Note that the map g corresponds to the idea of PSA proposed by Hoof and Patsch [1], because the physical meaning of v is the applied voltage when a discharge occurs.

4 First return map

Let E_+ and E_- be

$$E_+ = (V - h_+ + b_+, V] \times \{+\},$$
$$E_- = [-V, -V - h_- + b_-) \times \{-\}.$$

We can assume $h_+ - b_+ \leq b_- - h_-$ without any loss of generality, because, if the assumption does not hold, we can exchange the meaning of $+$ and $-$. In this section, we will consider the first return map of g on E_+.

Then every state in $[-V,V]\times\{+\}$ will be mapped into E_+ by g in a finite time; for $v\in[-V,V]$,

$$g^{\lfloor (v-V)/(b_+-h_+)\rfloor}(v,+)\in E_+.$$

Similarly, every state in $[-V,V]\times\{-\}$ will be mapped into E_- by g in a finite time; for $v\in[-V,V]$,

$$g^{\lfloor (v+V)/(b_--h_-)\rfloor}(v,-)\in E_-.$$

Note that $g(E_+)\subset[-V,V]\times\{-\}$ and that $g(E_-)\subset[-V,V]\times\{+\}$. This means that every orbit of g visits E_+ and E_- alternately.

For a point $(v,+)\in E_+$, consider the orbit that starts from $(v,+)$, and let $g_+(v,+)\in E_-$ be the state at the first visit of the orbit to E_-. Specifically, $g_+ : E_+\to E_-$ is defined as

$$\begin{aligned} g_+(v) &= g^{\lfloor (v+h_--b_++V)/(b_--h_-)\rfloor+1}(v)\\ &= (v+h_--b_++V) \bmod (b_--h_-)-V, \end{aligned}$$

where $x \bmod y = x-y\lfloor x/y\rfloor$ and trivial signs $+$ and $-$ are omitted. From the assumption that $h_+-b_+\le b_--h_-$,

$$g_+(v)=\begin{cases} v-C-V+b_--h_- & \text{if } v<C,\\ v-C-V & \text{if } v\ge C,\end{cases}$$

where $C=V-(2V+h_--b_+) \bmod (b_--h_-)$.

Similarly, for $(v,+)\in E_+$, define $g_- : E_-\to E_+$ by

$$\begin{aligned} g_-(v) &= g^{\lfloor (v+h_+-b_--V)/(b_+-h_+)\rfloor+1}(v)\\ &= (v+h_+-b_--V) \bmod (b_+-h_+)+V. \end{aligned}$$

Then, the first return map of g on E_+ is $g_-\circ g_+$. Therefore,

$$\begin{aligned} g|_{E_+}(v) &= g_-\circ g_+\\ &= V+\begin{cases}(v-V-B) \bmod (b_+-h_+) & \text{if } v<C,\\ (v-V-A) \bmod (b_+-h_+) & \text{if } v\ge C,\end{cases} \end{aligned}$$

where $A=V+C-h_++b_-$ and $B=V+C-h_++h_-$.

By the map $h(v)=(V-v)/(h_+-b_+)$, the first return map $g|_{E_+}$ is conjugate with the following map:

$$h\circ g|_{E_+}\circ h^{-1}(x)=\begin{cases}\{x+\alpha\} & \text{if } x\in[0,c],\\ \{x+\beta\} & \text{if } x\in(c,1),\end{cases} \qquad (1)$$

where $x=h(v)$, $c=h(C)$, $\alpha=A/(h_+-b_+)$, $\beta=B/(h_+-b_+)$, and $\{x\}$ denotes $x-\lfloor x\rfloor$.

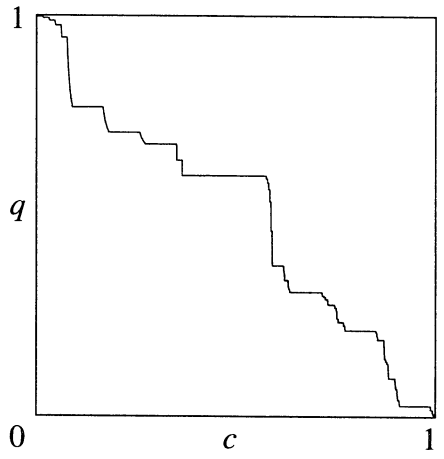


Figure 2: Discharge number

5 Double rotation

We have shown in the previous section that any equivalent circuit model can be reduced to the map

$$f_{(\alpha,\beta,c)}(x)=\begin{cases}\{x+\alpha\} & \text{if } x\in[0,c),\\ \{x+\beta\} & \text{if } x\in[c,1),\end{cases} \qquad (2)$$

where $(\alpha,\beta,c)\in[0,1)\times[0,1)\times[0,1]$. We call the map $f_{(\alpha,\beta,c)}$ a *double rotation.*

Note that there is a difference between the equations (1) and (2) at the point c, but this difference does not make any difference in global behavior of the model. This modification is made only for simplicity of the map.

Let the *discharge number* $q_{(\alpha,\beta,c)}(x)$ of a double rotation $f_{(\alpha,\beta,c)}$ for $x\in[0,1)$ be

$$q_{(\alpha,\beta,c)}(x)=\lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\chi_{[c,1)}(f^i_{(\alpha,\beta,c)}(x)),$$

if the limit exists. It can be shown that, for almost every (α,β,c), a limit exists and the value is independent of x.

A graph of $q_{(\alpha,\beta,c)}$ as a function of c is shown in Figure 2, where α and β are fixed. The graph is very complicated and like a devil's stair case.

In fact, this complex behavior of double rotations is general; it can be shown that, if α and β are linearly independent on $\mathbb{Q}$, there exists a measure-zero Cantor set Γ such that the discharge number as a function of c is (i) constant in every connected interval in $[0,1]\setminus\Gamma$ and (ii) monotonically non-increasing (Suzuki et al., in preparation).

6 Complex behavior of the equivalent circuit model

This complex behavior of double rotations also produces complex behavior in the equivalent circuit model. We will show in the following, that the discharge number of a double rotation is related to the average discharge rate of the equivalent circuit model.

Let $K = \lfloor (2V + h_- - b_+)/(b_- - h_-) \rfloor$. Then

$$C = -V - h_- + b_+ + K(b_- - h_-).$$

Hence,

$$\alpha = \frac{(K+1)(b_- - h_-)}{h_+ - b_+} - 1,$$
$$\beta = \frac{K(b_- - h_-)}{h_+ - b_+} - 1,$$
$$c = \frac{(2V + h_- - b_+) \bmod (b_- - h_-)}{h_+ - b_+}.$$

Therefore, when the value of V is changed, as far as K is constant, both α and β are constant, and only c depends linearly on V. This means that c corresponds to the amplitude of the applied voltage to the model.

Also note that, states in $[0, c)$ and $[c, 1)$ represent occurrence of $K + 1$ and K successive discharges in the $-$ direction, respectively. Therefore, if an equivalent circuit model is reduced to $f_{(\alpha,\beta,c)}$, the average $-$ discharge rate of the model is

$$\rho_- = K + 1 - q_{(\alpha,\beta,c)}.$$

Then the average $+$ discharge rate has to be

$$\rho_+ = \frac{b_- - h_-}{h_+ - b_+}\rho_-.$$

Consequently, the average discharge rate of the equivalent circuit model as a function of the amplitude of the applied voltage is like a devil's staircase as shown in Figure 2.

7 Conclusion

We have shown that the behavior of the equivalent circuit model is very complicated, even if it is a very simple and deterministic model.

Analysis of more realistic partial discharge models is an important future problem.

References

[1] Hoof M, Patsch R (1995), Pulse-Sequence Analysis: a new method for investigating the physics of PD-induced ageing. IEE Proc. Sci. Meas. Tech. 142:95–101.

[2] Hoof M, Freisleben B, Patsch R (1997), PD source identification with novel discharge parameters using counterpropagation neural networks. IEEE Trans. Dielectr. Electr. Insul. 4:17–32.

[3] Patsch R, Hoof M (1998), Physical modeling of partial discharge patterns. *Proc. 1998 IEEE International Conference on Conduction and Breakdown in Solid Dielectrics, June 1998, Västerås, Sweden*, pp. 114–118.

[4] Freisleben B, Hoof M, Patsch R (1998), Using counterpropagation neural networks for partial discharge diagnosis. Neural Computing & Applications 7:318–333.

[5] Suzuki H, Kimoto S, Mizukami Y, Okamoto T, Aihara K (1999), Stationary and deterministic analysis of partial discharge interpulse intervals. *Proc. 13th International Conference on Dielectric Liquids, Nara, Japan, July 1999*, pp. 353–356.

[6] Mizukami Y, Inoue T, Okamoto T, Aihara K (1999), Nonlinear dynamical study of partial discharge phenomena. *Proc. 13th International Conference on Dielectric Liquids, Nara, Japan, July 1999*, pp. 513–516.

[7] Lim YS, Kim SH, Ko JH, Park JJ, Kim JH (1999), Identification of chaotic characteristics in partial discharge. *Proc. 13th International Conference on Dielectric Liquids, Nara, Japan, July 1999*, pp. 556–559.

[8] Pawlowski T, Czaszejko T (1999), State-space analysis of partial discharge process. *Proc. 1999 Annual Report Conference on Electrical Insulation and Dielectric Phenomena, October 1999*, pp. 249–252.

[9] Suzuki H (2000), Deterministic analysis on electrical systems with thresholds. Ph.D. thesis, The Univ. of Tokyo.

[10] Whitehead S (1953), Dielectric Breakdown of Solids. Clarendon Press, Oxford.

[11] Austen AEW, Whitehead S (1941), Discharges in insulation under alternating-current stresses. J. Inst. Electr. Eng., 88(ii):88–92.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Dimensional analysis of the Hodgikin-Huxley equations with noise:

Effects of noise on chaotic neurodynamics

Hiroaki Tanaka and Kazuyuki Aihara

Department of Mathematical Engineering and Information Physics
The Univ. of Tokyo Bunkyo-ku, 113-8656, Japan

Abstract

The chaotic behavior of Hodgikin-Huxley equations modulated by noise [1] is studied with correlation dimensions, information entropies, and power spectra with changing noise components.

It is found that the noise component of compressing dynamics (CDN) increases the dimension up to the degree of freedom of the system (=4), when the bin size is reduced to the same order with the noise amplitude. On the contrary to this, the noise component of extending dynamics (EDN) increases the dimension only up to 2.5-3.0, with less steep dependence.

From the analysis of information entropy for those components, both CDN and EDN produce the noise induced order (NIO), though EDN's effect is a little bit larger than CDN's effect.

From the analysis of power spectra, there are two major peaks at 105 and 115Hz in the low frequency area. CDN enhances the 105Hz component with increasing its frequency. EDN enhances the 105Hz component only at small amplitude as 10^{-4}.

Comparing the effect on noise imposed on the variables of the Hodgikin-Huxley equations, the effect of noise components imposed on v and m seems as CDN, while those of h and n as EDN.

These suggest that (1) a fluctuation induced by EDN is restricted within some parts of the strange attractor, while (2) that by CDN shakes out the attractor from it. The shaking-out mechanism may have a significant role on the real neurodynamics.

Key words: Hodgikin-Huxley equations, noise, dimension, information entropy, noise-induced order

Introduction

The squid giant axon shows chaotic behavior under certain sinusoidal current stimulation [2,3,4]. The chaos in the neuron was well simulated by the numerical calculation of the Hodgikin-Huxley equations [5]. We found that the chaotic behavior of the Hodgikin-Huxley equations was modulated by small white noise [1] with two different mechanisms [6]. In addition, we studied the correlation dimensions and the information entropy at various bin sizes and found that (1) it takes about 30 times larger scale to compress the noise fluctuation and also (2) the noise induces order [7] when the box size is 1/30~1/3 of the phase space size[8].

Considering about the mechanism of chaotic dynamics, fluctuations imposed on the chaos trajectory may have two different effects. One is the fluctuation within the strange attractor unstable manifold and the other is that perpendicular to it. It seems adequate to decompose the noise into those two components and to impose either of them to elucidate this problem.

However, in the actual calculation, this decomposition is not easy at every iteration step. Thus in this study, we separate the noise component in a different way. That is, the dynamics of the variables are tested at every step by applying small perturbations to them and checking whether they are extended or compressed by the dynamics in the phase space. The noise components imposed on the variables are classified as extending dynamics noise (EDN) and compressing one (CDN) (Fig.1). Their effects on the correlation dimensions, the information entropies, and power spectra are analyzed.

Simulation

Detail methods and parametric conditions for the numerical calculation are based on Usami et.al. [5]. Gaussian noise is superimposed on the variables of the Hodgikin-Huxley equations, v, m, h and n, with certain normalization, respectively. The noise amplitude of 1.0 is adjusted to the level that the deviation of the noise is almost the same as the phase space size.

During the iterations of the calculations, small perturbation is imposed on the variables to test the sign of second derivative at every calculation step. In the case of EDN, the noise is added only to the variables with positive second derivative. In the case of CDN, it is added to the other variables.

In the analysis of the correlation dimensions and the information entropies, time series of 10,000 Poincare section data points at 240 degrees of the sinusoidal current phase are used [8]. For the power spectra, time series of 65,526 data points (0.04ms resolution) of v is applied to FFT program (8192 data points, 1.526Hz resolution).

Results

The slope of the relation between the correlation integration bin count versus the bin size in the logarithmic scale, gives the local correlation dimension (Figs. 1 and 2).

It is found that, if CDN is applied, the dimension increases at a smaller bin size region. The point of the bin size where local dimension starts to increase in the logarithmic scale is about -0.5, -1.0, -1.5, -2.0 and -2.5 for the noise amplitudes (logarithmic scale) of -2.0, -2.5, -3.0, -3.5 and -4.0, respectively. These show that the point is about 30 times larger than the noise amplitude. The maximum value of the local dimension is about 4.0, which is same to the noise dimension. This relation is almost same to the homogeneous noise [8].

If EDN is applied, the point of the bin size where local dimension starts to increase in the logarithmic scale is same to CDN. But the slope is less steep and the maximum value of the local dimension is 2.5-3.0

The box counting method is applied for calculate the information entropy. The excess entropy induced by noise shows that there is the noise induced order (NIO) [7] when the box size is about -1.5 to -0.5 in logarithmic scale (Fig. 4 and 5). Comparing Figs. 4 and 5, both EDN and CDN produce NIO, although EDN produces it slightly larger than CDN.

In the case of without noise, power spectra shows that there are two major peaks, 105 and 115Hz in the low frequency area. CDN enhances the 105Hz component with increasing its frequency (Fig. 6). On the contrary to this, EDN enhances 105Hz component at smaller amplitude as 10^{-4} (Fig. 7), although the spectra becomes broad with larger EDN.

To examine the neurodynamics, effects of noise on the variables, v, m, h and n, of the Hodgikin-Huxley equations, are analyzed by local correlation dimension versus the bin size (Fig. 8). It is found that the maximum values of the local dimension are about 4.0 for v and m, with steeper bin size dependence, while those are about 2.5-3.0 for h and m.

Discussion

The maximum local dimension of CDN, which is reached at the region when the bin size is smaller than the noise amplitude, is almost same to the system freedom (=4). This suggests that CDN shakes the strange attractor orbits out of its unstable manifold.

In the case of EDN, the maximum dimension is significantly smaller than the case of CDN. This smaller dimension seems to be related to the intrinsic character of the strange attractor. EDN fluctuate the strange attractor orbits within its unstable manifold. This may be related to the fact that NIO is more enhanced with EDN than CDN.

From the observation of the time course of v, the average frequency of major spike peaks of v is almost 105Hz. This frequency becomes a little bit increased to 115Hz and becomes significantly regular by applying larger noise [6]. The shift and enhancement of 105Hz peak by CDN seems to be consistent to this. If large CDN induces much regular spikes, it may be said that CDN shakes out the attractor from its unstable manifold to much stable orbit.

Comparing the effect of noise imposed on the variables of Hodgikin-Huxley equations, the effects of noise imposed on v and m are similar to CDN, while those of h

and n are as EDN. In the actual neuron in the brain, v may be mostly exposed to the noise. Thus, the CDN effect of noise imposed on v may have a significant role on the real neurodynamics.

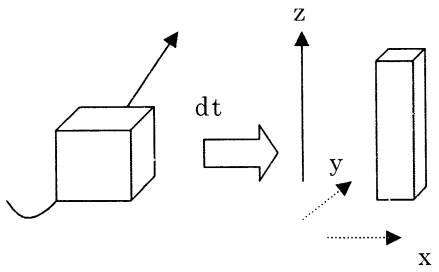


Fig. 1. Concept of the noise component. As a nature of chaos, small perturbations on the variables are extended or compressed according to the system dynamics in the phase space. The noise components imposed on the variables can be classified as extending dynamics noise (EDN), such as z, and compressing one (CDN), such as x and y.

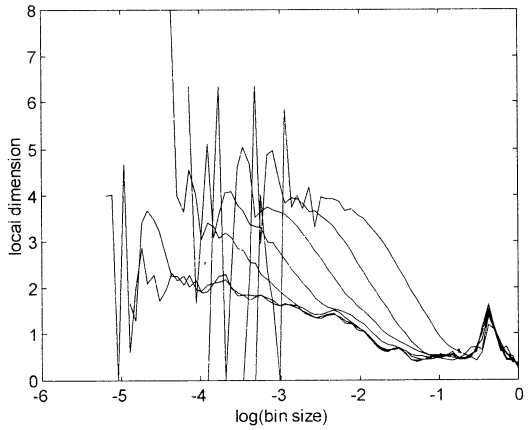


Fig. 2. The local correlation dimension versus the box size with CDN. The slope of the logarithmic relation between the bin count of the correlation length versus the bin size gives the local correlation dimension at the certain bin size. The curves from right to the left are the results of CDN amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4}, 10^{-6} and 0, respectively.

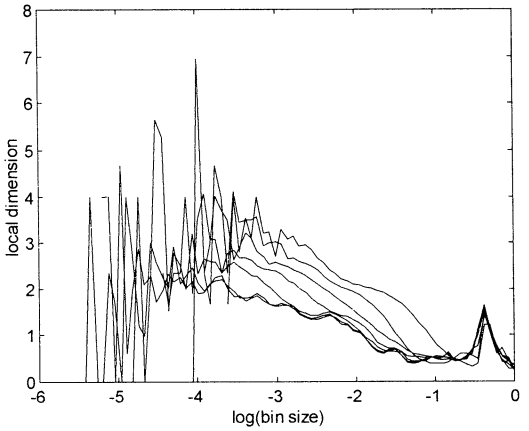


Fig. 3. The local correlation dimension versus the box size with EDN. Other details are same as Fig. 2.

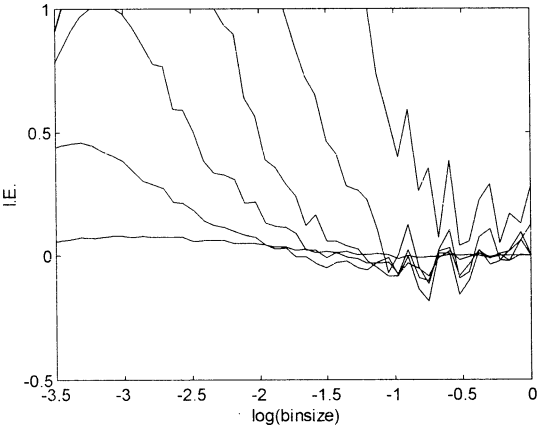


Fig. 4. The excess information entropy induced by CDN than without noise case. The curves from right to the left are the results of noise amplitudes of 10^{-2}, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$, 10^{-4} and 10^{-6}, respectively.

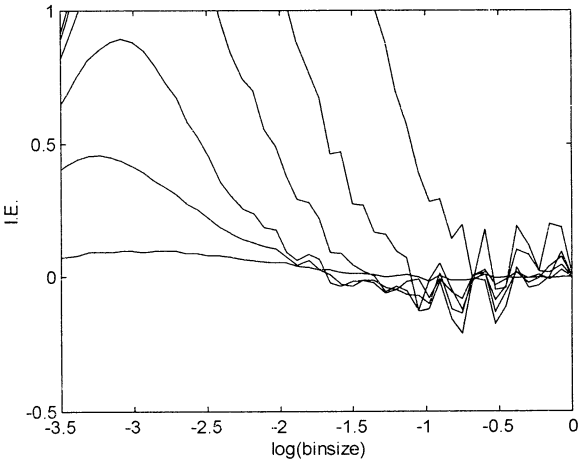


Fig. 5. The excess information entropy by EDN. Other details are same as Fig. 4.

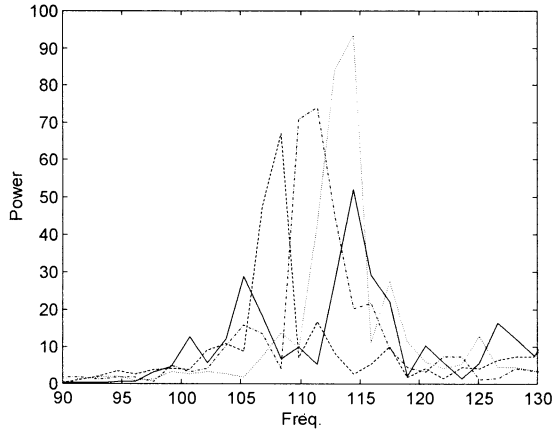


Fig. 6. The power spectra with CDN. The curves of dotted line, dash-dot line, dashed line and solid line are the results of CDN applied in amplitudes of 10^{-2}, 10^{-3}, 10^{-4} and without noise, respectively.

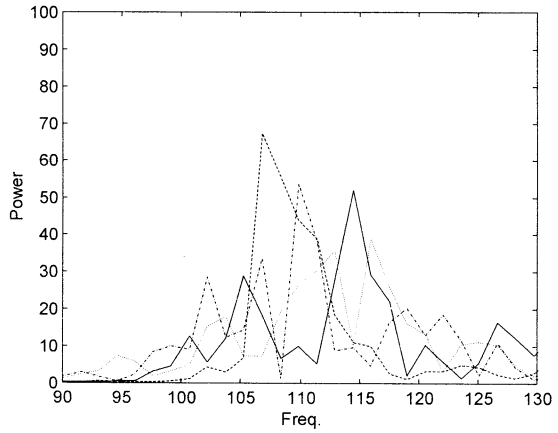


Fig. 7. The power spectra with CDN. Other details are same as Fig. 6.

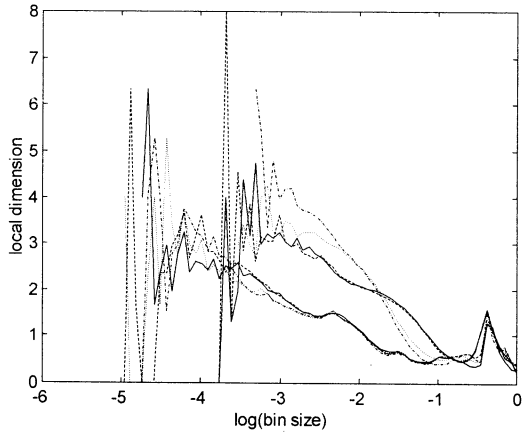


Fig. 8. The local correlation dimension versus the box size. Noises are imposed on the variables, v, m, h and n of the Hodgikin-Huxley equations. The curves of dotted line, dash-dot line, dashed line and solid line are the results of the noise imposed on v, m, h and n, respectively. The curves from right side group and left side one are the results of the noise applied in amplitudes of 10^{-2} and 10^{-4}, respectively.

Conclusion

It is found that that (1) a fluctuation induced by EDN is restricted within some parts of the strange attractor, while (2) that by CDN shakes out the attractor from it. This shaking-out mechanism may have a significant role on the real neurodynamics.

References

[1] Tanaka H, Aihara K (1997), Chaotic behavior of the Hodgikin-Huxley equations under small random noise. Complexity and Diversity. Springer-Verlag, Tokyo, pp.172-174

[2] Aihara K, Matsumoto G, Ikegaya Y (1984), Periodic and non-periodic responses of a periodically forced Hodgkin-Huxley oscillator. J. theor. Biol. 109:249-269

[3] Aihara K, Matsumoto G, Ichikawa M (1985), An alternating periodic-chaotic sequence observed in neural oscillators. Phys. Lett. A 111:251-255

[4] Aihara K, Numajiri T, Matsumoto G, Kotani M (1986), Structures of attractors in periodically forced neural oscillators. Phys. Lett. A 116:313-317

[5] Usami T, Yamada T, Ichinose N, Aihara K(1995), Hodgikin-Huxley equation and its response to the periodic stimulation. J.SICE. 34:769-774

[6] Tanaka H, Aihara K (2000), Effect of small random noise on the chaotic behavior of the Hodgikin-Huxley equations. Proc. of The 5th Int. Symp. on Artificial Life and Robotics. pp. 165-168

[7] Matsumoto K, Tsuda I (1983), Noise-induced order. J. Stat. Phys. 31:87-106

[8] Tanaka H, Aihara K (2002), Dimension analysis of the Hodgikin-Huxley equations with noise: Effect of random noise on chaos. Proc. of The 7th Int. Symp. on Artificial Life and Robotics. pp. 108-111

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Encoding Ternary Information using a Chaotic Neural Network

J. K. Ryeu
Dept. of Electronic Engineering
Dongyang University
Youngju, Korea, 750-711
e-mail: jkryeu@phenix.dyu.ac.kr

Abstract

We analyzed a model of a chaotic neural network consisting of a chaotically forcing neuron and two stable neurons in the previous study. In the study, we showed that the dynamics of the chaotically forcing neuron is embedded in the form of a code sequence on a fractal attractor of the two-neuron response system. As an engineering application on the information processing, we show that a desirable message can be encoded into an attractor space by using an chaotic neural network in the present study.

1 introduction

Many studies on the relations between chaos and fractals [4] [13] [6] have been done in recent years. Rössler et al. [9] reported some models representing chaos–driven contraction mapping. Their paper describes a hierarchy of models exhibiting fractal attractors including strange nonchaotic attractors found by Grebogi et al. and singular–continuous nowhere–differentiable(SCND) attractors, too. Tsuda [11] have also found the SCND attractor in a neural system consisting of chaotic neuron models proposed by Aihara et al. [1]. On the other hand, a lot of research on the use of chaos for nonlinear digital communications, especially for the encoding of digital information, has been reported [5] [3] [7]. Based on this research, we can expect that if a dynamical system has well–defined symbolic dynamics, the encoding of digital information is accomplished using the principle of controlling chaos [8]. Futhermore it has been showed by Hayes et al. [5] that a chaotic system can be manipulated to generate controlled chaotic time series whose symbolic representation corresponds to the digital information via arbitrarily small time–dependent perturbations.

The purpose of the present paper is to further develop fractal symbolic encoding in a chaotically–forced contraction system that exhibits fractal attractor. We generate a chaotic time series corresponds to a ternary desirable message by applying proper perturbations to the initial values of a chaotic neuron at each time [7]. At this time, we use two peaks of the return map of a chaotic neuron to assign a symbolic representation to the signal. And then, we observe a 2-dim. fractal attractor of two almost linear neurons forced by the chaotic orbits obtained by the time series for the desiarable message. To clarify the encoding property, we introduce hard–limit functions, or Heaviside functions as transfer functions from the forcing neuron to the response system, thereby the system is converted to an IFS (Iterated Function System)–like model which is composed of not affine but rather nonlinear transforms. According to Barnsley [2], if the IFS is totally disconnected and if the points on the attractor are distributed sparsely, it is possible to improve memories with a very high storage capacity and robustness against noise. Although the transfomation of the proposed system is nonlinear rather than affine, and it may not be completely invertible, a kind of coding of information may also be possible [11] [12].

In Section 2, we present the encoding of desired code sequences using a chaotic neuron model [1]. In Section 3, we investigate the relation between the structure of the code and the hierarchical structure of a fractal attractor for the chaotic orbits obtained in Sec. 2, and Section 4 is devoted to the discussion on the meaning of fractal encoding on the attractor and the possibility of its application to information processing.

2 Encoding code sequences using a chaotic neuron

2.1 Chaotic neuron model

We use a chaotic neuron model proposed by Aihara [1] to encode the desired code sequence into a trajectory of the chaotic neuron. The equation of the chaotic neuron is as follows:

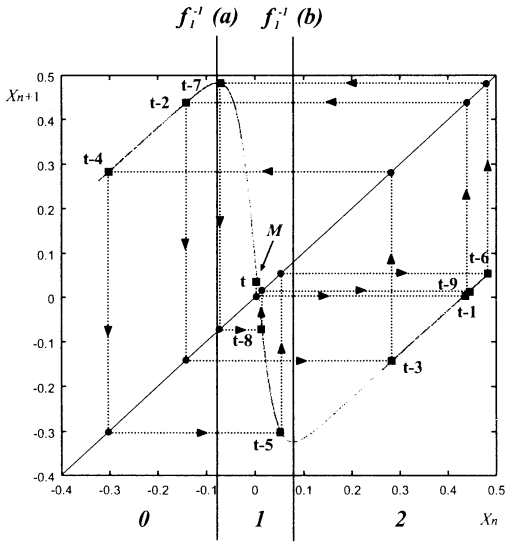


Figure 1: Definition of a code sequence based on the chaotic neuron map.

$$x_{n+1} = f_1(-\alpha_1 \sum_{r=0}^{n} k_1{}^r x_{n-r} + I_0) \tag{1}$$

where α_1 is a positive parameter, I_0 is the strength of the external input to the neuron x, k_1 is a decay parameter with $0 < k_1 < 1$, and the function $f_1(x)$ is the following sigmoidal function:

$$f_1(x) = \frac{1}{1 + e^{-x/\epsilon_1}}, \tag{2}$$

where ϵ_1 is a steepness parameter. We can represent the dynamics of the chaotic neuron as follows by defining new variables X_{n+1}:

$$X_{n+1} = k_1 X_n - \alpha_1 f_1(X_n) + I \tag{3}$$

where $I = I_0(1 - k_1)$ which is used to control chaotic dynamics of the neuron X.

Fig. 1 shows a chaotic neuron map and an example of generating a code sequence.

2.2 Encoding code sequences

We show an example of encoding digitized information into a trajectory of a chaotic neuron map. We use the algorithm of Lai [7] for the logistic map. The steps of encoding ternary codes are as follows.

1. Choose an initial value.
2. Determine m symbols corresponding to m points on the trajectory starting from the initial value x_0.
3. Examine whether the mth symbol is identical to the first message bit.
   - If so, iterate the process from x_0 to obtain x_1 and determine the $(m+1)$th symbol from x_0. Then examine whether it is identical to the second message bit.
   - Otherwise, apply some perturbation to x_0 for the mth symbol from x_0 to make accordance with the first message bit.
4. Continue this procedure by the time that all message bits are encoded into a chaotic trajectory.

The coding function [5] is used for calculating perturbations. The procedure obtaining the coding function is as follows [7]. At first, we divide the unit interval in x into N bins of size $\delta x = 1/N$, where $\delta x << 1/3^m$ and $1/3^m$ is the maximally allowed perturbation.Then we determine the symbol sequence of length $m : a_1 a_2 a_m \cdots (a_i \in 0, 1, 2)$ by choosing a point from each bin and performing m iterations. The symbolic value R can be calculated as follows:

$$R = \sum_{i=1}^{m} a_i/3^i, a \leq R \leq 1. \tag{4}$$

Fig. 2 shows the coding function for a chaotic neuron map.

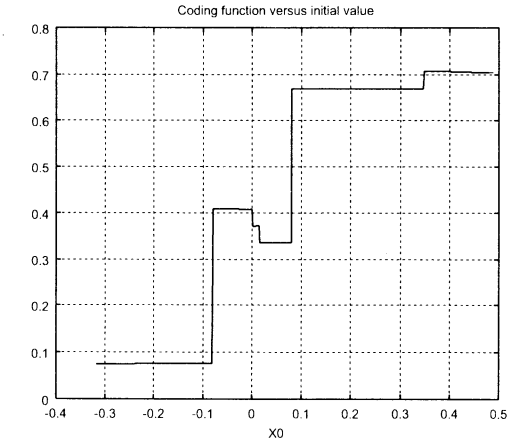


Figure 2: The coding function for the chaotic neuron map at $k = 0.98, \epsilon = 0.02$.

The values of perturnations are determined by the coding function as following procedures.

1. Let the m-bit symbol sequence a_m generated from x_0 be $a_1a_2\cdots a_{m-1}a_m$ and let the first message bit to be encoded be b_1.
2. Calculate $\delta R = (a_m - b_1)/3^m$ by comparing generated symbol sequence $a_1a_2\cdots a_{m-1}a_m$ with the desirable symbol sequence $a_1a_2\cdots a_{m-1}b_1$.
3. Compute the perturbation δx from the coding function $R(x)$.

An example of encoding the sequence "21011202 10202102 11212020 20111201" into a chaotic trajectory is shown in Fig. 3.

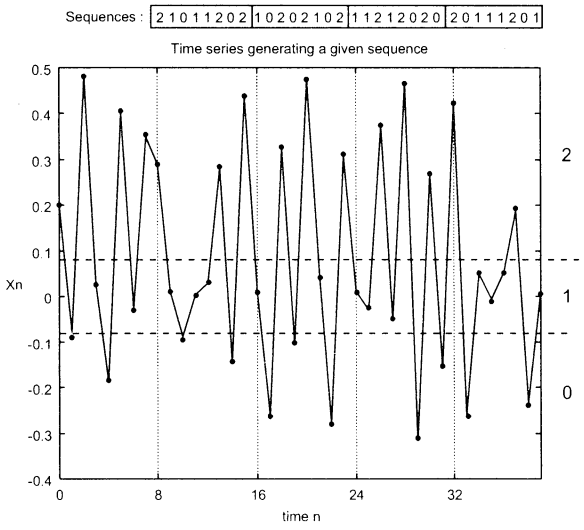


Figure 3: Times series for the perturbed initial values.

3 Fractal encoding

3.1 Chaotic neural network model

We have considered the nonlinear dynamics of a type of chaotic neural network [1], which consists of a chaotically forcing neuron and two almost linear neurons [10]. The latter stable neurons are forced by the chaotic neuron described in subsection 2.1 through transfer functions as shown in Fig. 4. In the model, we use two hard-limit functions, or Heaviside functions as transfer functions from the chaotic neuron X to the static neurons Y and Z to encode the dynamics of the chaotic neuron. The equations of the two stable neurons are as follows:

$$y_{n+1} = f_2(-\alpha_2 \sum_{r=0}^{n} {k_2}^r y_{n-r} + w_{yz} \sum_{r=0}^{n} {k_2}^r z_{n-r} + w_{yx} \sum_{r=0}^{n} {k_2}^r h_1(x_{n-r})) \quad (5)$$

$$z_{n+1} = f_3(-\alpha_3 \sum_{r=0}^{n} {k_3}^r z_{n-r} + w_{zy} \sum_{r=0}^{n} {k_3}^r y_{n-r} + w_{zx} \sum_{r=0}^{n} {k_3}^r h_2(x_{n-r})), \quad (6)$$

where α_i is a positive parameter, k_i is a decay parameter with $0 < k_i < 1$ $(i = 2, 3)$; w_{vu} is the connection weight from neuron u to v with $w_{zy} < 0$, and $w_{yz}, w_{yx}, w_{zx} > 0$; and the function $f_i(x)(i = 2, 3)$ is the sigmoidal function.

The hard-limit transfer functions $h_i(x)(i = 1, 2)$ are defined as follows:

$$h_1(x) = \begin{cases} 0 & (x < a) \\ 1 & (x \geq a) \end{cases} \quad (7)$$

$$h_2(x) = \begin{cases} 0 & (x < b) \\ 1 & (x \geq b), \end{cases} \quad (8)$$

where a and b are the threshold parameters assumed to be $a < b$ in this paper. The functions $h_i(x)(i = 1, 2)$ represent wave-shaping effect of axons producing all-or-none behaviors of propagating action potentials [1].

Then we can represent the dynamics of the network as follows:

$$Y_{n+1} = k_2 Y_n - \alpha_2 f_2(Y_n) + w_{yz} f_3(Z_n) + w_{yx} h_1(f_1(X_n)) \quad (9)$$

$$Z_{n+1} = k_3 Z_n - \alpha_3 f_3(Z_n) + w_{zy} f_2(Y_n) + w_{zx} h_2(f_1(X_n)). \quad (10)$$

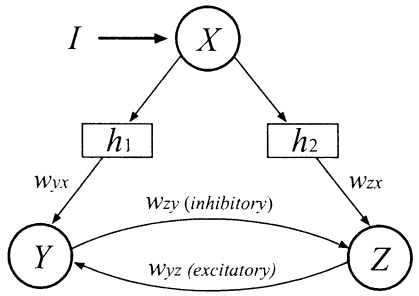


Figure 4: Network configuration of a chaotically forced contracting system.

3.2 Fractal attractor

We next consider the relation between the structure of a code and that of an attractor. We divide

the region of the state in the chaotic neuron map into three subintervals and label the symbols 0, 1, and 2 for each one as shown in Fig. 1. We then express the dynamical series $\{X_n\}$ as a code sequence consisting of the symbols 0, 1, and 2. For example, the point M on the chaotic map in Fig. 1, can be labeled as $1202012112\cdots$. It have been shown that the attractors on $Y-Z$ space have self-similar fractal structures where the hierarchy of the structure of the attractor corresponds to the hierarchy of the symbolic code generated by the forcing chaotic neuron in the previous study [10]. Fig. 5 shows the attractor drived by the chaotic time series of Fig. 3.

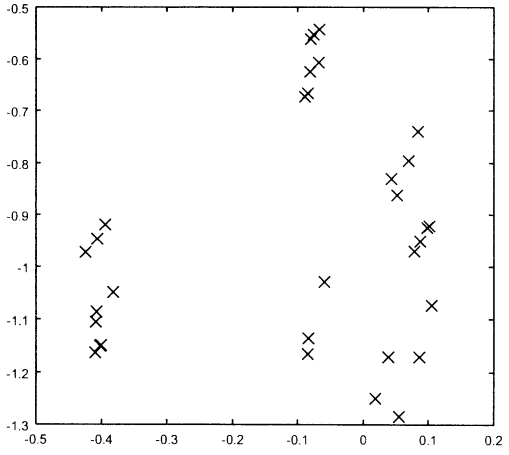


Figure 5: Attractor for the chaotic time series of Fig. 3.

4 Conclusion

In our previous study [10], the network composed of a chaotic neuron and two linearly static neurons has been proposed and implemented as a hardware system with analog discrete devices to investigate whether or not the fractal encoding is actually realized. An attractor structure with three regions 0, 1, and 2 has been clearly demonstrated in the study. The robustness of the fractal attractor for noise to a certain degree also has been observed. Futhermore, the LSI chip for the network has been designed and fabricated. The test of the chip will be done in time. From the viewpoint of engineering, it is interesting to apply the concept of encoding on the fractal attractor to practical information processing. In this respect, we showed that a desirable message can be encoded into an attractor space by using an chaotic neural network in the present study.

In the future study, the invertibility of the mapping for decoding must be examined. The relation between the distance in the sequence of code and the spatial distance on the attractor should be also investigated in detail.

References

[1] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks." *Phys. Lett. A*, Vol. A144, pp. 333-340, 1990.

[2] M. Barnsley, *Fractals Everywhere*, Academic Press, 1988.

[3] E. Bolt and Y. C. Lai, "Dynamics of coding in communicating with chaos," *Phys. Rev. E*, Vol. 58, pp. 1724-1736, 1998.

[4] M. Hata and M. Yamaguti, "Takagi function and its generation," *Japan J. Appl. Math.*, Vol. 1, pp. 186-199, 1984.

[5] S. Hayes, C. Grebogi, E. Ott, and A. Mark, "Experimental control for communication," *Phys. Rev. Lett.*, Vol. 73, pp. 1781-1784, 1994.

[6] J. L. Kaplan and J. A. Yorke, "Chaotic behavior of multidimensional difference equations," *Lecture Notes in Math.*, Vol. 730, pp. 204-227, 1979.

[7] Y. C. Lai, "Encoding digital information using transient chaos," *J. Bifurcation and Chaos 10*, Vol. 4, pp. 787-795, 2000.

[8] E. Ott, C. Grebogi, and J. A. Yorke, "Controlling chaos," *Phys. Rev. Lett.*, Vol. 64, pp. 1196-1199, 1990.

[9] O. E. Rössler, J. L. Hudson, C. Knudsen, and I. Tsuda, "Nowhere-differentiable attractors," *J. Intelligent Systems*, Vol. 10, pp. 15-23, 1995.

[10] J. K. Ryeu, K. Aihara and I. Tsuda, "Fractal encoding in a chaotic neural network," *Phy. Rev. E*, Vol. 64, pp. 2021-2026, 2001.

[11] I. Tsuda and A. Yamaguchi, "Singular-continuous nowhere-differentiable attractors in neural systems," *Neural Networks*, Vol. 11, pp. 927-937, 1998.

[12] I. Tsuda, "Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems," *Bihavioral and Brain Sciences*, Vol. 24, pp. 793-847, 2001.

[13] M. Yamaguti, "Schauder expansion by some quadratic base function," *Journal of Faculty of Science, The University of Tokyo*, Vol. IA36:1, pp. 187-191, 1989.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Human-Like Decision Making in an Autoassociative Neural Network with Dynamic Synapses

Zhijie Wang
Department of Electrical Engineering

DongHua University
Shanghai, 200051, China
wangzj@dhu.edu.cn

Kazuyuki Aihara
Department of Mathematical
Engineering and Information Physics
Graduate School of Engineering
the University of Tokyo
Tokyo,113-8656 Japan
aihara@sat.t.u-tokyo.ac.jp

Abstract

Synaptic depression is incorporated in a conventional autoassociative neural network. When several related patterns are stored in such a neural network and an external stimulus with properties shared by two of the stored patterns is applied to the neural network, the output of the neural network transits between the two stored patterns. The one of the two stored patterns which is closer to the external stimulus is visited more frequently by the network than the other one. This phenomenon is like the decision making process of human beings in some respects.

Key words: dynamic synapse, neural network, associative memory

1 Introduction

It is well known that an autoassociative neural network can store memories through its synaptic connections. In a conventional autoassociative neural network, once the weights of the synapses are learned by learning rules, they are fixed during the memory retrieval process. This implies that the synapses of the neurons are assumed to be 'static', i.e., that they change their weight only on the slow time scale. However, it has been discovered recently that synaptic plasticity occurs across many time scales-from the order of days to the order of milliseconds. It has been also reported that shorter-lasting forms of plasticity tend to depend only on the history of presynaptic stimulation, independent of the postsynaptic response [4].

The dynamic association in a neural network when a synaptic depression is included in the network is investigated recently [3]. The output of such a network transits among the stored patterns during the retrieval process. In this paper, we explore the dynamics of an associative neural network with synaptic depression when the network operates as a recognition system, i.e., explore the behavior of an autoassociative memory when it is applied with an external stimulus. Several related patterns are stored in the neural network, after that an external stimulus with properties shared by two of the stored patterns, for example two stored patterns A and B, is applied to the neural network. We found that the output of the network transits between the two stored patterns (patterns A and B). In other words, the network does not recognize the external stimulus as either of the two stored patterns crisply, though the network does know that the external stimulus is related to pattern A and pattern B. Moreover, if the external stimulus is closer to pattern A than to pattern B, the network will visit pattern A with higher probability. This behavior of the network is different from that of a conventional network with 'static' synapse that usually evolves to settle at a fix stable state. However, the behavior of the network maybe like the way of the decision-making of human beings in some respects.

2 Models of the neural network and the dynamic synapses

The autoassociate neural network is a fully connected network, in which every neuron is connected to all other neurons. The output of the neuron in this paper takes 1 (firing) or 0 (no firing). The dynamics of the neuron of the network is believed to be intrinsically noisy and is described as follows:

$$Prob[x_i(t+1) = 1] = 1/[1 + exp(-2\beta h_i(t))] \quad (1)$$

$$h_i(t+1) = \sum_{j=1}^{N} w_{ij} x_j(t) - \theta_i + a_i \qquad (2)$$

where x is the output of the neuron; θ_i is the threshold; a_i is the external input; the parameter $\beta = 1/T$ is an inverse temperature.

The weight of the dynamic synapse is determined by two parts as follows:

$$w_{ij}(t) = w_{ij}^c r_j(t), \qquad (3)$$

where w_{ij}^c is a static term that is determined by the stored patterns; $r_j(t)$ is the term that is caused by the synaptic depression.

w_{ij}^c is defined as follows:

$$w_{ij}^c = (1/N_s) \sum_{p=1}^{N_s} (2\mu_i^p - 1)(2\mu_j^p - 1), \qquad (4)$$

where N_s is the number of the stored patterns; μ_i^p is the ith component of the pth stored pattern.

Tsodyks and Markram [1,2] proposed a model for the synaptic depression. We used a simplified form of their model that can be described as follows (see [3] for details):

$$r_i(t+1) = r_i(t) + \delta t[\frac{1 - r_i(t)}{\tau} - U r_i(t) x_i(t)], \qquad (5)$$

where $r_i(t)$, which affects the weight of the dynamic synapse (see eqn. 3), is the recovered synaptic resources of the neuron i; τ is the recovery constant; the term $U r_i(t)$ represents a fraction of the recovered synaptic resources of the neuron i; δt is the time step of discretization.

3 Human-Like decision making in the neural network

Let us consider a simple example. As shown in Fig. 1, suppose that the output pattern of the neural network is displayed in a 16 by 16 matrix where the black-shaded areas correspond to neurons with the output of 1. Four patterns in Fig. 1 are stored in the neural network according to the learning rule of eqn. 4. The pattern in Fig. 1(a) stands for the number '6', the pattern in Fig. 1(b) stands for the number '9', the pattern in Fig. 1(c) stands for the reverse pattern of '3', and the pattern in Fig. 1(d) stands for the reserve pattern of 'E'. We explore the behavior of the network when an external stimulation in Fig. 2 that is shared by pattern '6' and pattern '9' is applied to the network.

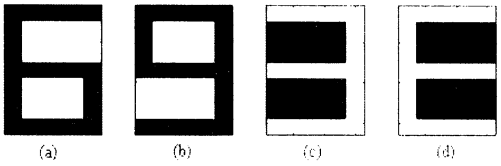


Figure 1: The stored patterns.

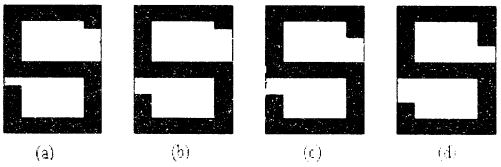


Figure 2: The external stimuli.

After the static part of the weight matrix $w_{ij}(t)$ is determined according to eqn. 4, the patterns are distributively stored in the neural network through the synaptic connections among the neurons. The neural network without synaptic depression ($r_i(t) \equiv 1$) shows conventional associative dynamics. For example, when a pattern close to pattern '6' (Fig. 2(a)) is presented to the neural network, the network can recall and stay at pattern '6' (see Fig. 3). The associative dynamics of the neural network with synaptic depression has been explored in [3]. We investigate the behavior of the network with synaptic depression when an external stimulus is applied. External inputs are applied only to the neurons representing the corresponding pattern by increasing a_i from 0 (no external input) to 0.6. When the parameters are carefully selected, the neural network will carry out a human-like decision making process. For example, Figure 4 shows the associative dynamics following the application of

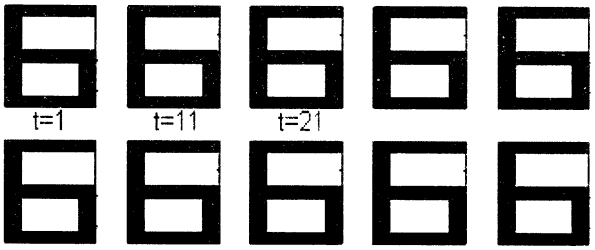


Figure 3: The associative dynamics without synaptic depression. The parameter values are $\beta = 2$, $\theta = 0$, $a = 0$, and $\delta t = 0.1ms$.

the external pattern of Fig. 2(c), which is between pattern '6' and pattern '9', to the neural network. The network does not stay in one state, instead, it visits pattern '6' then pattern '9', returns to pattern '6', and so on. The external stimulus is closer to pattern '6', so the network visits this pattern for longer periods. The dynamic neural network does not finish with a crisp recognition of the external pattern as either pattern '6' or pattern '9'; however, the network does recognize that the external pattern is related to both patterns and draws the relation between the external pattern and pattern '6' over a greater part of the total time. When confronting a problem of this type, the dynamic of the neural network is similar to the decision-making process of a human being.

Table 1 shows the simulation results when the external stimulus of Fig. 2(a), Fig. 2(b), Fig. 2(c), and Fig.2 (d) are applied to the network respectively. Simulation is stopped when the time step exceeds 5000. D_6 in Table 1 represents the Hamming distance between the external stimulation and the pattern '6', and D_9 in Table 1 represents the Hamming distance between the external stimulation and the pattern '9'. N_6 in Table 1 represents the number of the time steps of the network visiting pattern '6', and N_9 in Table 1 represents the number of the time steps of the network visiting pattern '9'. We recognize that the network is visiting pattern '6' when the output pattern of the network is exactly the same as pattern '6'. From Table 1, we can see that if the external stimulus is closer to pattern '6' than to pattern '9', i.e., that the Hamming distance of the external stimulus to the pattern '6' is less than to the pattern '9', the network visits pattern '6' for more time steps. We can also see that the ratio of N_6 to N_9 is reversely proportional to the ratio of D_6 to D_9. It is worth noting that the result of every run of the simulation is a little different due to the probability model of the neuron, however, we found that every run of the simulation has the property discussed above.

Table 1. The Hamming distances and the visiting time steps. The parameter values of simulation are $\beta = 2$, $\theta = 0$, $\tau = 15ms$, $U = 0.8$, and $\delta t = 0.1ms$. The initial value for r_i is 1.

	D_6	D_9	N_6	N_9	D_6/D_9	N_6/N_9
Fig. 2(a)	6	24	3317	1385	0.250	2.395
Fig. 2(b)	9	21	3248	1460	0.429	2.225
Fig. 2(c)	12	18	2517	2110	0.667	1.193
Fig. 2(d)	18	12	2032	2607	1.500	0.779

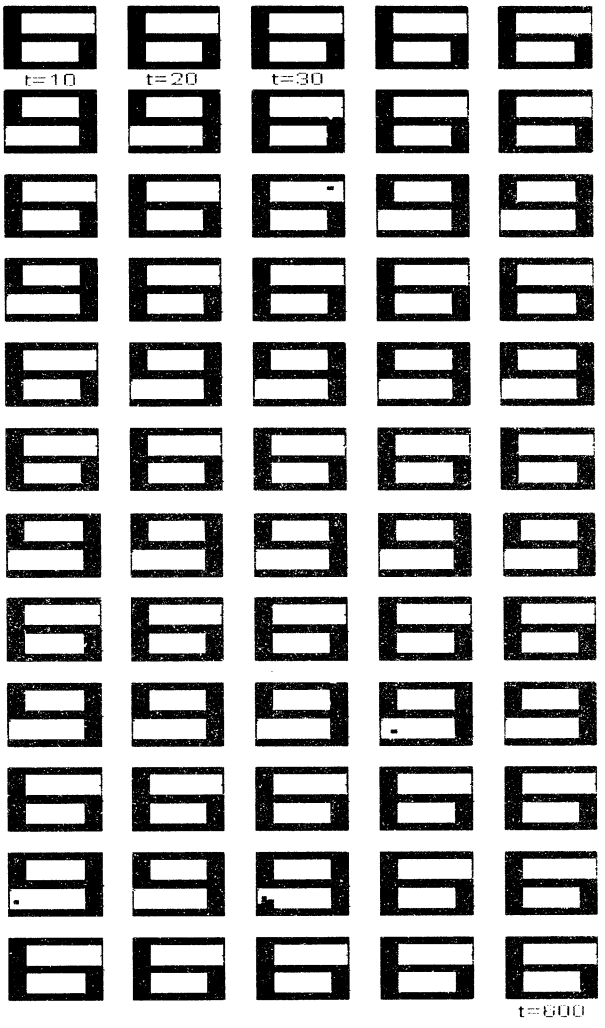


Figure 4: The associative dynamics when an external stimulus that is shared by patterns '6' and '9' is applied with the parameters $\beta = 2$, $\theta = 0$, $\tau = 15ms$, $U = 0.8$, and $\delta t = 0.1ms$ for the network. The initial value for r_i is 1. The time step t runs from 10 to 600.

4 Discussion

The human-like decision making in an association neural network is demonstrated with a simple example. We next explain this phenomenon qualitatively. As an example, we assume the external stimulus is the pattern in Fig. 2(c). Since the static part of the weight of the synapse w_{ij}^c is the same as those of the conventional autoassociative memory and the initial value of r_i is 1, the feedback inputs from other neurons through the recurrent interconnections have a 'force' (f_s) that attracts the network to a stored pattern (pattern '6') that is closest to the external stimulus (the pattern in Fig. 2(c)). When the network stays at pattern '6', the feed back inputs from other neurons through the recurrent interconnections have a 'force' (f_p) that prevent the network to move to pattern '9'. After the network stays at pattern '6' for some time, the effect of the synaptic depression makes r_i of those neurons in pattern '6' decrease according to eqn. 5. Accordingly, f_s and f_p decrease. Therefore, with the decrease of f_s, the network can not stay at pattern '6' no longer. On the other hand, with the decrease of f_p, those neurons in pattern '9' that correspond to external stimulus begins to fire because of the higher value of a_i in eqn. 2. Thus the network begin to transit to pattern '9'.

More formal analysis of the phenomenon is required to explain this phenomenon more clearly, and this is the future problem.

References

[1] Tsodyks, M. and Markram, H (1997), The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. Proc. Natl. Acad. Sci. USA, 94, 719-723.

[2] Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. Neural Comput., 10, 821-835.

[3] Pantic, L., Torres. J.J. and Kappen. H.J. Associative memory with dynamic synapses. To appear in Neural Computation 2002.

[4] Maass, W., and Bishop, C.M., (eds.), Pulsed Neural Networks, The MIT Press, 1998.

[5] Wang, Z.J. and Aihara, K., A Fuzzy-Like phenomenon in chaotic autoassociative memory. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E85-A, No. 3, pp. 714-722, March 2002.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Origins of Stochasticity in Gene Expression and Control of the Fluctuation

Yoshihiro Morishita[1] and Kazuyuki Aihara[1]

[1] *Department of Complexity Science and Engineering, The University of Tokyo, Hongo, Tokyo 113-8656, Japan*

Abstract

This paper considers origins of stochasticity in gene expression. We analyze them by using stochastic modeling and numerical simulation, and find that they are classified into two phases related to the number of proteins and the ratio of transcriptional and translational rates. In addition, we clarify that dimerization process and weak interaction between proteins and their surrounding molecules have a large effect on the fluctuation level. Based on the results, methods to control the fluctuation are suggested.

1 Introduction

Progress in molecular biology and experimental technology has enabled exploring the mechanism of genetic information processing in organisms. Recently, to develop an understanding of the dynamical behavior of genetic networks, artificial genetic networks - which are designed to have specific functions, such as a toggle switch[1] and an oscillator[2]- using well-known genes, such as lacZ and cI, have become one of the fundamental areas of genomic research. It is reported, however, that actual gene expression and network dynamics are very noisy, so to designe networks that operate with high stability and reliability, it is necessary to clarify the origins of stochasticity in gene expression and to explore methods to control the fluctuation. Thatti et al. [3] insisted that the transcriptional and translational rates are two main factors that decide the fluctuation level. They analyzed single gene expression using mathematical models and numerical simulations. Becskei et al. [4] showed experimentally that auto-regulation is effective as a method to control the fluctuation.

In this study, the origins of the fluctuation in gene expression are analyzed by using stochastic modeling and numerical simulation. From the results, it was found that the origins are divided into two phases related to the mean number of the synthesized proteins and the ratio of transcriptional and translational rates. In addition, it was clarified that the dimerization process has functions to reduce the fluctuation in gene expression itself as well as reduce the high-frequency noise. It was also found that the weak interaction between the proteins and their surrounding molecules has a large effect on the fluctuation. Based on the results, furthermore, methods to control the fluctuation are suggested.

2 Model

Stochastic Modeling

Stochastic modeling is used to describe gene expression. The basic concepts underlying stochastic modeling are that the quantity of each reacting matter, such as mRNA and protein, are represented not by the concentration of the matter but by the number of molecules, and that each chemical reaction occurs at every collision of reactants. The probability of the occurrence of a collision within a unit time, Δt, is considered to be proportional to the product of the numbers of reactants and the reaction rate constant. In the case of the reaction, for example,

$$A + B \xrightarrow{k} AB \qquad (k : rate\ constant). \tag{1}$$

when it is assumed that the numbers of reactants A and B at a certain time t are $[A]$ and $[B]$, then the probability of the occurrence of the collision between A and B is given by $k[A][B]\Delta t$ within the time interval, $[t, t + \Delta t]$.

Elementary Process

In this paper, the following are considered as elemental processes in gene expression (Fig. 1).

(a) Bonding between a promoter region and transcription related factors; e.g., RNAP, repressors and activators (k_f: forward, k_b: backward).

(b) Initiation of transcription and elongation of a mRNA (k_m: isomerization rate).

(c) Translation (k_p: translational rate).

(d) Multimerization (k_1: forward, k_2: backward).

(e), (f) Degradation of a mRNA and a protein (r_m,r_p).

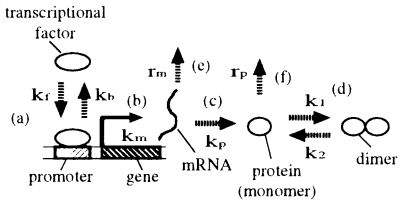


Figure 1: Elementary processes in gene expression

where $k_{\cdot}$ and $r_{\cdot}$ indicate the rate constants in each process. These parameters used in this paper take the values that are biologically valid (not shown).

Mean number of synthesized proteins
The mean number of synthesized proteins, $\langle P \rangle$, is focused on as one of the factors that decide the fluctuation level in gene expression. At a steady state, this value is approximately given by $\langle P \rangle = (k'_m k_p)/(r_m r_p)$ where k'_m indicates the mean rate of transcription, which is derived by simple caluculation using the additive property of Poisson under the condition that the number of transcriptional factors $[T]$ is constant, $k'_m = [T]/([T] + (k_b + k_m)/k_f)$.

3 Results

In the following, several factors that determine the fluctuation in gene expression level are considered. The time series of the gene expression level and the statistics of the expression level are calculated by using Gillespie's algorithm [5]. The fluctuations are mainly evaluated by using CV values since the relative quantity of the fluctuations that do not depend on the mean number of synthesized proteins is considered to be appropriate.

3.1 Mean Number of Synthesized Proteins

It is expected, by simple speculation, that the fluctuation in gene expression depends mainly on the number of synthesized protein molecules. As a first step, the relation between the fluctuation in single gene expression at a steady state and the mean number of the proteins, $\langle P \rangle$, is focused on and evaluated by using CV values (Fig. 2). The solid line shows the case where $\langle P \rangle$ was varied by controlling only the translational rate, on the other hand, the broken line shows the case where $\langle P \rangle$ was varied by controlling only the mean rate of transcription. In both cases, the fluctuation decreases when $\langle P \rangle$ increases. However the sensitivity of the fluctuation to $\langle P \rangle$, and the fluctuation under the condition that $\langle P \rangle$ is constant, strongly depend on the parameters controlled.

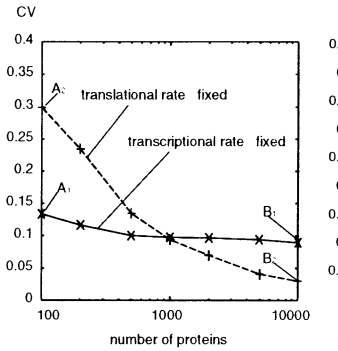


Figure 2: Effect of the number of the proteins

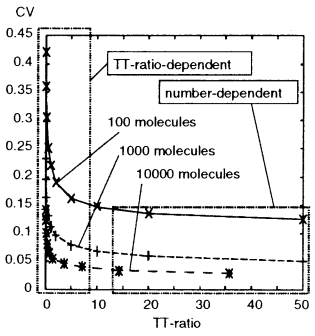


Figure 3: Effect of the TT-ratio

At points A_1 and A_2 in Fig. 2, the ratios of the mean rate of transcription and the translational rate, the TT-ratio, are 39.2 and 0.2, and at points B_1 and B_2 the ratios are 0.392 and 20.0. Hence it is suggested that under the condition that $\langle P \rangle$ has the same value, the fluctuation in gene expression is reduced with the increase of the TT-ratio.

3.2 Ratio of Mean Rate of Transcription and Translational Rate (TT-ratio)

Next, the relation between the fluctuation at a steady state and the ratios of the mean rate of transcription and the translational rate, TT-ratio, is examined under the condition that the mean number of the proteins, $\langle P \rangle$, is fixed. The three curves in Fig. 3 show the cases where $\langle P \rangle$s were fixed at 100, 1000, and 10000 molecules. When the TT-ratio was small, TT-ratio<5, the CV values became remarkably large, in the case of 100 molecules CV=0.15-0.45, and depended sensitively on the TT-ratio. On the other hand, when the TT-ratio was large, TT-ratio<10, the CV values were small, in the case of 100 molecules CV<0.15, and they had an almost constant value, which does not depend on the TT-ratio but on $\langle P \rangle$. Thus, it was found that the origins of fluctuation in gene expression can be classified into two phases: the parameter TT-ratio dependent phase and the number $\langle P \rangle$ dependent phase.

3.3 Input Noise

Next, the relation between the fluctuation at a steady state and the input noise, that is, the fluctuation of the number of the transcriptional factors is examined. Here, as an appropriate input noise, the cascade of the

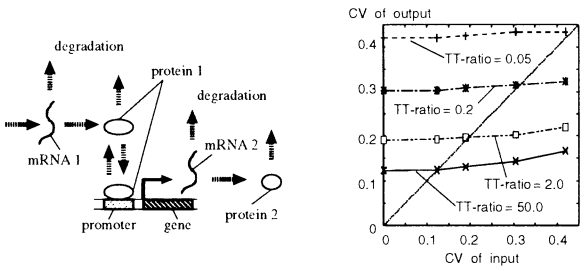


Figure 4: (a) Cascade of the expression and (b) effect of the input noise

gene expression is considered (Fig. 4 (a)). The input noise level is varied by controlling the TT-ratio of the first expression, and that of the second expression is fixed, by which the effect of only the input noise can be extracted. And $\langle P \rangle$ are fixed at 100 in both the first and second expressions.

Figure 4 (b) shows the input CV vs. the output CV, and each curve shows the case where the TT-ratio of the second expression was fixed at 0.05, 0.2, 2.0, and 50.0. Certainly the output CV becomes large when the input noise level is increased, but the effect of the input noise is totally small, and the fluctuation level is mainly determined by the TT-ratios.

If the cascade continues several times under the condition that the TT-ratio is fixed, then the CV values of the final expression converges to the intersection of each curve and the line y=x.

3.4 Dimerization Process

In the above, the fluctuation in gene expression was evaluated by using the CV value of the monomer proteins, however, a lot of proteins function as multimers in a cell. Here, to consider the biological meaning of the multimerization process from the viewpoint of the stochasticity in gene expression, the CV value of dimer proteins at a steady state is focused on (Fig. 1; k_1 and k_2 are the forward and backward rates). Parameters are set to satisfy the condition that the mean numbers of the synthesized monomer and the dimer at a steady state are 100.

Figure 5 shows the TT-ratio vs. the CV values of the monomer and the dimer at a steady state under the existence of the dimerization process. For comparison, the curve of the CV of the monomer without the dimerization process is also drawn in Fig. 5 (indicated by monomer* in the figure). It is found that the dimerization process suppresses the fluctuation of the monomer and the dimer; the CV of the monomer is reduced by 10-20% and that of the dimer is almost equal to but smaller than that of the monomer without the dimerization process. These results are given for the wide range values of k1 and k2, which determin the time to the steady state, but hardly affect the CV at the steady state. It is considered that this noise-reduction effect by the dimerization process is caused by the interaction between the monomer and the dimer state, by which the fluctuations are absorbed and cancel each other. Thus, the dimerization process plays the role of a kind of noise-buffer, which is considered to be one of the biological meanings of this process.

In addition, it was found that high-frequency noise is reduced by the dimerization process. Figure 6 shows the power-spectra of the time series of the monomer (top) and the dimer (bottom) at a steady state, and Figure 7 shows the difference of these spectra (i.e. *dimer − monomer*). It is considered that this phenomenon is caused by the fact that the variation of two molecules of the monomer corresponds to that of one molecule of the dimer, and that the fluctuation of the dimer becomes more moderate than that of the monomer. Thus, the reduction of high-frequency noise is considered to be also one of the biological meanings of the dimerization process.

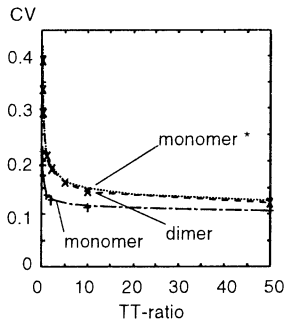


Figure 5: Effect of the dimerization

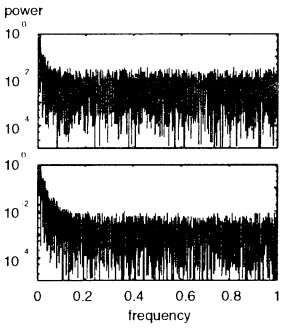


Figure 6: Power spectra of monomer (top) and dimer (bottom)

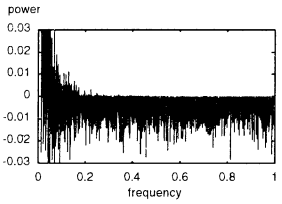


Figure 7: Difference of the powers (dimer - monomer)

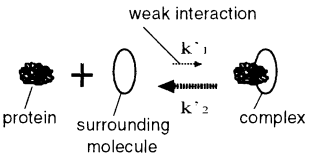


Figure 8: Interaction between proteins and surrounding molecules

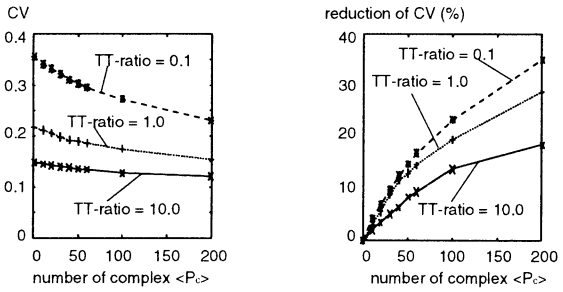


Figure 9: Effect of interaction with surrounding molecules; P_c-CV (left) and P_c-percentage of the reduction of CV (right)

3.5 Interaction with Surrounding Molecules

As an extension of the idea that the interaction between the monomer and the dimer states reduces the fluctuation, the relation of the fluctuation in gene expression and the interaction between synthesized proteins and surrounding molecules, which nonselectively bind with the proteins are examined.

Figure 8 shows the reaction of these molecules, where k_1' and k_2' indicate the association and dissociation rates. The mean number of the complex $\langle P_c \rangle$ is described as $\langle P_c \rangle = k_1' \langle P \rangle \langle N \rangle / k_2'$, where $\langle N \rangle$ indicates the mean number of the surrounding molecules. Here, when it is assumed that $\langle N \rangle$ is sufficiently large and k_1' is sufficiently small, and that $k_1' \langle N \rangle = const. \equiv K$, then $\langle P_c \rangle$ is given by $\langle P_c \rangle = K \langle P \rangle / k_2'$. In addition, when $\langle P \rangle$ is fixed at a constant value, e.g., 100 molecules, then $\langle P_c \rangle$ is determined by K and k_2'.

Figure 9 shows the relation between $\langle P_c \rangle$ and the CV of the proteins (left), and the percentage of reduction of the CV values (right) at the steady state under the condition that the TT-ratio of the protein synthesis is fixed. It is clear that the fluctuation is largely reduced with the increase of $\langle P_c \rangle$, which is caused by the similar mechanism of reduction of fluctuation that occurs in the dimerization process, that is, the equilibrium between the association and dissociation reactions works as a kind of noise-buffer.

4 Discussion

Control of the Fluctuation

From the above results, the following are regarded as practical methods to control the fluctuation in gene expression when artificial genetic networks are designed. First, it is considered to use the genes having a larger ratio of the transcriptional and translational rates if the number of synthesized proteins is the same.Practically, it is better to use or design genes having a strong promoter with a high transcriptional rate, or the genes of which mRNAs have a weak affinity to ribosomes. Secondly, it is considered to add, as a noise-buffer, molecules that bind with the synthesized proteins but their complex has no bioactivity and no effect on cellular functions. Practically, the mutant of the substrate of the synthesized protein is considered to be a candidate for the desired molecules; concretely, the binding site is kept normal and the active sites are mutated to have no bioactivity.

5 Conclusion

In this study, the origins of the fluctuation in gene expression were analyzed by using stochastic modeling and numerical simulation. As a result, the following were determined. First, the origins of stochasticity are divided into two phases related to the mean number of the synthesized proteins and the ratio of the mean rate of transcription and the translational rate. Secondly, the dimerization process has functions to reduce the fluctuation in gene expression itself and the high-frequency noise. Thirdly, weak interaction between the synthesized proteins and their surrounding molecules has a large effect on reducing the fluctuation. Furthermore, based on the results, methods to control the fluctuation were suggested.

Acknowledgements

This work was supported in part by the Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant 12208004.

References

[1] Timothy S. Gardner, Charles R. Cantor and James J. Collins (2000) "Construction of a genetic toggle switch in Escherichia coli" *Nature*, Vol. 403, p. 339

[2] Michael B. Elowitz and Stanislas Leibler (2000) "A synthetic oscillatory network of transcriptional regulators" *Nature*, Vol. 403, p. 335

[3] Mukund Thatti and Alexander van Oudenaarden (2001) "Intrinsic noise in gene regulatory networks" *PNAS*, Vol. 98, No. 15, p. 8614

[4] Attila Becskei and Luis Serrano (2000) "Engineering stability in gene networks by autoregulation" *Nature*, Vol. 81, p. 2340

[5] Gillespie D. T. (1977) "Exact stochastic simulation of coupled chemical reactions" *J. Phys. Chem.*, Vol. 405, p. 590

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Quality Evaluation of Transmission Devices Using the GA

Bingchen Wang, Omatu Sigeru
Compute and system Sciences
Graduate School of Engineering Osaka Prefecture University
Sakai city, 1-1 Gakuen-cho, 599-8531
{wangb,omatu}@sig.cs.osakafu-u.ac.jp

Abstract

In this paper, we propose a new method of evaluating the quality of a transmission device according to the acoustic data by using the genetic algorithm (GA). We consider that the "spectrum average" and the "frequency variation" reflect the characteristic of an acoustic data. In this paper, we first extract the "spectrum average" and the "frequency variation" from the acoustic data of operating transmission device. Then we use the GA to select the "significant frequencies" and determine the boundary between good and no good products. The experimental results show that the proposed method can perform the quality evaluation of transmission devices successfully.

1 Introduction

As is well known, transmission devices have been widely used in many kinds of machines. At present, most factories depend on skilled workers to test the quality of transmission machines by listening to the sound. Thus, developing a means of automatic and quantitative testing becomes an important problem. Moreover, an intelligent self-learning system is required in order to test new products effectively.

Research on acoustic recognition has made rapid progress in recent years [1]-[3]. Teranishi *et al.* [4] have classified new and used bills using the acoustic data from a bank machine. , In their research, they checked whether the bill is new or old according to the acoustic energy pattern by using a competitive neural network. In a previous study, Wang *et al.* [5] proposed a method of testing the quality of machines automatically by using a neuro-classifier trained by the learning vector quantization (LVQ). In that paper, we have tested the quality of machines by only using the "frequency variation". But in the practical production, we have found that the "spectrum average" was also important where the definition of "spectrum average" and "frequency variation" will be stated in the later Chapter. However, it is difficult to use the whole "spectrum average" to test the quality of a transmission device. We should determine some "significant frequencies" and give them proper levels to classify good and no good products. In this paper, we propose a new method of evaluating the quality of a transmission device according to the acoustic data by using the genetic algorithm (GA).

2 Quality Evaluation System

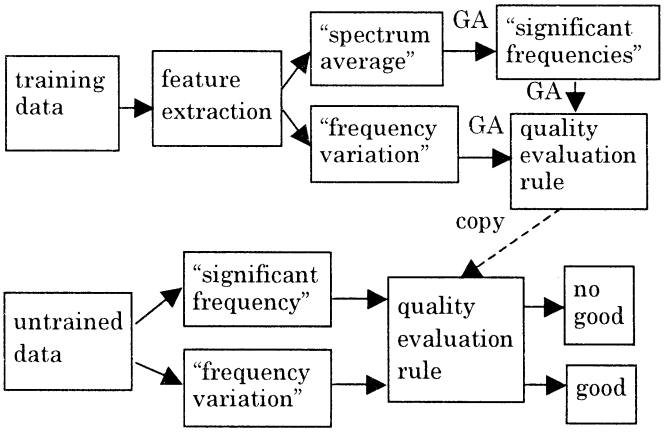


Fig. 1. Quality evaluation system.

Figure 1 shows the quality evaluation system. We first record the acoustic data while good and no good products are running. Then we compute the "spectrum average" and the "frequency variation" in the process of feature extraction. Next, we select some "significant frequencies" from the whole frequency field by using the GA. Then we use the GA to determine the boundary between good and no good products according to the selected frequencies and the periodicity of "frequency variation". At last, we test the untrained data by using this boundary.

3 Feature Extraction of Spectrum

In this paper, we use the following method to pre-process the acoustic data. First, we apply the fast Fourier transform (FFT) to the acoustic data. From the acoustic data, we take 4,096 data in order and compute their spectra. If we take Nx4,096 data, we can obtain N spectra. We compute the average of these N spectra which are called "spectrum average". Then instead of studying the entire frequency field, a frequency

range is selected according to the expert's knowledge and experience. We compute the average value of amplitude in the selected frequency range and draw the variation of the average value with time. In this paper, we call it "frequency variation".

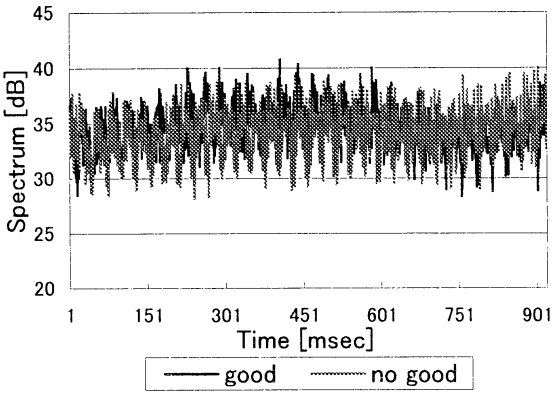


Fig.2 "Frequency variation" of good and no good products.

Figure 2 shows an example of "frequency variation" for good and no good products. It can be seen that the band vector of the good product is different from that of the no good product, but it is still difficult to classify them. From Fig. 2, we observe that both good and no good products have periodicities. From the mechanical structure of the transmission device, it can also be drew the same conclusion. We consider the "frequency variation" as a time series, and apply the FFT to it again. From the FFT result we select a part as the feature vector of "frequency variation".

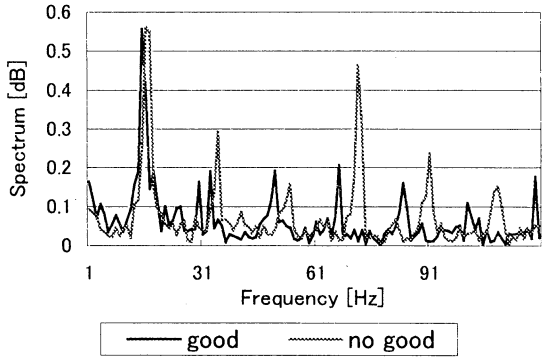


Fig.3. Feature vectors for good and no good products.

Figure 3 shows the feature vectors of good and no good products. It is obviously that the feature vectors of good and no good products have different characteristics.

4 Quality Evaluation with the GA

As mentioned above, we find that the "spectrum average" is also important. Especially, for such no good products as "LS" and "HS" where "LS" denotes the sound is too loud and "HS" denotes the sound has a high tone, they have large values in some particular frequencies but there are no obvious features in their "frequency variation". Moreover, from the "spectrum average" we can estimate the cause of trouble based on the abnormal power spectrum in some frequency ranges from theoretical viewpoint. Because the dimension of "spectrum frequency" is too large, it is difficult for applying it directly. In this paper, we select some "significant frequencies" and determine the boundary between good and no good products by using the GA.

4.1 Evaluation of the "Spectrum Average"

Figure 4 shows the process of the GA. we represent the frequency range as $[x_1, \ldots, x_n]$, and the corresponding power spectrum as $[y_1, \ldots, y_n]$.

The first step is evaluation of each frequency x_i. For the frequency x_i, if its amplitude y_i is larger than that of a good product, or its amplitude y_i is smaller than that of a no good product, we add one to its fitness. Using this evaluation function, we evolve the power spectrum y_i of each frequency x_i. At last for each frequency x_i we obtain a fitness f_i, which denotes its recognition ability of good and no good products, and a corresponding ranking o_i. From the frequency range $[x_1, \ldots, x_n]$, we select the best 20 of x_i as the "significant frequencies" according to the o_i.

The second step is evaluation of the selected frequencies. For all the "significant frequencies", if their amplitudes are larger than those of a good product, or if the amplitude of any frequency is smaller than that of a no good product, we add one to the fitness. Since we adopt two conditions: "spectrum average" and "frequency variation" to test the quality of products, we consider that the recognition of good products is more important than that of no good products. Therefore, if the good product is misclassified as a no good product, we subtract one from the fitness. We use the above evaluation function to evolve the selected frequencies. After evolution, we can obtain the best individual $\{(x_k,y_k',o_x'),\ldots,(x_l,y_l',o_y')\}$ and its fitness F_i.

The third step is exchanging the "significant frequencies". In the above steps, we obtain the best individual of "significant frequencies". Certainly, each frequency of "significant frequencies" has good performance in recognition. However, when we use them together, it is possible that they will conflict each other. This confliction will lead up to the result that some

frequencies have no effect on recognition of good and no good products. Therefore, we check each frequency x_i of present "significant frequencies", if the amplitude of any frequency x_i is larger than the amplitude of all no good products, we consider that the frequency x_i is not necessary to the quality evaluation. We discard this frequency x_i, and select the frequency which has the best fitness from the remaining frequencies. Repeat the operation of steps 1, 2 and 3, until the best 150 in the remaining frequencies of step 1 are used or all good and no good products are classified correctly.

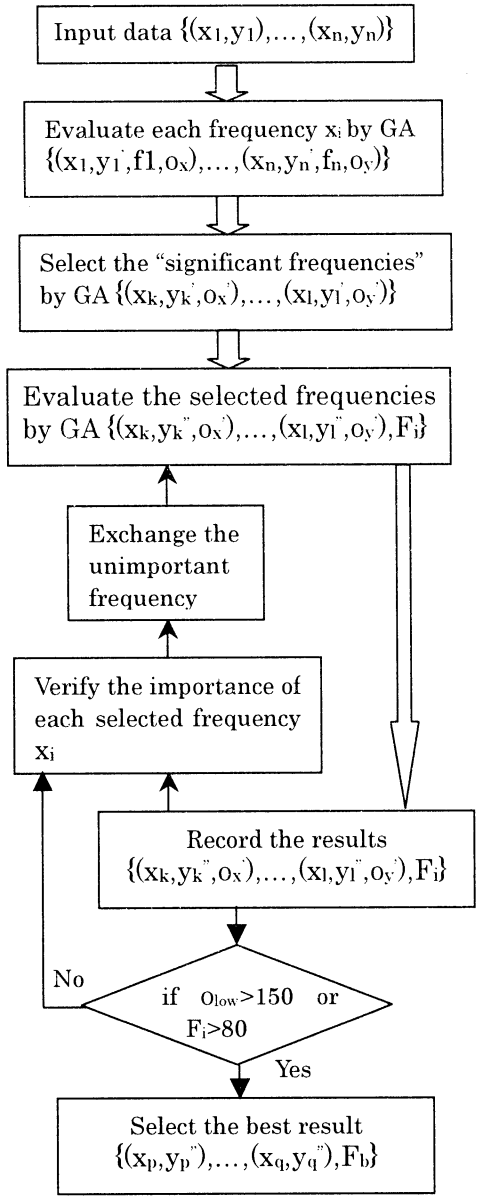


Fig.4. Process of the GA for "spectrum average".

4.2 Evaluation of the "Frequency Variation"

From Fig. 3, we can see that the peaks of feature vectors play an important role in discriminating between good and no good products. We have also obtained the same conclusion from the classification results by the LVQ in the previous study [5]. In this paper, we call these peaks as p1, p2, p3, p4, p5, and p6. As a good product, it must satisfy the following two conditions: One is that the value of each peak must be smaller than some value. The other is that the value of p4 cannot be too large comparing with the values of neighbor peaks p3 and p5. Here, we define a variable p7=p4/max(p3,p5). For a good product, its value must be smaller than some value. We use the GA to search the optimum value of data series {p1, p2, p3, p4, p5, p6, and p7}. The evaluation function is designed as following: if any value of the data series is smaller than that of a no good product, or if all the values of the data series are larger than those of a good product, we add one to the fitness. If the good product is misclassified as bad product, we subtract one from the fitness. After 10,000 generations evolution, we record the value of the best individual, and use it to classify the good and no good products.

5. Experimental Results

Tables 1 and 2 show the evolution results of "spectrum average" and "frequency variation", respectively.

Table 1. Evolution result 1.

Frequency	Amplitude
52	34.184
99	26.267
115	36.998
166	30.760
194	15.964
353	12.839
425	19.993
760	7.144

Table 2. Evolution result 2.

Frequency	Amplitude
P1	0.559
P2	0.401
P3	0.307
P4	0.366
P5	0.340
P6	0.217
P7	1.320

In order to verify the effectiveness of our proposed method, we use above evolution results to test 40 new data. Table 3 shows the part of test results which include 8 good products and 12 no good products where Human denotes the test

results by a skilled worker, Computer denotes the test results by a computer, Ave. denotes the "spectrum average", and Var. denotes the "frequency variation". From Table 3, it can be found that all the good and no good products are classified correctly. Sample BN10 is a no good product. The test result of the skilled worker is "LS". But as shown in Table 3, the test result of the "frequency variation" gives a wrong message.

Table 3 Test results.

File No.	Human	Computer	Ave.	Var.
GN1	1	1	1	1
GN2	1	1	1	1
GN3	1	1	1	1
GN4	1	1	1	1
GN5	1	1	1	1
GN6	1	1	1	1
GN7	1	1	1	1
GN8	1	1	1	1
BN1	0	0	0	0
BN2	0	0	0	0
BN3	0	0	0	0
BN4	0	0	0	0
BN5	0	0	0	0
BN6	0	0	0	0
BN7	0	0	0	0
BN8	0	0	0	0
BN9	0	0	0	0
BN10	0	0	0	1
BN11	0	0	0	0
BN12	0	0	0	0

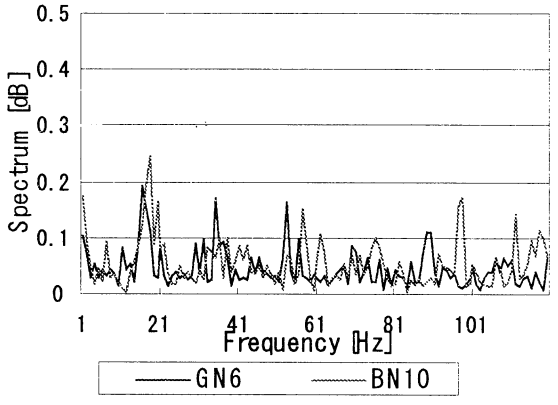


Fig. 5. Feature vectors of samples GN6 and BN10.

Figure 5 shows the feature vectors of "frequency variation" of samples GN6 and BN10. It can be seen that these two vectors are similar. As a matter of fact, "LS" has no any relation with "frequency variation". Figure 6 shows the "spectrum average" of samples GN6 and BN10. From Fig. 6, evidently we can get the conclusion that the problem of sample BN10 is too loud.

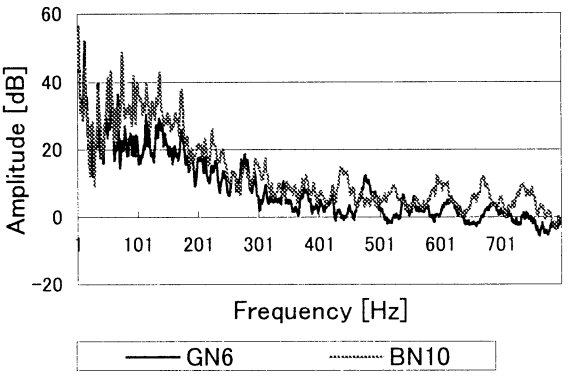


Fig. 6. "Spectrum average" of samples GN6 and BN10.

6 Conclusion

In this paper, we have proposed a new method of evaluating the quality of a transmission device according to the acoustic data by using the GA. We have stated how to extract the characteristic spectrum from the acoustic data of operating transmission device. We have solved the problem of selecting necessary frequencies from the entire frequency field. Experimental results show that the proposed method has a good performance in quality evaluation of transmission device. Using the proposed method, we consider that we can construct an intelligent system, which is able to give the proper quality evaluation for any transmission device automatically and have a self-learning ability to learn the new information while the environment is changing.

References

[1] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice-Hall, New Jersey, 1975.

[2] S. Furui, Digital Speech Processing, Synthesis, and Recognition, Marcel Dekker, New York, 1989.

[3] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1991.

[4] M. Teranishi, S. Omatu and T. Kosaka, "New and used bill money classification using spectral information based on acoustic data of banking machine", Trans. on IEE Japan, Vol. 117-C, No. 11, pp. 1677-1681, 1997

[5] B. Wang, S. Omatu and T. Abe, "Quality evaluation of machines using the LVQ", Journal of Signal Processing, 2002. (To publish)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A High Reliability Method for Classification of Paper Currency Based on Neural Networks

Ali Ahmadi, Sigeru Omatu
Division of Computer & Systems Sciences
Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
{ahmadi,omatu}@sig.cs.osakafu-u.ac.jp

Abstract

In this paper we use a PCA based method for increasing the reliability of paper currency recognition machines. The system is intended for classifying any kind of currency but in this work we examine only different kinds of US dollar (totally 10 bill types). The data is acquired through some advanced sensors and after preprocessing come as an array of pixels. The PCA algorithm is used to extract the main features of data and reducing the data size. An LVQ network model is applied as the main classifier of system. By defining a new method for rating the reliability, we evaluate the reliability of system for 1,200 test data. The result shows that reliability is increased up to 99% when the number of PCA components as well as number of LVQ codebooks are taken properly.

1 Introduction

In this paper we investigate an approach to Principal Components Analysis (PCA) method in order to increase the reliability of paper currency recognition machines, which use neural network classifier. The current classification rate of system is 100% for sample testing data but due to important task of such machines, the reliability must be improved over all various kinds of real input data. For this, PCA is employed to extract the main features of data using the correlation between data samples. The system is intended for classifying different kinds of currency papers but in current study we examine only US dollar bills. A Learning Vector Quantization (LVQ) neural network model is used as the main classifier and a total number of 10 bill types including 1, 2, 5, 10, 20, 50, and 100 dollar (new and old model) are considered as classification categories. We introduce a specific criteria for evaluating the reliability between output classes. The experimental results show a 30% growth in reliability after using extracted features. The paper contains six main Sections: Preprocessing data, PCA feature extracting, Classification, Reliability evaluating, Experimental results and discussion, and Conclusion as are described in below.

2 Preprocessing Data

The original image of bill money comes as a 10x170 array of data taken through three main image sensors and two auxiliary ones, using different waves lengths. Each pixel value varies between 0 and 255. A simple compression algorithm is then used to reduce the size of data from 170 pixels in each row to 30. Then a linear transformation is used as follow:

$$X_i = \frac{x_i - \bar{x}}{S_x} G + C$$

where x_i is the pixel value in each row, $\bar{x}$ is the mean value of pixels, S_x is deviation, and G and C are gain and offset respectively, taking proper values experimentally. Thus a matrix of 10x30 size is provided for using in feature extraction phase.

3 PCA Feature Extraction

PCA is an essential technique for data compression and feature extraction, and has been widely used in statistical data analysis and pattern recognition. The most common derivation of PCA is in terms of a standardized linear projection which maximizes the variance in the projected space [1,4]. For a set of observed d-dimensional data vectors $\{\mathbf{t}_n\}$, $n \in \{1,\ldots,N\}$, the q *principal axes* $\mathbf{w}_j$, $j \in \{1,\ldots,q\}$, are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors $\mathbf{w}_j$ are given by the q dominant eigenvectors (i.e. those with the largest associated eigenvalues λ_j) of the sample covariance matrix $S = \sum_n (t_n - \bar{t})(t_n - \bar{t})^T / N$, where $\bar{t}$ is the data sample mean, such that $Sw_j = \lambda_j w_j$. The q principal components of the

observed vector $\mathbf{t}_n$ are given by the vector $x_n = W^T(t_n - \bar{t})$, where $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q)$. The variables x_j are then uncorrelated such that the covariance matrix $\sum_n x_n x_n^T / N$ is diagonal with elements λ_i .

Hence, in this research we employ PCA to reduce the high dimension of data and finding the main components, which represent the features of data discriminatively. Before using PCA, a proper algorithm is applied to generate a new sort of data based on the original data. As mentioned before, a number of 5 sensors are used in the system, each of them generating two rows of pixels by using different values of wave-length. By using following function:

$$X_{new} = K\,(X_{wl2} / X_{wl1}) - L$$

where X denotes data array and X_{wl1} and X_{wl2} are the data collected by the sensors using wave-length *wl1* and *wl2* respectively, and K and L are constant coefficient. Thus this new data array together with original data make totally 15 channels (each contains 30 pixels) of data. Among these 15 we select 6 main channels, which represent the characteristic of data more effectively. Therefore an array of 6x30 (180 components) pixels is taken for each input image.

By approach to PCA and taking a number of 30 eigen values, a new data vector is then built which contains the main features of 180 dimensional data vector.

4 Classification

Kohonen's LVQ network is a supervised learning algorithm associated with the competitive network as shown in Fig.1 [2,3]. The network consists of an input layer and an output layer. A weight vector w_i is associated to the i-th node in the output layer. In learning phase, the LVQ network selects a weight vector closest to a given input vector and then compares the output of LVQ network with the output of training data. If they match, the selected weight vector is updated so that it approaches the input vector. Otherwise, the selected weight vector is updated so that it moves away from the input vector. After the learning, the LVQ network chooses the nearest weight vector to a given input vector, and output its label as the network output. Thus, a weight vector can be regarded as the center of a local region in the input

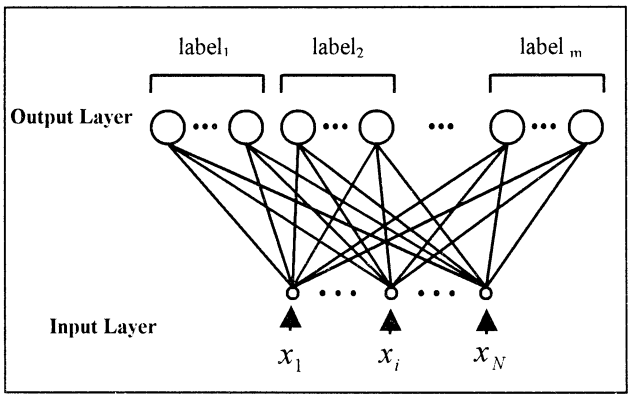


Fig.1 Structure of LVQ network.

space, which called Codebook.

Since the LVQ algorithm is beneficial in case of large number of the input and explanation of the misclassification, it is applied as the main classifier of this system. There exist 10 main categories of bill types and considering 4 direction for each bill (Fig. 2) totally 40 classes are taken. The system is trained by taking different number of codebook vectors for each class looking for the best result. A total number of 400 codebooks is taken finally (averagely 10 per class). Therefore the neural network has a number of 30 neurons (the number of extracted principal components) in the input layer and 400 neurons in the output layer.

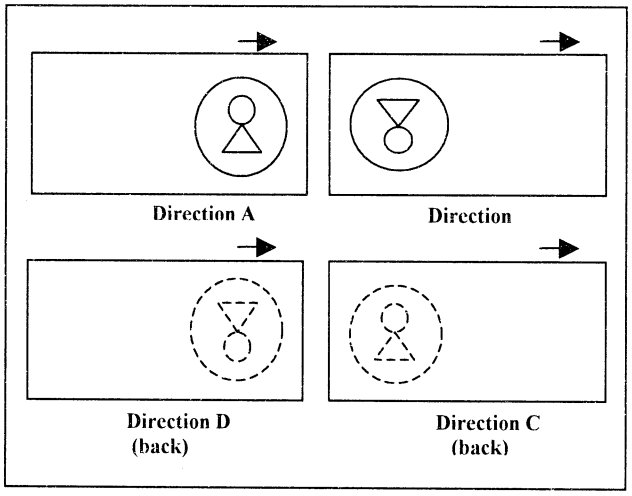


Fig. 2 Different four directions for inserting the bill into the machine.

Figure 3 shows the distribution of test data for different directions of a 100$ bill. It must be noticed that the four direction are finally considered as one class. As well the distribution of data after using PCA and taking the main 30 data

components are shown in Fig. 4. as it can be observed the variance in data distribution is highly reduced.

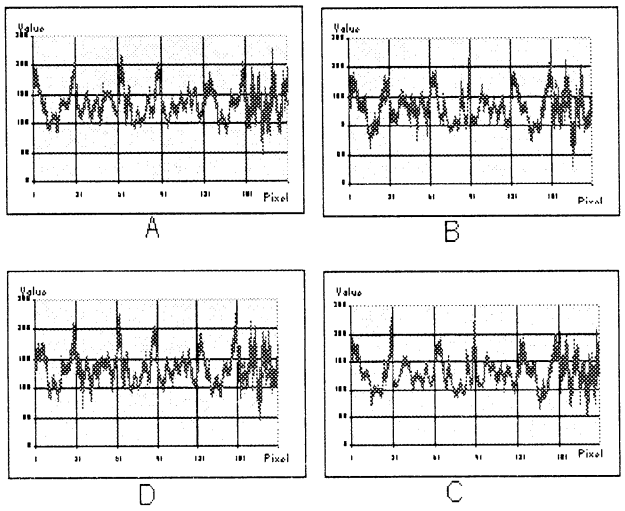


Fig.3 Distribution of data in four directions of 100$ bill (taken from 30 samples).

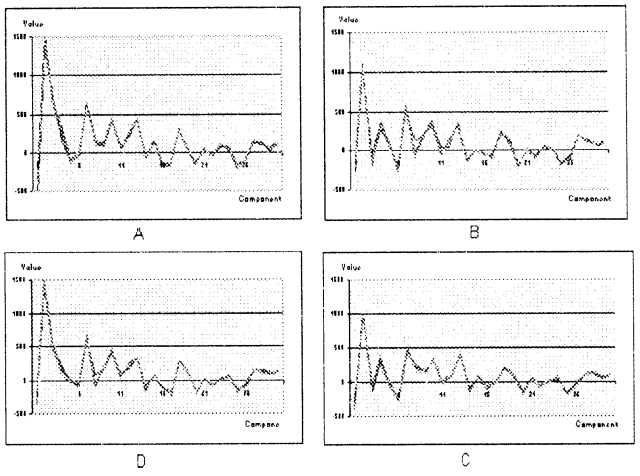


Fig.4 Distribution of data for 100$ bill by taking 30 main PCA components.

5 Reliability Evaluating

There are a number complicated algorithms already proposed for reliability evaluating [5] but usually can not be generalized simply. Here in order to evaluate the reliability of LVQ classifier we apply a simple algorithm. As the LVQ is trained and the codebooks are determined, the probability density histogram for each class is drawn supposing a Gaussian distribution for probabilities and taking a mixture model for each class which is estimated by use of codebooks densities parameters for each class according to the following relation:

$$p(x)=\sum_{i}^{G}\frac{1}{(2\pi)^{d/2}\left|\sum_i^{tr}\right|^{1/2}}\exp(-\frac{1}{2}(x-\mu_i^{tr})^T(\sum_i^{tr})^{-1}(x-\mu_i^{tr}))$$

where G is the number of kernels equal to number of codebooks in each class, d the dimensionality of the feature vectors, and μ_i^{tr} and $\sum_i^{tr}$ are the mean and covariance of kernel i in the training set respectively.

Considering the Gaussian density function for probabilities the interval $[\mu-4.5\sigma, \mu+4.5\sigma]$ can be considered as an area which covers almost 100% of probabilities. If the mixture has no conflict with other classes mixture in this area, then the area can be called as safe zone and the reliability for this class is supposed to be 100%. But in the case that this area is overlapped with distribution of another classes, the reliability can be determined as:

$$RM=1/(\sigma\sqrt{2\pi})\int_{2\mu-\theta}^{\theta}\exp\left(-(x-\mu)^2/2\sigma^2\right)dx$$

where μ is the mean value and σ is the variance of each class density and θ is the cross point with nearest class distribution as shown in Fig.5. As it can be realized any overlap between neighbor densities decreases the reliability rate significantly, so our goal is to make the mixtures separated enough. Next we show how the increasing number of codebooks as well as the PCA can affect this. Therefore, the reliability rate for all classes is calculated and then the system reliability is determined by averaging these reliability values.

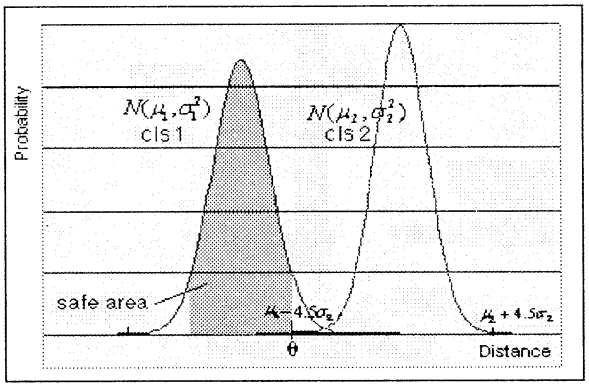


Fig. 5 The overlap between distributions of two class vectors.

6 Experimental Results and Discussion

For training the system a number of 3,570 sample data from 40 different classes was used (4 directions of 10 bill types including 1, 2, 5, 10, 20, 50, 100 dollar, new and old model). Then a number of 1,200 test input vectors (30 samples per class) is taken as for evaluating the system.

The results of classification and reliability of system are shown in Table 1. As it can be seen the recognition rate is 100% in all cases before and after PCA but the reliability is depended to the number of codebooks in LVQ classifier as well as number of components extracted by PCA. As it can be seen increase in number of codebooks will make a high increment in reliability. As well, PCA has a direct affect on reliability rate but when the number of components decreases so much, it will make decrement in the reliability rate. Figure 6 shows this relation clearly.

Table 1. The results of classification before and after using PCA.

No of Codebooks	No of Components	Recg. Rate %	Reli. Rate %
120	180 (Original data)	100	75.1
	15 (PCA)	100	79.6
	30 (PCA)	100	82.1
200	180 (Original data)	100	94.4
	15 (PCA)	100	96.5
	30 (PCA)	100	99.6
400	180 (Original data)	100	95.1
	15 (PCA)	100	93.7
	30 (PCA)	100	99.6

The reason can be explained in the way that increasing the number of codebooks will make the variance of data within each class and consequently the overlap zone between classes would be decreased. PCA also however increase the variance within the new components space, but as the distances between codebooks are increased, it makes the overlap between probability densities to be significantly declined and consequently the reliability of the system is improved. When the components are very few, there would be a very high variance in densities which will enlarge the overlap zones.

The number of codebooks and components can be optimized through an experimental process. In this system we found out that 200 codebooks and

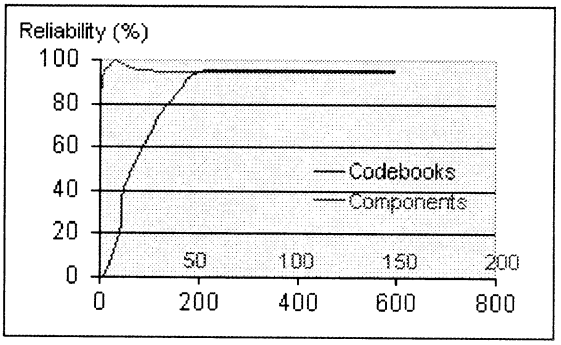


Fig.6 The relation between reliability and number of codebooks and PCA component.

30 components are the best selection and can increase the reliability value up to 99.8%.

7 Conclusion

In this paper we applied an LVQ neuro-classifier for classification of paper currency. A PCA based feature extraction is done on the data for selecting the most effective components. A new method is presented for evaluating the reliability of classification by using the mixture Gaussian densities for each class of data. The method is simple and easy to use in classifiers which are based on codebooks. The results show that by selecting a proper number of codebooks in classifier, and PCA components in features the reliability rate can be grown up to 99%.

In order to have a robust reliability rate further more, it seems that it is necessary to use an additional clustering stage prior to LVQ classifier which is subject of our future research.

References

[1] K. Fukunaga, *Introduction To Statistical Pattern Recognition*. New York: Academic Press, 1972.

[2] S. Haykin, *Neural Networsk*. New Jersey: Prentice Hall, 1999.

[3] T. Kohonen, *Self-Organization Maps*. Springer Series in Information Sciences, 1995.

[4] M.E. Tipping and C.M.Bishop, "Probabili-stic Principal Component Analysis," Journal of the Royal Statistical Society, Series B, 61, Part 3, pp. 611-622, 1999.

[5] L.P. Cordella et al., "Reliability Parameters to Improve Combination Strategies in Multi-Expert Systems," Pattern Analysis and Applications, vol. 3, no. 2, pp. 204-214, 1999.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

An Electronic Nose System Using Back Propagation Neural Networks with a Centroid Training Data Set

Bancha Charumporn, Michifumi Yoshioka, Toru Fujinaka, and Sigeru Omatu

Division of Computer and Systems Science, Graduate School of Engineering
Osaka Prefecture University
1-1Gakuen-cho, Sakai, Osaka 599-8531, Japan
E-mail: bancha@sig.cs.osakafu-u.ac.jp

Abstract

An electronic nose (EN) made from metal oxide gas sensors (MOGS) is applied to measure the smokes of three kinds of burning materials under various weather environments. The distribution of each data is scattering due to the sensitivity of the MOGS to the temperature and the humidity. If the training data are randomly selected for the error back-propagation (BP) neural networks, the randomized training data set may not be able to represent all scattering data and may contain a redundant data that increases the training time without improving the classification performance. Therefore, the method to generate a centroid training data set for a BP based on a similarity index (SI) value is proposed. The method groups the similar data together and the centroid data vectors of all groups are used as a training data set for a BP. The results show a high classification rate by using only a small number of centroid training data.

Keywords: electronic nose, back-propagation, smoke, centroid training data, neural networks, similarity index

1. Introduction

Nowadays, there are several commercial EN applied in food, drink, cosmetic industries, and environmental monitoring [1]. However, the prices of those EN are very expensive and most of them are too big to carry out for outdoor measurement. Therefore, we have developed an inexpensive EN using the commercial MOGS based on the concept of human olfactory system. This EN has been applied to measure three kinds of smoke under various weather environments. These smoke data have been successfully classified into their main categories, joss stick, cigarette, and mosquito coil, using a BP [2]. In this paper, we expand the analytical of these data into more detail, i.e. classify joss stick smoke from different brand and different concentration. Since the MOGS is sensitive to the temperature and the humidity, the data from the same kind of smoke are distributed into various zones. Generally, these data are randomly selected to be the training data. However, the randomized training data always contain a redundant data and sometimes may not be able to represent all scattering data. In order to find a proper training data set that can represent all scattering regions and contains no redundant data, the method to create a centroid training data based on the SI value is introduced. Two data with the SI value nearly 1 are highly correlated that means the information of one data can represent the information of the other. Thus, the data that have high SI value are grouped together by the proposed method. Then the average value of all data in each group is use as a training data for a BP to represent all data in its group. The results achieve high classification rates by using only a small number of training data.

The rest of the paper is organized as follows. Section 2 briefly describes the EN system. The experiment is explained in Section 3. Section 4 explains the method to generate the centroid training data. The results are discussed in Section 5. Finally, the conclusion is summarized in Section 6.

2. Electronic nose system

The electronic nose system shown in Fig.1 has been designed with reliable, long life, low cost and light weight qualifications based on the concept of human olfactory system.

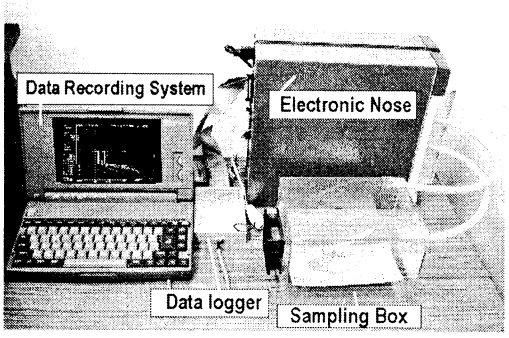


Fig. 1 The electronic nose system

A sampling box with an electric fan flows the tested odor into the EN just like human sniffs the smell. The EN contains various types of MOGS. The MOGS are used as the human olfactory receptors. When the metal oxide element on the surface of the sensor is heated at a certain high temperature, the oxygen is absorbed on the crystal surface with the negative charge. The chemical reaction between the negative charge of the metal oxide

surface and deoxidizing gas makes the resistance of the sensor vary as the partial of pressure of oxygen changes. Based on this characteristic, we can measure the total voltage during the sensors absorbing the tested gas. The signal data are converted from analog signal to digital data by the data logger and recorded by the software into the computer data media. Finally, the artificial neural networks (ANN) are applied just like human brain to analyze the data.

However, this EN does not have the mechanism to control the measuring environment. Therefore, a proper method to handle the data signal of this EN is necessary because the data signal is affected by the measuring environment.

3. Experiment

The experiments have tested three kinds of burning materials not only the different brands of the same kind of smoke, but also the same smoke in different concentration levels like the case of joss stick data. Each data set contains twelve data listed in Table 1. Forty data sets are measured under different weather environment from a winter season to a summer season. However, all twelve data in the same data set are measured consecutively on the same day under a similar environment.

Table 1. List of burning materials in this experiment

Burning Material	Type or Brand Name	Concentration Level	Symbol
Joss stick	Purple	3sticks	P3
		2 sticks	P2
		1 stick	P1
	Brown	3sticks	B3
		2 sticks	B2
		1 stick	B1
	Green	3sticks	G3
		2 sticks	G2
		1 stick	G1
Cigarette	Marlboro Light	1 cigarette	ML
	Caster Mild	1 cigarette	CM
Mosquito coil	Kincho	1 coil	MC

For each data, the voltage signal of the normal air is measured every second for one minute and its average value ($\bar{v}_{air}$) is used as an air reference point. After that the voltage signals of the sensors when absorbing tested smoke, $v_{smoke,\ t}$ are collected every two seconds for a period of two minutes on each smoke sample. Finally, the total change in signals on each period, $V_{smoke,t}$ is calculated by

$$V_{smoke,t} = v_{smoke,t} - \bar{v}_{air} \quad \text{.....(1)}$$

where t is the period from 1 to 60 .

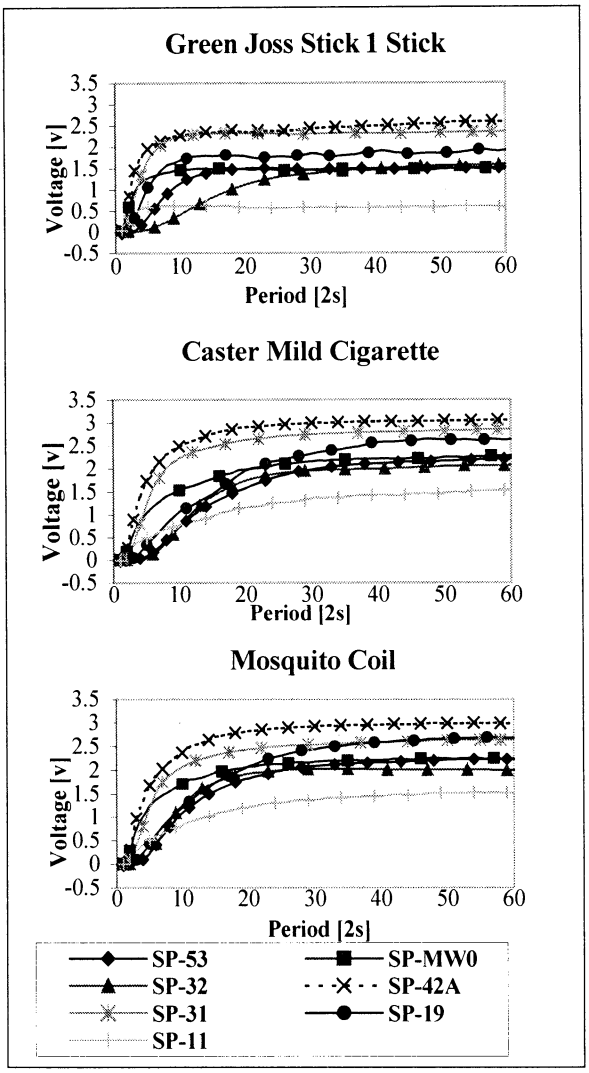


Fig. 2. Sample data from this experiment

Some sample data are plotted in Fig. 2. It shows that the signal from a Caster Mild cigarette smoke that was measured in the night time is quite similar to signal of the mosquito coil that was measured in the day time. Normally, these data will be used for training and testing by the ANN directly, but the results may not yield a good classification rate. Therefore, a method to reduce the effect from measuring environment is applied before generating the training data by the proposed method.

As mention previously that each data of a same data set is measured consecutively in a similar environment. Therefore, we make a hypothesis that all data in the same data set have the same effect from the environment. If we normalize these data, it should be able to reduce the effect of measuring environment. Thus, each data set is normalized by Eq.(2) before analyzing by the BP.

$$\text{norm}(V_{smoke,t}) = \frac{V_{smoke,t} - \bar{V}}{\sigma_V} \quad \text{.....(2)}$$

where $\bar{V}$ and σ_V are the mean and standard deviation of all data in each data set , respectively.

4. Centroid training data creation

4.1 Similarity index (SI)

In this paper, we have applied the correlation value developed by Karl Pearson to measure the similarity of the candidate data in order to select the most valuable training data for the BP. We call this correlation value as a similarity index (SI). The SI value varies from –1 to +1. Two random variables with a SI of either –1 or +1 are highly correlated because knowledge of one provides precise knowledge of the other. However, the SI provides information only about linear relationships between random variables. Random variables could have a nonlinear relationship but still have SI close to 0 [3]. Therefore, we make an assumption on this application that each data pattern has nearly linear relationship to other data patterns.

The similarity index matrix (**SIM**) of all candidate data is calculated by the following equation.

$$\mathbf{SIM} = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ r_{21} & \cdots & r_{2m} \\ \cdots & \cdots & \cdots \\ r_{m1} & \cdots & r_{mm} \end{bmatrix},$$

$$r_{xy} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\right)\left(\sum_{i=1}^{n} y_i^2 - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n}\right)}} \quad \text{.....(3)}$$

where x and y are the comparing data and n is the number of parameters in each data.

4.2 Generating centroid training data algorithm

Define D_c, and D_t as a candidate data set, and a training data set, respectively. The aim of this algorithm is to group the data in D_c and use the centroid vector of each group to be a training data. The following steps explain the processes.

Step1: Each data type contains m data patterns. Initialize the number of data in D_c, D_t equal to m, and 0, respectively. Calculate $m \times m$ size **SIM** using Eq.(3).

Step2: From **SIM,** calculate the mean of the SI for each data, $\bar{r}_i$ using Eq.(4).

$$\bar{r}_i = \frac{\sum_{j=1}^{m} r_{ij}}{m} \quad \text{.....(4)}$$

Step3: Compute the overall mean, $\tilde{r}$, and a standard deviation, σ_r using Eq.(5) and Eq.(6), respectively.

$$\tilde{r} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} r_{ij}}{m^2} \quad \text{.....(5)}$$

$$\sigma_r = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{m} (r_{ij} - \tilde{r})^2}{m^2}} \quad \text{.....(6)}$$

Step4: Find the threshold, T using Eq.(7).

$$T = \tilde{r} + \alpha\sigma_r \quad \text{.....(7)}$$

where α is the real value, recommended in the range of [0-1] and $T < 1$.

Step5: For all $i, j \in D_c$ and $i \neq j$, count the SI of data i by

$$Q_i = number\ of\ \{r_{ij} \mid r_{ij} > T\}$$

Step6: Group all data j that have $r_{kj} > T$ with data k, where $\bar{r}_k = max_i[\bar{r}_i]$ and $Q_k \geqq n$. Here, n is the minimum data that is worthwhile to form a group, recommended n higher than 2. After that, remove data k and all data j from D_c. The member of D_c is decreased by $1+Q_k$.

Step7: Calculate the centroid vector of the grouped data from step6 by Eq.(8)

$$\mathbf{C}_i = \frac{1}{N}\sum_{n=1}^{N} \mathbf{X}_n \quad \text{.....(8)}$$

where $\mathbf{C}_i$ and $\mathbf{X}_n$ are the centroid vector of group i, and data vector n in group i, respectively. N is the number of data in group i, which equals to $1+Q_k$ from the previous step. Then, put centroid vector $\mathbf{C}_i$ in D_t. Increase the number of data in D_t by 1.

This centroid vector will be used as a training data

Step8: Repeat Steps 5 and 7until $Q_k < n$.

The data that left in D_c are the data that have low correlation with the grouped data or the number of correlated data is not worthwhile to form a group. These data have a high possibility to be misclassified because there is no centroid data to represent them.

Steps 1 to 8 will be repeated for all data types before starting the BP method in Step 9.

Step9: Begin the training process of a BP method using all centroid data from Step 7 until a specific epoch or a specific mean square error (MSE) is triggered. Then test with all available data.

Step10: The misclassified data will be grouped to the other misclassified data if their SI value is nearly the threshold value, T in Step 4. Then their average value will be added to D_t and perform the training in Step 9 again.

Steps 9 and 10 will be repeated until the classification rate is satisfied or until the misclassified data cannot be grouped.

5. Classification results and discussion

5.1 Experiment results

The structure of BP in this application consists of three layers. The input layer consists of four hundred and twenty input neurons, which equals to the number of measuring period (60) times the number of sensors that response to the measured smoke (7). For the hidden

layer, we have tried with many values and finally thirty-five hidden neurons are the best value that shows a good accuracy with a reasonable training time. The output layer contains twelve output neurons. Each output neuron presents each data type shown in Table 1. The best training parameters getting from a trial and error method are, α=0.1 and μ = 0.001.

By using the method described in Section 4, six data types (P3, B1, B3, G3, ML, CM) can be grouped into four groups and six data types (P1, P2, B2, G1, G2, MC) can be grouped into three groups. Therefore, forty two centroid data are used as initial training data for BP. After that, some misclassified data that have reasonably high SI value are grouped together and their average vectors are added into the training set and retrain with the BP again.

We assume that a pattern is classified correctly if (output $\geq$0.6 and target =1) or (output $\leqq$ 0.3 and target =0). The final results of this experiment are shown in Table 2.

Table 2. Results from a BP method using centroid training data

	Classification Result													
	Joss Stick									Cigarette		Mos-quito coil	Total	Calssi-fication
Data	P3	P2	P1	B3	B2	B1	G3	G2	G1	ML	CM	MC		%
P3	39												40	98
P2		35		1									40	88
P1			34										40	85
B3	1			31			1						40	78
B2			2		37								40	93
B1						40							40	100
G3				2			35						40	88
G2					1			38					40	95
G1									40				40	100
ML										29	9		40	73
CM										8	29		40	73
MC												40	40	100

5.2 Discussion

The overall classification rates of all data in Table 2 are quite acceptable in most case however in case of cigarette smoke, both ML and CM can be achieved only 73% due to the similarity of both kinds of cigarette and the scattering of these data. It has been known that the level of classification highly depends on the quality of the training data [4]. During the grouping process by the method in the previous section, any kind of data that has a lot of ungrouped data trends to have a low classification rate. For example, ML data has 18 out of 40 data that could not be grouped. Therefore the classification rate of ML is much lower than the MC that has only 7 out of 40 data that could not be grouped.

This EN has been applied under various uncontrolled measuring environment, however the SI value of the data that have been measured under similar environment is quite high and they are grouped together by the proposed method. This could possibly mean that this EN has reproducibility qualification under a controlled environment.

The qualification of the burning material also has an effect to the signal data. The density of the ingredient of cigarette varies from one cigarette to another cigarette, therefore the signal of both CM and ML are more scattering than the joss stick and mosquito coil that have more constant density.

6. Conclusions

This paper has shown a proper method to handle the scattering data from an inexpensive EN that does not have a mechanism to control the measuring environment like the expensive EN in the market. By using a small number of effective centroid training data, the BP is able to classify not only the same kind of smoke from different brands, but also the same smoke from different concentration levels quite precisely.

The data that have been measured under similar environment have high SI value. Therefore, it could be summarized that these EN has reproducibility quail-fication under the controlled environment. This topic will be investigated in our further study.

Acknowledgement

We thank Mr. Nobuaki Murakami of FIS. Inc. for the technical information on making this electronic nose and for the data logger equipment.

References:

[1] P. Keller, L. Kangas, L. Liden, S. Hashem, R. Kouzes, "Electronic Noses and Their Applications," World Congress on Neural Networks (WCNN'96), San Diego, CA, USA, pp.928-931, 1996.

[2] Bancha Charumporn and Sigeru Omatu, Classifying Smokes using an Electronic Nose and Neural Network, SICE Annual Conference 2002, pp.2871-2875, Osaka, Japan, 2002

[3] William L. Carlson and Betty Thorne, Applied Statistical Methods, Prentice Hall International, 1997

[4] AP Engelbrecht, Sensitivity Analysis for Selective Learning by Feed forward Neural Networks, Fundamenta Informaticae, 45(1), pp.295-328, IOS press, 2001

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Web Page Classification Using Neural Networks Based on Augmented PCA

Ali Selamat, Hidekazu Yanagimoto and Sigeru Omatu
Computer and Systems Sciences,
Graduate School of Engineering,
Osaka Prefecture University,
Sakai, Osaka 599-8531, Japan.
aselamat@sig.cs.osakafu-u.ac.jp, {hidekazu, omatu}@cs.osakafu-u.ac.jp

Abstract

Automatic categorization is the only viable method to deal with the scaling problem of the World Wide Web (WWW). In this paper, we propose a news web page classification method (WPCM). The WPCM uses a neural network with inputs obtained by both the principal components and class profile-based features (CPBF). Each news web page is represented by the term-weighting scheme. As the number of unique words in the collection set is big, the principal component analysis (PCA) has been used to select the most relevant features for the classification. Then we augment the feature vectors generated from the PCA with the feature vectors from the class-profile which contains the most regular words in each class before feeding them to the neural networks for classification. These feature vectors are then used as the input to the neural networks for classification. The experimental evaluation demonstrates that the WPCM method provides acceptable classification accuracy with the sports news datasets.

1 Introduction

Neural networks have been widely applied by many researchers to classify the text documents with different types of feature vectors. Ruiz et al. [1] have used the X^2 measure to select the relevant features before classifying the text documents using the neural networks. Lam et al. [2] have used the principal component analysis (PCA) method as a feature reduction technique to the input data to the neural networks. However, if some original terms are particularly good when discriminating a class category, the discrimination power may be lost in the new vector space after using the Latent Semantic Indexing (LSI)[3]. Hence, the classification accuracy on the smaller classes can be degraded in the reduced dimensional space.

Here, we propose a web page classification method (WPCM), which is based on the PCA and the class profile-based features (CPBF). Each web page is represented by the term frequency-weighting scheme. As the dimensionality of a feature vector in the collection set is big, the PCA has been used to reduce it into a small number of principal components. Then we augment the feature vectors generated from the PCA with the feature vectors from the class-profile which contains the most regular words in each class before feeding them to the neural networks for classification.

The organization of this paper is as follows: The news classification using the WPCM is described in Section 2. The preprocessing of web pages and the PCA and WPCM approaches are explained in Section 3. The experiments and evaluation methods are discussed in Section 4. In Section 5, the conclusion of the classification results using the WPCM and other methods are stated.

2 Web page classification method

The high dimensionality of the news web pages dataset has made the process of classification difficult. This is because there are many categories of news in the web news pages such as sports, weathers, politics, economy, etc. In each category there are many different classes. For example, the classes that exist in the business category are stock market, financial investment, personal finance, etc. Our approach is based on the sports news category of web pages. In order to classify the news web pages, we propose the WPCM which uses the PCA and CPBF as the input to the neural networks. First, we have used the PCA algorithm to reduce the original data vectors to a small number of relevant features. Then we combine these features to the CPBF before inputting them to the neural networks for classification as shown in Fig. 1.

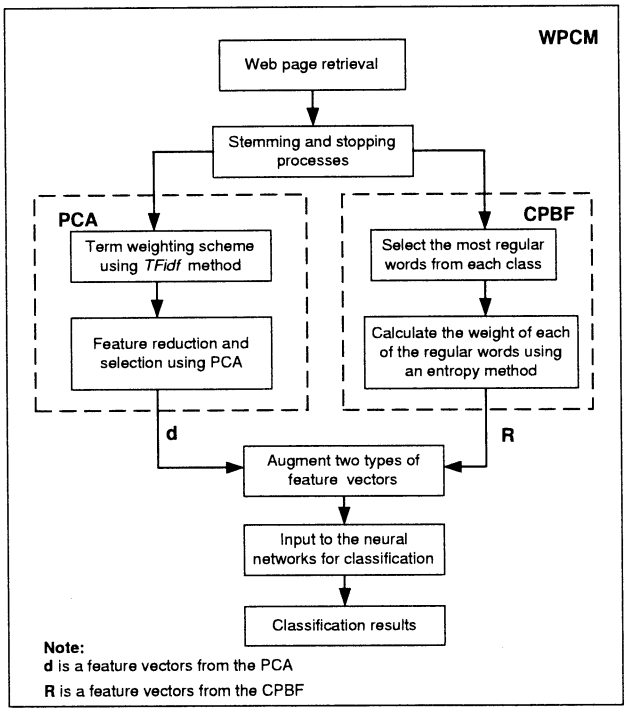


Figure 1: The process of classifying a news web page using the WPCM method.

3 Preprocessing of web pages

The process of classifying a news web page using the WPCM method is shown in Fig. 1. It consists of web news retrieval process, stemming and stopping processes, feature reduction process using our proposed method, and web classification process using error back-propagation neural networks (BP). After the stemming and stopping processes of the terms in each document, we will represent them as the document-term frequency matrix ($Doc_j \times TF_{jk}$). Doc_j is referring to each web page document that exists in the news database where $j = 1, \ldots, n$. Term frequency TF_{jk} is the number of how many times the distinct word w_k occurs in document Doc_j where $k = 1, \ldots, m$. The calculation of the terms weight x_{jk} of each word w_k is done by using a method as follows:

$$x_{jk} = TF_{jk} \times idf_k \tag{1}$$

where the document frequency df_k is the total number of documents in the database that contains the word w_k. The inverse document frequency $idf_k = log(\frac{n}{df_k})$ where n is the total number of documents in the database.

3.1 Feature reduction using the PCA

Suppose that we have $\mathbf{A}$, which is a matrix document-terms weight as below

$$\mathbf{A} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2k} & \cdots & x_{2m} \\ x_{31} & x_{32} & \cdots & x_{3k} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} & \cdots & x_{nm} \end{pmatrix}$$

where x_{jk} is the terms weight that exists in the collection of documents. The definitions of j, k, m, and n have been described in the previous paragraph. There are a few steps to be followed in order to calculate the principal components of data matrix $\mathbf{A}$. The mean of m variables in data matrix $\mathbf{A}$ will be calculated as follows

$$\overline{x}_k = \frac{1}{n}\sum_{j=1}^{n} x_{jk}. \tag{2}$$

After that the covariance matrix, $\mathbf{S} = \{s_{jk}\}$ is calculated. The variance s_{kk}^2 is given by

$$s_{kk}^2 = \frac{1}{n}\sum_{j=1}^{n} (x_{jk} - \overline{x}_k)^2. \tag{3}$$

The covariance s_{ik} is given by

$$s_{ik} = \frac{1}{n}\sum_{j=1}^{n} (x_{ji} - \overline{x}_i)(x_{jk} - \overline{x}_k) \tag{4}$$

where $i = 1, \ldots, m$. Then we determine the eigenvalues and eigenvectors of the covariance matrix $\mathbf{S}$ which is a real symmetric positive matrix. An eigenvalue λ and the eigenvector $\mathbf{e}$ can be found such that, $\mathbf{Se} = \lambda\mathbf{e}$.

In order to find the eigenvector $\mathbf{e}$ the characteristic equation $|\mathbf{S} - \lambda\mathbf{I}| = 0$ must be solved. If $\mathbf{S}$ is an $m \times m$ matrix of full rank, m eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_m)$ can be found. By using

$$(\mathbf{S} - \lambda_i\mathbf{I})\mathbf{e}_i = 0, \tag{5}$$

all corresponding eigenvectors can be found. The eigenvalues and corresponding eigenvectors will be sorted so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. Let a square matrix $\mathbf{E}$ be constructed from the eigenvector columns where $\mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_3\ \mathbf{e}_4\ \ldots\ \mathbf{e}_m]$. Also let us denote

$\mathbf{\Lambda}$ as

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ 0 & 0 & 0 & 0 & \lambda_m \end{pmatrix}.$$

In order to get the principal components of matrix $\mathbf{S}$, we will perform eigenvalue decomposition which is given by

$$\mathbf{E}^{\mathrm{T}}\mathbf{S}\mathbf{E} = \mathbf{\Lambda}. \tag{6}$$

Then we select the first $d \leq m$ eigenvectors where d is the desired value, e.g., 100, 200, 400, etc.

3.2 Feature selection using the CPBF

For the feature selection using a class profile-based approach, we have manually identified the most regular words that exist in each category and weighted them using the entropy weighting scheme before adding them to the feature vectors that have been selected from the PCA. For example, the words that exist regularly in a boxing class are 'Lewis', 'Tyson', 'heavyweight', 'fighter', 'knockout', etc. Then a fixed number of regular words from each class will be used as a feature vector together with the reduced principal components from the PCA. These feature vectors are then used as the input to the neural networks for classification. The entropy weighting scheme on each term is calculated as $L_{jk} \times G_k$ where L_{jk} is the local weighting of term k and G_k is the global weighting of term k. The L_{jk} and G_k are given by

$$L_{jk} = \begin{cases} 1 + \log TF_{jk} & (TF_{jk} > 0) \\ 0 & (TF_{jk} = 0) \end{cases} \tag{7}$$

and

$$G_k = \frac{1 + \sum_{k=1}^{n} \frac{TF_{jk}}{F_k} \log \frac{TF_{jk}}{F_k}}{\log n} \tag{8}$$

where n is the number of document in a collection and TF_{jk} is the term frequency of each word in Doc_j as mentioned previously. The F_k is a frequency of term k in the entire document collection. We have selected $R = 50$ words that have the highest entropy value to be added to the first d components from the PCA to be an input to the neural networks for classification as shown in Fig. 1.

3.3 Input data to the neural networks

The high dimensionality of feature vectors to be an input to the neural networks is not practical due to poor scalability and performance. Therefore, the PCA has been used to reduce the original feature vectors $m = 1,800$ into a small number of principal components. In our case, we have selected a few values of d (e.g., 100, 200, 300, 400, 500, and 600) together with $R = 50$ features selected from the CPBF approach because this parameter performs better for web news classification compared to the other parameters to be an input to the neural networks. The value of d contributes 81.6% of proportions from the original feature vectors. We have selected 400 features from the PCA and 50 features from CPBF methods.

Table 1: The number of documents that are stored in the news database. These are the classes that exist in sports category.

Class no.	Class name	Number of documents
1.	baseball	569
2.	boxing	210
3.	cycling	80
4.	football	550
5.	golf	456
6.	hockey	856
7.	motor-sports	405
8.	rugby	52
9.	skiing	233
10.	soccer	261
11.	swimming	169
12.	tennis	255
	Total	4,096

4 Experiments

We have used a web page dataset from the Yahoo sports news as shown in Table 1. The types of news in the database are baseball, boxing, cycling, football, golf, hockey, motor-sports, rugby, skiing, soccer, swimming, and tennis. The total of documents are 4,096. For training, we have selected randomly 1,000 documents from different classes. The rest of the documents are used as the test sets. For the PCA approach, we have selected 400 principal components and input them directly to the neural networks for classification. The PCA and WPCM feature selection methods for the input to the BP are evaluated using the standard information retrieval measures that are *Precision*, *Recall*, and $F1$. *Recall* is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. *Precision* is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. They are given by

$$Recall = (a/(a+b)) \times 100 \tag{9}$$

$$Precision = (a/(a+c)) \times 100 \tag{10}$$

where, a is the number of relevant record retrieved, b is the number of relevant record not retrieved, and c is the number of irrelevant record retrieved. The $F1$ measure is a kind of average of *Precision* and *Recall*. It is given by

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \times 100. \quad (11)$$

Table 2: The classification results using the PCA method.

Class no.	*Precision* (%)	*Recall* (%)	$F1$ (%)
1.	89.82	100.00	94.64
2.	84.75	100.00	91.74
3.	71.60	72.50	72.05
4.	97.06	99.00	98.02
5.	100.00	97.00	98.48
6.	90.83	99.00	94.74
7.	97.09	100.00	98.52
8.	100.00	100.00	100.00
9.	99.01	100.00	99.01
10.	71.26	67.00	67.07
11.	82.62	95.00	92.23
12.	69.44	75.00	72.12
Average	87.79	92.04	89.89

Table 3: The classification results using the WPCM method.

Class no.	*Precision* (%)	*Recall* (%)	$F1$ (%)
1.	97.14	68.00	80.00
2.	86.96	100.00	93.02
3.	98.68	93.75	96.15
4.	86.09	99.00	92.09
5.	89.91	98.00	93.78
6.	94.29	99.00	96.59
7.	84.03	100.00	91.32
8.	97.62	100.00	98.80
9.	89.29	100.00	94.34
10.	73.91	68.00	70.83
11.	90.09	100.00	94.79
12.	96.15	100.00	98.04
Average	90.35	93.81	91.65

4.1 Results

The classification results using the PCA and WPCM methods are shown in Tables 2 and 3. The average for *Precision*, *Recall*, and $F1$ measures using the PCA approach are 87.79%, 92.04%, and 89.89%, respectively. In comparison with the WPCM approach, the *Precision*, *Recall*, and $F1$ measures are 90.35%, 93.81%, and 91.65%, respectively. The $F1$ measure using the PCA approach for the cycling and rugby classes (class numbers 3 and 8) are 72.05% and 100%, respectively, as shown in Table 2. But for the WPCM approach the classification accuracies for both classes are 96.15% and 98.80%, respectively, as shown in Table 3. Furthermore, we have compared the WPCM with the TF-IDF and Bayesian methods using the F1 measure. They are 78.06% and 81.07% for the TF-IDF and Bayesian methods as shown in Table 4.

These indicate that the WPCM approach has been able to classify the documents correctly although the number of documents representing the class is small.

Table 4: The classification results using the $F1$ measure for the TF-IDF and Bayesian methods.

Class no.	TF-IDF (%)	Bayesian (%)
1.	83.04	83.04
2.	88.89	90.48
3.	100.00	100.00
4.	79.39	78.79
5.	86.86	81.75
6.	68.03	86.89
7.	56.46	59.49
8.	50.00	75.00
9.	91.30	89.86
10.	88.46	80.77
11.	64.00	56.00
12.	80.26	90.79
Average	78.06	81.07

5 Conclusions

We have overcome the limitation of the PCA in supervised data where the characteristic variables that describe smaller classes tend to be lost as a result of the dimensionality reduction by using the WPCM approach. Furthermore, the classification accuracy on the small classes can be improved although they have been reduced into a small number of principal components. Although the classification accuracy using the WPCM approach is high in comparison with other approaches, the time taken for training is relatively long compared with other methods.

References

[1] E.M. Ruiz, P. Srinivasan (2002), Hierarchical text categorization using neural networks, Information Retrieval 5:87-118

[2] S.Y. Lam and D.L. Lee (1999), Feature reduction for neural network based text categorization,Proceedings of the 6th ICDS, Hsinchu, Taiwan, April 1999

[3] F. Sebastini (2002), Machine learning in automated text categorization, ACM Computing Surveys 34:1-47

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Neural-Net Switching Controller for Partly Known Robot Systems with Guaranteed Tracking Performance

Sisil Kumarawadu, Keigo Watanabe, Kazuo Kiguchi, and Kiyotaka Izumi
Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University, 1-Honjomachi, Saga 840-8502, Japan
†E-mail : sisilkuma@ieee.org, {watanabe, kiguchi, izumi}@me.saga-u.ac.jp

Abstract

This paper presents a use of two types of neural networks (NNs) to optimize output tracking performance of partly known robot systems. A switching approach based on genetic algorithm optimization is adopted to best combine the different adaptation characteristics of the two neural architectures. Stability of this neural-net switching controller is guaranteed by using already stabilized NNs and a switching function that ensures minimum lumped tracking error. Results of simulations on our active binocular head are reported.

Keywords: Neural networks, Switching control, SoftMax functions, Gaussian-sum networks.

1 Introduction

Owing to their universal approximation property and efficient adaptation capabilities, artificial neural networks (NNs) have become popular to be used for adaptive control of nonlinear systems. Different NN architectures exhibit different adaptation properties when they deal with dynamic situations wherein robot's model parameter variations, changes in performance specifications or changes of the environment that the robot operates occur. Therefore, in an NN adaptive controller, to use few different network architectures, which can complement each other would definitely be advantageous than using only a single type of NN. Determining an optimum switching function, which best combines useful characteristics of different NNs to optimize the overall controller performance is always thorny.

We use two neurally-inspired adaptive compensating schemes to solve the output-tracking problem of partially known robotic systems. The controller consists of a model-based term and an NN on-line adaptive compensation term. The NN-based adaptive module, which is used in the underlined control strategy consists of two types of neural network, namely an altitude adaptive SoftMax function network [1] and a Gaussian-sum network. Genetic algorithm (GA) is then used to find the optimal combination of these two types (optimal switching function) to be used in the adaptive output-tracking scheme. The two aforementioned NNs outperform each other during different time periods of a given tracking epoch, with scenarios being different for different joints of the robot. As an on-line NN adaptive model, besides to its better performance during the initial stages of the on-line adaptation process, SoftMax function network is less vulnerable to robot's model parameter variations and sudden motion changes. On the other hand, a Gaussian-sum network outperforms the former during the steady state.

Each NN is decomposed into neural subnets to separately take care of each of the robot's degrees-of-freedom allowing different weight initializations and separate tuning making for a faster weight update procedure.

2 Implementation of the Controller

The control approach is based on Computed Torque Control (CTC) as seen in **Fig. 1**. With familiar notations, $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$ is the position error between the desired position and the sensed position, and $K_v, K_p \in R^{n\times n}$ are the diagonal gain matrices with constant positive elements. The purpose of the neural module is to bestow the controller with adaptation properties and to enhance the tracking performance of the partly known robotic system by adding an NN output signal vector, $\hat{\boldsymbol{v}}$ to modify the acceleration vector, $\boldsymbol{u}$. We assume that the robot model nonlinearities are completely unknown and use only a highly approximated inertia matrix, $\hat{M}(\boldsymbol{q})$, alone in the controller. Rigid body dynamics of the head can be written as [2]

$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \quad (1)$$

where M is the inertia matrix, $\boldsymbol{h}$ comprises centrifugal, Coriolis, gravitational and frictional terms. Substituting the controller torque command for $\boldsymbol{\tau}$ in (1), yields

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \hat{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_d + K_v\dot{\boldsymbol{e}} + K_p\boldsymbol{e} + \hat{\boldsymbol{v}}) \quad (2)$$

or

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \hat{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \hat{M}(\boldsymbol{q})(\ddot{\boldsymbol{e}} + K_v\dot{\boldsymbol{e}} + K_p\boldsymbol{e} + \hat{\boldsymbol{v}}) \quad (3)$$

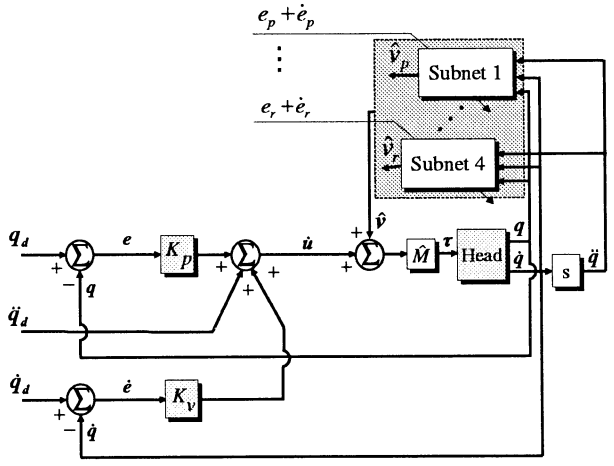


Figure 1: Overview of the control approach when the NNs are online adapted. s is the Laplace operator

allowing us to write

$$\begin{aligned} \ddot{e} + K_v\dot{e} + K_p e &= [\hat{M}^{-1}(q)M(q) - I]\ddot{q} \\ &\quad + \hat{M}^{-1}(q)h(q,\dot{q}) - \hat{v} \\ &= (v + \varepsilon) - \hat{v} \\ &= \tilde{v} \end{aligned} \tag{4}$$

where v is the NN signal vector, which corresponds to the ideal weights of the NN, and ε ($\|\varepsilon\| < \varepsilon_N$) may consist of desired $\tilde{v}$ and any approximation error. The goal of the NN model is to maintain a $\tilde{v}$ term in the nonhomogeneous differential equation (4) so that e is as close as possible to the null vector of its dimension. Understandably, the NN output signal vector is a complex function of robot's joint angle positions, angular velocities and accelerations. The complete mathematical formalism and analysis of the NNs module can be found in [1].

3 Neural Network Models

3.1 The activation functions

SoftMax function with variable altitude Gaussian functions has the analytic expression:

$$\Phi_i(x, \varphi_i, a_i) = \frac{G\left(\|x - \varphi_i\|, a_i\right)}{\sum_j G\left(\|x - \varphi_j\|, a_j\right)} \tag{5}$$

where

$$G(x, a) = \exp\left(f \ln a - \frac{x^2}{2\sigma^2}\right). \tag{6}$$

Here, f and a_i are the fan-in and the altitude of the ith neural unit respectively. In (6), the Gaussian functions, $G(\cdot)$s, are of equal variance but different altitudes, and $\|\cdot\|$ denotes Euclidian norm. φ_i is the feature's prototypical or center state, where $\Phi_i(\cdot)$, which is a type of hyperradial basis function has a maximum value and x is the f-dimensional input vector.

The analytic expression of Gaussian-sum function adopted here can be written by

$$\Psi_i(x, \varphi_i, \sigma_i) = \sum_{j=1}^{f} \exp\left(-\frac{(x_j - \varphi_{ij})^2}{2\sigma_i^2}\right) \tag{7}$$

where Ψ_i is the activation of the ith neuron and σ_i^2 is the variance of the Gaussian functions associated with the ith neuron. x_j is the jth element of f-dimensional input vector x. The SoftMax basis function architecture adopted in this work uses radial combination functions but activation function is SoftMax as described by (5) and (6) making the sum of activations of the hidden units to equal one. On the other hand, according to (7), maximum activation of a hidden unit of Gaussian-sum network can be as large as f when each element of x is equal to a value of center state φ_{ij}. Therefore the sensitivity of a Gaussian-sum hidden neuron is much higher than that of a SoftMax hidden neuron.

3.2 Adaptation laws of the NNs

Each NN module is designed in terms of neural subnets as shown in **Fig. 1**. Each subnet is a three-layered NN with linear processing units in the output layer. As dynamics of the binocular robotic head is highly coupled, each subnet is provided with the same input vector $[q^T, \dot{q}^T, \ddot{q}^T]^T$, where $q = [\theta_p, \theta_t, \theta_l, \theta_r]^T$; $\theta_p, \theta_t, \theta_l$ and θ_r are pan, tilt, left and right camera angles respectively.

The procedure involves separately tuning of individual subnets, making for a faster weight updates and independent weight initializations. The weight (current values of connecting weights $\hat{W}$ and altitudes $\hat{a}$) update laws for the kth subnet ($k = 1, 2, 3, 4$, representing the subnets for pan, tilt, left and right camera joints respectively) of altitude adaptive SoftMax net are as follows [1]:

$$\begin{aligned} \dot{\hat{W}}_k &= \rho_k r_k \hat{\Phi}_k^T - \kappa_k \rho_k \|E_k\| \hat{W}_k \\ \dot{\hat{a}}_k &= r_k \hat{W}_k \hat{\Phi}_k' H_k - \kappa_k \hat{a}_k \|E_k\| H_k \end{aligned} \tag{8}$$

where $E_k = [e_k^T, \dot{e}_k^T]^T$ and $r_k = (e_k + \dot{e}_k)$. $\rho_k > 0$ is a scalar parameter, $H_k = H_k^T > 0$ is any constant matrix and $\kappa_k > 0$ is a design parameter. $\hat{\Phi}_k$ represents the vector of activations of the hidden-layer neurons. The partitioned procedure makes it possible for each subnet to adaptively take care of the performance of the part of the robot, which it is dedicated to. $\hat{\Phi}_k' \in \Re^{N_k \times N_k}$, in which N_k ($k = 1, 2, 3, 4$) is the number of neural units in the kth subnet, is given by

$$\hat{\Phi}_k' = \begin{bmatrix} d\Phi_1/da_1 & d\Phi_1/da_2 & \cdots & d\Phi_1/da_{N_k} \\ d\Phi_2/da_1 & d\Phi_2/da_2 & \cdots & d\Phi_2/da_{N_k} \\ \vdots & \vdots & \ddots & \vdots \\ d\Phi_{N_k}/da_1 & d\Phi_{N_k}/da_2 & \cdots & d\Phi_{N_k}/da_{N_k} \end{bmatrix}.$$

The entries of $\hat{\Phi}'_k$ can be calculated using the generalized expressions

$$\begin{cases} d\Phi_i/da_i &= \dfrac{Y - X_i}{Y^2}\left(\dfrac{f}{a_i}X_i\right) \\ d\Phi_i/da_j &= \dfrac{-X_i}{Y^2}\left(\dfrac{f}{a_j}X_j\right); \quad i \neq j \end{cases}$$

where X_i and Y represent the numerator and denominator terms of Φ_i in (5) respectively.

Likewise, weight (current values of connecting weights $\hat{W}$ and standard deviations $\hat{\sigma}$) update laws for Gaussian-sum subnets are as follows:

$$\begin{aligned} \dot{\hat{W}}_k &= \rho_k r_k \hat{\Psi}_k^T - \kappa_k \rho_k ||E_k|| \hat{W}_k \\ \dot{\hat{\sigma}}_k &= r_k \hat{W}_k \hat{\Psi}'_k H_k - \kappa_k \hat{\sigma}_k ||E_k|| H_k \end{aligned} \tag{9}$$

where $\hat{\Psi}_k$ represents the vector of activations of the hidden-layer neurons, and

$$\hat{\Psi}'_k = \begin{bmatrix} d\Psi_1/d\sigma_1 & 0 & \cdots & 0 \\ 0 & d\Psi_2/d\sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d\Psi_{N_k}/d\sigma_{N_k} \end{bmatrix}.$$

4 Combinations of NNs Using GAs

This section describes the use of GA to optimize combinations between two types of aforementioned NN models, which result in an optimal tracking performance. In the simulation experiments, the robotic head tracks the centroid of a moving target to keep it in view at the image centers of the two cameras. The target moves along simulated elliptical trajectories with an angular velocity of π [rad/s] in space and completes 120 elliptical cycles. The total duration of an experiment is divided into 120 equal time sections and a binary string of length 4 is assigned to each time section, representing choices of types of network for each degree-of-freedom. This results in a binary chromosome of length (bit size) 480 with binary value 0 representing the choice of one type of the networks and binary value 1 representing the other type.

Sum of the mean squared errors between command set points and actual angular positions over four joints over entire 120 motion cycles is used as the index to the fitness function, which, for the ith individual, is described as follows:

$$(fitness)_i = \sum_{l=1}^{p} \frac{1}{m} \sum_{k=1}^{m} \sum_{j=1}^{n} e_{ij}^2(k)$$

where n is robot's number of degrees-of-freedom, p is the number of trajectory cycles, m is the number of samples per cycle and k is a time index, which resets to 1 at the beginning of each new cycle.

The population size for GA optimization is 100. The mutation rate is chosen to be 1/(bitsize). Crossover rate is 0.6 and selection criteria is "best 10".

5 Results and Discussion

Four tests are carried out on our four degrees-of-freedom active binocular head. The first involves the use of altitude adaptive SoftMax function network as the adaptive control module in the direct closed-loop controller. The second test involves the use of Gaussian-sum network to take on the same task. In each of those two tests, the NN output signal vectors to the controller at each sampling instant are recorded separately. In the third experiment, the best individual from the 100th generation of GA is used. NNs output signal vectors are read from the look-up tables according to the switching function from GA. The last test is conducted to visualize the amount of performance degradation in the absence of contribution of the NNs.

For a fair comparison between two types of network, the following constrains are made:

- Object trajectories and robot parameter changes used are the same.
- Adaptation law parameters are fixed constant and will remain unchanged during the experiments.
- The total number of weights, which change their values when NNs are adapted online, is the same.
- Center states are initialized similarly and remain unchanged during the experiments.

The parameters of the SoftMax subnets are selected as $\rho_1 = \rho_2 = 9.5$, $H_1 = H_2 = \text{diag}(0.01, 0.01, 0.01)$, $\kappa_1 = \kappa_2 = 0.001$. $\rho_3 = \rho_4 = 11.0$, $H_3 = H_4 = \text{diag}(0.02, 0.02, 0.02)$, $\kappa_3 = \kappa_4 = 0.001$. Three neural units are used in each subnet. Connecting weights are initialized in the range $[-0.7, 0.7]$ and altitudes in the range [0.8, 1.2] for all subnets. The standard deviation, σ, is constant at $20/\sqrt{2}$ for all the neurons.

The parameters of the Gaussian-sum subnets are chosen as $\rho_1 = \rho_2 = 0.17$, $H_1 = H_2 = \text{diag}(0.01, 0.01, 0.01)$, $\kappa_1 = \kappa_2 = 0.001$. $\rho_3 = \rho_4 = 0.22$, $H_3 = H_4 = \text{diag}(0.01, 0.01, 0.01)$, $\kappa_3 = \kappa 4 = 0.001$. Three neural units are used in each subnet. Connecting weights are initialized in the ranges $[-0.5, 0.5]$ and standard deviations σ in the range $[-0.5/\sqrt{2}, 0.5/\sqrt{2}]$ for all subnets.

The center states are set in the ranges $[-0.5, 0.5]$ similarly for the two types of network. CTC gain matrices are selected as $K_v = K_p = \text{diag}(15.0, 15.0, 15.0, 15.0)$.

Simulation experiments are carried out for simulated elliptical target trajectories with an angular velocity of π [rad/s] in 3D space. Sudden changes in target motion are made by changing the parameters of the ellipses. Although it is less relevant to the binocular tracking problem, robot's model parameter changes are made by changing the values of the masses of the head component parts to check on the adaptation properties of two types of NN. The desired joint angle trajectories to be used by the servo part of

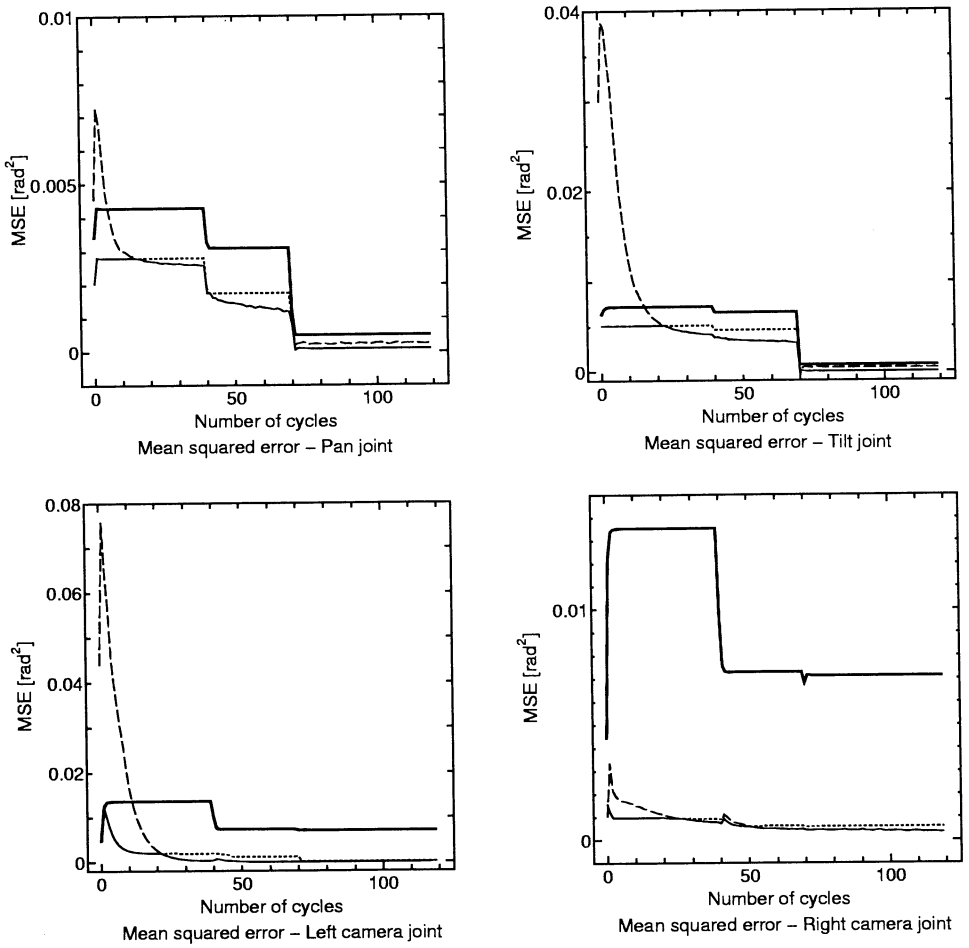


Figure 2: Tracking error in all four joints, where dotted line: altitude adaptive SoftMax NN; dashed line: Gaussian-sum NN; solid line (thin): best individual from GA; solid line (thick): without NNs.

the controller are computed using a suitable kinematic model [3].

The mean squared values of tracking errors (MSE) for all four experiments are shown depicted **Fig. 2**.

6 Conclusion

This paper has presented a use of two direct adaptive neural network architectures and genetic algorithms to enhance the output tracking performance of partly known robotic systems. Two NNs used in this work complement each other giving away much improved combined effect. The performance of the neural-net switching controller is guaranteed in the sense that the error convergence is guaranteed.

References

[1] S. Kumarawadu, K. Watanabe, K. Kiguchi, K. Izumi. "Adaptive Output Tracking of Partly Known Robotic Systems Using SoftMax Function Networks," in *Procs. of the IEEE World Congress on Computational Intelligence*, Honolulu, Hawaii: pp. 483–488 (IJCNN'02), 2002.

[2] J. Craig. Introduction to Robotics: Mechanics and Control. Reading, MA: Addision-Wesley, 1986.

[3] S. Kumarawadu, K. Watanabe, K. Kiguchi, K. Izumi. "An application of active vision head control using model-based compensating NNs controller," in *Procs. of the International Conference on Control, Automation and Systems*, Cheju, Korea: pp. 879–882, 2001.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Evolutionary Acquisition of Handstand Skill Using a Three-Link Rings Gymnastic Robot

Takaaki Yamada*,†, Keigo Watanabe*, Kazuo Kiguchi* and Kiyotaka Izumi*
*Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University, 1 Honjomachi, Saga 840-8502, Japan
†E-mail: t-yamada@ieee.org

Abstract

We have already proposed a "rings gymnastic robot" and have been mainly acquiring some skills. Handstand skill has been already acquired by using the three-link robot, in which the angles of the hip and shoulder joints throughout a handstand were not taken into account, though such angles are important factors in the skill and appearance of handstand. In this paper, handstand skill is newly acquired by considering the joint angles throughout the handstand.

1 Introduction

We have already proposed a "rings gymnastic robot" [1]. The purpose of studies on the rings gymnastic robot is to understand rings exercises through the robot and to apply the acquired skill to the gymnastic coaching. So far we have been mainly studying the skills acquisition of some performances. Performance skill of handstand, which is one of the compulsory requirements in the rings event, has been already acquired using the two- and three-link models in an evolutionary manner [1, 2]. In acquiring these handstand skills, joint angles throughout a handstand were not considered, although they are significant factors from the viewpoint of the skill and appearance of handstand. Therefore, a fitness function incorporating the joint angles throughout a handstand is introduced in this paper. Handstand skill of the three-link rings gymnastic robot is acquired as fuzzy control rules by using a genetic algorithm (GA) with the fitness function.

2 Three-Link Model

Although ring exercises are essentially complex three-dimensional motions, a three-link rings gymnastic robot that moves in two-dimensional plane is used in this paper as the preliminary step toward its analysis in a three-dimensional model and experiments. Note that Sprigings *et al.* [3] indicate that the performance skill to realize a backward giant circle appropriate for the transition to a handstand does not become easier, though arm abduction is not considered in a two-dimensional model.

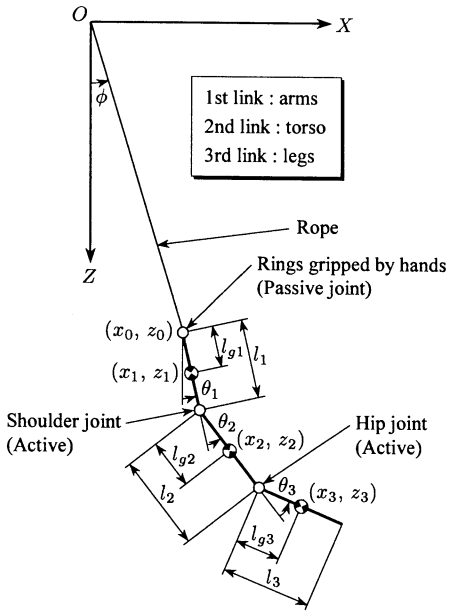


Figure 1: Three-link model of rings gymnastic robot

Assuming that the robot's arms, torso and legs are represented by one rigid link respectively leads to the model as shown in **Fig. 1**. This model corresponds to a triple physical pendulum suspended by a rope. In such a model, the shoulder and hip joints can be driven. In Fig. 1, (x_1, z_1) is the coordinate of mass center of link 1, θ_1 denotes the angle between Z-axis and link 1, θ_2 represents the angle of the shoulder joint,

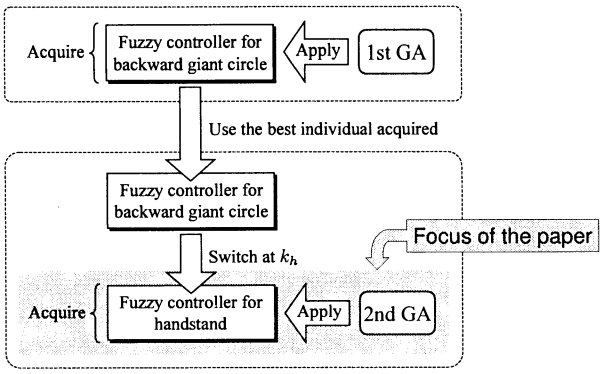


Figure 2: Procedure for acquiring parameters of controllers by two-stage approach and focus of the paper

i.e., the angle between links 1 and 2, and θ_3 is the angle of the hip joint, i.e., the angle between links 2 and 3.

Letting the generalized coordinate be $\boldsymbol{q} = [x_1\ z_1\ \theta_1\ \theta_2\ \theta_3]^T$, the dynamic equation of the model derived using the Lagrangian formulation is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Q} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is a 5×5 symmetric mass matrix, $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is a 5×1 vector of centrifugal and Coriolis terms, $\boldsymbol{g}(\boldsymbol{q})$ is a 5×1 vector of gravity terms, and $\boldsymbol{Q}$ is a 5×1 vector of generalized force terms. Elements of the vector of generalized force terms are described as follows:

$$\begin{aligned} &Q_1 = F_r \sin\phi, \quad Q_2 = F_r \cos\phi, \\ &Q_3 = F_r l_{g1} \sin(\theta_1 - \phi), \\ &Q_4 = \tau_2 + T_2, \quad Q_5 = \tau_3 + T_3. \end{aligned}$$

The first joint, i.e., the robot's end point is passive. τ_i $(i = 2, 3)$ is the input torque to the joint i. The tension F_r of the rope and the torque T_i caused by impedance at the joint i are given in the same way as [1].

3 Procedure for Acquiring Performance Skill

Although this paper is focused on handstand skill, a handstand from backward giant circle is adopted as the whole performance. In order to realize such a performance, it is divided into two basic exercises, i.e., a backward giant circle and a handstand. A fuzzy controller is prepared for each basic exercise. One of the controllers is selected according to the situation. Parameters of the fuzzy control rules are determined with a two-stage approach as shown in **Fig. 2**. In Fig. 2 k_h denotes the discrete-time instant at which the controller for the backward giant circle is switched to that for the handstand. In the first stage, parameters of the controller for the backward giant circle are acquired by the first GA. In the second stage, those for the handstand are obtained by the second GA. The backward giant circle before the handstand is performed by using the best controller acquired in the first stage. In this paper, the fuzzy control rules for backward giant circle obtained in [2] are used and rule parameters for the handstand are only determined by the fitness functions proposed, because the objective performance is only handstand.

4 Fuzzy Control Rules for Handstand

The realization condition for handstand has been already proposed in [1]. In this section, fuzzy control rules are described to do a handstand using its condition. From [1], the angle between Z-axis and force vector $\boldsymbol{F} = [F_x\ F_z]^T$ acting on the robot's mass center G is expressed by

$$\psi = \mathrm{atan2}(F_x, F_z). \tag{2}$$

Also, the angle between Z-axis and a straight line connecting $(x_0,\ z_0)$ to $(x_g,\ z_g)$ is written by

$$\psi' = \mathrm{atan2}(x_0 - x_g, z_0 - z_g). \tag{3}$$

Here, $(x_g,\ z_g)$ denotes the coordinate of robot's mass center and $(x_0,\ z_0)$ is the coordinate of robot's end point, i.e., the first joint. A handstand can be realized if ψ and ψ' are equal. Therefore $(\psi - \psi')$ is given as one of the inputs to the antecedent part of the fuzzy reasoning. The input torque depends on the angle ϕ between Z-axis and the rope and on its rate $\dot{\phi}$ as well as $(\psi - \psi')$, so that ϕ and $\dot{\phi}$ are also given as the inputs to the antecedent part.

The fuzzy control rules on input torques τ_i $(i = 2, 3)$ are described as follows:

Rule 1: If $(\psi - \psi')$ is ${}^1D_i$ and ϕ is ${}^1E_i$ and $\dot{\phi}$ is ${}^1F_i$ then $\tau_i = {}^1c_i$

$\vdots$

Rule 27: If $(\psi - \psi')$ is ${}^{27}D_i$ and ϕ is ${}^{27}E_i$ and $\dot{\phi}$ is ${}^{27}F_i$ then $\tau_i = {}^{27}c_i$.

If all inputs are zero, the handstand is stable; in this case, the input torque has to be zero. In order to satisfy such a condition, a triangular and two trapezoidal functions shown in **Fig. 3** are used as the antecedent membership functions for each input. The parameters

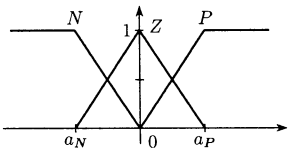


Figure 3: Antecedent membership functions for handstand

Table 1: Parameters of antecedent membership functions for handstand

Inputs	a_N	a_P
$\psi - \psi'$	$-\pi/90$	$\pi/90$
ϕ	$-\pi/18$	$\pi/18$
$\dot{\phi}$	-1	1

of the membership functions for each input are tabulated in **Table 1**. Note here that the antecedent grade is given as the product of the grade for each membership function.

Joint impedance is adjusted by changing the spring constants k_i $(i = 2, 3)$ at each joint according to the following equation:

$$k_i = \begin{cases} \gamma_i k_i, & \text{within range of motion} \\ k_i, & \text{otherwise} \end{cases} \quad (4)$$

where γ_i denotes a changing rate for the spring constant k_i. While doing a handstand, γ_i is kept to be an output value from the controller for the backward giant circle at the transition from the backward giant circle to the handstand.

5 Evolutionary Acquisition of Rule Parameters

5.1 Genetic algorithm

In GA, the number of individuals is 60. A uniform crossover is used with a crossover rate of 0.6 and a mutation rate is 0.01. A tournament strategy with 3 individuals is adopted in the selection. An elite strategy of 6 individuals is used in an alternation of generations.

In the fuzzy control rules for the handstand, the consequent constants ${}^{14}c_i$ $(i = 2, 3)$, whose fuzzy labels are all Z in the antecedent part, are set to zero. Other 52 constants, i.e. ${}^{1}c_i$–${}^{13}c_i$ and ${}^{15}c_i$–${}^{27}c_i$, are determined with GA. The search space of these consequent constants is set in the range of $[-200, 100]$.

5.2 Fitness functions

In GA for the handstand, the fuzzy controller for the backward giant circle is selected at the beginning of the simulation. It is switched to that for the handstand at k_h [2]. The handstand skill is evaluated successively while the controller for the handstand is selected.

The fitness function is given by

$$f_h = f_{h1} + f_{h2}. \quad (5)$$

Here, f_{h1} denotes a stability of the handstand and f_{h2} evaluates the joint angles throughout the handstand. By minimizing this fitness function, acquiring a desirable handstand as not only its skill but also appearance is expected. f_{h1} is written in detail by

$$f_{h1} = f'_{h1} + 10 f''_{h1} \quad (6)$$

where

$$f'_{h1} = \frac{1}{k' - k_h + 1} \sum_{i=k_h}^{k'} |s_h(i)| + \max\{|s_h(k_h)|, ..., |s_h(k')|\} \quad (7)$$

$$f''_{h1} = p. \quad (8)$$

f'_{h1} is the sum of the mean and maximum values of $|s_h| = |\psi - \psi'|$, which represents a stability of the handstand [4]. f''_{h1} denotes a penalty in the case where the handstand is unsuccessful. On the other hand, f_{h2} in equation (5) is further expressed by

$$f_{h2} = f'_{h2} + f''_{h2} \quad (9)$$

where

$$f'_{h2} = \frac{1}{k' - k_h + 1} \sum_{i=k_h}^{k'} \{|\theta_2(i)| + |\theta_3(i)|\} \quad (10)$$

$$f''_{h2} = \max\left[\max\{|\theta_2(k_h)|, ..., |\theta_2(k')|\}, \max\{|\theta_3(k_h)|, ..., |\theta_3(k')|\}\right]. \quad (11)$$

f'_{h2} is the sum of the mean angles of the shoulder and hip joints throughout the handstand. f''_{h2} denotes the biggest angle out of the maximum angles of the shoulder and hip joints throughout the handstand. p and k' in equations (7)–(11) are given as follows. If $|s_h(k)| > \pi/90$ [rad] is satisfied, then its handstand is treated as a failure. In this case,

$$p = (k_{fh} + 1) - k_p \quad (12)$$

$$k' = k_p \quad (13)$$

are used and the calculation of the individual is stopped. Here, k_p is the discrete-time instant at which

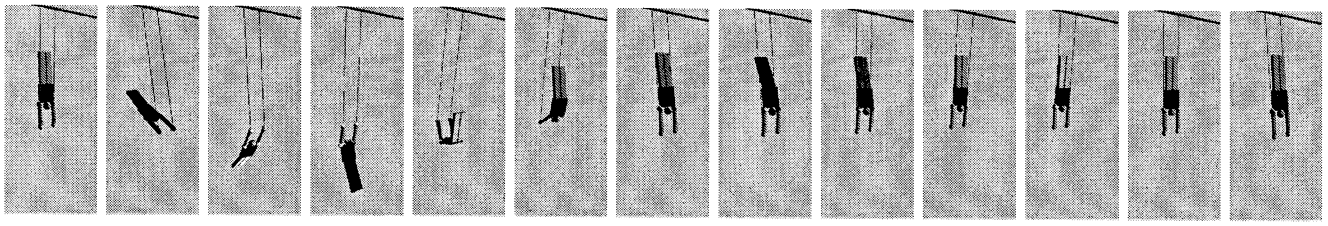


Figure 5: Graphics sequence of the rings gymnastic robot performing a handstand from backward giant circle

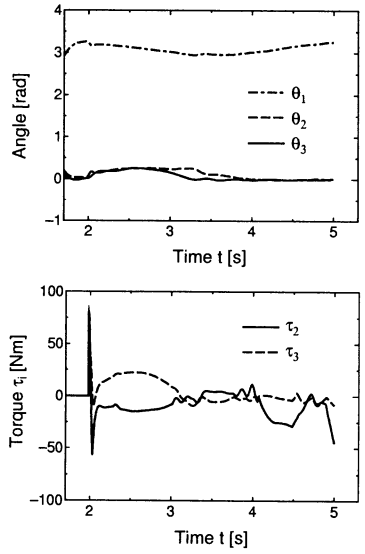


Figure 4: Simulation results

the above inequality is satisfied and k_{fh} denotes the predefined final discrete-time instant in simulation. On the other hand, if $|s_h(k)| \leq \pi/90$ [rad] is satisfied until k_{fh},

$$p = 0 \tag{14}$$

$$k' = k_{fh} \tag{15}$$

are given.

6 Simulation

In this simulation, the parameters and the initial states of the rings gymnastic robot shown in [2] are used. The simulation time is 5 [s] and the sampling period is 1 [ms]. The parameters of the fuzzy control rules are optimized by genetic operation of 5000 generations in GA. The simulation results performed by the obtained skill are shown in **Fig. 4**. The graphics sequence based on the results is depicted in **Fig. 5**. From these results, it is found that the handstand is realized and the joint angles throughout the handstand are small.

7 Conclusions

In this paper, the parameters of the fuzzy control rules for a handstand has been acquired by the evolutionary method using the proposed fitness functions, which took both a stability and an appearance of the handstand into consideration by incorporating the joint angles throughout the handstand. The effectiveness of the obtained skill and the proposed evaluation was illustrated through a simulation.

References

[1] T. Yamada, K. Watanabe, K. Kiguchi, and K. Izumi, "Acquisition of Fuzzy Control Based Exercises of a Rings Gymnastic Robot," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2584–2589, 2001.

[2] T. Yamada, K. Watanabe, K. Kiguchi and K. Izumi, "A Handstand from Backward Giant Circle by a Three-Link Rings Gymnastic Robot," *Proc. of the 20th Annual Conf. of the Robotics Society of Japan*, 2H19, 2002. (in Japanese)

[3] E. J. Sprigings, J. L. Lanovaz, L. G. Watson, and K. W. Russell, "Removing swing from a handstand on rings using a properly timed backward giant circle: a simulation solution," *J. of Biomechanics*, Vol. 31, No. 1, pp. 27–35, 1998.

[4] T. Yamada, K. Watanabe, K. Kiguchi, and K. Izumi, "Acquisition of Exercise Skill Represented by Fuzzy Control Rules for a Rings Gymnastic Robot," *Proc. of IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, pp. 368–373, 2001.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Dynamic Potential Field Method for Local Obstacle Avoidance of Mobile Robot

Xin Yang*, Keigo Watanabe**, Kiyotaka Izumi** and Kazuo Kiguchi**
*Department of Production Control Technology,
Division of Engineering Systems and Technology,,
**Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University, 1-Honjomachi, Saga 840-8502, Japan
†E-mail: xinyang@yahoo.co.jp, {watanabe, izumi, kiguchi}@me.saga-u.ac.jp

Abstract

In this paper, a dynamic potential field method (DPFM) is proposed for the local obstacle avoidance of mobile robot. The magnitude of the DPFM is defined by the kinetic energy of a robot relative to an obstacle/goal. It is assumed that a desired trajectory or a sub-goal of a mobile robot has defined previously by a high-level path-planning algorithm. The DPFM is used to accelerate or decelerate the mobile robot to avoid the unexpected or moving obstacle when the robot tries to achieve its target. Combining with a high-level path planer, this method makes the mobile robot be able to move in a dynamic environment and need not adjust any parameter of the artificial potential force in a new environment. Simulation results show the effectiveness of the proposed method.

1 Introduction

Autonomous navigation of mobile robots is widely recognized as a fundamental research issue. Many mobile robot systems combine a global path planning model (high-level planning control) with a local obstacle avoidance model (reactive control) to perform navigation. With planning, before moving, the robot determines an optimal trajectory or a series of sub-goals using the model of environment to achieve a specified goal. With local obstacle avoidance, the robot determines the suitable motion just before executes it based on recent sensor data while the robot tries to track the desired trajectory or to achieve the next sub-goal.

The potential field method (PFM) suggested by Khatib [1] is one of the most popular approaches for implementing real-time obstacle avoidance. One of the reasons for the popularity of this method is its simplicity and elegance. In this approach obstacles exert repulsive force onto the robot, while the goal applies an attractive force to the robot. The sum of all forces determines the subsequent motion of the robot. One commonly used expression of PFM is:

$$\boldsymbol{F}^*_{art} = \boldsymbol{F}^*_{xd} + \boldsymbol{F}^*_o \quad (1)$$

$$\boldsymbol{F}^*_{xd} = -k_p(\boldsymbol{x} - \boldsymbol{x}_d) \quad (2)$$

$$\boldsymbol{F}^*_o = \eta\left(\frac{1}{\rho} - \frac{1}{\rho_o}\right)\frac{1}{\rho^2}\frac{\partial \rho}{\partial \boldsymbol{x}} \quad (3)$$

where $\boldsymbol{F}^*_{art}$, $\boldsymbol{F}^*_{xd}$ and $\boldsymbol{F}^*_o$ represent the artificial force that the robot suffered, the attractive force that draws the robot to reach the goal position and the repulsion force that pushes the robot away from the obstacle, respectively. $\boldsymbol{x}$ is the position of the robot, $\boldsymbol{x}_d$ denotes the goal position of it, and k_p is the position gain. η is a constant gain, ρ_o represents the limit distance of the potential filed influence and ρ denotes the shortest distance to the obstacle.

Most of the existing works focused on the inherent limitations of PFM [2][3], such as, the local minimum problem. In fact, how to scale the potential field is an actual problem of PFM in implementation. In details, on the one hand how to define the magnitude of gain k_p and η in equations (2) and (3). On the other hand the force generated by equations (2) and (3) is only related to the distance between the robot and obstacle/goal, but it is clear that these forces should also be related to the access velocity. For example, in a repulsive potential field the obstacle should exert a bigger force on a robot with bigger access velocity than that on a robot with smaller access velocity even if the two robots have the same distance with the obstacle.

In this paper, we propose a dynamic potential field method (DPFM). Different to other velocity-dependent PFM [4], in DPFM the magnitude of potential force is decided by the kinetic energy of the robot which is relative to an obstacle/goal; in this way the magnitude of potential is related not only to the distance between the robot and obstacle/goal, but also to the access velocity between them. Furthermore, different to other local obstacle avoidance methods, in this research the DPFM is just used as an alternative with a trajectory tracking method rather than a local path planner or a local path explorer.

2 Kinetic Energy Based Potential Field Method

Essentially, the DPFM is a kinetic energy based potential field method and it also includes two kinds of force field: a repulsion potential field and an attractive potential field.

In a repulsion potential field, the kinetic energy of the robot relative to the object should be the same

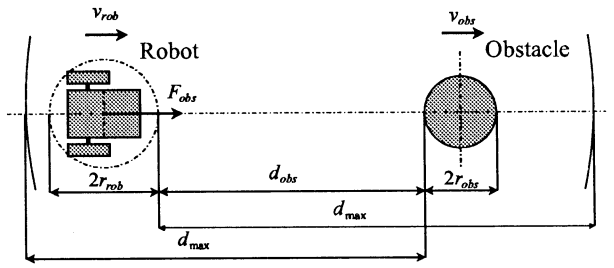


Figure 1: The repulsion potential field

with or less than the work that the repulsion force acts on it. This means the access velocity between the robot and the obstacle will be reduced to zero before the cleaning boundary of the robot reaches the obstacle.

In an attractive potential field, the kinetic energy of the robot relative to the goal should be the same with the work that the attractive force acts on it. The attractive potential force not only draws the robot to the goal but also reduces the access velocity between the robot and goal to zero when the robot reaches the goal.

As shown in **Fig. 1** and **Fig. 2**, two one-dimensional scenarios illustrate the basic idea of DPFM. In **Fig. 1**, the robot with the absolute velocity v_{rob} enters the repulsion potential field of an obstacle with velocity v_{obs}. $d_{\max}$ denotes the maximum range of the repulsion potential field or to say the effective range of the sensor fixed on the robot. The radii of cleaning boundary of the robot and the obstacle are r_{rob} and r_{obs}, and d_{obs} represents the cleaning distance between the robot and the obstacle.

According to the definition of DPFM, the work that the repulsive force acts on the robot:

$$U_{rep} = \int_0^{d_{obs}} F_{obs}(x)dx \tag{4}$$

should be the same with or bigger than the kinetic energy of the robot relative to the obstacle:

$$E_{rel} = \frac{1}{2} m_{rob} \dot{d}_{obs}^2 \tag{5}$$

where $\dot{d}_{obs}(= v_{rob} - v_{obs})$ is the access velocity between the robot and the obstacle, m_{rob} is the mass of the robot. The virtual repulsive force F_{obs} is a function of the relative distance d_{obs}, so that using equation (3) the DPFM can be expressed by

$$\frac{1}{2} m_{rob} \dot{d}_{obs}^2 \leq \int_0^{d_{obs}} \eta \left(\frac{1}{d_{obs}} - \frac{1}{d_{max}} \right) \frac{1}{d_{obs}^2} dd_{obs}. \tag{6}$$

In this way, the gain η can be calculated rather than be defined. In this paper, we simply assume that F_{obs} is a constant value, and therefore we have

$$\begin{aligned} \frac{1}{2} m_{rob} \dot{d}_{obs}^2 &\leq |F_{obs} d_{obs}| \\ &\leq |m_{rob} a_{obs} d_{obs}| \end{aligned} \tag{7}$$

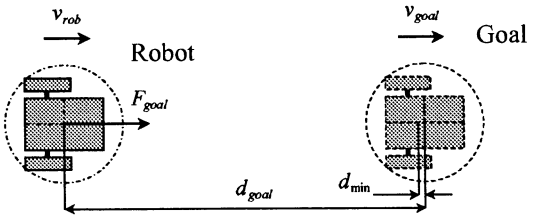


Figure 2: The attractive potential field

where a_{obs} is the acceleration of the robot generated by the virtual repulsion force. Rearranging equation (7), it is easy to obtain that

$$a_{obs} = \begin{cases} -\frac{1}{2} \frac{\dot{d}_{obs}^2}{d_{obs}} & \dot{d}_{obs} > 0 \text{ and } d_{obs} < d_{\max} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In the same way, for an attractive potential field (**Fig. 2**), the acceleration of the robot generated by the virtual attractive force, a_{goal}, is expressed by

$$a_{goal} = \begin{cases} 0 & d_{goal} \leq d_{\min} \\ \frac{1}{2} \frac{\dot{d}_{goal}^2}{d_{goal}} - g_{att} & d_{goal} > d_{\min} \text{ and } \dot{d}_{goal} \geq 0 \\ -\frac{1}{2} \frac{\dot{d}_{goal}^2}{d_{goal}} - g_{att} & d_{goal} > d_{\min} \text{ and } \dot{d}_{goal} < 0 \end{cases} \tag{9}$$

where $\dot{d}_{goal} = v_{rob} - v_{goal}$ and g_{att} is a virtual gravity, which should be defined bigger than an acceleration limit of the robot, $a_{\max}$. It should be noticed that any accelerations of the robot such as a_{obs} and a_{goal} are all restricted by $a_{\max}$. Usually, the magnitude of $a_{\max}$ is related to the structure of the robot or to the friction coefficient between the wheel and the ground. Moreover, $d_{\min}$ is a threshold value. We here assume that the robot is controlled by DPFM when $d_{goal} > d_{\min}$, whereas it is controlled by a conventional controller like PI controller when $d_{goal} \leq d_{\min}$. In this way, after avoiding obstacles, the robot can converge to its desired trajectory or achieve its goal without overshoot finally.

In conventional PFM, the artificial force that acts on a robot is the summation of the repulsive forces generated by all the obstacles around and the attractive force generated by the goal. But in the proposed method, a robot just tries to avoid the most dangerous obstacle.

3 Using DPFM in Local Obstacle Avoidance of Mobile Robot

3.1 Kinematic model of the mobile robot

Fig. 2 shows the kinematic model of the mobile robot. Since the reference point o of this robot offsets from the central point of two driving wheels with a distance s along the central line of the robot, we can control the position of point o rather than control the

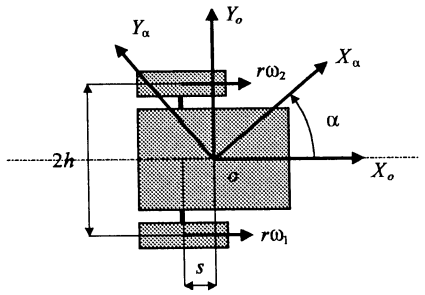


Figure 3: Kinematic model of the mobile robot

translational and rotational speed of the robot. The distance between two driving wheels is $2h$, the radius of each wheel is r and $\boldsymbol{\omega}(= [\omega_1\ \omega_2])$ denotes the angular velocity vetor of them. The local coordinate system $\sum_o$ is set at point o so that its X_o axis is along the central line of the robot. After defining the velocity of the robot ${}^o\dot{\boldsymbol{x}}_o(= [{}^o\dot{x}_o\ {}^o\dot{y}_o]^T)$ in $\sum_o$, the kinematic model of the mobile robot is expressed by

$$\boldsymbol{\omega} = A_o^{-1}\,{}^o\dot{\boldsymbol{x}}_o \quad (10)$$

$${}^o\dot{\boldsymbol{x}}_o = A_o\boldsymbol{\omega} \quad (11)$$

where

$$A_o^{-1} = \frac{1}{r}\begin{bmatrix} 1 & \frac{h}{s} \\ 1 & -\frac{h}{s} \end{bmatrix}$$

and

$$A_o = \frac{r}{2}\begin{bmatrix} 1 & 1 \\ \frac{s}{h} & -\frac{s}{h} \end{bmatrix}.$$

If another local coordinate system $\sum_\alpha$ fixed on the robot is defined in such a way that the origin of $\sum_\alpha$ is located on point o so that the angle between X_o and X_α axes is α, then the motion of the robot in $\sum_\alpha$ can be mapped into space $\sum_o$ by

$${}^\alpha\dot{\boldsymbol{x}}_o = {}^o_\alpha\boldsymbol{R}\ {}^o\dot{\boldsymbol{x}}_o \quad (12)$$

where

$${}^o_\alpha\boldsymbol{R} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}.$$

3.2 Implementation of DPFM in local obstacle avoidance

When the robot enters the potential fields of some obstacles, the motion of the robot is decided by the effect of both repulsive and attractive potential forces. As shown in **Fig. 4**, the local coordinate system $\sum_\alpha$ is defined in such a way that the X_α axis always points to the most dangerous obstacle, which means only the biggest repulsive force generated by obstacles around the robot effects on the robot.

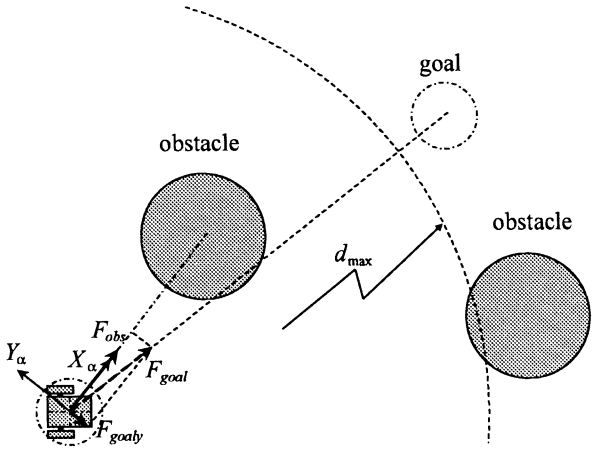


Figure 4: The robot staying in repulsive and attractive potential fields

The motion of the robot along X_α is controlled by the repulsive force F_{obs} generated by the obstacle and the motion along Y_α is controlled by the Y_α directional component F_{goaly} of attractive force F_{goal} generated by the goal. Then we have

$${}^\alpha\ddot{x}_o := \begin{cases} a_{obs} & |a_{obs}| < a_{\max} \\ \frac{a_{obs}}{|a_{obs}|}a_{\max} & |a_{obs}| \geq a_{\max} \end{cases} \quad (13)$$

$${}^\alpha\ddot{y}_o := \begin{cases} a_{goaly} & |a_{obs}| < a_{\max} \\ \frac{a_{goaly}}{|a_{goaly}|}a_{\max} & |a_{obs}| \geq a_{\max} \end{cases} \quad (14)$$

and the robot will accelerate or decelerate based on its current velocity, which can be calculated by equation (11). A special situation for a case of F_{obs} and F_{goal} lying on the same line means that the robot falls into a local minimum. In this case, DPFM can not help the robot to jump out of the local minimum; the only thing that the robot can do is to implement a wall-following motion or to ask help from a high-level path planner.

If there is no any obstacle that appears in the range of sensors equipped on the robot, the robot is only attracted by the goal. In this case, the X_α axis of local coordinate system $\sum_\alpha$ points to the goal as shown in **Fig. 5**. The velocity along Y_α will be reduced to zero as fast as possible, such as

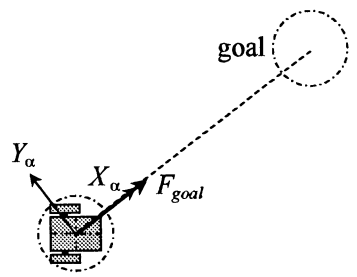


Figure 5: The robot staying in attractive potential field

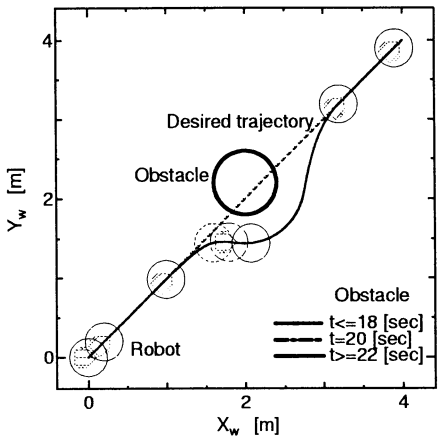


Figure 6: The tobot avoiding a static obstacle (Simulation I)

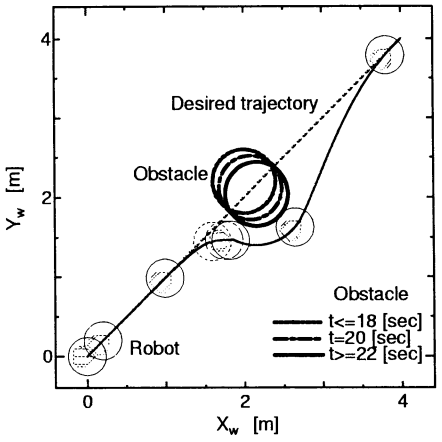


Figure 7: The robot avoiding a moving obstacle with low speed (Simulation II)

$$ {}^{\alpha}\ddot{x}_o := \begin{cases} a_{goal} & |a_{obs}| < a_{\max} \\ \frac{a_{goal}}{|a_{goal}|} a_{\max} & |a_{obs}| \geq a_{\max} \end{cases} \quad (15) $$

and

$$ {}^{\alpha}\ddot{y}_o := \begin{cases} {}^{\alpha}\ddot{y}_o - \frac{{}^{\alpha}\dot{y}_o}{|{}^{\alpha}\dot{y}_o|} a_{\max} dt & |{}^{\alpha}\dot{y}_o| > a_{\max} dt \\ 0 & \text{otherwise.} \end{cases} \quad (16) $$

4 Simulations

In simulations, one robot is set to follow a desired trajectory. In fact, the situation that the robot runs toward to a static goal is just a special case of running toward to a moving goal or following a trajectory. In simulation I, a static obstacle was set near the desired trajectory of the robot. The mobile robot avoided the obstacle successfully and finally converged to the desired trajectory. In simulation II, the obstacle began to move when the robot closed to it ($t = 18$[sec]), and consequently the robot could still move in front of the obstacle and avoid a collision. The speed of obstacle in simulation III was higher than that in simulation II. For this case, the robot turned left and avoided the obstacle from its back.

5 Conclusion

The classical potential field has been extended with relative kinetic energy. In this way, the parameters of potential field are calculated automatically in a new or a dynamic environment. In this paper, the DPFM was just used as a reactive control method for local obstacle avoidance. Different to other modified PFM, the DPFM has to work together with a high-level path planning method and it is assumed that the local minimum problem, which is the inherent limitations of PFM, can be solved by the high-level path planning method. Simulation results showed that a robot was able to avoid an obstacle dynamically when it tried to track a desired trajectory by using DPFM.

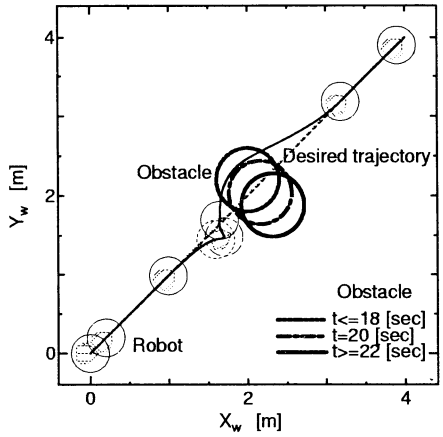


Figure 8: The robot avoiding a moving obstacle with high speed (Simulation III)

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 500–505, 1985.

[2] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Con. on Robotic & Automation*, pp. 1398–1404, 1991.

[3] C. Q. Liu, M. Ang Jr, et. al., "Virtual obstacle concept for local-minimum-recovery in potential-field based navigation," in *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 983–988, 2000.

[4] B. H. Krogh, "A generalized potential field approach to obstacle avoidance control," in *Proc. of the Int. Robotics Research Conference*, pp. 1150–1156, 1984.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Neural Network Based Expectation Learning in Perception Control: Learning and Control with Unreliable Sensory System

Sherwin A. Guirnaldo*,[1], Keigo Watanabe**, Kiyotaka Izumi** and Kazuo Kiguchi**
*Department of Production Control Technology,
Division of Engineering Systems and Technology,,
**Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University, 1-Honjomachi, Saga 840-8502, Japan
†E-mail: xinyang@yahoo.co.jp, {watanabe, izumi, kiguchi}@me.saga-u.ac.jp

Abstract

In this paper, we investigate the viability of our proposed neural network(NN)-based extension of the "perception" control concept introduced by Randløv and Alstrøm. In their work, each of the expectation elements is linearly acquired such that the expectation tells only the dominant information in the recent past. This handicap could become a serious problem when the "perception" process is applied to real physical systems. Their approach has no capability to sense the trend and the dynamics in the information. Here, we introduce an extension of the "perception" control process by using a radial basis function feedforward NN to learn the trend and the dynamics in the information. Through our simulations, we will show that our NN based method is far better.

1 Introduction

The basics of intelligent agents design tell us that intelligent machines should be capable of knowing how their world changes with time and for each action taken [1]. Such a capability is very crucial specially when the sensory system is incapable of giving full access to the complete state of the world (e.g., perceptual aliasing [2,3], limited field of view and occlusions). In addition to the above, sensor failure can make the situation even worst. In practice, it is possible that sensors may fail during critical operation or there will be cut in the communication from sensory system such that the information received by the controller is in incomplete form. It is often not desirable to disrupt the process until the critical period is over or to maintain control, despite of the absence of some vital information.

In connection to sensor failure, Randløv and Alstrøm [4] introduced a motivation-based control system coupled with a process in which they called "perception" (shown in Fig. 1). The "perception" process was used to select and to complete the information received from the system or environment to be controlled. They were able to show that their "perception" paradigm is valuable when parts of the information from the system being controlled are sometimes missing. The "perception" process completes the information, s^{rcc}, from the system to be controlled before passing it (as s^{com}) to the controller (or action system) so that control is not lost. At every time steps the process produces an expectation (not shown in Fig. 1) of the next (i.e., time step) set of information to be received from the system. These expectations are not driven by any predefined success criteria, but are formed in an ongoing irradiation process [5] of one activity pattern initiating another. But their "perception" process approach has a major drawback, each of the expectation elements are linearly acquired such that the expectation tells only the most dominant information in the recent past. This handicap could become a serious problem when the "perception" process is applied to real physical systems. Sensing the trend and the dynamics of the information will be a problem.

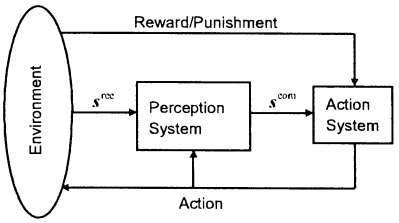


Figure 1: Perception control system.

In this work, we introduce an extension of [4] by using a radial basis function (RBF) feedforward neural network (NN) as the irradiation process. The NN will learn the trend of the information as a consequence for every action taken to produce an expectation of the next observation that will be used to complete the information.

2 Randløv and Alstrøm Perception Control

Given the perception or short term memory matrix $\Psi(t) \in \Re^{m\times\tau}$, where m and τ represent the number of discrete sensory elements and the number of action choices respectively, and the set of weights $U_i(t) \in \Re^{m\times\tau}$, $i = 1, \cdots, m$. Each element $\Psi_{jk}(t)$ in the matrix $\Psi(t)$ is computed according to:

$$\Psi_{jk}(t) = \psi_j^*(t-1)a_k(t-1) \qquad (1)$$

[1]Permanent address: Dept. of Mechanical Engineering, Mindanao State University, Marawi, Philippines

where $\boldsymbol{\psi}^*(t-1) \triangleq [\psi_1^*(t-1), \cdots, \psi_m^*(t-1)]^T$ is the new percept vector from the previous time step and $\boldsymbol{a}(t-1) \triangleq [a_1(t-1), \cdots, a_\tau(t-1)]^T$ is the previous control action vector. Here,

$$a_k(t) = \begin{cases} 1 & \text{if the agent chose action } k \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

1. At every time step t, compute the expectation vector $\boldsymbol{\psi}(t) \in \Re^m$ according to:

$$\psi_i(t) = \sum_{j=1}^{m} \sum_{k=1}^{\tau} U_{ijk}(t) \Psi_{jk}(t). \tag{3}$$

   Apply a winners-take-all (WTA) function, f_{WTA}, such that

$$\boldsymbol{\psi}(t) \quad := \quad f_{WTA}[\boldsymbol{\psi}(t)]. \tag{4}$$

   Winners are set to 1 and losers will take 0. A winner is defined as the element in a component with the highest numerical value.

2. Obtain the new sensor readings $\boldsymbol{s}^{\text{rcc}}(t) \in \Re^m$.

3. Determine the new percept vector $\boldsymbol{\psi}^*(t) \in \Re^m$:

$$\psi_i^*(t) = \begin{cases} \psi_i(t)\kappa & \text{if the component that} \\ & i \text{ belongs to is missing,} \\ s_i^{\text{rcc}}(t) & \text{otherwise.} \end{cases} \tag{5}$$

   Note that since an expectation element is used here as a substitute when a sensor fails to function as intended, the current expectation will influence the next expectation (i.e., the expectation is recurrent by nature). The constant κ here is the confidence for the expectation and the range of its value is between $[0, 1]$.

4. Complete the observation by replacing missing parts in the observation with the corresponding expectation elements, i.e.,

$$s_i^{\text{com}}(t) = \begin{cases} \psi_i(t) & \text{if the component that} \\ & i \text{ belongs to is missing,} \\ s_i^{\text{rcc}}(t) & \text{otherwise.} \end{cases} \tag{6}$$

5. Update each of the weights U_{ijk}. This step is known as the learning part of the expectation dynamics. When the expectation wrongly predicts the next sensor readings then the learning process should take place. To this end, an error vector $\delta_i(t) \in \Re$ is calculated:

$$\delta_i(t) = s_i^{\text{rcc}}(t) - [\boldsymbol{\psi}(t)]_i \tag{7}$$

   where $[\cdot]_i$ denotes the ith element of the expectation vector. If a particular component in the sensor vector is missing, then all the corresponding elements for that component in the error vector is set to zero. The weight update rule is:

$$\Delta U_{ijk}(t) = \alpha_u \delta_i(t) a_k \begin{cases} \alpha_u \psi_j(t-1) & \text{if the component that } j \text{ belongs to is missing,} \\ s_j^{\text{rcc}}(t-1) & \text{otherwise} \end{cases} \tag{8}$$

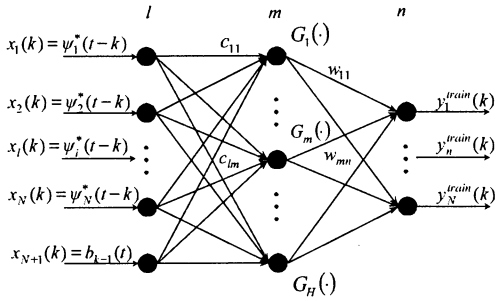


Figure 2: Structure of RBFNN based expectation dynamics.

$$U_{ijk}(t+1) = U_{ijk}(t) + \Delta U_{ijk}(t). \tag{9}$$

where α_u is the adaptation rate.

3 RBFNN Based Perception Control

Given the perception or short term memory matrix $\Psi(t) \in \Re^{N \times \tau}$, where N and τ represent the number of continuous sensory elements (i.e., the number of sensors) and the length of the short term memory respectively. Note here that τ is no longer necessarily equal to the number of available action choices; it can be set to higher values if longer trail of experiences is desired. In addition to that, an action memory vector $\boldsymbol{b}(t) \triangleq [b_1(t), \cdots, b_{\tau-1}(t)]^T \triangleq [a(t-1), \cdots, a(t-\tau+1)]^T$ will be introduced to store a finite amount of previous actions taken by the agent, in which $a(t)$ refers to the control action taken at time t. Figure 2 shows the structure of the RBFNN used to produce the expectation by learning how the actions in the past changed the information that the agent received. During the expectation learning, the inputs of the network, $\boldsymbol{x}(k) \triangleq [x_1(k), \cdots, x_{N+1}(k)]^T$, are the elements of the new percept vector $\boldsymbol{\psi}^*(t-k) \triangleq [\psi_1^*(t-k), \cdots, \psi_N^*(t-k)]^T$ stored in the kth row of $\Psi(t)$ and the control action stored in $b_{k-1}(t)$. These inputs are then used to produce the training expectation vector $\boldsymbol{y}^{train}(k) \triangleq [y_1^{train}(k), \cdots, y_N^{train}(k)]^T$. See Fig. 3 for a clear illustration on how the NN function will access the vectors and scalars in $\Psi(t)$ and $\boldsymbol{b}(t)$ respectively, through the use of k to produce the training expectation vector $\boldsymbol{y}^{train}$. In this case, $\boldsymbol{\psi}^*$, $\boldsymbol{\psi}$, $\boldsymbol{y}^{train}$ and Ψ are no longer discrete; they contain actual sensor readings. H is the number of RBF hidden nodes, w's, c's, and σ's are the parameters of the network. In the learning process, we use the basic gradient descent approach as depicted in the equations of step 6.

1. At every time step t, compute the expectation vector $\boldsymbol{\psi}(t) \in \Re^N$ according to:

$$\boldsymbol{\psi}(t) = f_{RBFNN}(\boldsymbol{\psi}^*(t-1), a(t)) \tag{10}$$

   where $f_{RBFNN}(\cdot)$ is the RBFNN function.

2. Obtain the new sensor readings (non-discrete) $\boldsymbol{s}^{\text{rcc}}(t) \in \Re^N$.

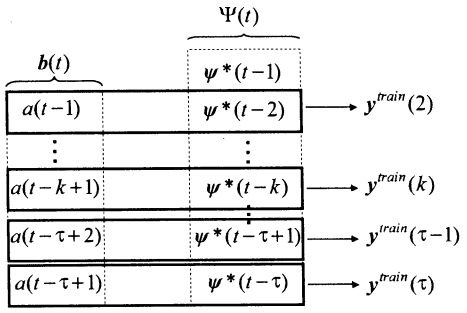


Figure 3: Training expectations.

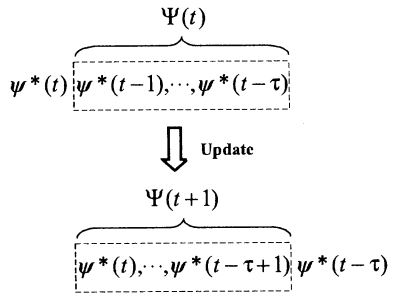


Figure 4: Update of the short term memory.

3. Determine the new percept vector $\boldsymbol{\psi}^*(t) \in \Re^N$:

$$\psi_i^*(t) = \begin{cases} \psi_i(t)\kappa & \text{if sensor } i \text{ reading} \\ & \text{is missing,} \\ s_i^{\text{rcc}}(t) & \text{otherwise.} \end{cases} \quad (11)$$

4. Complete the observation by replacing missing parts in the observation, i.e.,

$$s_i^{\text{com}}(t) = \begin{cases} \psi_i(t) & \text{if the sensor information that } i \\ & \text{represents is missing,} \\ s_i^{\text{rcc}}(t) & \text{otherwise.} \end{cases} \quad (12)$$

5. Update the short-term memory matrix Ψ by pushing the new percept vector $\boldsymbol{\psi}^*(t)$ into it (see Fig. 4). The detailed update is that the oldest perception vector will be thrown out from the short-term memory matrix, the rest of the remaining perception vector will be pushed back one step in the memory. The update procedure for vector $\boldsymbol{b}$ uses the same rule with that of Ψ.

6. Update the neural network parameters. This is the expectation learning part of this "perception" control dynamics. The radial basis function or activation function is given by

$$G_m(\boldsymbol{x}(k)) = \exp\left(\frac{\|\boldsymbol{x}(k) - \boldsymbol{c}_m\|^2}{-2\sigma_m^2}\right) \quad (13)$$

where $\boldsymbol{c}_m \triangleq [c_{m1}, \cdots, c_{m(N+1)}]^T$. An output $y_n^{train}(k)$ is computed according to

$$y_n^{train}(k) = \sum_{m=1}^{H} w_{mn} G_m(\boldsymbol{x}(k)) \quad (14)$$

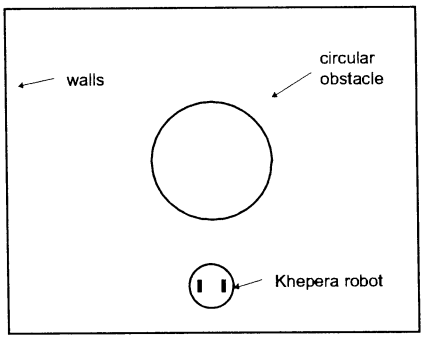


Figure 5: The Khepera robot world.

or equivalently

$$y_n^{train}(k) = \sum_{m=1}^{H} w_{mn} \exp\left(\frac{\sum_{l=1}^{N+1} (x_l(k) - c_{ml})^2}{-2\sigma_m^2}\right). \quad (15)$$

An error vector $\boldsymbol{e}(k) \triangleq [e_1(k), \cdots, e_N(k)]^T$ is defined for each training expectation vector $\boldsymbol{y}^{train}(k)$ by

$$e_n(k) = \Psi_{n(k-1)} - y_n^{train}(k) \quad (16)$$

where the training expectation vector $\boldsymbol{y}^{train}(k)$ is generated using the RBFNN function,

$$\boldsymbol{y}^{train}(k) = f_{RBFNN}(\boldsymbol{\psi}^*(t-k), b_{k-1}(t)) \quad (17)$$

and $\boldsymbol{\psi}^*(t-k)$ is a vector stored in the kth row of $\Psi(t)$. The squared error for all the elements of a training expectation vector is given by

$$E(k) = \frac{1}{2}\sum_{n=1}^{N} e_n^2(k). \quad (18)$$

Errors of the expectation are computed by subtracting the elements of $\boldsymbol{y}^{train}(k)$ from the corresponding elements in the $(k-1)$th row of Ψ. This is performed recursively for all the rows (or k) of $\Psi(t)$ except for the first row (i.e., $k = 1$).

4 Obstacle Avoidance with Khepera Mobile Robot

Figure 5 shows the mobile robot Khepera and the world where it operates. The Khepera robot is equipped with eight infrared sensors that can be used to detect the proximity of objects in front of it, behind it and to the right and left side of it. Each sensor returns a value ranging between 0 and 1023, in which 0 means that no object is perceived while 1023 means that an object is very close to the sensor (almost touching the sensor). To simulate the robot, we used the freely available Khepera Simulator version 2.0 software written by Michele [6].

In the simulation study we conducted two tests. The goal of the first test was to learn a collision avoidance

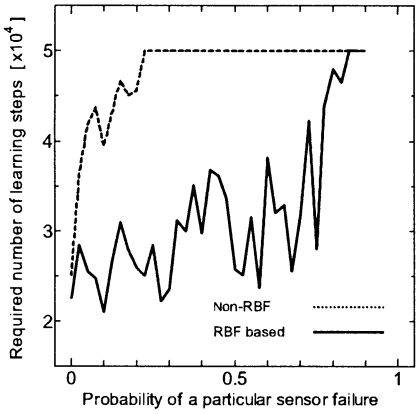


Figure 6: The Khepera robot learning curve (Test 1).

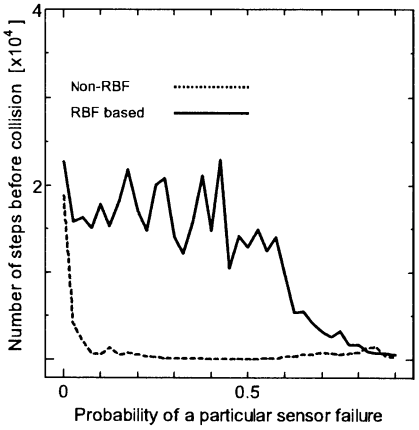


Figure 7: Average number of steps before any collision (Test 2).

behavior through Q–learning while some vital information are randomly missing. If the robot successfully avoids collision for 10,000 time steps, then the objective behavior is considered to be learned already; while if the robot is unable to do that within 50,000 time steps, then the condition is considered to be unlearnable. The second test's goal was to determine how long the robot can avoid obstacles after learning the desired avoidance behavior while some of the information are missing at random. The robot is given 12,500 steps to learn the behavior and after that we allow the robot to roam around its world for additional 37,500 steps offline (i.e., no Q-learning). During the last 37,500 steps, the information received by the robot is sometimes incomplete (at random).

The results of the simulations are shown in Fig. 6 for the first test and Fig. 7 for the second test. The data plotted are the average of 20 runs. The results clearly showed that the RBF–based "perception" control is far better than the other method. With 20 percent chance that a particular sensor fails (Fig. 6), the other method starts to fail to learn the desired behavior. On the other hand, the RBF-based "perception" control can still bare the condition up to 85 percent probability of failure. On the second test, the RBF-based "perception" control was able to avoid collision for a longer time and maintain it for a wider spectrum of sensor failure probabilities.

In here, the RBFNN based "perception" control is to learn how the information from the proximity sensors changes for each of the action taken. For a large unstructured environment and with limited size of the NN, global convergence will be unlikely. In cases like this the "perception" control should produce the expectation greatly based on the most recent experiences and less on non-recent one. Consequently, this can suggest the use of bigger adaptation rates.

5 Conclusion

We have introduced and examined the advantages of using RBFNN as "perception" irradiation process that completes the information from the environment to be controlled. We presented the "perception" control of [4] and showed how the concept can be extended using RBFNN. We implemented the two approaches with Q–learning and applied them to a learning and obstacle avoidance problems where environmental information is missing (at random). The probability of missing information may be substantial, but still it must be available for some time, at least, to allow the "perception" control to build up an expectation.

Our simulations showed that the RBFNN based "perception" control works better than the one presented in [4]. The improved "perception" control dynamics made it possible for the Q–learning to learn an appropriate control policies or maintain control even when substantial amounts of environmental information are missing most of the time.

References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[2] R. McCallum, "Instance-based utile distinctions for reinforcement learning with hidden states," in *Proc. of the 12th Int. Conf. on Machine Learning*, vol. A. no. 290, 2001, pp. 251–267.

[3] L. Chrisman, "Reinforcement learning with perceptual aliasing: the perceptual distinctions approach," in *Proc. of the 10th National Conf. on Artificial Intelligence*, San Jose, CA, 1992, pp. 183–188.

[4] J. Randløv and P. Alstrøm, "Perception control" *Physica A*, vol. 289, pp. 561–573, 2001.

[5] I. Pavlov, *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*, London, UK: Oxford University Press, 1927.

[6] O. Michele, "Khepera Simulator version 2.0," http://wwwi3s.unice.fr/ om/, 1996.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Control of 3-DOF Underactuated Manipulator Using Fuzzy Based Switching

Lanka Udawatta, Keigo Watanabe, Kiyotaka Izumi, and Kazuo Kiguchi
Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University, 1-Honjomachi, Saga 840-8502, Japan
†E-mail : lanka@ieee.org

Abstract

A novel concept for designing a fuzzy logic based switching controller in order to control underactuated manipulators is presented. The proposed controller employs elemental controllers, which are designed in advance. Parameters of both antecedent and consequent parts of a fuzzy indexer are optimized by using evolutionary computation. Design parameters of the fuzzy indexer are encoded into chromosomes, i.e., the shapes of the Gaussian membership functions and corresponding switching laws of the consequent part are evolved to minimize the angular position errors. Then, these trained fuzzy rules can be brought into the online operation of underactuated manipulators. Simulation results show that the new methodology is effective in designing controllers for underactuated robot manipulators.

Keywords: Underactuated robots, Fuzzy reasoning, Switching control, Evolutionary learning.

1 Introduction

Design of control algorithms for underactuated manipulators is more complex than that for regular robot manipulators. Such manipulators consist of both active and passive joints and the passive joints result in a lack of controllability of the total system. Moreover, there is an inertial coupling between the motions of active and passive joints, so that mapping such as Jacobian matrix, depends not only on the kinematic properties, but also on the inertia properties of the links. Examples of nonholonomic control systems have been studied in the context of robot manipulators, mobile robots, wheeled vehicles, and space robotics [1]. On the other hand, humans commonly utilize underactuation to perform tasks more precisely or more easily (see **Fig. 1**) [2]. Thus, controlling of this class of robots is a challenging task and still remains as an open problem; therefore it still needs exploring this interesting topic [2, 3].

When the system has complex nonlinear dynamics and nonholonomic properties it makes difficult to control the nonlinear system using conventional methods, therefore fuzzy-learning controllers are an attractive alternative. Blending of GA with fuzzy rules, in order to capture the hidden nonlinearities of the system, will be useful in developing learning controllers. In this study, a robust fuzzy rule extraction method via GA is considered for controlling three-degrees-of freedom (DOF) planar manipulators using switching computed torque approach. The rest of the paper is organized as follows: In Section 2, the concept of fuzzy logic based switching controller is presented. Design of elemental controllers for a three-DOF underactuated manipulator and evolutionary learning for obtaining fuzzy rules are discussed in Section 3 and Section 4 respectively. Finally, control results and conclusions are given in Section 5 and Section 6 respectively.

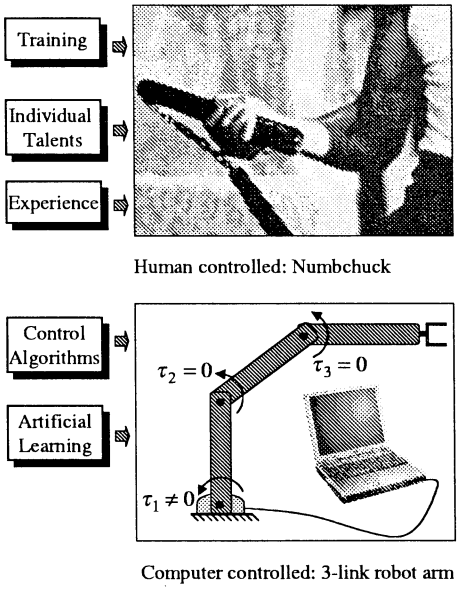


Figure 1: From human controlled underactuated systems to computer controlled underactuated systems

2 Concept of Fuzzy Logic Based Switching Controller

In this section, we present the design procedure of the proposed fuzzy logic based switching controller. The controller design concept explained in [4] uses partly stable controllers (PSC), which can be applied for controlling underactuated robot systems with appropriate switching of the available set of elemental controllers, i.e. PSC. These PSC can be synthesized by analyzing the dynamic equation of an underactuated manipulator (1) given below:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = F \qquad (1)$$

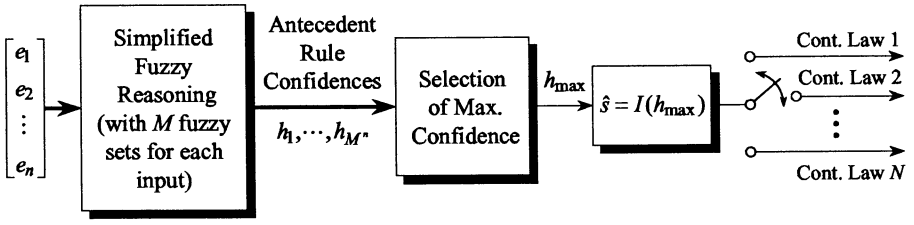


Figure 2: Fuzzy indexer for switching elemental controllers

Here, the generalized coordinate vector $\boldsymbol{q}$ and input force/torque vector F are given by $\boldsymbol{q} \in \Re^n$ and $F \in \Re^n$ respectively. $M(\boldsymbol{q})$ is the $n \times n$ inertia matrix and $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represents Coriolis, centrifugal, frictional and gravitational components. Suppose that an underactuated robot system has m_u number of actuators to control and n-DOF ($1 \leq m_u < n$) links. Then, we have ${}_nC_{m_u}$ number of combinations of m_u-dimensional controllers. To actuate the robot system, one of the above combinations of available controllers can be selected via fuzzy logic based switching.

Let us consider a simplified fuzzy reasoning that generates an index number for the selection of a suitable control law. That is, the proposed fuzzy indexer has a set of input-output data from an underactuated system with n position error variables, $\{e_i \mid i = 1, 2, \cdots, n\}$, and a scalar index $s \in \mathcal{S} = \{1, 2, \cdots, N\}$, where $N = {}_nC_{m_u}$ denotes the number of control laws. The input space is created with a fuzzy rule base such that the position error is taken and converted into a grade of membership $\{\mu_{ij}(e_i) \triangleq \mu_{A_{ij}}(e_i) \mid i = 1, 2, \ldots, n\}$ employing a fuzzy set A_{ij} at the jth rule. Here, antecedent linguistic values, i.e., labels for fuzzy sets in each position error are assigned such as P: positive, Z: zero, N: negative, etc., totally $l = 1, 2, \ldots, M$. Then, one of the elemental controllers (i.e., PSCs) will be assigned in the consequent part as below:

$$\begin{aligned} \mathrm{R}_j: \quad & \text{IF } e_1 \text{ is } A_{1j} \text{ AND } e_2 \text{ is } A_{2j} \text{ AND } \ldots e_n \text{ is } A_{nj} \\ & \text{THEN} \quad \text{Control law is } s_j, \quad j = 1, 2, \cdots, M^n \end{aligned} \tag{2}$$

where $s_j \in \mathcal{S}$.

Following the conventional weighted average method, we can readily obtain the reasoning result of s, which is denoted by $\hat{s}$ such as

$$\hat{s} = \sum_{j=1}^{M^n} w_j s_j \tag{3}$$

$$w_j = \frac{h_j}{\sum_{k=1}^{M^n} h_k}, \quad h_j \triangleq \prod_{i=1}^{n} \mu_{ij}(e_i). \tag{4}$$

Note however that in this case the reasoning result generates a real number that is inconvenient for selecting an integer number in $\mathcal{S}$. From this fact, we here use the following reasoning result:

$$\hat{s} = I(h_{\max}), \quad h_{\max} \triangleq \max\{h_1, \cdots, h_{M^n}\} \tag{5}$$

where $I(h_{\max})$ means that the reasoning result is taken as the direct consequent index value which is assigned to the rule number, whose antecedent confidence has a maximum among them. **Figure 2** shows the block diagram of the proposed fuzzy indexer for switching elemental controllers.

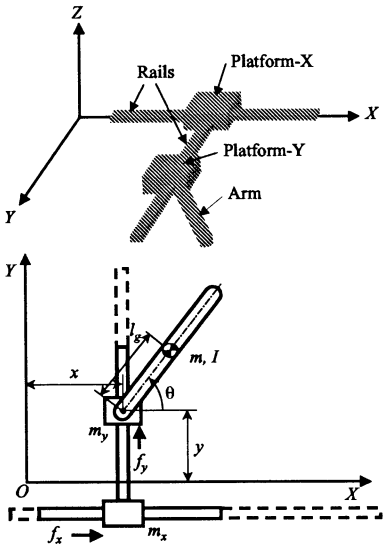


Figure 3: 3D view of PPR robot system and its parameters

3 Design of Elemental Controllers for a 3-DOF Underactuated Manipulator

The underactuated robot arm in **Fig. 3** shows the basic configuration of the robot manipulator which has three-DOF. A single arm with a passive joint is fixed on to a platform such that the complete system travels along a rail. Generalized coordinates and inputs of the robot manipulator in Fig. 3 are defined as $\boldsymbol{q} \triangleq [q_1 \;\; q_2 \;\; q_3]^T = [x \;\; y \;\; \theta]^T$ and $F = [f_x \;\; f_y \;\; 0]^T = [f_1 \;\; f_2 \;\; 0]^T$ respectively. The manipulator has the following parameters and variables:

m_x:	Mass of the platform X, 0.5 [kg]
m_y:	Mass of the platform Y, 0.8 [kg]
m:	Mass of the arm, 0.5 [kg]
l_g:	Distance to m from passive joint, 0.2 [m]
I:	Moment of inertia of arm, 6.7×10^{-3} [kgm^2]
x:	Distance of m_x along the X axis [m]
y:	Distance of m_y along the Y axis [m]
θ:	Angle measured from X axis [rad]
f_x:	Force on the platform X [N]
f_y:	Force on the platform Y [N]
d_x, d_y:	Damping coefficients of platforms of X and Y
d_θ:	Damping coefficient of passive joint

The equation of motion of the robot system is given by (1) and it has the following $M(\boldsymbol{q})$ and $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$:

$$\mathrm{M}(q) = \begin{bmatrix} m_x + m_y + m & 0 & -ml_g \sin\theta \\ 0 & m_y + m & ml_g \cos\theta \\ -ml_g \sin\theta & ml_g \cos\theta & ml_g^2 + I \end{bmatrix}$$

$$\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -ml_g\dot{\theta}^2 \sin\theta + d_x\dot{x} \\ -ml_g\dot{\theta}^2 \sin\theta + d_y\dot{y} \\ d_\theta\dot{\theta} \end{bmatrix}$$

Taking D as $D = \det(M)$ and $\hat{M}$ as cofactor of M gives

$$\begin{aligned} \ddot{\boldsymbol{q}} &= M^{-1}(\boldsymbol{q})\{-h(\boldsymbol{q}, \dot{\boldsymbol{q}}) + F\} \\ &= \frac{1}{D}\hat{M}(\boldsymbol{q})\{-h(\boldsymbol{q}, \dot{\boldsymbol{q}}) + F\} \end{aligned} \tag{6}$$

By analyzing the equation (6) and using the concept of computed torque method, we can synthesize the first control law for the sub system:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = -\begin{bmatrix} \frac{\hat{M}_{11}}{D} & \frac{\hat{M}_{12}}{D} & \frac{\hat{M}_{13}}{D} \\ \frac{\hat{M}_{21}}{D} & \frac{\hat{M}_{22}}{D} & \frac{\hat{M}_{23}}{D} \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} \frac{\hat{M}_{11}}{D} & \frac{\hat{M}_{12}}{D} \\ \frac{\hat{M}_{12}}{D} & \frac{\hat{M}_{22}}{D} \end{bmatrix}\begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{7}$$

- Control law 1

$$\begin{aligned} \begin{bmatrix} f_x \\ f_y \end{bmatrix} &= \begin{bmatrix} \frac{\hat{M}_{11}}{D} & \frac{\hat{M}_{12}}{D} \\ \frac{\hat{M}_{12}}{D} & \frac{\hat{M}_{22}}{D} \end{bmatrix}^{-1}\left\{\boldsymbol{s}^* + \mathcal{N}\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}\right\} \\ &= \frac{1}{\mathcal{D}_A}\hat{\mathcal{M}}\left\{\boldsymbol{s}^* + \mathcal{N}\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}\right\} \end{aligned} \tag{8}$$

here,

$$\boldsymbol{s}^* = \begin{bmatrix} \ddot{q}_1^* \\ \ddot{q}_2^* \end{bmatrix} \tag{9}$$

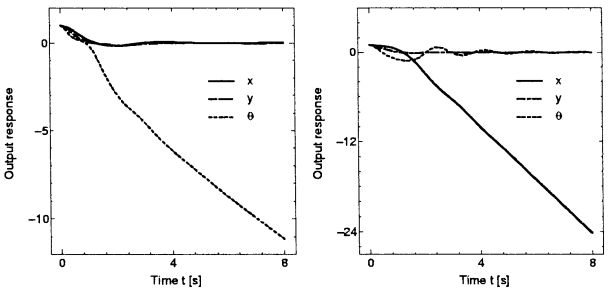
$$\mathcal{N} = \begin{bmatrix} \frac{\hat{M}_{11}}{D} & \frac{\hat{M}_{12}}{D} & \frac{\hat{M}_{13}}{D} \\ \frac{\hat{M}_{21}}{D} & \frac{\hat{M}_{22}}{D} & \frac{\hat{M}_{23}}{D} \end{bmatrix} \tag{10}$$

$$\hat{\mathcal{M}} = \begin{bmatrix} \frac{\hat{M}_{22}}{D} & -\frac{\hat{M}_{12}}{D} \\ -\frac{\hat{M}_{12}}{D} & \frac{\hat{M}_{11}}{D} \end{bmatrix} \tag{11}$$

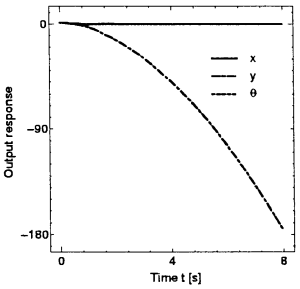
where $\mathcal{D}_A = (\hat{M}_{11}\hat{M}_{22} - \hat{M}_{12}^2)/D^2$ and modified accelerations $[\ \ddot{q}_1^*\ \ \ddot{q}_2^*]$ values are determined by a simple PD controller:

$$\begin{bmatrix} \ddot{q}_1^* \\ \ddot{q}_2^* \end{bmatrix} = \begin{bmatrix} \ddot{q}_{d1} + K_{v1}(\dot{q}_{d1} - \dot{q}_1) + K_{p1}(q_{d1} - q_1) \\ \ddot{q}_{d2} + K_{v2}(\dot{q}_{d2} - \dot{q}_2) + K_{p2}(q_{d2} - q_2) \end{bmatrix} \tag{12}$$

Similarly, we can design Control law 2 and Control law 3. Applying control laws 1, 2 and 3 to the robot manipulator seperately, it can be obtained the three graphs (see **Fig. 4**) respectively. In this study, we selected K_p, K_v values as: $K_{p1} = 4.0$, $K_{p2} = 4.0$, $K_{p3} = 1.0$ and $K_{v1} = 5.0$, $K_{v2} = 5.0$, $K_{v3} = 1.0$.



(a) control law 1 (b) control law 2



(c) control law 3

Figure 4: Time response of x, y, and θ using each PSC independently

4 Evolutionary Learning for Obtaining Fuzzy Rules

In the training phase, for a given initial configuration, we try to extract a common fuzzy rule base, which is optimized off-line by using GA. The membership functions for ith input are calculated by

$$\begin{aligned} \mu_{ij} &= \exp\{\ln(0.5)(e_i - w_{cij})^2/w_{dij}^2\} \\ &\quad i = 1, \ldots, n, \quad j = 1, \ldots, M \end{aligned} \tag{13}$$

where w_{cij} and w_{dij} represent the center and deviation of the jth Gaussian function of the ith input variable, respectively. Parameters of such Gaussian membership functions and the controller indices are encoded into chromosomes in order to obtain the optimum rule base of (2). **Figure 5** shows a typical untrained Gaussian functions associated with position error. Here, NB, N, Z, P, and PB give the linguistic labels, negative big, negative, zero, positive, and positive big respectively. Fitness function J is based on minimizing the following total error,

$$J = \sum_{k=1}^{T_N} \|\boldsymbol{x}_d(k) - \boldsymbol{x}^{\Phi_j}(k)\|_{W(k)}^2 \tag{14}$$

i.e., shapes of the Gaussian functions and the corresponding best assigned elemental controllers are evolved, where $\boldsymbol{x}^{\Phi_j}$ is the state vector starting from the initial configuration Φ_j. The angular position errors are used as the inputs to the fuzzy membership functions in the antecedent part and the one index of the PSCs is assigned in the consequent part of the fuzzy reasoning. The example, three-DOF underactuated planar manipulator that we focused in Section 3 is brought into to illustrate the training methodology. In this way, the Gaussian fuzzy membership

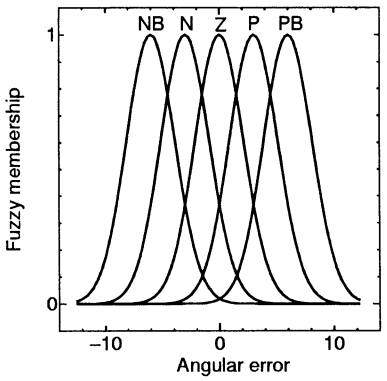


Figure 5: Untrained membership functions for an angular error (M=5)

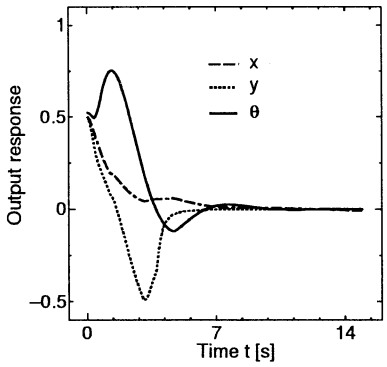


Figure 6: Time response of variables x, y and θ

functions with some linguistic variables were selected as in Fig. 5, setting the M value as $M = 7$. Note here that the termination condition of the GA process was set to 100 generations.

5 Control Results

We applied the concept explained in Section 4 to control three-DOF planar robot system, starting from $\boldsymbol{x}_0 = [0.5 \;\; 0.5 \;\; \pi/6 \;\; 0 \;\; 0 \;\; 0]^T$. The desired value was $\boldsymbol{x}_d = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$ and the total time frame t_{T_N} was set to 15 [s] with 10 [ms] sampling. According to **Fig. 6** and **7**, the state variables of the manipulator have converged to the desired values within an acceptable level of time frame. Moreover, 3-dimensional error plot is given in **Fig. 8**.

6 Conclusions and Discussion

We have proposed a fuzzy logic based switching controller the online operations of underactuated manipulators. To illustrate the design procedure, we applied the concept to control a three-DOF planar manipulator and the simulation results showed that the present control methodology was effective for obtaining good control results of underactuated manipulators, which gave a practical stability like a uniformly ultimate bounded stability. One of the major advantages of this method is that the entire system can be controlled

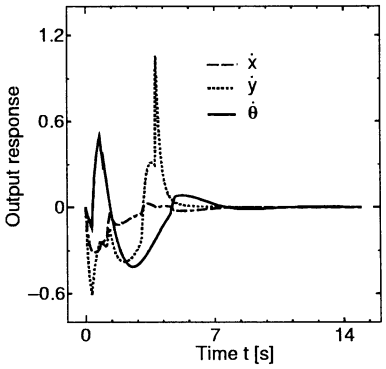


Figure 7: Time response of velocities $\dot{x}$, $\dot{y}$ and $\dot{\theta}$

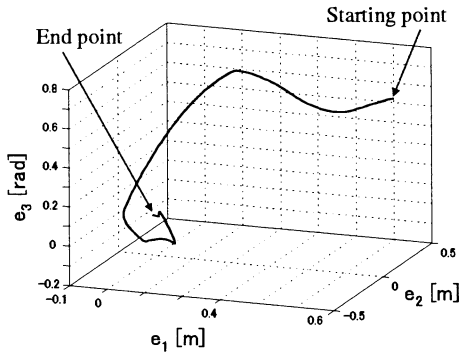


Figure 8: 3D Error trajectory of e_1, e_2 and e_3

without using rigorous linearizations or deformations of the original nonlinear system and employing simple computed torque controllers as elemental controllers in the control system. In addition, the following factors can be considered for further improvements:

- Include velocity error as another input to the antecedent part for complicated systems.
- Train the system for more trajectories.
- Compensate system uncertainties such as friction in the training process, including them as constraints.

References

[1] I. Kolmanovsky and N.H. McClamroch, "Development in Nonholonomic Control Problems," *IEEE Control Systems Magazine*, vol. 15, no. 6, pp. 20–36, 1995.

[2] L. Udawatta, K. Watanabe, K. Kiguchi, and K. Izumi, "Developments in underactuated manipulator control techniques and latest control using AI," in *Proc. of 7th AROB,* vol. 2, 2002, pp. 425–428.

[3] H. Arai, K. Tanie, and N. Shiroma, "Nonholonomic Control of a Three DOF Planar Underactuated Manipulator," *IEEE Trans. on Robotics and Automation,* vol. 14, no. 5 pp. 681–695, 1998.

[4] L. Udawatta, K. Watanabe, K. Izumi, and K. Kiguchi, "Control of Underactuated Robot Manipulators Using Switching Computed Torque Method: GA Based Approach," *Journal of Soft Computing,* to be appeared, 2003.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Identification of Periodic Function Using Dynamical Neural Network

Kunihiko Nakazono
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
nakazono@tec.u-ryukyu.ac.jp

Kouhei Ohnishi
Keio University
Hiyoshi 3–14–1, Yokohama. 222–8522
ohnishi@sd.keio.ac.jp

Hiroshi Kinjo
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
kinjo@tec.u-ryukyu.ac.jp

Tetsuhiko Yamamoto
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
yamamoto@tec.u-ryukyu.ac.jp

Abstract

In this paper we propose a dynamical neural network with the characteristics of inertia, viscosity, and stiffness and its training algorithm based on the backpropagation method. The simulation results show that the dynamical neural network with characteristics of inertia, viscosity, and stiffness trained by the backpropagation method obtains good training performances for time series patterns generated from sine functions. In this paper, we compare the dynamical neural network with a feed-forward neural network and recurrent neural network in order to verify the validity of the dynamical neural network.

1 Introduction

Recently, recurrent neural networks have been researched more actively than layered neural networks which have static mapping capability[1, 2]. The recurrent neural network is a possible candidate to improve the dynamic characteristics of a system because it has a structure with feedback combination in the neuron unit and time delay is taken into consideration. There are several methods for training recurrnet neural networks, for example, the backpropagation through time (BPTT) method which was expanded from the error backpropagation method, the real-time recurrent learning (RTRL) method, and genetic algorithms[3, 4]. However, there is a problem that the network structure in the recurrent neural network becomes complex in comparison with the layered neural network and the training algorithm.

We propose a dynamical neural network that can achieve dynamics of a system and has a network structure with the characteristics of inertia, viscosity, and stiffness without taking time delay into consideration. Moreover, the proposed dynamical neural network can be built with the training algorithm based on the error backpropagation method.

In order to verify the validity of the proposed dynamical neural network, it was made to identify the periodic function, a simple one periodic sine waveform and several periodic sine waveforms. Simulation results show that the proposed dynamical neural network has higher performance than the feed-forward neural network and recurrent neural network.

2 Dynamical neural network

In this study, a dynamical neural network (DNN) is configured using a neuron which has the characteristics of inertia, viscosity, and stiffness. In this model, we consider the output from neuron to be given the characteristics of inertia, viscosity, and stiffness. The proposed DNN is composed of three hierarchy layers and the neruon adopts a hidden layer and an output layer. The structure of the DNN is shown in figure 1.

The equations for the DNN are expressed as follows.

$$y_i = u_i, \quad (i = 1, \cdots, I) \tag{1}$$

$$y_j = K_j f_j(net_j) + D_j \dot{f}_j(net_j) + M_j \ddot{f}_j(net_j) \tag{2}$$

$$net_j = \sum_{i=1}^{I} w_{ij} y_i, \quad (j = 1, \cdots, J) \tag{3}$$

$$y_k = K_k f_k(net_k) + D_k \dot{f}_k(net_k) + M_k \ddot{f}_k(net_k) \tag{4}$$

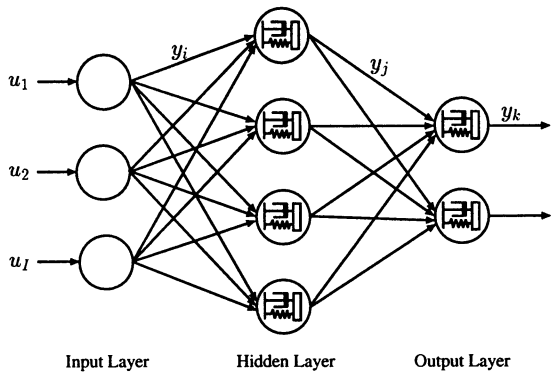


Figure 1: Structure of dynamical neural network

$$net_k = \sum_{j=1}^{J} w_{jk} y_j, \quad (k = 1, \cdots, K) \tag{5}$$

Here, u shows input to the neural network, and y shows output from each neuron. net is the total sum of products of the connecting weight w and the output y from each neuron. The subscripts i, j, and k are the numbers of input layers, hidden layers, and output layers, respectively.

3 Training algorithm

First, we derive a minimizing sequence of the measurement of error function E:

$$E = \frac{1}{2}\sum_k e_k^2 = \frac{1}{2}\sum_k (d_k - y_k)^2 \tag{6}$$

where d_k is the desired signal. In order to minimize the measurement of error function E of equation (6), the connecting weights of the dynamical neural network are modified.

Based on the steepest descent method, the derivation of the error backpropagation algorithm is shown in this section.

$$w^{\text{new}} = w^{\text{old}} + \Delta w \tag{7}$$

$$\Delta w = -\eta \frac{\partial E}{\partial w} \tag{8}$$

η is the training rate ($\eta > 0$).

The derivations of $\partial E/\partial w_{jk}$ and $\partial E/\partial w_{ij}$ are described as follows.

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}} = \frac{\partial E}{\partial net_k} \cdot y_j \tag{9}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}} = \frac{\partial E}{\partial net_j} \cdot y_i \tag{10}$$

In the upper expression, the derivations are defined as

$$\delta_k = \frac{\partial E}{\partial net_k} \tag{11}$$

$$\delta_j = \frac{\partial E}{\partial net_j} \tag{12}$$

and δ_k and δ_j are calculated, respectively. First, δ_k is expanded as

$$\begin{aligned} \delta_k &= \frac{\partial E}{\partial net_k} = \frac{\partial E}{\partial e_k} \cdot \frac{\partial e_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial net_k} \\ &= e_k \cdot (-1) \cdot \frac{\partial y_k}{\partial net_k}. \end{aligned} \tag{13}$$

The derivation of $\partial y_k/\partial net_k$ is described as

$$\begin{aligned} \frac{\partial y_k}{\partial net_k} &= K_k \frac{\partial f_k(net_k)}{\partial net_k} + D_k \frac{\partial \dot{f}_k(net_k)}{\partial net_k} \\ &\quad + M_k \frac{\partial \ddot{f}_k(net_k)}{\partial net_k} \end{aligned} \tag{14}$$

$$\frac{\partial \dot{f}_k(net_k)}{\partial net_k} = f_k''(net_k) \cdot \dot{net}_k \tag{15}$$

$$\frac{\partial \ddot{f}_k(net_k)}{\partial net_k} = f_k'''(net_k) \cdot \dot{net}_k^2 + f_k''(net_k) \cdot \ddot{net}_k. \tag{16}$$

δ_k finally becomes equation (17).

$$\begin{aligned} \delta_k &= -e_k [K_k f'(net_k) + D_k f_k''(net_k) \cdot \dot{net}_k \\ &\quad + M_k (f_k'''(net_k) \cdot \dot{net}_k^2 + f_k''(net_k) \cdot \ddot{net}_k)] \end{aligned} \tag{17}$$

When δ_j is calculated in the same way, it becomes

$$\begin{aligned} \delta_j &= \frac{\partial E}{\partial net_j} \\ &= \sum_k \left(\frac{\partial E}{\partial e_k} \cdot \frac{\partial e_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial net_k} \right) \cdot \frac{\partial net_k}{\partial y_j} \cdot \frac{\partial y_j}{\partial net_j} \\ &= \sum_k \delta_k \cdot w_{jk} \cdot \frac{\partial y_j}{\partial net_j} \\ &= [K_j f'(net_j) + D_j f_j''(net_j) \cdot \dot{net}_j \\ &\quad + M_j (f_k'''(net_j) \cdot \dot{net}_j^2 + f_j''(net_j) \cdot \ddot{net}_j)] \\ &\quad \times \sum_k w_{jk} \delta_k. \end{aligned} \tag{18}$$

It becomes possible for the DNN to train the neural network utilizing the concept of error backpropagation.

4 Simulation

The effectiveness of the dynamical neural network proposed in this paper is verified by the numerical simulation of identification of the periodic function. The DNN is structured to have a single input and single output (SISO). The method by which a time series signal is identified is shown in figure 2.

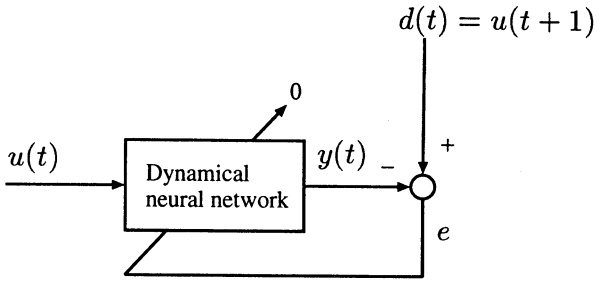


Figure 2: Identification of periodic function

The desired signal, that is the training data $d(t)$, is the signal which passed one sampling time ahead of the input signal $u(t)$.

4.1 One periodic function

In order to facilitate analysis, simulation shows that the DNN identified the time series signal of the one sine periodic function of cycle T as

$$u(t) = \sin\left(\frac{2\pi t}{T}\right). \tag{19}$$

At this time, the number of neurons in the hidden layer is made five. The inertia parameter is fixed to 1.0, and viscosity and stiffness parameters are sought by trial and error, and set as follows.

$$\begin{aligned}(M_j, D_j, K_j) &= (1.0, 8.00, 9.87),\ (1.0, 0.98, 2.02),\\ &\quad (1.0, 0.78, 9.38),\ (1.0, 1.24, 9.95),\\ &\quad (1.0, 3.77, 1.90)\\ (M_k, D_k, K_k) &= (1.0, 5.45, 5.96)\end{aligned}$$

The training involved 1,000 iterations. The process of the error function is shown in figure 3. In order to show the effectiveness of the DNN, it is compared with the RNN-trained BPTT method. The result of the simulation clearly shows that the DNN has better performance than the RNN. The result of regenerating the signal utilizing the trained DNN is shown in figure 4.

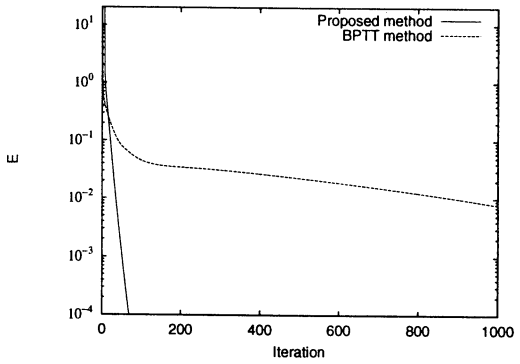


Figure 3: Error function ($T = 16$)

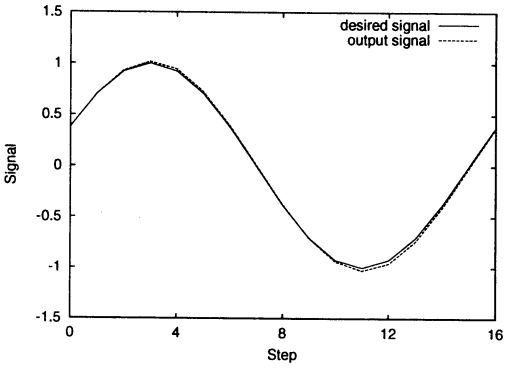


Figure 4: Regenerated waveform

The output of the DNN negligibly deviates from the desired signal.

4.2 Several periodic functions

In this section, the simulation shows that a signal which has more than one cycle was identified. The number of neurons in the hidden layer is nine. Each parameter was set as follows, and training involved 10,000 iterations.

$$\begin{aligned}(M_j, D_j, K_j) &= (1.0, 4.16, 8.43),\ (1.0, 0.78, 0.32),\\ &\quad (1.0, 4.00, 0.33),\ (1.0, 1.61, 8.26),\\ &\quad (1.0, 7.51, 2.75),\ (1.0, 1.35, 5.70),\\ &\quad (1.0, 6.91, 3.30),\ (1.0, 9.86, 9.50),\\ &\quad (1.0, 8.92, 1.88)\\ (M_k, D_k, K_k) &= (1.0, 0.27, 2.07)\end{aligned}$$

The trend of the error function is shown in figure 5. In the same way, it is compared with the RNN-trained BPTT method. The result of the simulation shows that the DNN has better performance than the RNN. The signal regenerated utilizing the trained DNN is shown in figure 6.

The outputs of the DNN negligibly deviate from the desired signal.

5 Conclusion

In this study, the dynamical neural network, which had the effectiveness of a neuron with characteristics of inertia, viscosity, and stiffness passed on to each neurons, was configured. The validity of the proposed dynamical neural network was shown by comparison with a conventional neural network.

References

[1] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (1989), Parallel Distributed Processing, *The MIT Press.*

[2] S. Omatu, M. Khalid, and R. Yusof (1995), Neuro–Control and Its Applications, *Springer.*

[3] R. J. Williams and D. Zipser (1989), A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, 1, No.2, pp.270–280.

[4] H. Kinjo, K. Nakazono, and T. Yamamoto (1997), Pattern Recognition for Time Series Signals Using Recurrent Neural Networks by Genetic Algorithms (in Japanese), *Trans. of ISCIE*, Vol. 10, No. 6, pp.304–314.

[5] K. Kiguchi and T. Fukuda (1998), Neural Network Controllers for Robot Manipulators – Application of Damping Neurons, *Advanced Robotics*, Vol. 12, No. 3, pp.191–208.

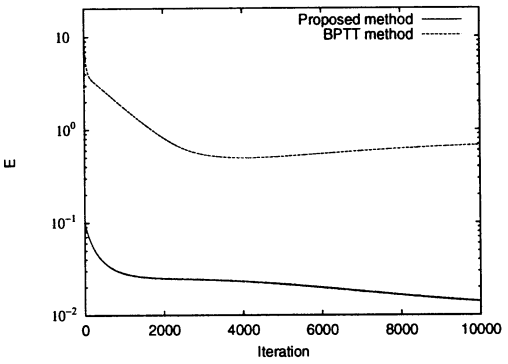
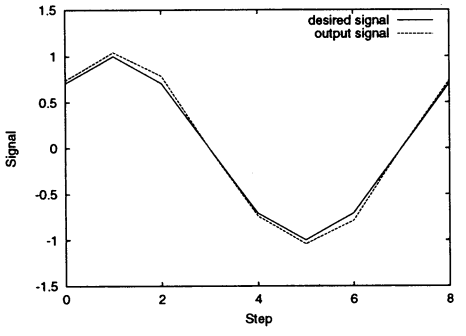
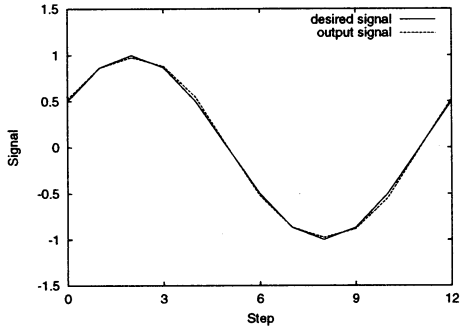


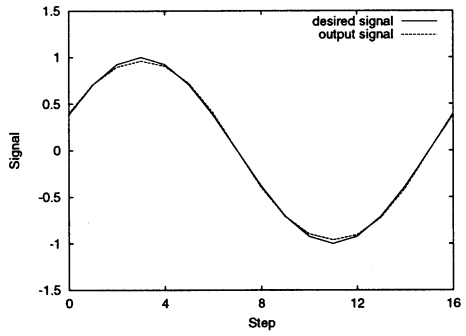
Figure 5: Error function($T = 8, 12, 16$)



$T = 8$



$T = 12$



$T = 16$

Figure 6: Regenerated waveforms

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Training of Pulse Interval for Spiking Neural Networks Using Genetic Algorithm

Susumu Kamoi
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
kamoi@mibai.tec.u-ryukyu.ac.jp

Hiroshi Kinjo
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
kinjo@tec.u-ryukyu.ac.jp

Kunihiko Nakazono
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
nakazono@tec.u-ryukyu.ac.jp

Abstract

In this paper, a training method of a spike train for spiking neural networks (SNNs) by the use of genetic algorithms (GAs) is proposed. There have been some reports on training methods of artificial neural networks. However, there are only a few reports on SNNs. SNNs are known to be highly similar to biological neural networks because they treat spike trains. SNNs process information based on pulse signals. In SNNs, spiking neurons receive spike pulses from and fire spike pulses to other neurons. The spiking neurons have a characteristic of suddenly changing the membrane potential immediately before and after firing. The characteristics of the potential behavior cause some difficulties in training SNNs. Many currently used training methods of SNNs apply the Hebb rule or gradient methods. However, under the application of the Hebb rule, SNNs sometimes failed the training test. Furthermore, the gradient methods include complicated calculations. In this study, GAs are applied to the training of SNNs. In the GA method, it is not required to consider the characteristics of membrane potentials of the spiking neurons. By applying the GA to the training of SNNs, we obtain a superior SNN that is able to process input pulses and fire the desired spike trains. Simulations of SNN training for spike trains are shown.

1 Introduction

Recently, a research on spiking neural networks (SNNs) has been initiated [1]. SNNs are known to be highly similar to biological neural networks [2]. SNNs treat spike pulses and process the signals based on spike trains. In the SNNs, spiking neurons receive spike pulses from and fire spike pulses to other neurons. The spiking neurons have a characteristic of suddenly changing the membrane potential immediately before and after firing. The characteristics of the potential behavior cause some difficulties in training SNNs. Many currently used training methods of SNNs apply the Hebb rule or gradient methods [1],[3]. However, under the application of the Hebb rule, the SNNs sometimes failed the training test. Furthermore, gradient methods include complicated calculations.

In this paper, the genetic algorithm (GA) [4] is applied to the training of SNNs. The GA method does not require complex calculations. The pulse intervals of the spike train are trained for SNNs by using the GA. Finally, simulation shows that the GA method is very effective.

2 Spiking Neural Network

2.1 Spiking Neuron

Figure 1 shows an integrate-and-fire spiking neuron model. The neuron model consists of two resistor R and capacitor C circuits. The first RC circuit models the synaptic connection between neurons, and the second RC circuit models the membrane potential in the neuron. When the neuron receives a pulse signal from another neuron, the former RC circuit outputs current $I(t)$:

$$I(t) = \sum_{j=1}^{N} \omega_j \sum_{f=1}^{F_j(t)} \alpha(t - t_j^f), \qquad (1)$$

where N is the number of presynaptic neurons; $F_j(t)$ is the number of spikes until time t on the synapse j; and

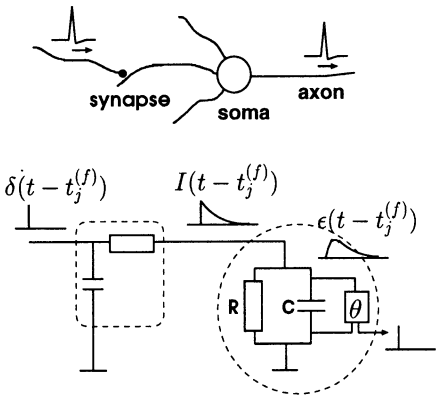


Fig. 1 Integrate-and-fire neuron model

ω_j is the connecting weight. $\alpha(s)$ denotes the current response of the postsynaptic neuron after receiving a spike from the presynaptic neuron. $t_j^{(f)}$ is the input time of spike f on synapse j.

Current $\alpha(s)$ based on a single pulse is

$$\alpha(s) = \frac{1}{\tau_s}\exp(-\frac{s}{\tau_s})H(s), \tag{2}$$

where τ_s is a synaptic time constant and $H(x)$ is the Heaviside step function which vanishes for $x \leq 0$ and has a value of one for $x > 0$.

On receiving current $I(t)$, capacitor C of the second RC circuit is charged and outputs voltage as membrane potential $u(t)$. Membrane potential $u(t)$ is described by

$$\tau_m \frac{du}{dt} = -u(t) + RI(t), \tag{3}$$

where $\tau_m = RC$ is the membrane time constant.

When a neuron receives many signal pulses, the membrane potential of the neuron gradually increases. When the voltage of the membrane potential exceeds threshold level θ, the neuron fires a spike signal to other neurons. After firing, the membrane potential of the neuron is reset to the initial voltage.

2.2 Structure of SNN

Figure 2 shows a single-input, single-output recurrent spiking neural network. All of the neurons are connected to each other. The input neuron receives the spike train and is activated, and sometimes, the input neuron fires. Upon receiving the spike pulses from the input neuron, the other neurons in the network are activated and some of them fire. If the membrane potential of the output neuron exceeds the threshold level, the neuron fires a spike pulse to the other neurons and also out of the network.

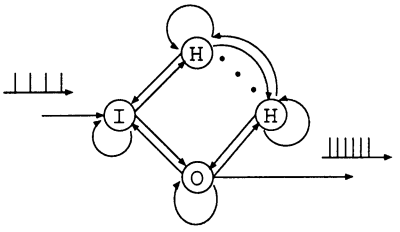


Fig. 2 SISO recurrent SNN

We want the SNN to output a certain spike train. In this study, the SNN is trained to output the spike trains. The training method is to use the GA. The GA is applied to modify the network connections. Moreover, in order to improve the training performance, we propose to modify the time constants τ_s and τ_m by using the GA.

3 Utilizing the GA

3.1 Training SNN by the GA

The algorithm of SNN training has the following sequence.

Process 1: Generate the initial SNNs at random. Connecting weights of SNNs are transformed to the chromosome.
Process 2: Rank the SNNs based on the fitness function.
Process 3: Select the parent SNNs.
Process 4: Perform crossover operation for the chromosome to produce new SNNs.
Process 5: Perform mutation for some new SNNs.
Process 6: Rank all of the SNNs including new SNNs. Go to process 3.

The connecting weights of SNNs are transformed to chromosome g. The transform equation is

$$g = [(\frac{w}{N_r} + 1)\frac{65535}{2}], \tag{4}$$

where the symbol $[\cdot]$ denotes a Gaussian function which transforms decimals to integers and N_r is a coefficient that indicates the range of w.

All of the SNNs are ranked based on the fitness function f_i:

$$f_i = \frac{E_{\max} - E_i}{S}, \quad i = 1, 2, \cdots, N_s, \tag{5}$$

where E_i is an error function of each i of SNNs; $E_{\max}$ is the maximum error of SNNs; S is the total error of all SNNs; N_s is the number of SNN individuals.

The selection of a pair of SNNs is by roulette-wheel parent selection based on the fitness function f_i. The probability P_i that SNN individual i is selected to be a parent SNN is

$$P_i = \frac{f_i}{\sum_{j=1} f_j}. \quad (6)$$

We use a uniform crossover to produce new SNNs. The chromosome of the new SNNs is changed using the bit mutation operator at any rate.

3.2 Evaluation function

Consider the SNN learning problem for a spike train using the single-input, single-output recurrent SNN. When spike pulses with constant interval I_i are input to the SNN, the SNN outputs a spike train with firing times of

$$F_o = \{t_o^{(1)}, t_o^{(2)}, \cdots, t_o^{(F_N)}\}, \quad (7)$$

where F_N is the number of firing events. We want the SNN to output a spike train with constant time interval I_d. The evaluation function of SNNs that have firing times F_o is defined as

$$E = \sum_{f=2}^{F_N} \left(1 - \frac{I_o^{(f)}}{I_d}\right)^2, \quad (8)$$

$$I_o^{(f)} = t_o^{(f)} - t_o^{(f-1)}, \quad (9)$$

where $I_o^{(f)}$ denotes an output pulse interval at any f firing event.

4 Simulations

4.1 Training exercises

We examine the following three exercises for SNN with three hidden neurons. Tables 1, 2 and 3 show the training pulse intervals.

Table 1 Desired interval I_d (Exercise 1)

Pattern number	I_i[ms]	I_d[ms]
1	5	5
2	10	10
3	15	15

Table 2 Desired interval I_d (Exercise 2)

Pattern number	I_i[ms]	I_d[ms]
1	5	10
2	10	20
3	15	30

Table 3 Desired interval I_d (Exercise 3)

Pattern number	I_i[ms]	I_d[ms]
1	5	15
2	10	10
3	15	5

4.2 Training results

Table 4 Parameters of spiking neuron

Range of membrane time constant τ_m [ms]	$[1, 20]$
Range of synaptic time constant τ_s [ms]	$[0.2, 4.0]$
Firing threshold θ [mV]	20.0
Magnitude of pulse [mV]	100

Table 5 Parameters of GA process

Weight range	$[-50, 50]$
Population of SNN	40
Number of produced SNN	20
Generation	500
Mutation probability	10 %

Tables 4 and 5 show the parameters of the spiking neuron and the GA process. Table 6 shows the rate of successful evolution. The values of the rate are taken to be valid when the total error of patterns is less than 0.1 in 20 trial simulations. In the second line in the table, the constant neuron model parameters used are $\tau_m = 5.0$[ms] and $\tau_s = 1.0$[ms]. The results in the table indicate that Exercises 1 and 2 result in well-trained SNNs by both methods. However for Exercise 3, the training problem is very difficult, and the method of using the changing w and constant parameters does not successfully train the SNNs. The method of using changing w, τ_s and τ_m is only slightly successful in training SNNs. In these results, we can see that the modification of τ_s and τ_m is useful for SNN evolution.

Table 6 Rate of successful evolution [%]
(Total error of the patterns < 0.1)

	Exercise 1	Exercise 2	Exercise 3
Changing w and constant τ_s, τ_m	90	65	0
Changing w, τ_s, τ_m	100	95	15

Figure 3 shows an example of training SNNs using the GA for Exercise 3. It can be seen that the evolution of SNN is successful.

4.3 Performance of SNN

We checked the SNN trained for Exercise 3. Table 7 shows the values of the output pulse intervals.

From the table, three input intervals $I_i = 5$[ms], 10[ms] and 15[ms] were used for training intervals. Considering Tables 3 and 7, we can see that the SNN could output spike pulses for the trained pulse intervals.

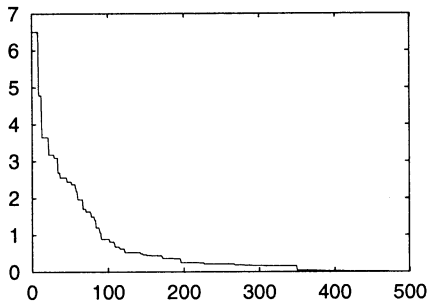


Fig. 3 Evolution results

Table 7 Time intervals of the output neuron

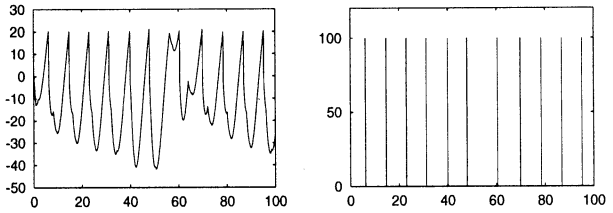
I_i [ms]	Output time intervals [ms]								
	I_o^2	I_o^3	I_o^4	I_o^5	I_o^6	I_o^7	I_o^8	I_o^9	I_o^{10}
5	15.2	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
6	12.1	12.1	12.0	12.0	12.0	12.0	12.0	12.0	12.0
7	12.0	9.4	11.7	11.8	11.7	9.3	11.7	9.3	11.7
8	8.5	8.3	8.2	8.8	8.0	12.5	8.4	8.6	8.4
9	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0
10	9.9	9.9	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	10.4	5.6	8.2	8.8	5.3	7.0	5.1	5.0	5.0
12	5.4	7.2	5.1	7.8	5.2	7.1	5.1	5.0	5.0
13	5.0	7.5	5.2	5.0	5.0	5.0	5.0	5.0	5.0
14	5.4	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
15	5.4	5.0	5.0	4.9	5.0	5.0	5.0	5.0	5.0

For the untrained input pulse intervals, we can observe three phenomena of the output intervals: a constant interval, an oscillating interval, and a leading interval. For example, when a spike train with time interval $I_i = 8$[ms] is input to the network, the SNN outputs the spike pulses of approximately $I_o = 8.4$[ms]. When the input pulse interval is $I_i = 7$[ms], the output pulse interval oscillates between $I_o = 9.3$[ms] and 11.7[ms]. When the input pulse interval is $I_i = 11$[ms], the time intervals of output pulses are leading to $I_o = 5$[ms].

Figures 4, 5 and 6 show the reactions of the output neuron when time intervals of input pulses are $I_i =$ 8[ms], 7[ms] and 11[ms], respectively.

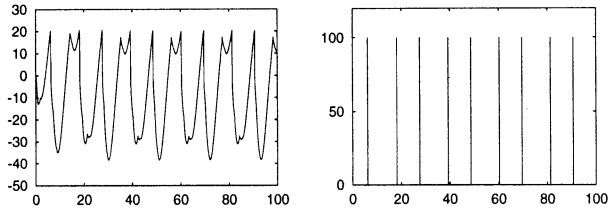
5 Conclusion

In this study, GA was applied to the training of pulse intervals for SNNs. The neuron model used was a simple integrate-and-fire model. The SNN used was a single-input, single-output recurrent neural network. The connecting weights of SNN were modified, by the GA, to fire spikes that have the desired pulse intervals. In order to improve the training performance, we propose that the parameters of the spike neuron should also be changed by GA when training the SNN. The



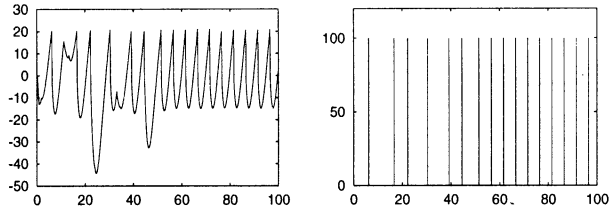
(a) Membrane potential (b) Output pulse

Fig. 4 Reaction of the SNN ($I_i = 8$ [ms])



(a) Membrane potential (b) Output pulse

Fig. 5 Reaction of the SNN ($I_i = 7$ [ms])



(a) Membrane potential (b) Output pulse

Fig. 6 Reaction of the SNN ($I_i = 11$ [ms])

evaluation function of SNN training was defined as a simple error function. Simulation results showed that the GA is effective for SNN training. After training, the SNN could output the desired spike trains. The characteristics of the SNN were analyzed.

References

[1] For example, 2001 Special Issue: Spiking Neurons in Neuroscience and Technology, Neural Networks, Vol. 14, Nos 6-7, 2001

[2] W. Mass and C. M. Bishop: Pulsed Neural Networks, MIT Press, 1999

[3] K. Selvaratnam, Y. Kuroe and T. Mori: Learning Methods of Recurrent Spiking Neural Networks, Transactions of ISCIE, Vol. 13, No. 3, pp. 95-104, 2000

[4] L. Davis: Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Information Separation of Position and Direction of a Robot by Self-Organizing Map

K. Kurata, N. Oshiro
Faculty of Engineering, University of the Ryukyus,
Senbaru 1, Nishihara, Okinawa. 903–0213
kurata@mibai.tec.u-ryukyu.ac.jp

Abstract

A model to self-organize a map for visual recognition of position and direction of a robot moving autonomously in a room. The robot is assumed to have visual sensors. The model is based on Kohonen's SOM (Self-Organizing Map), which was proposed as a model of self-organization of cortex. Ordinary SOM consists of two-dimensional array of neuron-like feature detector units. In our model, however, units are arranged in three-dimensional array and periodic boundary condition is assumed in one dimension. Also, some new learning rules are added. Our model is shown by computer simulation to form a map which can extract from the visual input two factors of information separately, which are position and direction of the robot. This is an example of so-called *two-factor problems*. In our algorithm, the difference of topology of information is used to separate two factors of information.

Key Words: robot navigation, self-organization, SOM, information separation

1 Introduction

We, human beings, can find our position and direction instantly from what we see in the environment familiar to us. We learn form experience and memorize the scenes and are supposed to have some kind of map in our brain. This ability is obviously desirable for robots to share. We suppose a robot with visual sensors around it to move autonomously in a room. We propose an algorithm with which the robot can self-organize a map to recognize the position and direction of itself in the room.

In some preceding works, SOM was used for localization of a robot[2]. Also, a kind of associatue neural network was used for robot navigation [7]. In this study, we focus on *separation of two factors of information*. The visual input obtained by the sensors is a function of two factors of information or the position and direction of the robot. With a map properly organized with our model, it will be shown, the two factors can be extracted separately from the visual input.

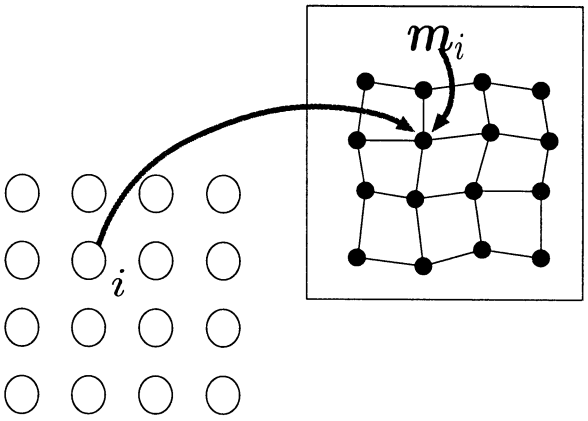


Figure 1: Two-dimensional SOM with two-dimensional input vector space

2 Kohonen's SOM Algorithm

SOM (Self-Organizing Map) was proposed by Kohonen [3, 4] as a model of cerebral cortex and its self-organization. It was successful for SOM to reproduce the functional map on visual cortex [1, 6], and was applied to many kinds of data as a statistical tool of nonlinear auto-regression [4].

SOM usually consists of two-dimensional array of neuron-like units, each of which has an *reference vector* $\boldsymbol{m}_i$, which is n-dimensional as well as input vectors $\boldsymbol{x}$ (Figure 1). SOM algorithm is as follows:

(SOM1) Assign random values for the reference vec-

tors.

(SOM2) Choose or generate an input vector $\boldsymbol{x}$ in some random manner.

(SOM3) Find the unit whose reference vector is the closest to the input vector $\boldsymbol{x}$ in the sense of n-dimensional Euclidian distance. The unit is called the *winner*.

(SOM4) Apply the following learning rule for the reference vectors:

$$\boldsymbol{m}_i := \boldsymbol{m}_i + h_{ci}(\boldsymbol{x} - \boldsymbol{m}_i), \tag{1}$$

$$h_{ci} = \alpha \exp\left(-\frac{\|\boldsymbol{r}_c - \boldsymbol{r}_i\|^2}{2\sigma^2}\right), \tag{2}$$

where $\boldsymbol{r}_i$ is the two-dimensional position of the unit i on the unit array, and c is the index assigned to the winner unit.

(SOM5) Return to (SOM2) and repeat (SOM2) - (SOM4) many times.

This is called *neighborhood learning*, because modification of the reference vectors takes place mainly in the neighborhood of the winner, as is represented by h_{ci}, which is called the *neighborhood function*. As the result of this learning rule, the reference vectors are scattered all over the input signal region and neighboring units tend to grow similar reference vectors, which means a topological map of the input signal region is *self-organized* on the unit array.

3 Input Data

The robot is placed in a square room with four walls of four different colors. The robot has 60 sensors all around it, and each of the sensors can distinguish the color of the wall in front of it (Figure 2). Thus, the information obtained by one sensor is represented by one of the following four-dimensional vectors, (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), so that the whole visual input consists of 240 bits, which are divided into 16 sets and summarized into a 16-dimensional input vector for SOM. Therefore, the reference vectors in our model are 16-dimensional, also.

4 Separation of Direction and Position

We used SOM consisting of three-dimensional unit array, for the input vectors carry essentially three-dimensional information, that is two-dimensional position and one-dimensional direction. We design our

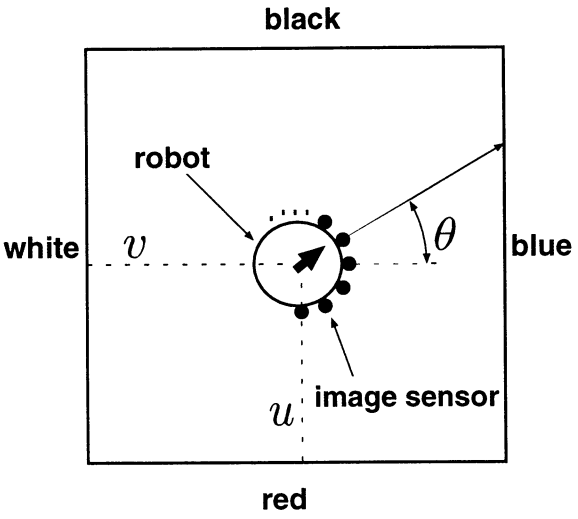


Figure 2: Robot in a room and its sensors

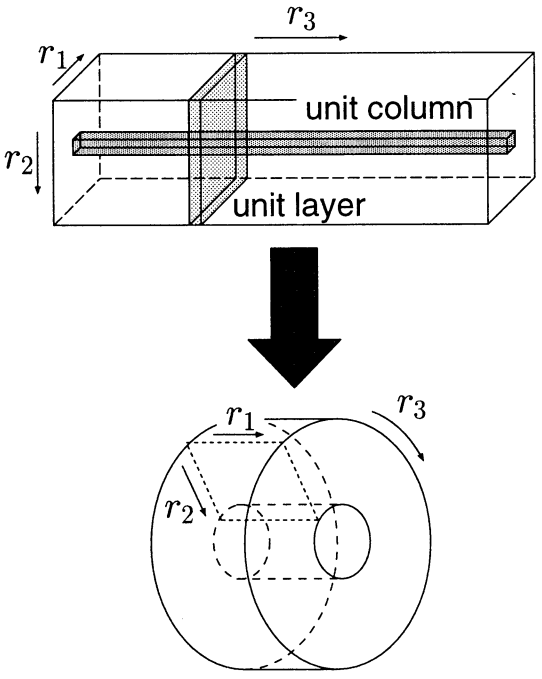


Figure 3: Three-dimensional SOM with periodic boundary condition

model so that the position of the robot should be mapped along the two dimensions, r_1 and r_2 of the array and the direction along r_3. Let two-dimensional sub-arrays parallel to r_1 and r_2 axes called *unit layers*, and one-dimensional unit sub-arrays parallel to r_3 axis called *unit columns* (Figure 3).

The robot moves randomly within the room. Here, we assume the robot changes one and only one of its position and direction at each step. Taking advantage of this crucial assumption, we formulate our new algorithm, which can assign the two factors of information to proper dimensions of unit array:

(RN1) Assign random values for the reference vectors.

(RN2) Set the position and direction of the robot randomly. Calculate 16-dimensional input vector.

(RN4) Find the winner and apply the neighborhood learning rule.

(RN5) Change the position of the robot randomly or change the direction of the robot randomly, and calculate 16-dimensional input vector.

(RN6) Choose the winner and apply the neighborhood learning rule. If the position of the robot was changed in (RN5), the new winner must be searched for within the unit layer to which the former winner belongs. That is the new winner is defined as the unit whose reference vector is the closest in the layer to the input vector $\boldsymbol{x}$. If the direction was changed, the new winner must be searched for within the unit column to which the former winner belongs.

(RN7) Return to (RN5) and repeat (RN5) and (RN6) many times.

Simulations have shown (RN6) is necessary to separate the two factors of information. We used $7 \times 7 \times 20$ units for simulation. Therefore, our model has 20 unit layers and 49 unit columns. Array is designed to be longer in the dimension of r_3, which is also helpful for the information separation, because input changes more drastically when the direction in changed. Because the information of direction has topology of circle, we assumed the periodic boundary condition in the third dimension of array. This boundary condition makes each unit column to form circular array. This means that, in the neighborhood function (Equation 2), the distance in the third dimension r_3 should be measured along the shorter arc of the circle.

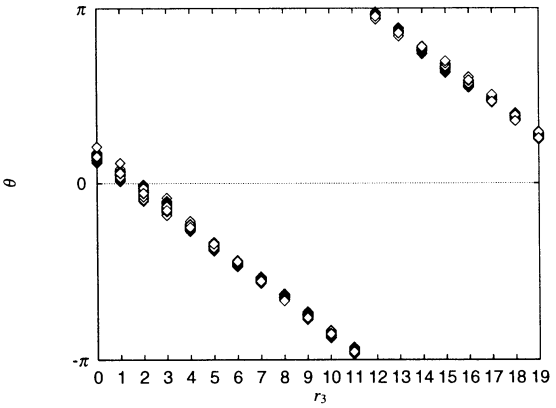


Figure 5: Preferred direction of units

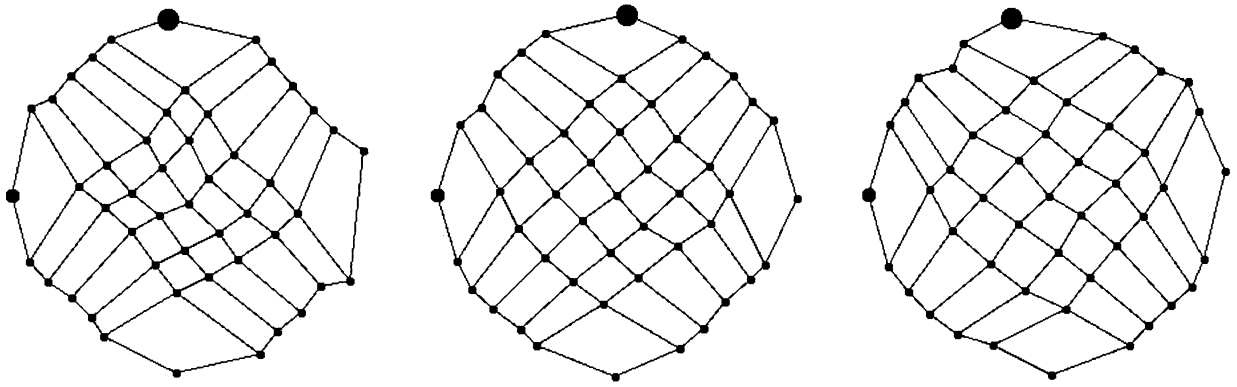
5 Simulation Results

After the leaning, each of the units acquires its *preferred position* and *preferred direction*, that is, a unit becomes the winner only when the robot comes into some small region of the room with some specific direction. Figure 4 shows almost the same distributions of the preferred positions of the units in three different unit layers. This means that units in one layer have different preferred positions each and units in one column have a same preferred position, which means the position of the robot is mapped along r_1 and r_2, the first and the second dimension of the array, and each unit column corresponds to a different point in the room. Figure 5 shows the preferred direction of the units as a function of r_3. We can see units in one layer have a same preferred direction, and each unit layer corresponds to different directions of the robot. The direction of the robot, therefore, is mapped along r_3, the third dimension of the array.

6 Discussion

After the self-organization of the map, r_1 and r_2 coordinates of the winner unit tell us the position of the robot in the room, and r_3 coordinate tells the direction. Moreover, neighboring relation of the units in SOM will be helpful to develop a path-finding algorithm based on this map.

This algorithm may not be successfully applied to a robot in a room whose shape is unknown, because the unit layers in our model are rigidly square. Our model will be improved by adopting *neural gas* model for the unit layers. Neural gas, proposed by Martinetz [5], is more flexible and can fit for wide variety of topology.



(A) $r_3 = 0$ (B) $r_3 = 10$ (C) $r_3 = 15$

Figure 4: Preferred positions of units in three different unit layers

7 Conclusion

We have proposed a model to self-organize a map or array of units, which can learn to recognize from visual input the position and direction of a robot in a room. During the learning phase, the robot is assumed to move randomly within the room. The model is based on three-dimensional SOM. Position and direction are recognized separately or mapped along different dimensions of the array.

References

[1] Erwin, E., Obermayer, K., Schulten, K. (1995). Models of Orientation and Ocular Dominance Columns in the Visual Cortex: A Critical Comparison. *Neural Computation*, **7**, pp.425-468.

[2] Gerecke, U., Sharkey, N. (1999). Quick and Dirty Localization for a Lost Robot. In *Proceedings of IEEE International Conference on Computational Intelligence for Robotics and Automation (CIRA-99)*, Monterey, CA.

[3] Kohonen,T. (1982). Self-Organized Formation of Topologically Correct Feature Map. *Biological Cybernetics* **43**, pp.59-69.

[4] Kohonen,T. (1994). *Self-Organizing Maps*, Springer-Verlag, Berlin.

[5] Martinetz, T., Berkovich, S., Schulten, K. (1993). Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Trans. on Neural Networks*, **4**, pp.558-559.

[6] Shouno, H., Kurata, K. (2001). Formation of a Direction Map using Kohonen's Self-Organization Map by Projection Learning. *Biological Cybernetics*, **85**, pp.241-246.

[7] Mizutani K., Omori T.(1999). On-line Map Formation and Path Planning for Mobile Robot by Associative Memory with Controllable Attention, In *Proc. of IJCNN'99*

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Backward Control of Multitrailer Systems Using Neurocontrollers Evolved by Genetic Algorithm

Ayaki Kiyuna
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
kiyuna@mibai.tec.u-ryukyu.ac.jp

Hiroshi Kinjo
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
kinjo@tec.u-ryukyu.ac.jp

Kunihiko Nakazono
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
nakazono@tec.u-ryukyu.ac.jp

Tetsuhiko Yamamoto
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa. 903–0213
yamamoto@tec.u-ryukyu.ac.jp

Abstract

In this paper, we propose a design method for neurocontrollers (NCs) evolved by a genetic algorithm (GA) for the backward movement control of multitrailer truck systems. The difficulty of the backward movement control depends on the number of connected trailers. In order to search for the best NCs for the multitrailer systems, we propose a step-up training method. The step-up training sequence is as follows. First, the initial NCs, that are set to random values, are trained for an easy control object. Second, the set of NCs is trained for more difficult control objects. In this study, firstly, the initial NCs are trained for a two-trailer-connected truck system, secondly the NCs are trained for a three-trailer system, and finally the NCs are trained for a four-trailer system. The step-up training method is able to advance NCs which can successfully control multitrailer systems. The simulation results show that the step-up training method is useful for multitrailer systems.

1 Introduction

Backward movement control of a trailer-truck system is known to be a typical nonlinear control problem. The difficulty of control system design not only causes the dynamics to be nonlinear but we also have to consider inherent physical limitations of the system such as the "jackknife" phenomenon.

Many control design methods for multitrailer systems have been reported [1], [2]. With regard to neurocontrol, first, Nguyen and Widrow [3] have successfully solved the backward control problem using the back-propagation (BP) algorithm. Second, Jenkins and Yuhas [4] report a small neurocontroller (NC). However, when utilizing the BP method it is necessary to compute the partial derivative of the control object which is usually a very complex and mathematically difficult process and sometimes, the partial derivative cannot be obtained.

Recently, a method of evolutionary computation [5] has been fully researched and applied to many industrial problems [6]-[8]. If evolutionary computation such as a genetic algorithm (GA) is applied to the NC training, then the design of the control system becomes simplified. In a previous study, Kinjo *et al.* [9] proposed a control method using NCs evolved by a GA to solve the backward movement control problem. But in ref. [9], the truck is connected to only a single trailer. It is clear that the control problem becomes more difficult when the number of connected trailers increases.

The difficulty of the problem depends on the number of control variables. When the truck system has three or more trailers, then the method of NCs with GA evolution is often not able to produce a better controller, or the GA process takes too much time to obtain NCs that can control the trailer-truck system successfully.

In this paper, we present a step-up training method for NCs evolved by a GA that is able to successfully control three or more trailer connected truck systems.

2 Model of Trailer-Truck System

Figure 1 shows the model of a truck system with four trailers. We designed a controller for the multitrailer system.

The control purpose is to back up the trailer-truck

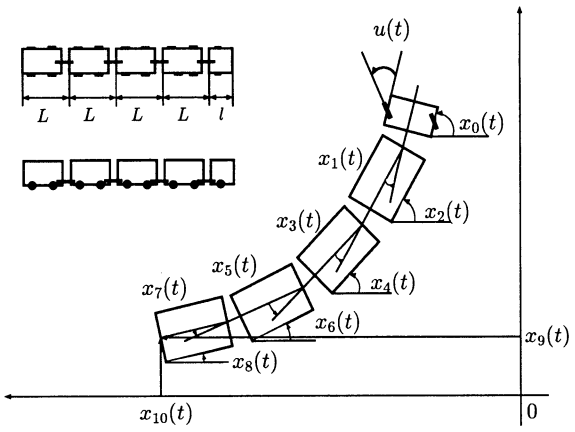


Fig. 1 Model of a truck system with four trailers

system along a straight line ($x_9(t) = 0$) without forward movement. That is, $x_1(t) \to 0$, $x_3(t) \to 0$, $x_5(t) \to 0$, $x_7(t) \to 0$, $x_8(t) \to 0$, $x_9(t) \to 0$.

3 Control System

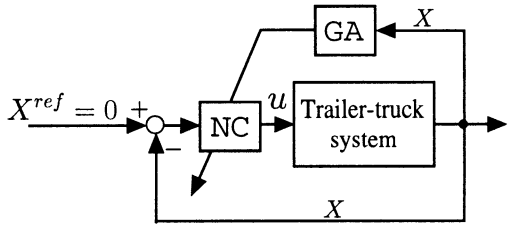


Fig. 2 Control system using NC with GA evolution

Figure 2 shows the control system using NC with GA evolution. In the figure, NC denotes a neurocontroller which receives the error of angles x_1, x_3, x_5, x_7, x_8 and position x_9 as inputs and outputs steering angle u. The trailer-truck system receives the steering angle u and outputs the state variables of the next step while referring to the present configuration. GA denotes the genetic algorithm procedure.

By applying the GA to NC evolution, we obtain the "best individual" from among the evolved NCs.

4 Genetic Algorithm

4.1 Evolution of NCs

Figure 3 shows the flow chart of the evolution procedure for the NCs. The procedure for adapting NCs is as follows. First we produce some NCs, whose connecting weights are chosen initially at random. Each NC has a genetic code which is transformed from the connecting weights. The transform equation

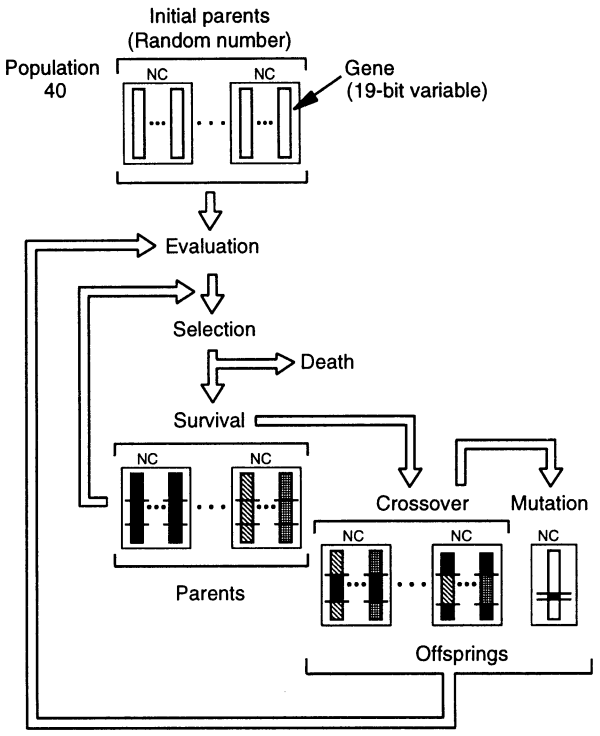


Fig. 3 Flow chart of NC evolution with GA

$g = [(\frac{w}{N_r} + 1)\frac{524287}{2}]$ is used, where g and w are the genetic code and connecting weight, respectively; the symbol $[\cdot]$ denotes a Gaussian function which transforms decimals to integers; N_r is a coefficient that indicates the range of w.

In the evaluation process, the control performances of NCs are gauged. The NC evaluation is performed as follows. The trailer-truck system is set to an initial configuration. The truck backs up using the NC, undergoing many individual cycles of backing up, until it stops. The final error of the trailer-truck system is recorded. Next, we place the trailer-truck system in another initial configuration and allow it to back up until it stops. Table 1 shows the initial configurations.

When control trials starting from eight configurations are completed, the control performance of the NC is evaluated. All of the NCs in the population are evaluated by the same methods.

After the evaluation process, some pairs of NCs are selected and produce new NCs by two-point crossover and mutation operations.

After the NC production process, the new NCs go to the evaluation process, and the GA processes are repeated until the best individual, with good control performance, is obtained.

Table 1 Initial configurations for NC evolution

Pattern No.	x_0, x_2, x_4, x_6, x_8 [rad]	x_9 [m]
1	0	0.0
2	$\pi/6$	
3	$\pi/3$	
4	$\pi/2$	
5	0	0.4
6	$\pi/6$	
7	$\pi/3$	
8	$\pi/2$	

4.2 Evaluation Function

During the GA-based training process of NCs, we use an error function E to evaluate the control performance of each NC. When the number of trailers is four, E is described by the following equation.

$$E = \sum_{p=1}^{P} \{ q_1(x_1^{\mathrm{ref}} - x_{1p}^{\mathrm{end}})^2 + q_3(x_3^{\mathrm{ref}} - x_{3p}^{\mathrm{end}})^2 + q_5(x_5^{\mathrm{ref}} - x_{5p}^{\mathrm{end}})^2 + q_7(x_7^{\mathrm{ref}} - x_{7p}^{\mathrm{end}})^2 + q_8(x_8^{\mathrm{ref}} - x_{8p}^{\mathrm{end}})^2 + q_9(x_9^{\mathrm{ref}} - x_{9p}^{\mathrm{end}})^2 \}, \quad (1)$$

where x_p^{end} is the final value of the state vector which starts from any initial configuration of p; x^{ref} is the reference vector; and q is the weight factor which adjusts the importance of control variables. P denotes the number of initial configurations, in this case $P = 8$.

The above GA method is very effective when the control object is an easy system, i.e., a single- or two-trailer-connected truck system. However, if the number of connecting trailers increases, the GA cannot perform well. Therefore, we propose a step-up training method in the next subsection.

4.3 Step-up training method

When the number of connecting trailers increases, the evolution of NCs becomes difficult and the GA process cannot produce better individuals.

In order to produce better NCs that can control the multitrailer system successfully, we propose a step-up training method. The method is described in the following sequence. As a first step, the connecting weights are set at random and the initial NCs that are trained for an easy object, e.g. two-trailer-connected truck system, using the GA. Next, as the second step, the NCs are trained by the GA for the three-trailer-connected truck system. After the second step of training, the set of NCs is trained by the GA for the four-trailer-connected truck system that is the final training step. Figure 4 shows the flowchart of the step-up training method.

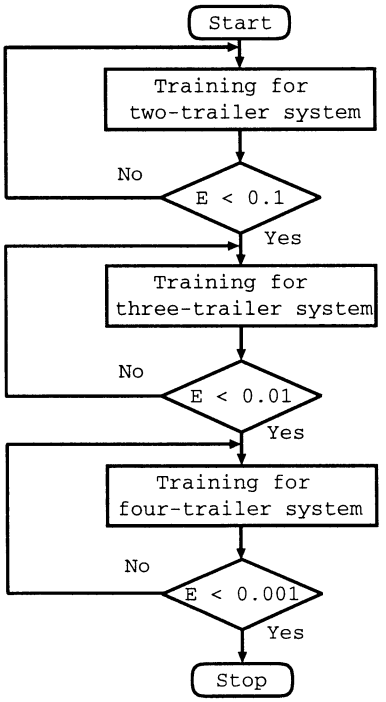


Fig. 4 Flow chart of step-up training

5 Simulation Results

5.1 Evolution Results

In the simulation, we set the parameters of the lengths of the truck and trailer as $l = 0.129$ [mm] and $L = 0.124$ [mm], respectively; the backward speed is $v = -0.2$ [m/s]; sampling time is $\Delta t = 0.5$ [s]. The weights of E are set to $q_1 = q_3 = q_5 = q_7 = q_8 = q_9 = 1.0$. The number of NCs treated in the GA is set to 30 % and the probability of crossover is 0.6. We set $N_r = 50$, so that the range of connecting weight w is $[-50, 50]$. Figure 5 shows the evolution results for the four-trailer-connected truck system in the final training step. It is observed that the evolution of NCs is successful. In this case, initial individuals are sets of NCs that can successfully control the two- and three-trailer-connected systems through the previous first and second training steps. If the initial individuals of NC were set to only random values without the first and second steps, the GA could not search for the best NCs for the four-trailer system.

5.2 Control Results

Figure 6 shows the control trajectory of the truck system with four trailers using the trained NC. In this

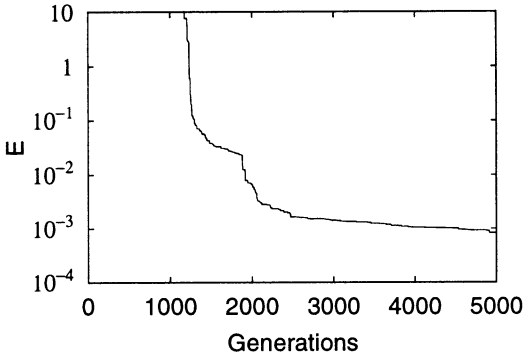


Fig. 5 Evolution results for four-trailer system

case, the initial angles are set to training pattern 4 in Table 1. It can be seen that the NC is able to control the trailer-truck system successfully. Figure 7 shows the steering angle u. It can be seen that the angle u ranges from -1 [rad] to 1 [rad].

6 Conclusion

In this paper, we propose a method of backward movement control for truck systems connected with multitrailers using neurocontrollers evolved by a GA. When the truck is connected to three or more trailers, the design method of neurocontrollers using GA does not perform well. In order to search for the best neurocontroller, for the multitrailer systems, we proposed a step-up training method that firstly trains the NCs for an easy system, and after the training, the set of NCs is trained for more difficult systems. The simulation results show that the step-up training is effective for the training of the controllers for multitrailer truck systems.

Acknowledgments

This paper was supported by an endowment of NTT Data Creation Corporation.

References

[1] K. Tanaka, T. Kosaki and H. O. Wang: "Backing Control Problem of a Mobile Robot with Multiple Trailers: Fuzzy Modeling and LMI-Based Design," IEEE Trans. on Systems, Man, and Cybernetics, Part C, Vol. 28, No. 3, pp. 329–337 (1998)

[2] K. Tanaka and T. Kosaki: "Fuzzy Backward Movement Control of a Mobile Robot with Two Trailers," Trans. of SICE in Japan, Vol. 33, No. 6, pp. 541–546 (1997) (in Japanese)

[3] D. Nguyen and B. Widrow: The Truck Backer-Upper: "An Example of Self-learning in Neural Networks," Proc. of Int. Joint Conf. on Neural Networks (IJCNN-89), Vol. 2, pp. 357–363 (1989)

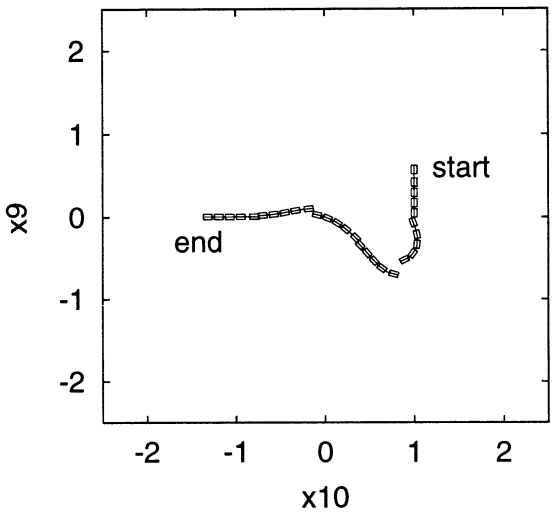


Fig. 6 Trajectory of trailer-truck system

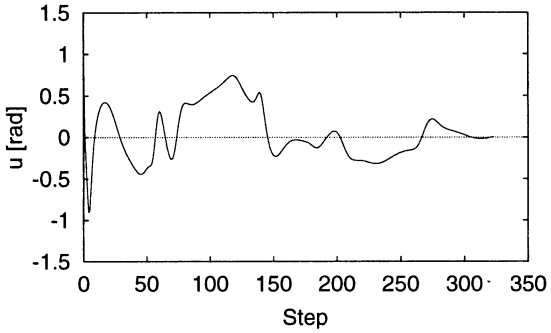


Fig. 7 Steering angle u

[4] R. E. Jenkins and B. P. Yuhas: "A Simplified Neural Network Solution Through Problem Decomposition: The Case of the Truck Backer-Upper," IEEE Trans. on Neural Networks, Vol. 4, No. 4, pp. 718-720 (1993)

[5] D. B. Fogel: Evolutionary Computation, IEEE Press (1995)

[6] J. H. Holland: "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence," MIT Press (1992)

[7] D. E. Goldberg: "Genetic Algorithms in Search, Optimization & Machine Learning," Addison-Wesley (1989)

[8] L. Davis: "Hand Book of Genetic Algorithms," Van Nostrand Reinhold (1991)

[9] H. Kinjo, B. Wang, K. Nakazono and T. Yamamoto: "Backward Movement Control of a Trailer-Truck System Using Neruocontrollers Evolved by Genetic Algorithm," Trans. of IEE Japan, Vol. 121-C, No. 3, pp. 631-641 (2001) (in Japanese)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Reaching Control of the Humanoid Robot by using Linear Visual Servoing

Kazuya Okamoto, Kengo Yamaguchi, Noriaki Maru

Wakayama University
Faculty of Systems Engineering
Sakaedani 930 Wakayama, JAPAN

Abstract

This paper presents a method for the arm reaching control of the Humanoid Robot by using linear visual servoing (LVS). Reaching control is realized by the approaching motion of the body to the target object within a certain distance and reaching motion of its hand. LVS is based on linear approximation of the transformation between binocular visual space and joint space. It is very robust to calibration error, especially to camera turning, because it uses neither camera angles nor joint angles to calculate feedback command. Some experimental results are presented to show the effectiveness of the proposed method.

1. Introduction

Vision is indispensable for the Humanoid Robots that coexists with human being and perform specified tasks in a changing environment. Various kinds of mechanism of visual feedback have been proposed and they are called visual servoing [1]. Especially, visual servoing is proposed as a robust control method for calibration error. But conventional visual servoing schemes are not so robust to calibration error, because they use both image Jacobian matrix which includes camera angles and robot Jacobian matrix which includes joint angles to calculate feedback command.

We proposed a simple visual servoing scheme called linear visual servoing for the Humanoid robots [2][3]. LVS is based on linear approximation between the binocular visual space and joint space. The method is very robust to calibration error, especially to camera turning, because it uses neither camera angles nor joint angles to calculate feedback command. Hence, it is especially suitable for the Humanoid Robots which use active stereo vision. That is, it is possible to turn cameras to facilitate visual processing, even if the manipulator is under control by visual servoing. Furthermore, the amount of the calculation is very small compared to the conventional visual servoing schemes, because it only needs both the time-invariant constant matrix and the image coordinates of the feature points.

In this paper, we propose a method for reaching control of the Humanoid Robot by using LVS. Reaching control is realized by the approaching motion of the body to the target object within a certain distance and reaching motion of its hand. LVS is based on linear approximation of the transformation between binocular visual space and joint space. Approaching motion by using LVS is based on linear approximation of the transformation between binocular visual space and motion space of the Humanoid Robot. Motion space is defined by translational velocity and rotational velocity of the robot coordinate system which is attached at the center of gravity of the Humanoid Robot. Reaching motion by using LVS is based on linear approximation of the transformation between binocular visual space and joint space of the arm of the Humanoid Robot. We will show some experimental results which demonstrate the effectiveness of the proposed method.

2. Linear Approximation of the Inverse Kinematics

2.1 Model of the Humanoid Robot

Fig.1 shows the Humanoid Robot. The Humanoid Robot consists of two independently driven wheels and two casters which are attached forward and backward and arm with two links and three joints. The elbow joint has 1DOF, and the shoulder joint has 2DOF. The two cameras pan and tilt independently. We show the parameters of the Humanoid Robot in **Table.1**.

The binocular visual space is defined as the vergence

angle γ and the viewing direction θ , δ (see **Fig.2**). The binocular visual coordinates $\mathbf{V} = (\gamma, \theta, \delta)^T$ of a feature point projected on the camera image planes are described as

$$\mathbf{V}= \begin{pmatrix} \alpha_L - \alpha_R \\ (\alpha_L + \alpha_R)/2 \\ \alpha_D \end{pmatrix} + \begin{pmatrix} (X^L - X^R)/f \\ (X^L + X^R)/2f \\ (Y^L + Y^R)/2f \end{pmatrix} \quad ,(1)$$

where α_L, α_R and α_D are the camera angles and (X^L, Y^L) and (X^R, Y^R) are the coordinates of a feature point in the left and right image respectively.

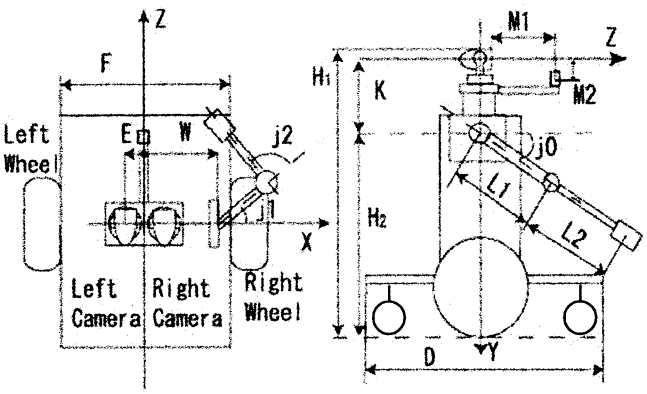


Fig.1 Model of Humanoid Robot

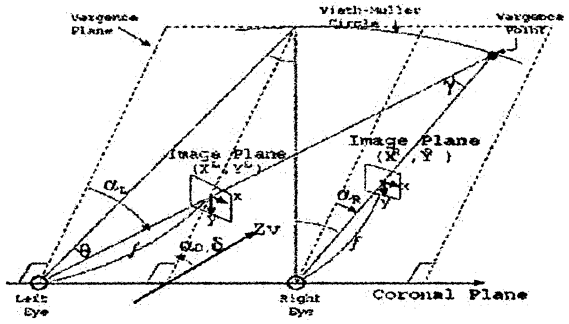


Fig.2 Binocular visual space

Table.1 Parameters of humanoid robot

Parameter	Size
H_1	1000mm
H_2	650mm
L_1	300mm
L_2	300mm
D	830mm
F	430mm
W	200mm
E	60mm
K	300mm
M_1	300mm
M_2	90mm

2.2 Linear Approximation of the Inverse Kinematics of arm

Fig.3, Fig.4 shows the joint space of the humanoid robot projected onto Cartesian space. We linearize these transformation using the least squares approximation within a region defined as $-20[deg] \leqq j_0 \leqq 20[deg]$, $20[deg] \leqq j_1 \leqq 60[deg]$, $60[deg] \leqq j_2 \leqq 100[deg]$, $1000[mm] \leqq r \leqq 3000[mm]$, $-30[deg] \leqq \phi \leqq 30[deg]$.

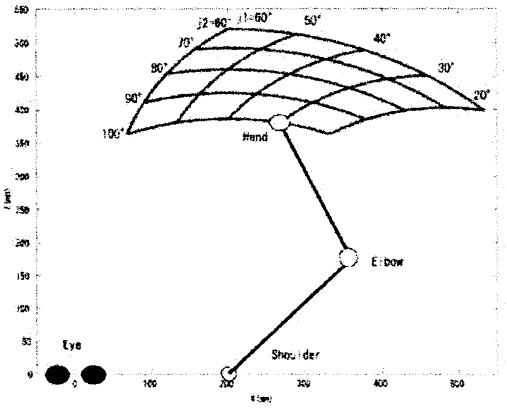


Fig.3 Joint space projected onto Cartesian space (j_1, j_2)

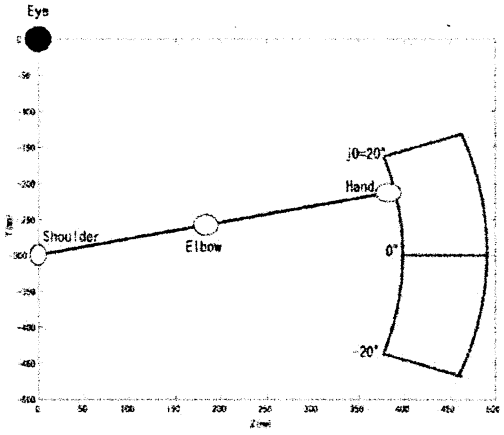


Fig.4 Joint space projected onto Cartesian space (j_0)

2.3 Linear Approximation of the Inverse Kinematics of Motion

Fig.5 show the motion space of the humanoid robot projected onto Cartesian space. We linearize these transformation using the least squares approximation within a region defined as $1000[mm] \leqq r \leqq 3000[mm]$, $-30[deg] \leqq \phi \leqq 30[deg]$.

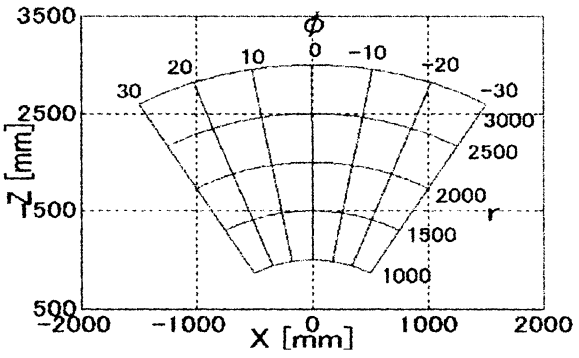


Fig.5 Motion space projected onto Cartesian space (r, ϕ)

Then the transformation from the binocular visual space are to joint space and motion space given by

$$\mathbf{j_v} = \mathbf{R_v V} + \mathbf{C_v} \quad , \tag{2}$$

$$\mathbf{j_m} = \mathbf{R_m V} + \mathbf{C_m} \quad , \tag{3}$$

where $\mathbf{V} = (\gamma, \theta, \delta)^T$ and $\mathbf{j_v} = (j_0, j_1, j_2)^T$, $\mathbf{j_m} = (r, \phi, 0)^T$ $\mathbf{R_v}$, $\mathbf{R_m}$: matrix (3×3), $\mathbf{C_v}$, $\mathbf{C_m}$: vector(3×1), j_0, j_1, j_2: joint angle, r: translational velocity, ϕ:rotational velocity.

The least-squares approximation using binocular visual space result in

$$\mathbf{R_v} = \begin{pmatrix} -3.505 & -0.674 & -1.023 \\ -3.622 & -1.854 & 0.752 \\ 10.943 & 1.471 & -1.213 \end{pmatrix} , \tag{4}$$

$$\mathbf{C_v} = \begin{pmatrix} 13.530 \\ 153.656 \\ -113.012 \end{pmatrix} , \tag{5}$$

$$\mathbf{R_m} = \begin{pmatrix} 489.204 & 0.000 & 0.000 \\ 0.000 & 1.130 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{pmatrix} , \tag{6}$$

$$\mathbf{C_m} = \begin{pmatrix} 301.75 \\ 0.000 \\ 0.000 \end{pmatrix} . \tag{7}$$

2.4 Linear Visual Servoing of arm

Linear visual servoing for arm control is given by

$$\mathbf{u_v} = -\lambda_1 \mathbf{R_v} (\mathbf{V} - \mathbf{V_d}) ,$$

$$= -\lambda_1 \mathbf{R_v} \begin{pmatrix} \{(X^L - X^R) - (X^L_d - X^R_d)\}/f \\ \{(X^L + X^R) - (X^L_d + X^R_d)\}/2f \\ \{(Y^L + Y^R) - (Y^L_d + Y^R_d)\}/2f \end{pmatrix} , \tag{8}$$

where $\mathbf{u_v}$ are control signals to joint velocity controllers, $\mathbf{V}$ is the binocular visual coordinates of the end tip of the Humanoid Robot, $\mathbf{V_d}$ is the binocular visual coordinates of a target and λ_1 is scalar gain, $\mathbf{R_v}$ is the linear approximation matrix of the inverse kinematics obtained in the previous section.

2.5 Linear Visual Servoing of motion

Linear visual servoing for motion control is given by

$$\mathbf{u_m} = -\lambda_2 \mathbf{R_m} (\mathbf{V} - \mathbf{V_d}) ,$$

$$= -\lambda_2 \mathbf{R_m} \begin{pmatrix} \{(X^L - X^R) - (X^L_d - X^R_d)\}/f \\ \{(X^L + X^R) - (X^L_d + X^R_d)\}/2f \\ \{(Y^L + Y^R) - (Y^L_d + Y^R_d)\}/2f \end{pmatrix} , \tag{9}$$

where $\mathbf{u_m}$ is control signals to mobile velocity controllers, $\mathbf{V}$ is the binocular visual coordinates of the end tip of the humanoid robot, $\mathbf{V_d}$ is the binocular visual coordinates of a target and λ_2 is scalar gain, $\mathbf{R_m}$ is the linear approximation matrix of the inverse kinematics obtained in the previous section.

2.6 The change of arm control and move control

The change of arm control and move control use a relation with a work space. An arm control space is defined as 300[mm] ≦ Z ≦500[mm], -10[deg] ≦ ϕ ≦40[deg], and the other space is considered as move control. (**Fig.6**)

3 Experiment

3.1 System Construction

Fig.7 shows the experimental system. We used two EVI-D100 (SONY Co. Ltd) as a stereo cameras which pan and tilt independently and IP5000 (HITACHI Co. Ltd) as an image processing and DOS/V computer. We attached a yellow marker at the end tip of the arm and use a red object to simplify image processing. The stereo

images are converted in an image using a picture separator (Video Device Co. Ltd.) and input to IP5000. The stereo images are binarized by each color and the center of gravity is calculated to obtain the image coordinates of both the end of the Humanoid Robot and target. Then the feedback commands are calculated and sent to the motor driver. We show the specification of the Humanoid Robot in **Table.2**.

Table.2 Specification of Humanoid Robot

Dimension (W×D×H)	300(mm) × 800(mm) × 1000(mm)
Move speed	About 10(km/h)
Actuator	Dc servo actuator with harmonic drive ×5
	DC motor×2
Sensor	CCD camera×2
	Acceleration sensor×3
	Gyroscope sensor×3
Degree of freedom	Neck part 2 DOF
	Arm part 3DOF
	Wheel part 2DOF

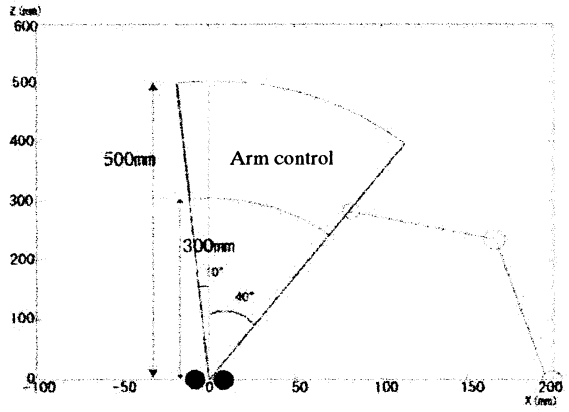


Fig.6 Work Space

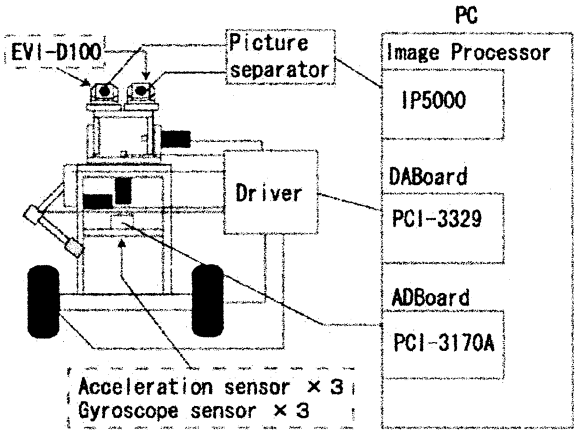


Fig.7 Experimental system

3.2 Experimental Results

Fig.8 show error pixel of center position of image at the arm reaching control of the Humanoid Robot by using LVS. This figure indicates that LVS using binocular visual space is also effective in arm reaching control is possible, moving.

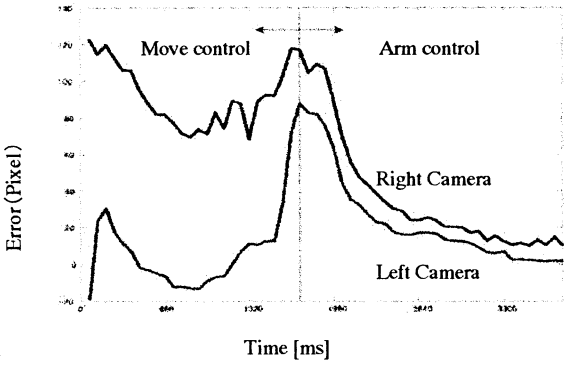


Fig.8 Error pixel

4 Conclusion

This paper presents a method for the arm reaching control of the Humanoid Robot by using LVS. It is based on linear approximation of the transformation between binocular visual space and joint space. Approaching motion by using LVS is based on linear approximation of the transformation between binocular visual space and motion space of the Humanoid Robot. Reaching motion by using LVS is based on linear approximation of the transformation between binocular visual space and joint space of the arm of the Humanoid Robot. We have shown some experimental results which demonstrate the effectiveness of the proposed method.

References

[1] L.E.Weiss, A.C.Sanderson and C.P.Neuman, "Dynamic sensor – based control of robots with visual feedback", IEEE J. Robotics Automation, vol.RA – 3,No.5, pp404-417, 1987

[2] T.Mituda, N.Maru, K.Fujikawa, F.Miyazaki" Visual Servoing based on Linear Approximation of the Inverser Kinematics" Journal of the Robotics Society of Japan vol.14 No.5 pp734 – 750 1996. (in Japanese)

[3] T.Mituda, N.Maru, K.Fujikawa, F.Miyazaki" Linear Approximation of the Inverser Kinematics by using a Binocular Visual Space" Journal of the Robotics Society of Japan vol.14 No.8 pp1145 – 1151 1996. (in Japanese)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Real-Time Visual Servoing for Laparoscopic Surgery

Min-Seok Kim
Dept. of Mechanical Engineering
Korea Advanced Institute of Science and Technology,
Daejon, Korea 305-701
kithnkin@kaist.ac.kr

Jin-Seok Heo
Dept. of Mechanical Engineering
Korea Advanced Institute of Science and Technology,
Daejon, Korea 305-701
dandyheo@kaist.ac.kr

Jung-Ju Lee
Dept. of Mechanical Engineering
Korea Advanced Institute of Science and Technology,
Daejon, Korea 305-701
jjlee@mail.kaist.ac.kr

Abstract

In this paper, a real-time visual servoing unit for laparoscopic surgery is presented using two-staged CONDENSATION (Conditional Density Propagation) Algorithm and damping system. This unit can control the laparoscope manipulator automatically tracking the image of laparoscopic surgical tool. We modified CONDENSATION [1], [2] algorithm to get the accurate position of the surgical tool tip from a surgical image sequence in real-time and this algorithm can be used on the condition that the intensity of illumination is changed abruptly. And we adopted the imaginary damping system to control laparoscope manipulator safely and stably. This visual servoing unit operates and controls the manipulator in order to locate the surgical tool in the center of the image at real-time. The experimental results show that the proposed visual tracking algorithm is highly robust and the controlled manipulator can present stable and safe view.

1. Introduction

Laparoscopic surgery is minimally invasive surgery (MIS), a new kind of surgery that gets more and more common nowadays. In this method, a surgical operation is performed by the help of laparoscope and several long, thin, rigid instruments through small incisions. In recent years, laparoscope manipulator, for example, AESOP[1] that substitutes a camera assistant in a surgical operation is used more widely in laparoscopic surgeries. But it has a bothersome task for surgeon to control the laparoscope manipulator manually or with his voice.

There have been some attempts to control laparoscope automatically because the purpose of laparoscope manipulator is aiming at the target of operation. To control laparoscope automatically, controller must have information about the position of tool tip in surgical operation. Several methods of obtaining position information of tool tip in surgery have been suggested. Among those methods, the visual tracking is an efficient one, because it needs not additional sensor system but the information of surgical image sequence.

Casals, A [3] and Cheolwhan Lee [4] use shape information of surgical tool as a feature of tracking target. This method detects edge-information and then verifies the result with pre-defined tool shape. This method is a reliable one but it may not work well in the case that the some part of tool is covered by obstacles. It is a time consuming and computing resources consuming method, and it is hard to meet real-time tracking.

Guo-Qing Wei [5] and Omote, K [6] use color feature of artificial mark which is attached to surgical tool. This method has simpler structures than those of the previous method, so it costs less computational resources. However it requires additional task, attaching artificial mark. It has many problems such as choice of attaching method, and sterilization of mark. In addition, surgeons' concern is not on the mark but on the surgical tool tip, so tracking failure can be happened when the mark is covered by obstacles or it is out of image.

In this paper, we present a new visual tracking algorithm which has the advantages of two methods, reliability and real-time property above mentioned. This method is based on the CONDENSATION algorithm which detects the color information of image. We modified the CONDENSATION to adapt the changes of intensity of illumination and then apply the modified CONDENSATION tracker to two stages. In the first stage, this new tracker finds out the position and shape information of the tool. In the second stage, it can obtain exact position of the tool tip

Once the tool tip position is obtained by the tracking part, the controlling part of the visual servoing unit controls the laparoscope to be located in the center of surgical image. Therefore, the damping system is presented for stable and safe motion of the laparoscope image.

2. Visual Tracking Algorithm

2.1 CONDENSATION algorithm

The CONDENSATION is sampling based algorithm. It uses the stochastic propagation of conditional density, probability distribution

[1] Automated Endoscope System for Optimal Positioning, Computer motion, Inc. 1993

function (PDF) is constructed by weights of sampling positions in image. We take the sampling positions of surgical tool for the state x and take the color feature of surgical tool for the observation z of the CONDENSATION for the purpose of simpler structure and faster speed in computation. The algorithm composed of select, predict and measure phases.

In the select phase, it selects the sampling positions through given prior density

$$p(\mathrm{x}_{t-1} \mid Z_{t-1}) \tag{1}$$

, among a set of N random samples $S_{t-1}^{(n)} = \{s_{t-1}^{(1)}, \ldots, s_{t-1}^{(n)}\}$ at time t-1. The selected sampling positions S_{t-1} have high probability that they locate on the surgical tool in image.

In the predict phase, the tracker predicts the sampling positions of surgical tool in an image at time t using dynamic characteristics.

$$p(\mathrm{x}_t \mid \mathrm{x}_{t-1} = S_{t-1}^{\prime(n)}) \tag{2}$$

In the measure phase, the tracker calculates the weights by using the difference between the given color feature of surgical tool and the predicted sampling positions' color values in image.

$$\pi_t^{(n)} = p(z_t \mid \mathrm{x}_t = S_t^{\prime(n)}) \tag{3}$$

By using the weights at time t, dynamic characteristics of the state and prior density at time t-1, we can make posterior density at time t through

$$p(\mathrm{x}_t \mid Z_t) = k_t p(z_t \mid \mathrm{x}_t) p(\mathrm{x}_t \mid Z_{t-1})$$
$$p(\mathrm{x}_t \mid Z_{t-1}) = \int_{\mathrm{x}_{t-1}} p(\mathrm{x}_t \mid \mathrm{x}_{t-1}) p(\mathrm{x}_{t-1} \mid Z_{t-1}) \tag{4}$$

, where $Z_t = \{z_t^{(1)}, \ldots, z_t^{(n)}\}$ and k_t is a normalization constant.

After the measure phase the tracker goes back the select phase and repeat recursively. Using color feature of surgical tool as the measure of CONDENSATION tracker, we can make the tracker simple and fast. But the tracker can fail in tracking in the case that the color of surgical tool is altered by illumination change of surgical environment or that the surgical tool is covered by obstacles such as the tissue of internal organs or supplementary surgical tool. To overcome those defects of the CONDENSATION tracker, we modified the CONDENSATION algorithm named 'two-stage adaptive CONDENSATION.'

2.2 Adaptive Color Model

As safety is most important matter in surgery, surgical tool tracker must be strong and adaptive to change of target feature value on account of possible environment changes. Because there is no other light source except laparoscope's inherent light in laparoscopic surgery, we consider possible illumination changes during surgical operation.

Many types of laparoscopic surgical tools is used in laparoscope surgery. But most of those are composed of two-part, guide rod and movable appendages and they have different color. Therefore we choose different color components as color features of each part of the tool. To be strong and also to be adaptive to illumination change, we choose hue and value components among HSV (hue, saturation and value) color model as the color features of the tool's rod part. Figure 1 shows the histogram of hue and value components of the tool's rod part in normal light. For the movable appendages part, we choose hue and r component because it has metallic luster. The value component of that color can be changed too much by different illumination, but r component can be changed little and it help the tracker to easily discriminate between that part and image backgrounds.

$$H_1 = \cos^{-1}\left(\frac{\frac{1}{2}[(R-G)+(R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$
$$Hue = \begin{cases} H_1 & (B \le G) \\ 360 & (B > G) \end{cases} \tag{5}$$
$$Value = \frac{Max(R,G,B)}{255}$$
$$r = \frac{R}{R+G+B}$$

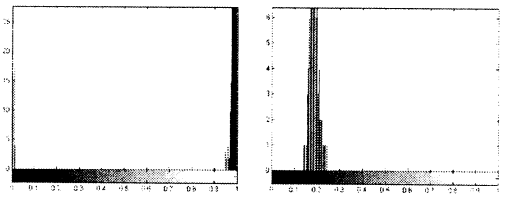


Figure 1 Histograms of surgical tool (H, V)

We examine each hue and value components of the tool's rod part in normal light and in minimum light for laparoscopic surgery. And then we calculate median and standard deviation (Std.). Same procedure is performed for the tool's movable appendages part. The median values represent the color feature of the surgical tool and the standard deviation values are utilized in the weight calculation in adaptive color model. Those values are shown in Table 1. We modified the concept of the adaptive color model [7] with the examined color feature values. The concept is

realized by applying the CONDENSATION tracker to the each color feature of surgical tool to adapt the change of color features. Once the tracking is success, the tracker goes to the select phase.

Surgical tool \ Environment			Norma l light	Dark light
Rod part	Hue	Med.	1.0	0.1592
		std.	0.0137	0.0137
	Value	Med.	0.4196	0.1843
		Std.	0.1589	0.0170
Movable appendages	Hue	Med.	1.0	0.1250
		std.	0.0137	0.0225
	r	Med.	0.8627	0.3686
		Std.	0.0226	0.0213

Table 1 Color features of surgical tool

In the select phase, we select the highly weighted sampling positions $S_{t-1}^{(n)}$ and examine the color features of those positions'. Then we can calculate the median and standard deviations of those values and set the median value as a new color feature.

And then, in the measure phase, we can calculate the weights, difference between the median values and the predicted sampling positions' color feature values as

$$\pi_t^{(n)} = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{(S_t'^{(n)} - m)^2}{2\sigma^2}\} \qquad (6)$$

where m is median value of $S_{t-1}^{(n)}$ and σ is the standard deviation calculated in previous phase.

2.3 Two-Stage CONDENSATION

We apply the adaptive CONDENSATION tracker by two stages, one for the tool's rod and the other for the tool's movable appendages part. The first stage is the preparation for the second one. In this stage, we select the highly weighted sampling positions. Using the weighted least square fitting method, we generate a line equation which represents the direction of the surgical tool. We can also know the thickness of tool that is same to the diameter of the rod by the calculation of the length between that line and the farthest position from the line among the selected positions.

At the second stage, we estimate the tip position of the movable appendages. We can imagine a rectangle on the left of tool's rod part. The width of rectangle is thickness of the tool and the length of rectangle is a third of distance between two points on the center line. One is leftmost point of tool's rod and the other is left intercept point of image. We set the length by the iterative test of this tracker. Therefore we can restrict the sampling region to the rectangle.

Finally among the highly weighted sampling positions of the second tracker, we set leftmost position as the tip position of surgical tool. The whole procedure is depicted in Figure 2.

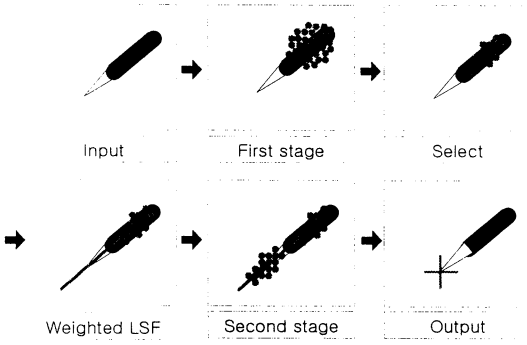


Figure 2 Two-stage CONDENSATION scheme

Using the two-stage adaptive CONDENSATION tracker, we can get the exact position of tool tip. However, we become suffer from higher computational cost than that of the existing CONDENSATION tracker. To compensate the time expense, we adapt the master element concept of finite element method in predict phase of second stage of the tracker. On account of using this concept, we don't need to calculate all the sampling positions but just calculate four edge positions of the imaginary rectangle. Initially, we create sampling positions in the square of master element once for all for whole image sequence as shown in Figure 3, transform the positions to the inside of rectangle in image.

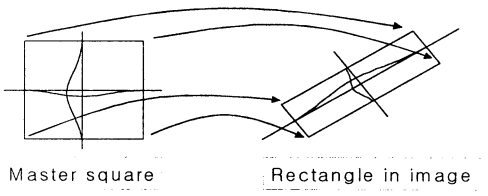


Figure 3 Transformation scheme

3. Manipulator Control Strategy

Laparoscope plays a role of surgeon's eyes in laparoscopic surgery. Surgeon always wants laparoscope manipulator to aim at the right hand's surgical tool tip on condition that he or she is a right-hand person. Therefore our objective is to locate the tool tip on center of image. We set safety zone and couple up imaginary dampers to that zone as shown in Figure 4.

The surgical tool can move freely in the safety zone, but the tool feel the restoring force when it

is out of the zone. Because the damper is a kind of spring, the magnitude of force is determined by the distance between the tool tip position and the margin of safety zone as $F=kx$ where F is force, k is spring constant and x is elongated length. This force changes into controller's input voltage which is driving signal of laparoscope manipulator. Therefore long elongated length means high voltage input of controller and it make the manipulator move fast.

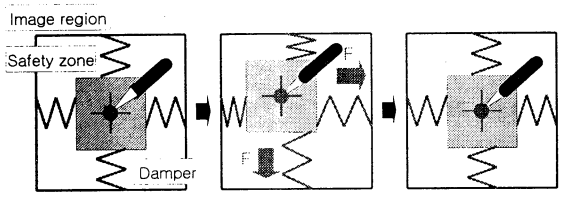


Figure 4 Manipulator control strategy scheme

4. Experimental Results

We have implemented this visual servoing unit on a Pentium IV 1.7GHz PC with 320 by 240 image size and used a Sensoray 626 DA board. We test this unit in both bright and dark lighting conditions and also test in the lighting condition, alternating abruptly between two lighting conditions using the conventional laparoscope manipulator AESOP.

When we use 1000 samples each stage for one frame of image, the tracker's net time of one image grabbing and processing is about 40ms. Therefore processing speed of our system is 25frame/sec. This result satisfies sufficient real-time tracking.

Figure 5 shows an example of servoing result with change of illumination and position. In the second row's leftmost image illumination change abruptly, even though the tracking is successful without any disturbance. And in the center images surgical tool cross the safety zone and then the manipulator move upward. In the next image surgical tool is inside of the safety zone.

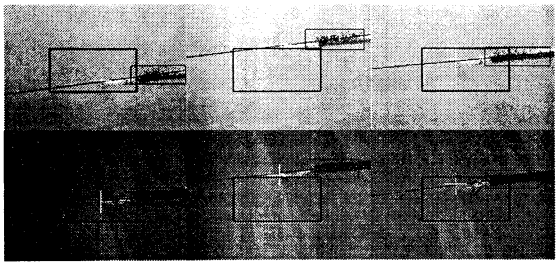


Figure 5 Visual servoing results

5. Summary

In this paper, we present a new real-time visual servoing unit for laparoscopic surgery. The two-stage adaptive CONDENSATION algorithm shows real-time and robust tracking result, because it uses master element concept in computation and it uses color features of surgical tool as the measure of the CONDENSATION algorithm. The manipulator control strategy and its performance are highly evaluated by our co-working surgeon because the visual serving system shows stable view in the safety region.

We expect to test the system on humans in near future following evaluation of results of experiments on animals.

Acknowledgement

We gratefully acknowledge the financial support of the Korea Science and Engineering Foundation (HWRS-ERC).

References

1 Michael Isard, Andrew Blake, "CNDENSATION-Conditional Density Propagation for Visual Tracking", Int. J. Computer Vision, 29(1), 5-28, 1998

2 Michael Isard, Andrew Blake, "Contour tracking by stochastic propagation of conditional density", In Proc. European Conf. Computer Vision, pp.343-356, 1996

3 Casals, A., Amat, J., Laporte, E., "Automatic guidance of an assistant robot in laparoscopic surgery", IEEE International Conf. on Robotics and Automation, Vol. 1, pp. 895–900, 1996

4 Cheolwhan Lee, Yuan-Fang Wang, Uecker, D.R., Yulun Wang, "Image analysis for automated tracking in robot-assisted endoscopic surgery", Proceedings of the 12th IAPR International Conference on , Vol. 1, pp. 88–92, 1994

5 Guo-Qing Wei, Arbter, K., Hirzinger, G., "Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation", IEEE Eng. in Med. and Bio. Magazine , Vol. 16, Issue 1 , pp. 40–45, 1997

6 Omote, K., Feussner, H., Ungeheuer, A., Arbter, K., Guo-Qing Wei, "Self-guided robotic camera control for laparoscopic surgery compared with human camera control", The American journal of surgery, Vol. 177, pp. 321-324, April, 1999

7 Gi-jeong Jang, In-so Kweon, "Robust Object Tracking Using an Adaptive Color Model", IEEE International Conf. on Robotics and Automation, pp1677-1682, 2001

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Compensatory Eye Movement for Translational Motion of Robot Head

Hiroaki Tsuji , Noriaki Maru
Dept. of Systems Engineering,
Wakayama Univ.
930 Sakaedani,Wakayama, Japan, 640-8570

Abstract

We propose the Linear Vestibulo Ocular Reflex (LVOR) which is a compensatory eye movement for translational motion of robot head. We realize LVOR using kinematics and 3 accelerometers. Furthermore, we use Feedback Error Learning (FEL) to compensate for dynamics of accelerometer and motor. FEL uses the output of Opto-kinetic Reflex(OKR) as the error signal. We use acceleration, velocity and a position of the head as the input of FEL. Some experimental results are presented to demonstrate the effectiveness of the proposed method.

1 Introduction

We can recognize the environment while walking on rough terrain. This is because there are compensatory eye movements which compensate movement of a head to stabilize the retinal image in human visual system.

Compensatory eye movements are also indispensable in a robot visual system. Vestibulo Ocular Reflex (VOR) is in the typical one of compensation eye movement. VOR is human vision system to stablize the retinal image during movement of the head. There are 2 types of VOR. One is VOR to the rotation of the head. The other is VOR to the translation of the head. We call it LVOR. Although there are many researches on VOR to rotation [2], there is no research of LVOR. LVOR is necessary to gaze at an object while a robot moves on rough terrain. Although we proposed a method to realize LVOR using kinematics and accelerometer[1],the performance of LVOR was low. It is because the dynamics of motor and sensor considered was not.

Opto Kinetic Reflex (OKR) is the other compensatory eye movement known well. The system is to move eye at the time that optical flow is found on retina. There are various researches in the conventional research. But, it is difficult for OKR to move at high speed because there is a problem of a video rate, when a commercial vision system is used. This problem don't arise at LVOR using high sampling speed sensor. OKR is feed-back system whereas that LVOR is feed-forward system. Therefore, OKR moves perfectly at no-error. But, if the error of a sensor arises, LVOR cannot gaze target completely.

In this paper, we propose a method to improve the performance of LVOR by using FEL. FEL uses the output of OKR as the error signal to compensate for dynamics of motor and sensor. We use acceleration, velocity and a position of the head as the input of FEL. Some experimental results are presented to demonstrate the effectiveness of the proposed method.

2 Linear VOR

2.1 Analysis of LVOR

First, we consider about the movement in X-Z plane. Let us consider that two cameras gazing at $P_0(0, 0, Z_0)$ as show in **Fig.1**. And let the rotational angle of the left and right camera be θ_l and θ_r respectively. We suppose that a head moves l, k in the direction of the X,Z-axis duaring the very small time Δt. That is we suppose that each camera moved to L', R'. To keep gazing at P_0, robot needs to rotate the left camera $\Delta\theta_l$.

In this time, an angular of the left camera is θ_l and the variation of the angular is $\Delta\theta_l$. θ_l, $\Delta\theta_l$ given by $\triangle OLP_0$, $\triangle HL'P_0$ in **Fig.1**.

$$\tan\theta_l = \frac{Z_0}{X_0} \quad (1)$$

$$\tan(\theta_l + \Delta\theta_l) = \frac{Z_0 - k}{X_0 - l} \quad (2)$$

$$\tan\Delta\theta_l = \frac{kX_0 - lZ_0}{(l - X_0)X_0 + (k - Z_0)Z_0} \quad (3)$$

Therefore, the angular velocity of the left camera is

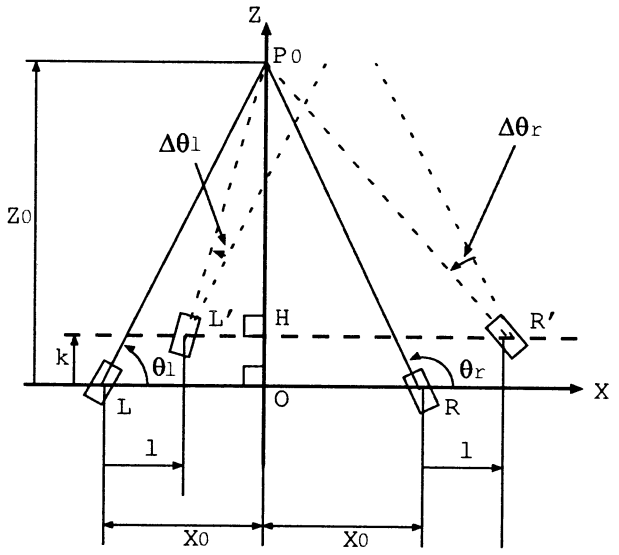


Figure 1: Motion in X-Z plane

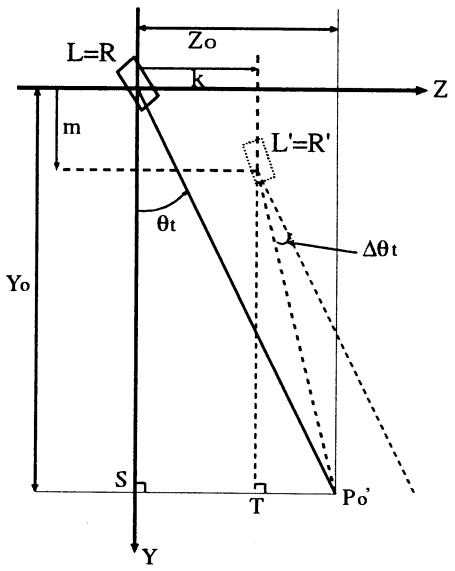


Figure 2: Motion in Y-Z plane

given by

$$\omega_l = \frac{(dk/dt)(l - X_0) - (dl/dt)(k - Z_0)}{(l - X_0)^2 + (k - Z_0)^2}. \quad (4)$$

Similarly, the angular velocity of the right camera is also given by

$$\omega_r = \frac{(dk/dt)(l + X_0) - (dl/dt)(k - Z_0)}{(l + X_0)^2 + (k - Z_0)^2}. \quad (5)$$

Next, we consider about the movement in Y-Z plane. From **Fig.2**, The angular velocity ω_t in Y-Z plane is also given by

$$\omega_t = \frac{(dk/dt)(m - Y_0) - (dm/dt)(k - Z_0)}{(m - Y_0)^2 + (k - Z_0)^2}. \quad (6)$$

X_0 is a constant value of this system. Z_0, Y_0 can be measured from stereo cameras. Consequently, if $l, m, k, dl/dt, dm/dt, dk/dt$ are known, we can obtain $\omega_l, \omega_r, \omega_t$. Then, we use accelerometer to know those values.

3 Compensation of Dynamics

3.1 LVOR-OKR model

FEL predicts that the climbing fiber inputs encode the error signal in the motor-command coordinates, and the cerebellar cortex acquires the inverse dynamics model by changing synaptic weights between parallel-fiber inputs and Purkinje cells. We use FEL controller to compensate for dynamics of sensor and motor. A block diagram is shown in **Fig.3**.

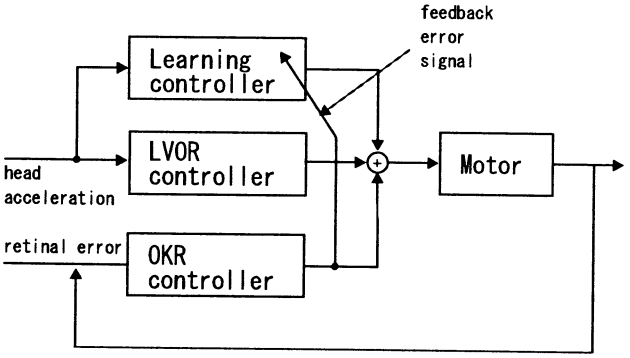


Figure 3: LVOR-OKR model

The LVOR controller in **Fig.3** is the feed-forward controller based on kinematics[1]. The OKR controller is the feedback controller that use optical flow on the image. The output signal of OKR controller is a teaching signal of FEL.

3.2 Feedback Error Learning

The input of a FEL are the acceleration obtained by of the accelerator of three axes attached in the head, the velocity and move distance which are calculated by

time integration of it, and move distance. The output signal of OKR controller is used as a teaching signal. An output is the angular velocity to each camera. The number of middle layer is 70.

4 System

The robot head consists of two sets of camera, and six sets of DC servo motor(Harmonic Drive Systems.Inc, RH-5A, RH-8D), as shown in **Fig.4**. Both cameras have 2 DOF. Neck has 2 DOF. The focal length of The cameras(ELMO, UV-100) is 4.5[mm] . The motion of the cameras are controlled by sending pause via pulse generator(Nippon pulse motor,PCL5022) form host computer(Gateway2000,G6-300). The motion of the neck are controlled via motor drivers(Okazakisangyo,TITech Driver PC-0143-2 and IG-0138-1) form host computer. We use three accelerometers (Japan Aviation Electronics Industry,JA-5VC2). The accelerometer can measure acceleration up to 2G and frequency band width is from DC to 500[Hz]. The acceleration is measured through A/D converter (Contec,AD12H(PC)) whose sampling rate is 1[ms]. The performance of VOR is measured by tracking the fixed object using vision (Fujitsu Tracking Vision). This vision is also used for OKR. The Tracking Vision can track the target object in real time (33[ms]). The image of each camera become one image through image parting machine(TV-350).

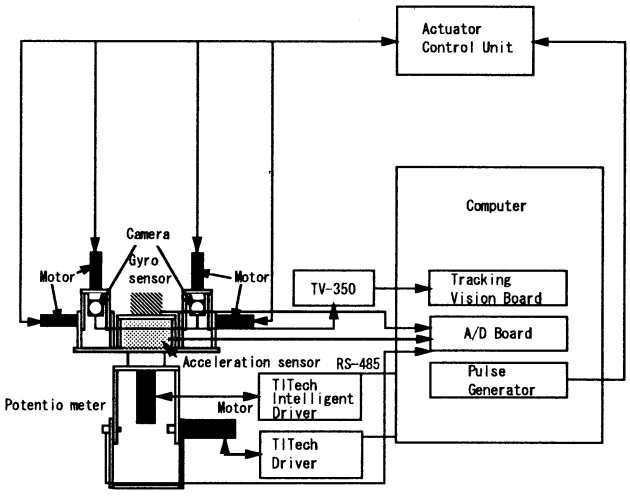


Figure 4: Construction

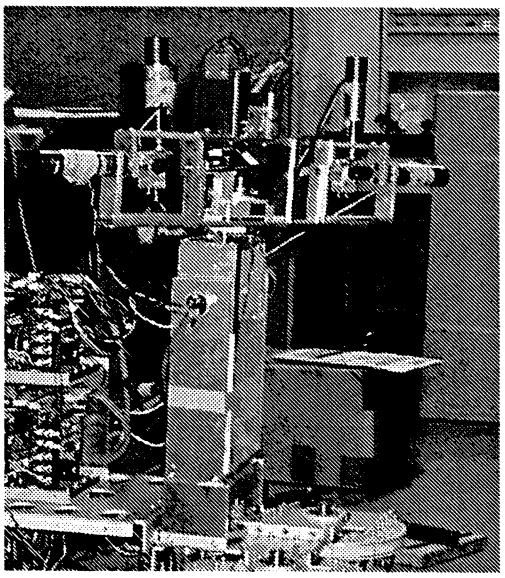


Figure 5: An overview of robot head

5 Experiment

5.1 Experimental set up

The robot head is set on the slider which can be moved manually.

The experiment method is described below. First, a gazing target is placed 50cm ahead of the robot, and the robot gazes it. Next, we move the robot on slider.

The learning method is described below. First, The above-mentioned experiment is conducted at not-learning system. Next, a neural network learns using the output data of OKR at that time. The learning data is built into a system and the above-mentioned experiment is conducted.

6 Result

Fig.6, **Fig.7**, **Fig.8** denotes the result when performing LVOR, OKR, LVOR with FEL. The horizontal axis denotes time[msec] and the vertical axis denotes the error.

2 compensated data and 2 uncompensated data is shown by **Fig.6**, **Fig.7**. Each data shows the time of moving slow and the time of moving fast.

Fig.6 shows faster reaction than OKR. But, it doesn't show better final errors.

Fig.8 denotes a result of LVOR with FEL. Compared with non-learned data, reaction velocity has be-

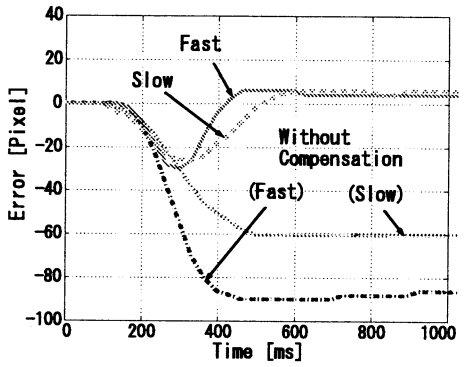


Figure 6: LVOR

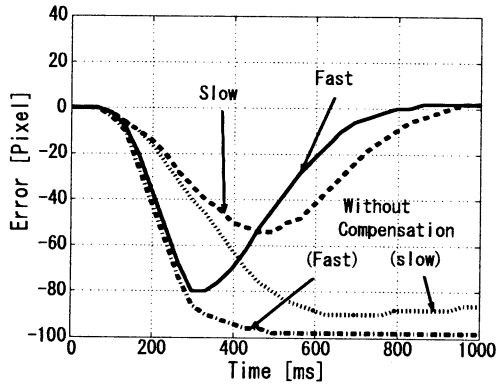


Figure 7: OKR

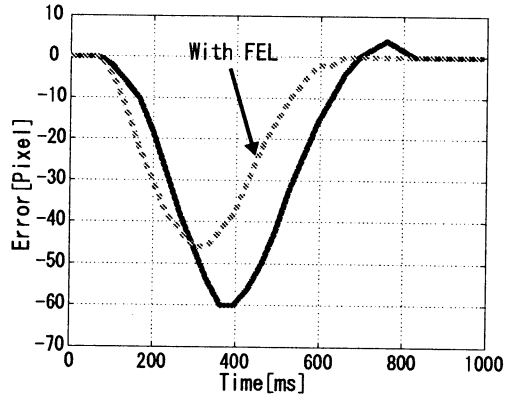


Figure 8: LVOR with FEL

come early clearly. Furthermore final error is 0 unlike **Fig.6**.

7 Conclusion

In this paper, we have proposed a method to improve the performance of LVOR by using FEL. FEL uses the output of OKR as the error signal to compensate for dynamics of motor and sensor. We used acceleration, velocity and position of the head as the input of FEL. Some experimental results are presented to demonstrate the effectiveness of the proposed method. A future research is adaptation to the robot which actually walks.

References

[1] Hiroaki Tsuji, Noriaki Maru, "Compensatory Eye Movement for translational Motion of Robot Head" *Proceeding of the 19th Annual Conference of the Robotics Society of Japan*,2001 (in japanese)

[2] Kenichi Murata, Fumio Miyazaki, Noriaki Maru, "Compensational of Head Rotation in Active Stereo Vision System" *Proceeding of the 34th Annual Conference of the Society of Instrument and Control Engineers*,Vol.1 pp31-32, (1995).

[3] Tomohiro Shibata, Stefan Schaal, "Fast Learning of Biomimetic Oculomotor Control with Nonparametric Regression Networks," *Journal of the Robotics Society of Japan*,Vol.19 No.4 pp468-475, 2001

[4] Mitsuo Kawato, "Feedback-error-learning neural network for supervised motor learning" *In R.Eckmiller,Advanced Neural Computers*,pp.365-372.1990

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Space and Time Sensor Fusion Using an Active Camera For Mobile Robot Navigation

Tae-Seok Jin[1], Jae-Mu Yun[2], Kwon-Soon Lee[3], and Jang-Myung Lee[4]

[1,2,4]Dept. of Electronics Engineering, Pusan Nat'l Univ., Pusan, 609-735, korea
[3]Dept. of Electrical Engineering, Dong-A Univ., Pusan, 604-714, korea

Abstract

This paper proposes a sensor-fusion technique where the data sets for the previous moments are properly transformed and fused into the current data sets to enable accurate measurement, such as, distance to an obstacle and location of the service robot itself. In the conventional fusion schemes, the measurement is dependent on the current data sets.

As the results, more of sensors are required to measure a certain physical parameter or to improve the accuracy of the measurement. However, in this approach, instead of adding more sensors to the system, the temporal sequence of the data sets are stored and utilized for the measurement improvement. Theoretical basis is illustrated by examples and the effectiveness is proved through the simulations. Finally, the new space and time sensor fusion (*STSF*) scheme is applied to the control of a mobile robot in an unstructured environment as well as structured environment.

Keywords: Multi-sensor data fusion, Image processing, Localization, Navigation, Mobile robot

1. Introduction

So far many of researches have been done on the spatial fusion technique. That is, multiple sensor data are utilized either for the purpose of providing complementary or redundant data to measuring physical parameters. That is, all of the current data from the sensors are integrated and fused to obtain a correct set of data.

In this new approach, the data obtained by the sensors are utilized until they do not have any efficiency for the measurement decision. The data set can be either redundant to improve the accuracy or complementary for the measurement. For the later case, this space and time sensor fusion is essential for the measurement.

The space and time fusion is inevitable for the complementary case. Therefore the effectiveness is very clear and the utilization method will be determined by the sensory data structure. However for the redundant case, it is required to define that how to fuse the previous data sets to the current data set. In this paper, we are basically going to utilize the minimum square solution for the fusion scheme without considering the error variance in the measurement for simplicity.

2. Space and Time Sensor Fusion

The space and time fusion consists of combining information acquired at different instants and then deciding the data. It implies that the system must be able to predict objects state at each instant (see Figure 1).

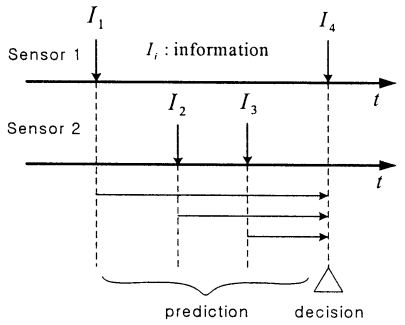


Fig. 1. An example of Space and Time fusion.

2.1 Sensor Fusion Transformation

Let us define the k-th moment data set provided by i-th sensor as, $\mathbf{z_i(k)}$, and the k-th measurement vector as $\mathbf{x(k)}$. Then the conventional sensor fusion technique provides the measurement as

$$\hat{x}(k) = \sum_{i=1}^{n} W_i x_i(k) \tag{1}$$

where $x_i(k) = H_i z_i(k) \in R^m$,

H_i represents transformation from the sensory data to the measurement vector, and $W_i \in R^{m \times m}$ represents the weighting value for i-th sensor.

Note that in the measurement of $\mathbf{z_i(k)}$, the low-level fusion might be applied with multiple sets of data with known statistics[2]. The determination of H_i is purely dependent on the sensory information and the decision of W_i can be done through the sensor fusion process. Later this measured data are provided to the linear model of the control/measurement system as current state vector, $\mathbf{x(k)}$. In this approach, we propose a multi-sensor data fusion using sensory data, $Tz_i(j)$, as

$$\hat{x}(k) = \sum_{i=1}^{n} W_i \{ \sum_{j=1}^{k} P_j Tz_i(j) \} \tag{2}$$

where $\sum_{j=1}^{k} P_j = 1$.

Note that when each of sensor information can provide the measurement vector, that is, the redundant case, $Tz_i(j)$ can be expanded as

$$Tz_i(j) = T_j + H_i z_i(j) \tag{3}$$

where T_j represents the homogeneous transformation from the location of the j-th to the k-th measurements.

However, when the multi-sensors are utilized in the complementary mode, the transformation relationship cannot be defined uniquely; instead it will be defined depending on the data constructing algorithm from the measurements. For example, a single image frame captured by a camera on a

mobile robot cannot provide the distance to an object until the corresponding object image is provided again from a different location. This algorithm will be described in detail in the section 3.1.

Figure 2 illustrates the concept of this space and time sensor fusion. Estimation of parameter may provide the measurement vector at each sampling moment. The verification of significance and adjustment of weight steps are pre-processing stages for the sensor fusion. After these steps, the previous data set will be fused with the current data set, which provides a reliable and accurate data set as the result of space and time sensor fusion.

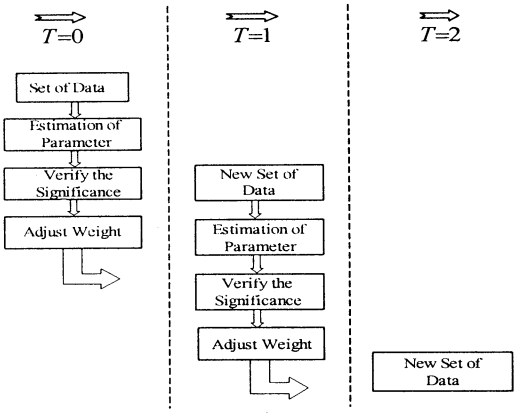


Fig. 2 Concept of Space and Time Sensor Fusion.

2.3. Auto-correlation for Estimation Techniques

Each previous data set is transformed to the k-th (current) sampling location, and represented by the measurement vector, $Tz_i(j)$. Now how can we fuse the k data sets into a reliable and accurate data set? In the Eq. (2), W_i can be determined by the geometrical relationship among sensors, in other words, by the spatial sensor fusion.

While the estimating sensor is tracking a feature, it generates a stream of measurements. When there is relative motion between the feature and the sensor, the processes then cease to be stationary. As an illustration of gathering the model information from the sensor, we shall only consider the stationary case, in order words, there is no motion between the sensor and the feature being tracked. Our interest in the proceeding analysis lies only in determining whether the process noise is white or not.

Random processes are defined in terms of their ensemble averages and these can be estimated. Our model shall be in terms of such averages. In practice, we require to estimate these averages from finite sequences. We consider a process y_k as realized (estimated) by the finite sequence $y(k)$, for $0 \le k \le N-1$. That $y(k)$ is an estimate of the random process y_k is made plausible by a consideration of ergodic processes. From $y(k)$, we can, therefore, estimate the averages for the process, the mean is estimated by

$$\hat{\mu}_x = \frac{1}{N}\sum_{k=0}^{N-1} x(k) \quad \text{and the variance by} \quad \hat{\sigma}_x^2 = \frac{1}{N}\sum_{k=0}^{N-1}(x(k)-\mu_y)^2,$$

A biased estimate of the autocorrelation is given by

$$\tilde{\phi}_{xx}(m) = \frac{1}{N}\sum_{k=0}^{N-|m|-1} x(k)\cdot x(k+m) \tag{4}$$

with a variance of $\tilde{\sigma}_{\tilde{x}} = \frac{1}{\sqrt{N}}\tilde{\sigma}_x^2$, (5)

where $|m| < N$. Similarly, an unbiased autocorrelation is estimated by

$$\hat{\phi}_{xx}(m) = \frac{1}{N-|m|}\sum_{k=0}^{N-|m|-1} x(k)\cdot x(k+m) \tag{6}$$

and the variance of the unbiased autocorrelation is given by

$$\hat{\sigma}_{\hat{\phi}} = \frac{\sqrt{N}}{N-|m|}\hat{\sigma}_x^2 \tag{7}$$

for the biased autocorrelation and similarly for the unbiased one. Therefore, determination of P_j is the final step for the temporal sensor fusion. Note that this expands the dimension of sensor fusion from one to two.

As one of solid candidate, we propose here to use the auto-correlation as an index for the weight adjustment and have the form,

$$\Psi_j = \sum_{k=-\infty}^{+\infty} x_i(k)x_i(j-k). \tag{8}$$

Depending on the correlation, P_j will be determined as

$$P_j = \frac{\Psi_j}{\sum_{j=1}^{k}\Psi_j}. \tag{9}$$

3. Applications to Mobile Robots

3.1 Complementary Usage for 3D Vision

If the image for an object is well matched to one model in the database, the position of the object can be obtained directly. In a well-structured environment, it may be a usual case. However, when the mobile robot is navigating in an unstructured environment, it needs to recognize the position/orientation of an object located in the middle of its path, which is not known to the robot a priori.

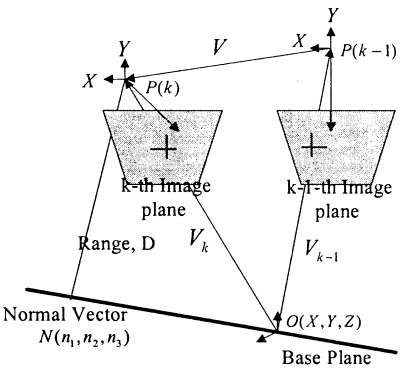


Fig. 3. Transformation of camera coordinates.

As a typical geometrical model for camera, a pinhole model is widely used in vision application fields as shown in Fig. 3. At the k-th sampling moment, a scene point O(X,Y,Z) is captured by a camera on the mobile robot. The vectors from the scene point to the k-th and *(k-1)*th camera perspective center are represented by V_k and V_{k-1}, respectively. The motion of mobile robot from *(k-1)*th moment to k-th moment is represented by V. Now we can write the vector relationship as

$$V_{k-1} = V_k - V. \tag{10}$$

This can be represented as a matrix form,

$$\alpha\begin{bmatrix} x_{k-1} \\ y_{k-1} \\ -f \end{bmatrix} = \beta\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} x_k \\ y_k \\ -f \end{bmatrix} - \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (11)$$

where $(x_k, y_k, -f)$ and $(x_{k-1}, y_{k-1}, -f)$ represent the projection of the scene point onto the camera image planes; $V(v_1, v_2, v_3)$ represents the translational motion of the mobile robot; r_{ij} is an element of the rotation matrix, R represents the relative rotation between the two camera frames; α and β are constants.

Now consider the reference base plane passing through the scene point P with a direction vector $N(n_1, n_2, n_3)$; then the range value, D, can be represented as

$$D = V_k \cdot N \ . \quad (12)$$

This can be represented again as

$$D = \beta(n_1 x_k + n_2 y_k - n_3 f) \ . \quad (13)$$

Now, Eq. (11) is reformulated as

$$(\alpha/\beta)\begin{bmatrix} x_{k-1} \\ y_{k-1} \\ -f \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} x_k \\ y_k \\ -f \end{bmatrix} \quad (14)$$

where $a_{ij} = r_{ij} - (v_i \cdot n_j / D)$.

Expanding the matrices and dividing rows one and two by row three gives

$$D(R_3 x_{k-1} + R_1 f) = C_3 x_{k-1} + C_1 f \quad (15)$$

$$D(R_3 y_{k-1} + R_2 f) = C_3 y_{k-1} + C_2 f \quad (16)$$

where $R_i = r_{i1}x_k + r_{i2}y_k - r_{i3}f$ and $C_i = v_i(n_1 x_k + n_2 y_k - n_3 f)$.

In matrix form, these equations can be expressed as

$$AD = B \quad (17)$$

where $A^T = [a \quad b], B^T = [c \quad d]$, $a = R_3 x_{k-1} + R_1 f$,

$b = R_3 y_{k-1} + R_2 f$, $c = C_3 x_{k-1} + C_1 f$, and $d = C_3 y_{k-1} + C_2 f$.

Use of the pseudo-inverse matrix enables computation of the range value, D which is associated with image point (x_k, y_k), and is written as,

$$D = \frac{(ac + bd)}{a^2 + b^2} \ . \quad (18)$$

So far, we have shown that using the consecutive two image frames, the distance information of the scene point can be obtained as using the stereo images at a certain moment.

3.2 Space and Time Fusion Filter

Given two estimators $\hat{\Theta}_1$ and $\hat{\Theta}_2$ of Θ, and the task is to fuse then together to form one single "optimal" estimate $\hat{\theta}$. Here, estimators are stochastic variables and are denoted by capital letters. Assume that $\hat{\Theta}_1 - \Theta$ and $\hat{\Theta}_2 - \Theta$ are independent Gaussian distributed with zero mean and covariances P_1 and P_2, respectively. Now let $X = \Theta - \hat{\Theta}_1$ and $Y = \hat{\Theta}_2 - \hat{\Theta}_1$. Then $\sum_{xy} = P_1$ and $\sum_{yy} = P_1 + P_2$. Hence

$$\hat{x} = P_1(P_1 + P_2)^{-1}(\hat{x}_2 - \hat{x}_1) \quad (19)$$

$$\hat{\theta} = [P_1^{-1} + P_2^{-1}]^{-1}[P_1^{-1}\hat{\theta}_1 + P_2^{-1}\hat{\theta}_2], \quad (20)$$

with covariance

$$P_1 - P_1(P_1 + P_2)^{-1}P_1 = [P_1^{-1} + P_2^{-1}]^{-1} \ . \quad (21)$$

The fusion formula just means that estimates should be weighted together, with weights inversely proportional to their qualities/variances. It is easy to modify the fusion filter to handle correlated estimators.

4. Robot Type in Experiments Setup

The mobile robot used in the experiments is shown in Figure 5 along with some of its sensory components. Its main controller is made on system clock 1 GHz, Pentium IV Processor. Basically, the sensors, 16-ultrasonic and a robust odometry system are installed on the mobile robot. For visual information, a CCD camera is mounted on the top of the mobile robot in order to sense obstacles or landmarks of the side and the rear of mobile robot.

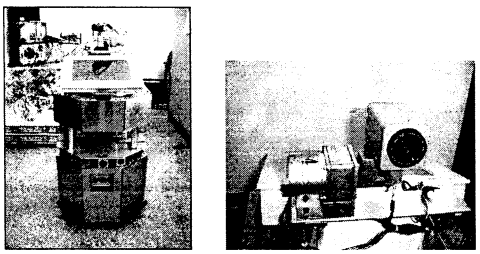


Fig. 5. *IRL*-2001 robot and CCD camera.

5. Experimental Results

5.1 Robot Localization using a Landmark

The service robot(*IRL*-2001) is commanded to follow the environment as shown from (a) to (f) of Figure 7. We performed the experiment for two cases.

To begin with, the 2-D landmark used by *IRL-2001* is shown in Figure 6. The primary pattern of landmark is a 10cm black square block on white background and a 5cm square block. The major reasons for choosing the square blocks are

- The projection of a square block in the image plane can always be approximated by an ellipse, which is easy to recognize using the elliptical Hough transformation technique.
- A square pattern is more robust to noise and occlusion than circular, polygonal patterns during template matching process, even though all these patterns can be detected by using Hough transformation technique.

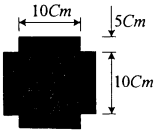


Fig. 6. The landmark pattern and size used by *IRL-2001*.

The image corners are then automatically extracted by camera parameters, and displayed on Figure 6 and the blue squares around the corner points show the limits of the corner finder window. The corners are extracted to an accuracy of about 0.1 pixel. The extrinsic parameters, relative positions of the landmark with respect to the camera, are then shown in a form of a 3D plot as Figure 8. And on Figure 9, every camera position and orientation are represented by red pyramid, therefore we can see the location and the orientation of a mobile robot in the indoor environment. To measure the relative distance of the landmark from the mobile robot, we first measure the distance of image from the fixed position in *IRL*-lab corridor.

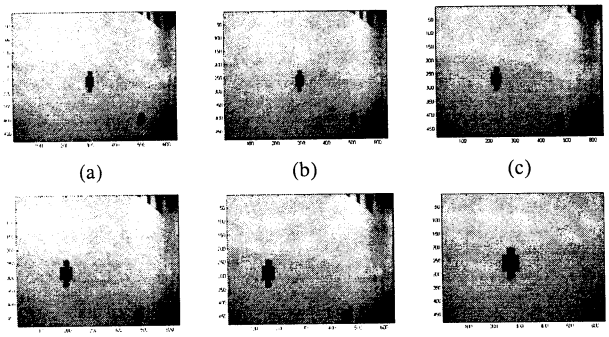


Fig. 7. A landmark locations detected by camera.

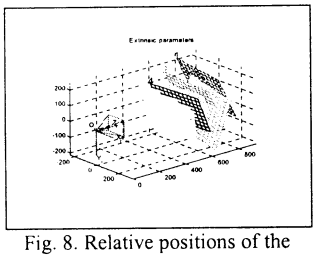


Fig. 8. Relative positions of the landmark w.r.t the camera.

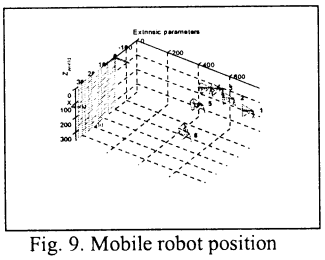


Fig. 9. Mobile robot position and orientation.

The predefined values of the landmark defined in this section are given as follows the origin of coordinates is equal to the origin of mobile robot, a Y-axis is fit to the front of mobile robot and an X-axis is perpendicular with Y-axis.

Table 1. The result of relative distance (Dim.:m).

Frame Number	World Coordinate Distance	Image Coordinate Distance	Error
1	7.81	8.13	0.32
2	7.02	7.30	0.28
3	6.28	6.53	0.25
4	5.06	4.89	0.17
5	5.52	5.39	0.13
6	6.32	6.46	0.14

Table 1 lists the comparison data between world coordinate distance and image coordinate distance measured in *IRL*-lab corridor. From table 1, we find the maximum and the minimum error on distance is 0.32 m and 0.13m, respectively.

It shows that the distance error becomes less and less by frames, which composes the environment map. And so, we can use it to measure the relative distance of the mobile robot.

Finally, the robot is tested to follow the whole trajectory from start position to final position as shown in Figure 10 and Figure 11.

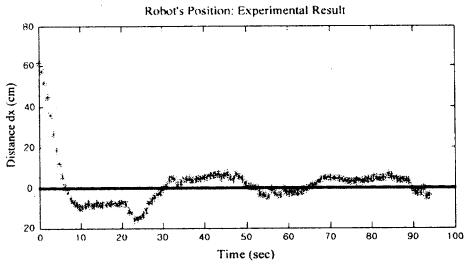


Fig. 12. Robot's position experiment result.

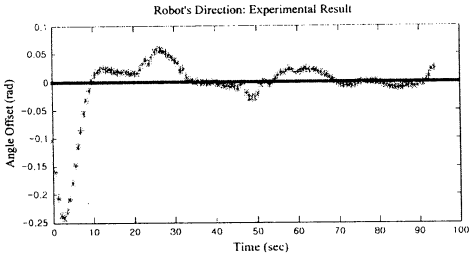


Fig. 13. Robot's direction experiment result.

The experimental results of the robot status under such control strategy are given in Figure 12 and 13.

6. Conclusions

In this paper, a new sensor fusion concept, STSF(space and time sensor fusion), was introduced. The effectiveness of STSF was demonstrated through the examples, simulations and experiments. To generate complete navigation trajectories without *a prior* information on the environment, not only the data from the sensors located at different places but also the previous sensor data are inevitably utilized. Although we have tried using the sonar system for map building and navigation in indoor environment, the result from the above experiments clearly shows that by utilizing both systems and applying active sensing to adapt to differing situation, a high level of competent collision avoidance behavior by STSF can be achieved.

Sonar system and visual systems are cooperatively utilized for collision avoidance based upon STSF such that a mobile robot was successfully navigated in an unstructured environment as well as in a structured environment. Based on these results, further experiments will aim at applying the proposed tracking technique to the multi-sensor fusion scheme which is applied to the control of a mobile robot in an unstructured environment.

Acknowledgements

The author would like to acknowledge financial support from Center for Intelligent & Integrated Port Management Systems[CIIPMS] at Dong-A university.

References

[1] R. C. Ruo and K. L. Su, "A Review of High-level Multisensor Fusion: Approaches and applications," *Proc. Of IEEE Int'l. Conf. On Multisensor Fusion and Integration for Intelligent Systems*, pp. 25-31, Taipei, Taiwan, 1999.

[2] J. M. Lee, B. H. Kim, M. H. Lee, M. C. Lee, J. W. Choi, and S. H. Han, "Fine Active Calibration of Camera Position/Orientation through Pattern Recognition," *Proc. of IEEE Int'l. Symp. on Industrial Electronics,* pp. 100-105, Slovenia, 1999.

[3] M.e Kam, X. Zhu, and P. Kalata, "Sensor Fusion for Mobile Robot Navigation," *Proc. of the IEEE*, Vol. 85, No. 1, pp. 108-119, Jan. 1997.

[4] P. Weckesser and R. Dillman, "Navigating a Mobile Service-Robot in a Natural Environment Using Sensor-Fusion Techniques," *Proc. of IROS*, pp.1423-1428, 1997.

[5] J. Llinas and E. Waltz, *Multisensor Data Fusion*. Boston, MA: Artech House, 1990.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Face Detection System by Camera Array that Satisfies both Wide View and High Resolution

Kosuke Okabe* Hiroshi Mizoguch**
Takaomi Shigehara* Kazuyuki Hiraoka* Masaru Tanaka* Taketoshi Mishima* Shuji Yoshizawa*
*Department of Information & Computer Sciences
Saitama University
255 Shimo Okubo, Saitama 338-8570, Japan

**Department of Mechanical Engineering
Tokyo University of Science
2641 Yamazaki, Noda 278-8510, Japan

Abstract

This paper presents a vision system that is possible to detect human face in wide view. The system has an advantage that it is capable to simultaneously satisfy incompatible requirements of wide view and high resolution. The authors have studied human symbiotic and collaborative robot. As a part of our effort to realize such robot, the authors have been developing face recognition system. During the development, we have faced a trade-off between wide view and high resolution. When the resolution is increased, i.e. image is closed up, so that a face can be stably detected, field of view becomes narrower. On the contrary, when the filed of view is widened, then the resolution becomes lower and the face detection becomes fail. To resolve this trade-off, the authors introduce camera array of more than one cameras. The camera array enables to satisfy the above mentioned incompatible requirements simultaneously. The authors have already implemented a working prototype system of the camera array based upon the above idea. And we have conducted experiments using the prototype. The experimental results demonstrate feasibility and effectiveness of the idea behind the system. In our experiment, the field of view is widened at 1.7 times.

1 Introduction

The authors have studied human symbiotic and collaborative robot. As a part of our effort to realize such robot, the authors have been developing face recognition system [1]. During the development, we have faced a trade-off between wide view and high resolution. When the resolution is increased, i.e. image is closed up, so that a face can be stably detected, field of view becomes narrower. On the contrary, when the filed of view is widened, then the resolution becomes lower and the face detection becomes fail.

For example, this problem cannot be ignored when combining a speaker array [2] and the face recognition system. Since the view range of a single camera is relatively narrow, it is limited to the view range that the target person can hear sound. Thus, in this paper, the authors solve this problem by using the camera array which consists of two or more set cameras to extend the view range. The camera array satisfies both wide view and high resolution simultaneously.

2 Camera-Array System

2.1 Problem

Fig. 1 shows the relation between resolution and a view. Here, the method of carrying out face discovery is considered in this case using the face discovery system using template matching based on the correlation operation which authors developed. The size of a template is 16x16 pixels. As for the size of a picture, both Fig.1(a) and Fig.1(b) are 120x90 pixels. As shown in fig.1(a), since the resolution of the face for a screen falls, the problem that a face cannot be discovered occurs. On the contrary, if resolution is raised to the grade which can discover a face in the state of Fig.1, the view range of a camera becomes narrow. That is, the view range and resolution have the relation of a trade-off.

2.2 Solution

There is a method of pursuing a candidate person using two or more cameras [3]. In this paper, solution of the problem of the trade-off stated to the foregoing paragraph by using the camera array which consists of two or more fixed cameras is aimed at. Fig.2 shows the Configuration of Camera Array. As shown in a Fig.2, two or more cameras are aligned out so that a mutual view may overlap. A camera is connected with a computer, respectively and performs face detection independently. Each camera sends the information on a success or failure of face discovery in a charge view to the integrated server in a figure. In addition to this, when a face is discovered,

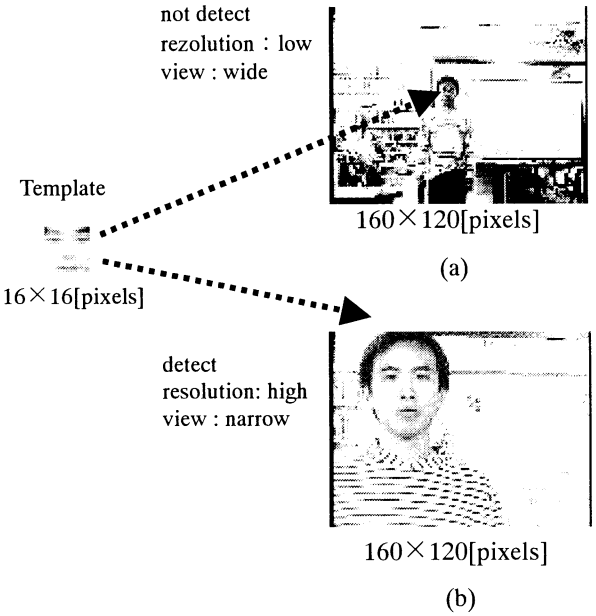


Fig. 1 : Relation between resolution and a view

face position information is transmitted. Based on a face discovery success and failure, an integrated server judges the validity of the sent data. When people exist in the domain which the view of a camera overlaps, although a face will be found with two or more cameras, naturally the average value of the sent data in that case is determined as a face position.

Thus, the range of a view can be extended by using a camera array. Since two or more cameras share a view, it is not necessary to extend the view of each camera, therefore resolution does not fall. In consequence of this, the phenomenon in which a face cannot be discovered by the low resolution is avoidable.

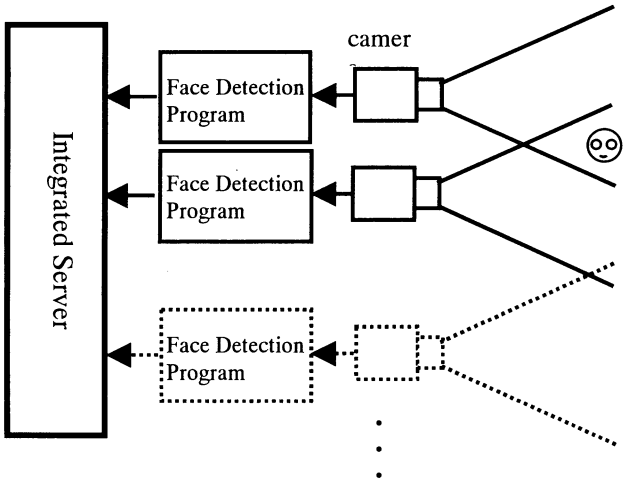


Fig. 2 : Configuration of Camera Array

3 Experiment

3.1 Method

The authors have built a prototype camera array system in order to prove the validity of the above-mentioned idea. Fig. 3 shows the appearance of an experiment system.

Fig.4 illustrates the block diagram of the implemented camera array system. As shown in fig. 4, the method of determining the position of a face passed the information on the position of the face discovered by each face discovery program to the face position output program, and it realized by sending each position information to an integrated server using Socket.

The contents of an experiment are shown below. The field of view of one camera in camera array conform to the field of view of a single camera. In other words, as the number of the cameras in a camera array increases, a view becomes larger from a single camera. In this state, It investigates whether a face can be detected or not. The number of the cameras which constitute camera array is two. Face detection uses the face detection system by the vision which the author implemented. Furthermore, Table.1 shows the conditions for execution of face detection.

Table 1 : Experimental Condition.

OS	Linux
CPU	Pentium4 2.8GHz
Multiple resolution	10 level
Skin Color Extraction	use
Size of Template	16×16[pixels]

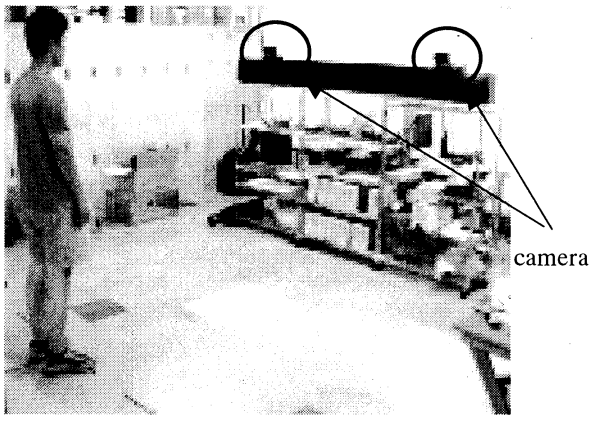


Fig. 3 : Figure of System

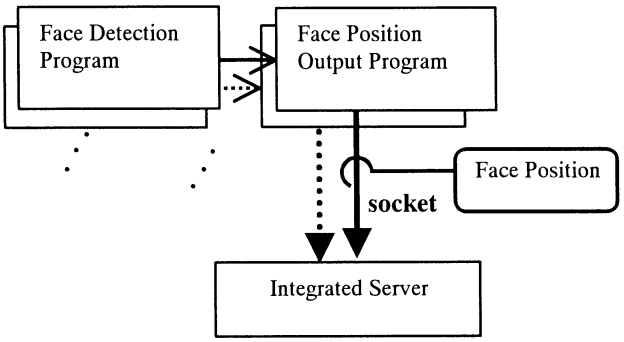


Fig. 4 : Block Diagram of Camera Array

3.2 Experimental Result

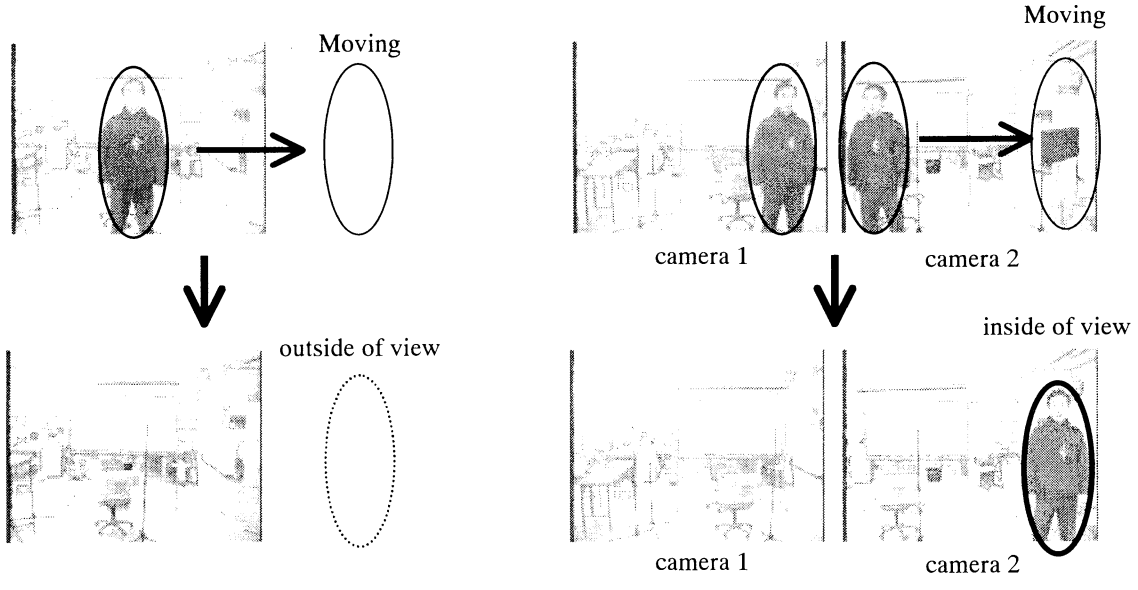
The result of the experiment is presented. Fig.5 shows comparison of the view of camera array and a single camera in actual operation environment. At this time, the size of the image of the single camera and camera array is 320x240 pixels. In the case of single camera of Fig. 5(a), if the person moves in the direction indicated by arrows , a single camera loses sight of person. To the contrary, in the case of Camera Array of Fig. 5(b), person is in the view of a camera 2. For this reason, when resolution is the same, it is the view that the camera array is larger than a single camera. There we measure actually width of a view. Fig.5 shows the actual measurement of the view range. Fig.5(a) is a view range in the case of one camera. and Fig.5(b) is a view range in the case of a camera array. In consequence of this, in the case of same resolution, the view of a camera array is 1.7 times as large as a single camera.

Next, the face discovery by the camera array is described in the following. Fig. 8 shows time transition of the face detection in each camera. In an experiment, person walks from the left end of a camera 1 to a camera 2. the position of detected face among Fig.8a is given a superimposition indication of the frame. Fig. In 8 (a), a candidate person moves to (6) from (1). In addition to this, (3) and (4) are each picture of the cameras 1 and 2 of same time. In consequence of this, Even if it disappears from the view of a camera 1, it turns out that it has detected in a camera 2 continuously.

Fig.8 (b) is the graph showing time transition of the position of the detected face. The vertical axis of Fig. 8(b) shows the position where a face is horizontal. namely, shows the value of the x-coordinate of Fig. 8(a). The value of the horizontal position of a face is a value which averaged the value acquired from cameras 1 and 2 in the integrated server. As shown in a figure, the center of a camera 1 is the points of origin. The horizontal axis of Fig.8b shows time. The unit of a horizontal axis is the number of frames. In consequence of this, A camera array can perform face detection in the range larger than a single camera.

4 Conclution

The single camera has trade-off that a large-scale field of view and big high resolution are incompatible. Then, the problem can be solved using the camera array. As a result, it succeeds in realizing face detection covering a 1.7 times



(a) Single Camera

(b) Camera Array

Fig. 5. Comparison of a view

as many large area as this compared with a single camera. The future works are expansion of much more view range with three or more cameras and detection and simultaneous tracking of the face of two or more person.

Acknowledgements

This work has been partly supported by CREST of JST (Japan Science and Technology Corporation). The workalso has been supported in part by Grant-in-Aid for Scientific Research by JSPS.

References

[1] HIDAI Ken-ichi, et al, "Robust Face Detection Against Brightness Fluctuation and Size Variation", *Proc. of IROS2000*, pp. 1379-1384 ,2000.

[2] MIZOGUCHI Hiroshi, et. al, "Implementation of Invisible Messenger System to Whisper in a Person's Ear Remotely by Integrating Visual Face Tracking and Speaker Array", *Proc. of SMC2002*, pp.WA2N4(1)-(5), 2002.

[3] Norimichi Ukita, et al , "Real-Time Cooperative multi-Target Tracking by Communicating Active Vision Agents", *Proc. of ICPR2002*, pp.14-19, 2002.

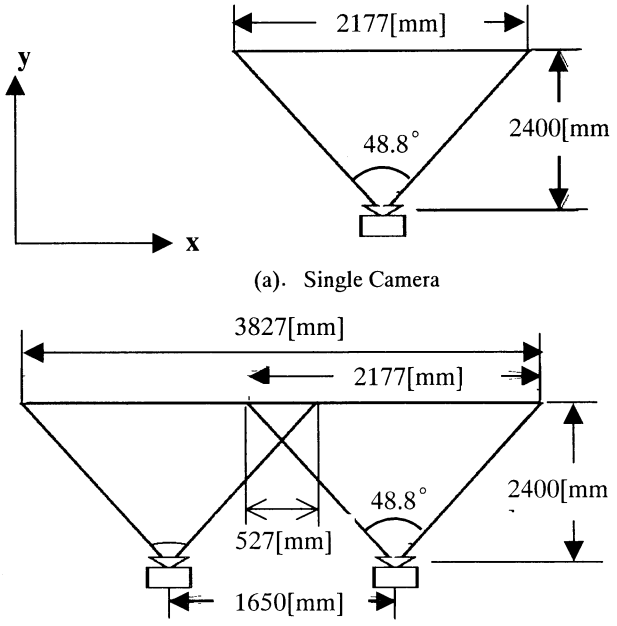


Fig. 5. View of Camera

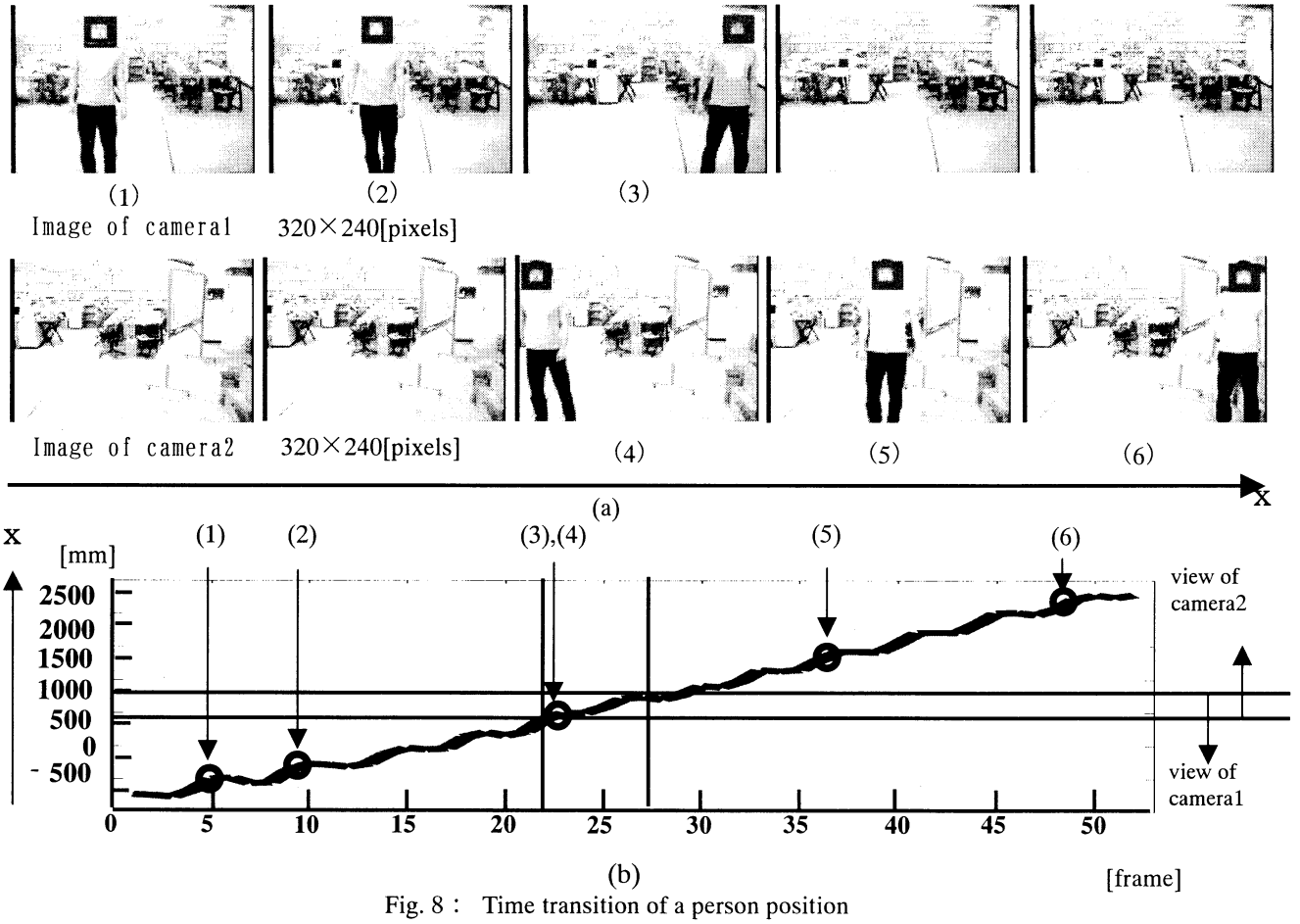


Fig. 8 : Time transition of a person position

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

2D Artificial Life System Using Network-type Assembly-like Language: A Comparative Study with Linear-type Assembly-like Language

Yuhki Shiraishi* Kotaro Hirasawa† Jinglu Hu* Junichi Murata*
*Graduate School of Information Science and Electrical Engineering, Kyushu University
Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, Japan
†Graduate School of Information, Production and Systems, Waseda University
2-2 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan
e-mail: yuhki@cig.ees.kyushu-u.ac.jp

Abstract

Evolutionary dynamics of digital creatures were investigated by T. Ray in Tierra, and evolutionary processes and complex symbiotic relationships between creatures were observed; but, they have reached an evolutionary stable state. It is considered as an open problem to unlock the full potential of evolution in digital media.

We have proposed a new system using network-type assembly-like language to solve the problem and to unlock the full potential of evolution in a computer, where creatures, i.e., self-replicating programs, live in a discrete 2D torus space and an interaction between creatures is restricted locally like C. Adami's Avida.

In this paper, we compare the network-type assembly and linear-type assembly using Simpson's and Shannon's diversity. And it is confirmed that the diversity of the network system is greater than that of the linear system. These show the possibility that the network-type assembly-like system has more potential of evolution.

Keywords: Self-replication, Network structure, Diversity, Evolvability, Intron

1 Introduction

In the early work of Artificial Life, T. Ray has created Tierra motivated by a desire to observe the evolutionary process in a medium other than carbon chemistry [1]. In his system, a digital creature is a self-replicating assembly-like program, because he consider self-replication and open-end evolution as the essential properties of life. And, evolution is caused by random mutation and interaction among creatures.

He investigated the evolutionary dynamics of the digital creatures in Tierra, and observed the evolutionary process and complex symbiotic relationships between creatures; but, it has reached an evolutionary stable state, i.e., the number of species doesn't increase quickly and explosively as it does in the Cambrian period.

That means, this system probably does not have the same potential for evolutionary innovation as physical systems. It is considered as an open problem to unlock the full potential of evolution in digital media [2]. He and his college are trying to increase the richness of its evolution appending new features: computer networks and multi-threads architectures [3].

We have proposed a new system using network-type assembly-like language to solve the problem and to unlock the full potential of evolution in a computer [4], where creatures, i.e., self-replicating programs, live in a discrete 2D torus space and an interaction between creatures is restricted locally like C. Adami's Avida [5].

We constructed a brand new network-type assembly-like language because (1) a network-type assembly forms a higher hierarchical level of structure and (2) physical world's creatures probably take the advantages of their own networks structure (e.g. proteins). We belive that complexity is one of the most important properties of life.

Our network-type assembly have some linear assembly's properties, because it is difficult for the network-type assembly to self-replicate. This is not so curious because we know genes form a linear structure and proteins' reactions form a network structure.

The aims of this paper are to investigate the basic properties of our system comparing to Avida-like linear-type assembly-like system and to study the evolvability in our system. In this case, a basic instruction set is used which has no command to parasite other creatures.

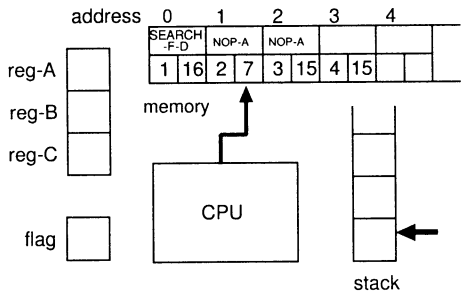


Figure 1: Structure of the virtual CPU.

Table 1: Instruction sets.

Network	Linear
NOP-A NOP-B NOP-C	NOP-A NOP-B NOP-C
IF-EQU-MOD IF-NOT-EQU-MOD	IF-NOT-EQU
JUMP-F	JUMP-F JUMP-B
SHIFT-R SHIFT-L INC DEC	SHIFT-R SHIFT-L INC DEC
PUSH POP	PUSH POP
ADD SUD NAND	ADD SUB NAND
ALLOCATE DIVIDE	ALLOCATE DIVIDE
COPY-E COPY-N1 COPY-N2	COPY
SEARCH-F-D	SEARCH-F SEARCH-B
FLIP-FLAG	

2 System Description

2.1 Overview

As we said above, we use a network-type self-replicating assembly-like language, creatures, i.e., self-replicating programs live in a discrete 2D torus space. And we expect to observe evolutionary processes by random mutation and an implicit dynamic fitness function. In this paper, our system is similar to Avida but a network-type assembly-like language.

Each grid in the space has an virtual CPU, a local memory, three registers (`reg-A`, `reg-B`, and `reg-C`), a `flag`, and a stack (Figure 1); and each grid has one creature at most. Therefore, MIMD (Multiple Instruction Multiple Data) parallel computers are suitable for our system. In this paper, we assume that each CPU has the same power of processing,

Figure 1 shows that a memory cell (or *node*) has an instruction and other two values that are the addresses of the node to be executed next step. The method of an exact execution is as follows:

step 1. execute the instruction at address 0.
step 2. if `flag` equals 0,
step 2-1. move to the node whose address is written at lower left in the current node modulo the *size*[1] of the creature.
step 2-2. otherwise, move to the node at lower right modulo the size of it.
step 3. execute the instruction of the current node.
step 4. jump to step 2.

The above process is repeated until a creature die out. In this case, the *lifespan*[2] of a creature is proportional to the size of it, and *time-slice*[3] is a constant.

[1]The number of nodes.
[2]The number of instructions executed after which creatures die.
[3]The number of instructions a creature execute every *update*.

When it copies itself to one of the 4-neighborhood cells (up, down, left, and right); (1) if there is no space, i.e., all four cells are already occupied by creatures, the oldest creature is killed and the new creature is copied to this cell, (2) otherwise, it is copied to one of the free cells selected randomly.

2.2 Features of Network Assembly

Instructions[4] were created in the similar mind with Avida [6] (Table 1) but not the same: (1) our system have a `FLIP-FLAG`[5] command, (2) some instructions are executed modulo the size of the creature, and (3) `SEARCH` and `JUMP` commands are executed only forward direction using `flag` as it is natural for our network structure.

Creatures have nodes' values unused[6] in execution but used in evolution. In other words, our network-type system's creatures have introns naturally but ordinary linear-type system's don't have ones. This realizes the diversity of species in our system.

2.3 Mutation and Evaluation

Evolution needs variations of creatures and evaluation of these.

For variations, (1) when a creature executes a `COPY` command, a random copy-error occurs (*copy mutation*); (2) when it executes a `DIVIDE` command, random insertions and/or deletions of the node occur (*divide mutation*). Even after the divide mutation, each node keeps indicating the same nodes, because of not destroying the structure of the creatures.

[4]The details of the instructions is shown in [4].
[5]Flipping the value of a flag.
[6]We can find the unused nodes by executing the creature.

Table 2: Ancestor creatures. ("*": intron)

Network				Linear
node no.	Instruction	next node		Instruction
		flag=0	flag=1	
0	SEARCH-F-D	1	16*	SEARCH-F
1	NOP-A	2	7*	NOP-A
2	NOP-A	3	15*	NOP-A
3	ADD	4	15*	ADD
4	INC	5	18*	INC
5	ALLOCATE	6	3*	ALLOCATE
6	PUSH	7	6*	PUSH
7	NOP-B	8	15*	NOP-B
8	POP	9	5*	POP
9	NOP-C	10	11*	NOP-C
10	POP	11	9*	SUB
11	NOP-B	12	12*	NOP-B
12	COPY-E	13	13	COPY
13	COPY-N1	14	14	INC
14	COPY-N2	15	15	IF-NOT-EQU
15	INC	16	16	JUMP-B
16	IF-EQU-MOD	17	12*	NOP-A
17	DIVIDE	18	7*	DIVIDE
18	NOP-B	19	10*	NOP-B
19	NOP-B	5	19*	NOP-B

Evaluation is whether creatures survive or not. We can guess the size of the creature decreases through a evolutionary process, because the smaller one usually can copy itself faster.

2.4 Configuratin

In this paper, a creature dies out after it executes commands 20 times the size of it steps, copy mutation rate is 0.005, divide mutation rate is 0.05, world's size is 60×60, and *time-slice* is 30. At the first step, one man-made ancestor creature is copied to one cell (Table 2).

2.5 Measured Indices

We measure the diversity of species(or genes) in the network-type and linear-type system to study the possibility of evolution.

In this paper, two popular diversity's indices are used. One is Simpson's diversity:

$$D = 1/\Sigma p_i^2, \tag{1}$$

and the other is Shannon's diversity:

$$H = -\Sigma p_i \log p_i, \tag{2}$$

where p_i is the proportional abundance of species i [7].

Now, we redefine two words, *gene* and *species* to compare the network system to the linear one. (1) If every node's instruction and two values are the same between two creatures each other, they have the same *genes*. (2) Under the same condition but unused values (or *intron*) as (1), they belong to the same *species*.

To study the effect of the modulo calculation, we use the species in two contexts. This means we call the above species *species in the narrow sense*, and two creatures belong to the same *species in the broad sense* when they have the same instructions and next node's values modulo the size of them. This is introduced because creatures which belong to the same species in the broad sense execute almost the same instructions.

Table 3: The mean(m) and standard deviation(σ) of Simpson's diversity and those of Shannon's diversity over 10 runs (500,000 *updates* per run).

Simpson's Diversity			
		m	σ
Network	genes	1249.4	561.4
	species in the narrow sense	41.4	8.3
	species in the broad sense	15.3	5.6
Linear	genes	4.21	0.52

Shannon's Diversity			
		m	σ
Network	genes	7.32	0.24
	species in the narrow sense	4.88	0.24
	species in the broad sense	3.92	0.72
Linear	genes	2.75	0.25

3 Results

Network system's species and linear system's genes are considered to have a quite similar concept as for phenotype, because the creatures belonging to the same species(or genes) behave similar[7].

Fig. 2 and 3 shows the time series of Simpson's diversity and those of Shannon diversity. Both results shows the diversity of the network system is greater than that of the linear system.

The mean and standard deviation of Simpson's diversity and those of Shannon's diversity over 10 runs (500,000 *updates*[8] per run) are shown in Table 3. These help the above results.

[7] As for genotype, these are not similar. Because the creatures belonging to the same species consist of different codes as the result of introns, and the creatures belonging to the same genes consist of the same codes. This causes a significant difference in evolution.

[8] An artificial unit of time.

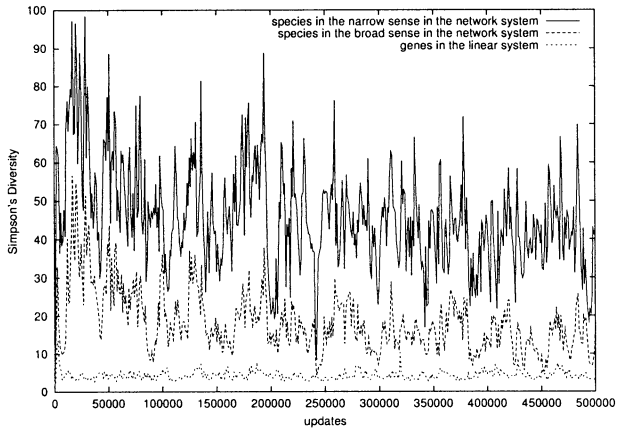


Figure 2: Simpson's Diversity vs. updates.

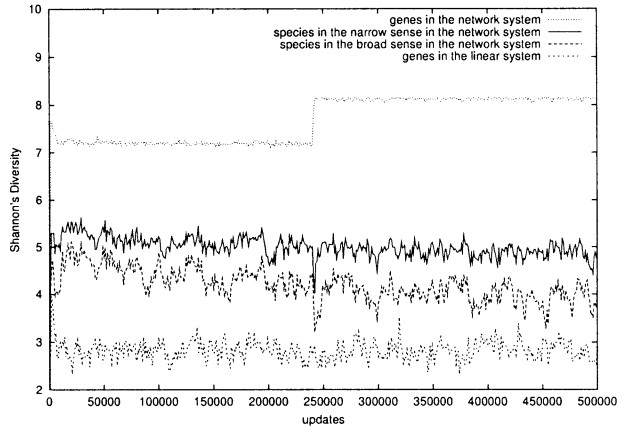


Figure 3: Shannon's Diversity vs. updates.

Figure 4 shows the time series of the size of creatures belonging to dominant species in the network system, and that of creatures belonging to dominant genes in the linear system. Then, the decrease of the size, as we discussed in section 2.3, is confirmed, where a hunting occurs in network system, but doesn't in the linear system, that is interesting.

4 Conclusions

We have proposed a new system using network-type assembly-like language.

In this paper, we compare the network-type assembly and linear-type assembly using Simpson's diversity and Shannon's diversity. And it is confirmed that the

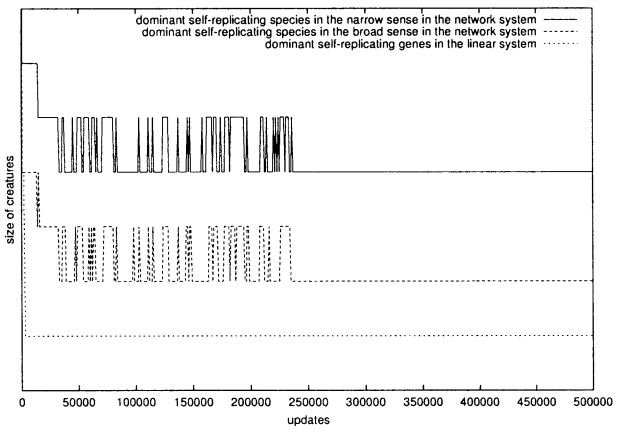


Figure 4: Size of creatures belonging to dominant species(genes) vs. updates.

diversity of the network system is greater than that of the linear system. These show the possibility that the network-type assembly-like system has more potential of evolution.

References

[1] T. S. Ray: An Evolutionary Approach to Synthecic Biology, Artificial Life, pp.179-209, The MIT Press (1995)

[2] Bedau et al.: Open problems in Artificial Life, Artificial Life VI, pp.363-376, the MIT Press (2000)

[3] T. S. Ray and J. F. Hart: Evolution of Differentiation in Multithreaded Digital Organasms, Artificial Life, pp.132-140, The MIT Press (2000)

[4] Y. Shiraishi, K. Hirasawa, J. Hu and J. Murata: The Basic Study of Artificial Ecosystem Models Using Network-type Assembly-like Language, Late-Breaking Papers of GECCO-2002, pp.412-418 (2002)

[5] C. Adami and C. T. Brown: Evolutionary Learning in the 2D Artificial Life System "Avida", Artificial Life IV, pp.377-381, The MIT Press (1994)

[6] C. Adami: Introduction to Artificial Life, Springer-Verlag New York, Inc. (1998)

[7] E. P. Odum: Basic Ecology: CBS College Publishing. (1983)

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Adaptive trail formation under dynamic feeding

Tomomi Tao , Hiroyuki Nakagawa[a],Masato Yamasaki[b]
and Hiraku Nishimori[a]

Department of Systems Engineering, Shizuoka University,Hamamatsu 432-8561,Japan
[a]Department of Mathematical Sciences, Osaka Prefecture University,Sakai 599-8531,Japan
[b]Department of Mathematical Sciences, Ibaraki University,Mito 310-8512,Japan

Abstract

Using a simple model for the trail formation of ants, the relation between i)the schedules of feeding which represent the unsteady natural environment, ii)emerging patterns of trails which connect between a nest and foods, and iii)the foraging efficiency is studied. Simulations results show that the emerging trails are classified into several types of characteristic geometries and each of them appears so as to realize the efficient foraging tactics for each feeding regime.

1 introduction

So far, various studies have been made on the collective behavior of social insects [1, 2, 3, 4, 5, 6, 7, 8, 9]. Specially, the trail formation of ants [1, 2, 3, 4, 5, 6, 7, 8, 9] have called a wide interest as one of the remarkably synergetic behavior fulfilled by presumably innocent individuals [10, 11]. Still, many points are left unconfirmed on the trail formation process, like the detailed ingredient of almost all types of chemicals(pheromone) [1, 2] secreted by individual ants and their roles in each socio-biological regime [1, 2, 3, 4, 5, 6]. In spite of such inadequate understanding of the real counterpart, various models on ants [6, 7, 8] have been proposed to imitate their behaviors using heuristic approaches. Here, setting a simple set of rules several parts of which are following previous studies [6, 7, 8], we investigate the relation between i)the schedules of feeding which represent the unsteady natural environment, ii)emerging patterns of trails which connect between a nest and foods, and iii)the foraging efficiency accompanied by each schedule of feeding. Our final purpose is to connect the pattern formation dynamics of ants to the consequently emerging functionality with which they can flexibly survive in the unsteady environment.

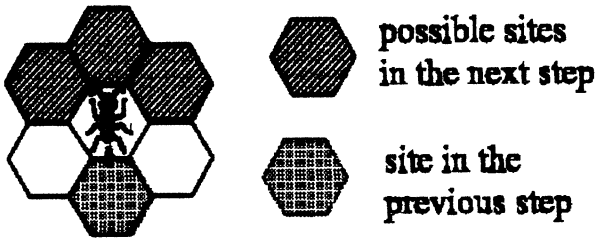


FIG. 1 Each ant is, at each time step, facing one of the six nearest sites. This facing direction corresponds to the moving direction in the previous time step. The possible moving directions in each time step are: the forward (=facing) direction and its neighboring (right and left) directions. To choose one of them, a stochastic rule with the weight, $P_\alpha = exp(-\Delta^\alpha/T)/Z$ is adopted where Z is the normalization factor. Here α indicates forward, right, or left direction relative to the facing direction, and $\Delta^\alpha \equiv \rho_\beta^\alpha - \rho_\beta(\mathbf{x}, t)$ is the gradients of pheromone density to the candidate directions where β is the index to indicate the recruit pheromone(for mode-II ants) or the foot pheromone(for mode-III ants). Note the above weight is not applicable for ants in mode-I which is making random walk independent of pheromone field. Moreover,the amount of pheromone emitted by ants depends on their 'energy(=activity)' which varies in the following manner; They get 'energy' when they reach foods or the nest. The value E_β of this energy decrease as they walk like $E_\beta(n+1) = BE_\beta(n)$ here $0 < B < 1$. If the energy is wasted to go below critical values $E_\beta^{critical}$ the ants is forced to come back to the nest to restart the foraging behavior in the mode I. The last rule is introduced to avoid the endless random walk or the endless circular walk though the process in the corresponding reality is not clear.

2 Model

The Model consists of a colony of N movable agents (we call them 'ants') which are situated on a triangular 2D lattice with periodic boundary condition. Each of these ants is located at one site in the lattice and is facing one of the six nearest sites(fig1) to which he may move according to the dynamical rule described below.

The only task for the ants is foraging, that is, finding foods and carrying them back to a nest, where the feeding sites are located at two corners of an equilateral triangular the third corner of which is the nest(fig2).

The amount and the schedule of feeding are controllable through a set of control parameters. Specifically,

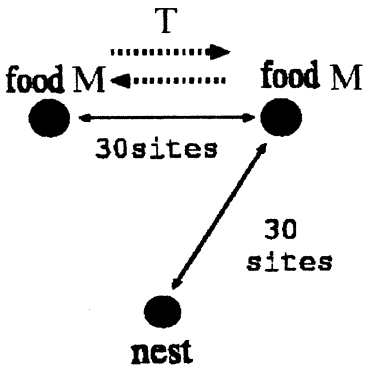


FIG. 2 The method of feeding. Feeding sites are located at two corners of an equilateral triangular the 3rd corner of which is the nest site. The amount M of foods are supplied, by turn, from one of these sites to the other at every feeding interval T. After each feeding event, they decreases as taken away by ants.

at every constant interval T, a certain amount M of foods are supplied at one of two feeding sites. The site of supply will change, by turn, from one feeding site to the other feeding site. Until the next feeding, the amount of foods at the corresponding site monotonically decrease as ants bring them away. As the mean for the communication among ants, two types of attractive chemicals,i.e., pheromones, are secreted and perceived by individual ants according to their temporal modes which consist of: i)random walk mode (modeI), ii)exploring mode (modeII) and iii)homing mode (modeIII). Pheromones introduced in the present model consists of: i)recruit pheromone and ii)foot pheromone, the density of them are expressed as, respectively, $\rho_{rec}(\mathbf{x}, n)$ and $\rho_{foot}(\mathbf{x}, n)$ where $\mathbf{x}$ is the position of the underlying site and n is the corresponding Monte Calro time step.

Detailed behaviors of ants in individual modes and the transition rule between different modes are described below:

[modeI]modeI is the random walk mode. In this mode every ant starts/restart from the nest, thereafter, before each ant reaches on a site with more than a critical density of recruit pheromone such that, $\rho_{rec}(\mathbf{x}, n) > \rho_c$, he keep himself in this mode. Specific dynamics of ants in this mode is as follows: Each ant is, at each time step, facing one of the six nearest sites(fig.1). This facing direction corresponds to the moving direction in the last walking event. The moving direction in the present walking event is among the forward (=facing) direction and its neighboring (right and left) directions, to choose one of which, a stochastic rule with an equivalent weight is applied, in such sense, random walk is realized. During this mode after each walking event, constant amount of foot pheromone is secreted at corresponding sites. Remember modeI is the random walk mode of ants all of whom start from the nest, therefore, density field of foot pheromone secreted by large number of ants forms rather monotonic gradient field around the nest which serves as the landmark for homing(modeIII) ants though some degree of deformation is unavoidable because of the behavior of ants in modeII which also secrete same pheromone as explained soon. On reaching the site with $\rho_{rec}(\mathbf{x}, n) > \rho_{rec-c}$, the ant is excited into mode-II, or, on arriving at sites with foods, his mode changes into III

[modeII]ModeII is the exploring mode for foods, here, ants walk perceiving the local gradient of recruit pheromone density. The possible moving directions in each time step are same as the case at mode-I, while stochastic rule with the weight $P_\alpha = exp(-\Delta^\alpha/T)/Z$ is adopted where $\Delta^\alpha \equiv \rho^\alpha - \rho(\mathbf{x}, t)$ is the gradients of pheromone density to individual directions, Z is the normalization factor, and, α indicates forward, right, or left direction relative to the facing direction. During this mode, as is in modeI, after each waking event, a certain amount of foot pheromone is secreted at the corresponding sites. On arriving at sites with foods, the mode changes into modeIII, or, if the ant gets lost from the sites with $\rho_{rec}(\mathbf{x}, n) > \rho_{rec-c}$ his mode turns back to the previous mode-I.

[mode-III]ModeIII is the homing mode after getting foods. On arriving at foods, regardless of his previous mode, his mode changes into modeIII in which mode, perceiving the local gradient of foot pheromone he tries to walk back to the nest. During this mode recruit pheromone is kept secreted at each site of the walk.

In addition to the above dynamics of ants, pheromones evaporate in the air. Namely pheromones density at site $\mathbf{x}$ decay as

$$\rho_\beta(\mathbf{x}, n+1) - \rho_\beta(\mathbf{x}, n) = -A_\beta \rho_\beta(\mathbf{x}, n) \qquad (1)$$

where A_β are the evaporation rates and β is the index for recruit/foot pheromones. Moreover, pheromones diffuse to the nearest sites[?]

$$\rho_\beta(\mathbf{x}, n+1) - \rho_\beta(\mathbf{x}, n) = D(<< \rho_\beta(\mathbf{x}, n) >> - \rho_\beta(\mathbf{x}, n)) \qquad (2)$$

where $<<>>$ means the average over the nearest sites of $\mathbf{x}$, D_β are diffusion constants. Furthermore, the amount of pheromone secreted by ants depends on their 'energy' which varies with time, that is, they get the 'energy' on reaching foods or the nest. The value $E_{\beta'}$ of this energy decrease as they walk like

$E'_\beta(n+1) = B_{\beta'} E_{\beta'}(n)$ here $0 < B_{\beta'} < 1$. If the energy falls below threshold values $E^{th}_{\beta'}$ the ants is forced to come back to the nest to restart the foraging behavior in the mode I. This rule is introduced to avoid the endless random walk or the endless circular walk which are hardly observed in nature.

3 Simulation

All the simulations are performed on a 150×150 triangular lattice with periodic boundary condition. At the initial time step, N=500 ants are located at the nest, and no pheromone are distributed in the field, thereafter, all ants are simultaneously released from the nest. Note the unit time in our simulation corresponds to one Monte Calro step within which N elementary randomly chosen ants will sequentially take one step.

Under a proper combination of fixed parameters *on ants* (i.e.,total number of ants, evaporation rates and diffusion constants of pheromones, the location of the nest, and, energy supply at the sites of foods and nest, the critical energy to continue the foraging behavior), with a certain range of the values of control parameters $\{M, T\}$ *on feeding*, formation processes of trails and the subsequent efficient foraging are realized. In such regime we see:i)the geometries of trails and ii)the accompanying foraging efficiency.

Now, before showing the several characteristic geometries of trails and the conditions for the appearance of each of them, we introduce (or reconfirm) three different time scales relevant for this model: i)T:switching time, i.e., the interval between two successive feedings which take place at alternate feeding sites. , ii)T':residence time of food after each event of feeding. This interval is roughly proportional to M, that is, $T' \simeq aM$ where a is constant to represent the time required for the colony of ants to carry away a unit amount of food. iii)τ: characteristic time for ants to construct a new trail to a food. Unlike T and T', τ exclusively depends on the parameters *on ants* and the distance between the nest and feeding sites which are kept constant through our simulations.

The outcoming geometries of trails and their dependency on the pair of control parameters $\{T, M\}$ are displayed in fig.3 which give the spatial distributions of ants averaged over sufficiently longer time than T, there, trails' geometries are, roughly categorized into three types:i)V-shaped trails, ii)Y-shaped trail and iii)straight-trail.

The detailed situations for the appearance of these

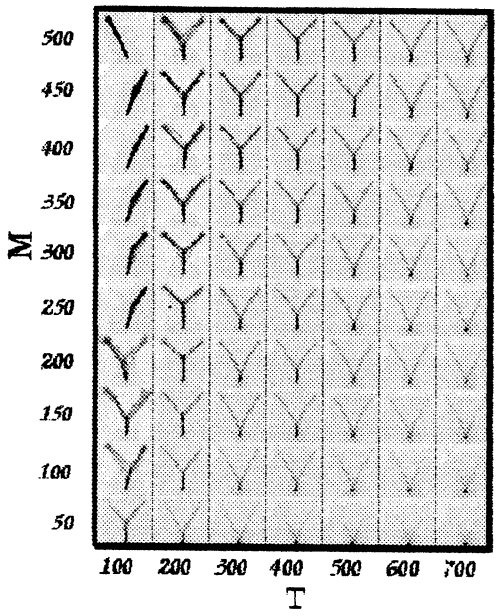


FIG. 3 The relation between emergent trail patterns and the combinations of feeding schedule parameters, $\{M,T\}$. Here darkness in each figure means ants density averaged over sufficient longer time than T, the interval between successive feedings. The trail geometries are, while fluctuating (or switching) with time, roughly categorized into three types after averaged over sufficiently longer period than T:
i)V-shaped trail: This shape is recognized if we take the average of the ants density of straight trails one end of which are sticked at the nest and the other is alternating between two sites of foods depending on the temporal food supply.
ii)Y-shaped trail: It connects the nest and both sites of foods with a junction located in-between. Temporally, it emerges as an un-forked path with a bend, while Y-shape is recognized after averaged over sufficiently longer time than T.
iii)Straight trail: It almost steadily connects the nest and the only one site of food. Here, temporal trail and its time average collapses each other.

geometries are like followings:

i)V-shaped trail: Typically in the right-bottom part of the figure, When the interval T between successive feedings is sufficiently long and the amount M of foods supplies at each feeding event is not adequate enough to remain until the next supply, a straight trail to the last feeding sites is made up and, thereafter, dissolves as foods are exhausted. The same process is repeated after each feeding event. Remember the feeding site switches from one to the other by turn, therefore, such trails are, averaged over sufficiently longer time than T, recognized as the V-shaped trail.

ii)Straight-shaped trail: If the amount of supplied foods at each feeding event is adequate and the switching time T is short, at the upper part of the left-most column in fig,3, ants will encounter with foods almost whenever they arrive at whichever site of foods. Then, once straight trail which connects between the nest

and an odd feeding site is established, no driving force acts on ants to make the alternative trail to the other feeding site. This configuration remains unchanged even after averaged over period sufficiently longer than the switching time. So that we classify this type of trail as the straight trail.

iii)Y-shaped trail:At the intermediate region between i) and ii), in the figure the upper-left part except the left-most column, the following is the case. In this regime, at any period of time, foods are remaining on, at least, one feeding sites, however, the amount of foods' supply is not enough for each site to be a permanent source of foods. Here, as time proceeds, ants find the Y-shaped path which has a junction used as the 'base-camp' to explore new foods while keeping the trunk trail from the nest to the camp. In this sense, the Y-shaped trails looks as the outcome of the intermediate tactics to realize the globally efficient foraging though not the optimizes path in the short time scale.

The figure indicates the most relevant quantity for the trail geometry is $M/T(i.e.T'/T)$, the amount of feeding per unit time. As M/T increases, the shape of trail varies roughly from the V-shape to the Y-shape and finally the straight trail is obtained.

To investigate the relation between the efficiency of the foraging and the trail geometry, the total mass F_{eff} of foods carried by ants into the nest per unit time is measured under various M with T fixed. Fig.4 is the case for $T = 100$. Here, monotonically increasing relation between M and F_{eff} is seen until F_{eff} reaches the saturation value at which most ants is involving in the foraging along the straight trail to one site of food. In the $M - F_{eff}$ relation, four characteristic zones are recognized as indicated by symbols A, B, C and D in fig.4. Zone D is the saturation phase mentioned above. In zones A and C are seen the almost linear $M - F_{eff}$ relations whereas the inclination in A is approximately twice as that in C. These inclinations correspond, respectively, to the foraging tactics with the Y-shaped trail and that with the straight trail, while a crossover tactics appears at the zone B where the trail shape is varying from one to the other depending on time. In such a way the change of trail geometry is accompanied with that of foraging efficiency. Especially, across the zone B ants are forced to switch their tactics from one to the other as shown in fig.3. At the zone A, the 'total transport capacity' of ants exceeds the total supply of foods, thus trails will extend to both sites of foods to take all of them. On the contrary, at the zone D the 'total transport capacity' of ants can not afford to cover both sites of foods, in other words, foods at each site are more than enough for all

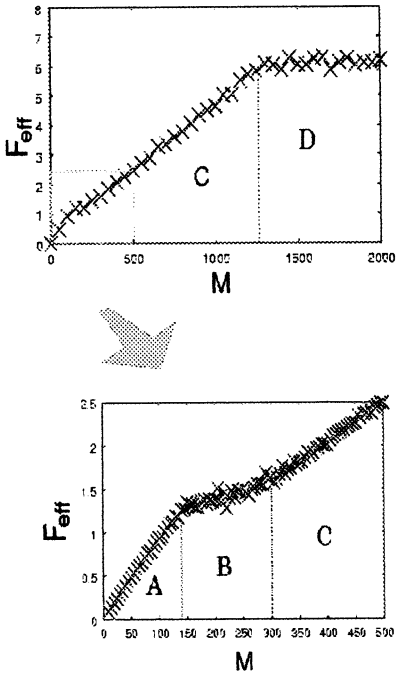


FIG. 4 The symbols × show the the relation between the amount of food supply M at each feeding and the efficiency of foraging F_{eff} by ants. The figure at the bottom is the inset of the above. There are four characteristic zones A,B,C and D for the $F_{eff} - M$ relation. Each zone is accompanied by each foraging tactics corresponding to each characteristic geometry of trail shown in fig.3. It means ants make adaptive foraging patterns according to the regime of feeding.

ants. Also in the zone C, ants are insisting to only one feeding site though food run out a short time before the next feeding at the corresponding feeding site. In the zone B where the food of each site becomes less than the above case, ants come to explore the other sites and Y-shaped tactics appears though irregularly varying into and back from the straight tactics. As noted in the below, it is not simply concluded that the crossover zone B in our simulation falls on the most efficiently range for the switching from Y-shaped trail to the straight trail. It is remarkable, however, that through the above switching of trail geometry ants will naturally change tactics from effective one for inadequate environment(with Y-shaped trails) to that for abundant environment(with straight trails).

The similar is the transition from the V-shaped trail to the Y-shaped trail under the variation of T with fixed value of M/T. This transition also is accompanied by the switching of the foraging efficiency. It connects two extreme cases, i)the case for $T >> \tau$ where V-shaped trails is formed as the superposition of the temporally optimized two straight trails. ii)the case for $T < \tau$ where Y-shaped trail is recognized. The

details of these cases are left to be reported.

4 Discussion

Here, we showed that the geometry of trails, which emerges from the local tactics of ants, has a deep relation to the efficient foraging. That is, ants as a group, while each of them perceive only local gradient of pheromones, can realize efficient foraging through a flexible change of the geometry of trail according to the feeding environment.

Still many problems remain to be solved; Firstly, the more precise relation between trail geometries and their foraging efficiency should be investigated which includes the detailed discussion on the location of crossover zone B. Also a wider class of studies varying other parameters than the present case. Secondly the comparison of 'our' ants to real ants is required though several tough problems should be overcome, like the specification of pheromones relevant to the trail formation. However, in this stage, it has a certain meaning to discuss possible mechanism how the adaptive tactics emerges among the innocent agents according to dynamically changing environment.

Acknowledgements

The authors thank the participants of a research project "Creation and Sustenance of Diversity" organized by International Institute of Advanced Studies for fruitful discussions.

References

[1] B.Holldobler B.and Wilson E.O, *The Ants*, Belknap, Cambrige, 1990.

[2] Gotwald,jr. W.H, *Army Ants*, Cornell University press, 1995.

[3] Hölldobler B. and Möglich M. *Insectes Sociaux*, Vol. 27,pp. 237, 1980.

[4] Haefner J.W. and Crist T.O, *J. Theor. Biol*, Vol. 166,pp. 299, 1994.

[5] Deneubourg J.L.and Goss S.and Franks N. and Pasteels J.M, *J. Insect Behavior*,Vol. 2,pp. 719, 1989.

[6] Deneubourg J.L. and Goss S, *Ethology Ecology & Evolution*,Vol. 1,pp. 295, 1989.

[7] Helbing D.and Schweitzer F. and Molnár P, *Phys. Rev. E*,Vol. 56,pp. 2527, 1997.

[8] Schweitzer F. and Lao K. and Family F, *BioSystems*,Vol. 41,pp. 153, 1997.

[9] Bonabeau E and Dorigo M and Theraulaz G, *Swarm Intelligence*, Oxford University press, Oxford, 1999.

[10] Haken H, *Advanced Synergetics, 2nd ed*, Springer, Berlin, 1987.

[11] Nicolis G and Prigogine I, *Self-Organization in Nonequilibrium Systems. From Dissipative Structures to Order Through Fluctuations*, Wiley,New York, 1977.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Learning of animal behavior strategy by neural network and genetic algorithm

K. Hayashi H. Kanoh
Dept. of Mechanical Eng. and Informatix
Meiji University
Kawasaki City, 214-8571 JAPAN

Abstract

This paper develops a learning process for a dynamic behavior in a dynamic environmental field where an insect and its enemy are moving around searching for food, a bait for the insect randomly appears in the field and when it is eaten another bait appears at some different position and the enemy always moves around and chases after the insect to catch when he sense the existence of the insect. The insect can take a bait and escapes from the enemy, but sometimes the insect is captured by the enemy during the turn to escape which gives the insect one step behind. More complicated situation where two enemies and one insect exist and point out the problems to be considered for this case is considered.

1 Introduction

Recently, the research for the artificial life has become popular[1]. The research is widely advanced on the theme how to imitate the action observed to the organism of the nature. This paper develops using the neural network action strategies for an insect that survives by searching bait while escaping from the enemy. To this end, we make a dynamic environmental field where an insect and its enemy are moving around searching for food, a bait for the insect randomly appears in the field and when it is eaten another bait appears at some different position and the enemy always moves around and chases after the insect to catch when he sense the existence of the insect[2],[3].

The insect and the enemy have the neural networks of feed forward type to decide the action. The neural nets are four layer neural networks with three inputs and one output and 30 nodes for each layer, the relationships between input-output are regulated by sigmoid functions only output layer is linear.

The existences of the bait, insect and enemy in the virtual world are expressed by the numerical value patterns from which the insect and the enemy can get information about existences of their foods.

2 Field and environment setting

The screen of 15 × 15 squares is set as a field where the bait is located and real time movements of the insect and the enemy are visualized. As information of the locations of them, bit patterns are given to indicate the densities of their existence as shown the next figures.

0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	2	1	0	0	0	0	0
0	0	0	0	1	2	3	2	1	0	0	0	0
0	0	0	1	2	3	4	3	2	1	0	0	0
0	0	1	2	3	4	5	4	3	2	1	0	0
0	1	2	3	4	5	6	5	4	3	2	1	0
1	2	3	4	5	6	7	6	5	4	3	2	1
0	1	2	3	4	5	6	5	4	3	2	1	0
0	0	1	2	3	4	5	4	3	2	1	0	0
0	0	0	1	2	3	4	3	2	1	0	0	0
0	0	0	0	1	2	3	2	1	0	0	0	0
0	0	0	0	0	1	2	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0

Fig.1 Pattern of bait

0	0	1	0	0
0	1	2	1	0
1	2	3	2	1
0	1	2	1	0
0	0	1	0	0

Fig.2 Pattern of enemy

In Fig.1, the bait is located at the number 7 and the insect goes toward 7 sensing the neighboring smell of the bait to get it, but he have to escape from the enemy when he senses in neighborhood the existence of the enemy the density pattern of which is shown in Fig.2. The enemy also can sense the density of the existence of the insect indicated by the same pattern as Fig.2. The insect and the enemy can sense only the values in front, right and left positions. Initial locations of the insect, the bait and the enemy are given at random. When the bait is gotten by the insect, another bait appears at random position. The insect has initially life span of 100 steps and he loss one step by each movement and gets more 5 steps when he catches the bait. One episode terminates when the insect spend all steps or the insect is captured by the enemy.

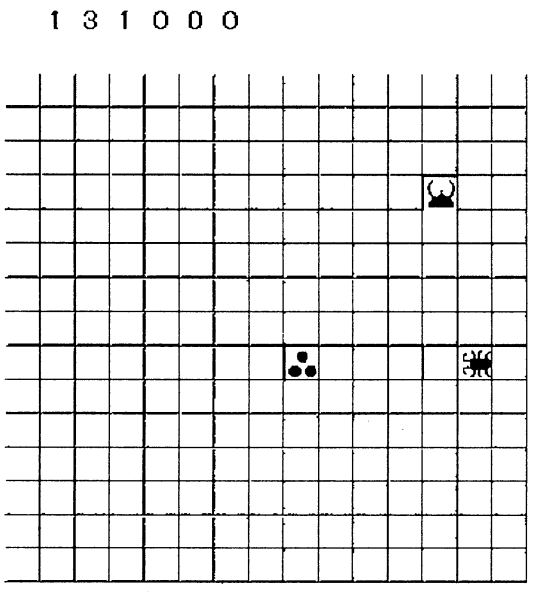


Fig.3 Field setting

Fig.3 shows the field setting on the monitor where represents the insect, represents the enemy and indicates the bait, the numbers on the upper part is insect's information on bait and enemy. At the present position, the insect senses the smell of bait 3 at front 1 at both sides and no indication of enemy in the neighborhood.

3 Learning with teacher

Learning of the insect and the enemy is performed using four layer neural networks[1] with three inputs and one output and 30 nodes for each layer as shown in Fig.4.

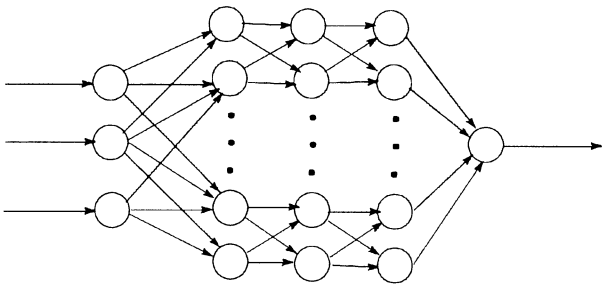


Fig.4 Neural Net used for insect and enemy

The inputs are the sensed patterns and the output is an action signal where 0 for turn to the left, 1 go straight and 2 for turn to the right. The neural net is trained to follow teacher signal which is made by the human judgment. The teacher signal for the insect consists of 39 situations, and the teacher signal for enemy consists of 17 situations.

3.1 The environment only with insect and bait

After sufficient training, the insect is put on the environment only with a bait. When there is no bait in his sight, the insect does random walk and when he senses the smell of the bait, the insect surely catches the bait and continually extends his life; he can possibly survive for ever.

3.2 The environment only with insect and enemy

After sufficient training, the insect and the enemy are put on the environment only without baits. When the mutual figures enter the visual field, the enemy single-mindedly chases the insect and the insect also intently tries to escape from the enemy, which cause endless run and chase.

3.3 Tripartite environment

We consider the environment as shown in Fig.3 where the insect, the enemy and the bait exist. In this environment, the insect has to survive by

taking a bait, while escaping from the enemy. For this case, the insect has two neural networks; one for getting food and one for escape while the enemy has one neural network for chasing after the insect. When there is no information about the objects, both the insect and the enemy make random walk and do not use the networks. When the enemy senses the existence of the insect, the enemy starts to chase after the insect using the neural network which is trained by teacher signals in advance. When the insect senses the existence of the enemy, he starts to escape from the enemy regardlessly of the existence of the bait, using the neural network trained for escape. When the insect senses the existence of the bait and does not sense the existence of the enemy, he starts go towards the bait to take it using alternative neural network trained to get the bait in advance.

As a result of the experiment, the insect can take a bait and escapes from the enemy, but sometimes the insect is captured by the enemy during the turn to escape which gives the insect one step behind. The cause of this failure is dependent on the teaching signal.

4 Learning without teacher

We consider the environment only with insect and bait. In this case, we do not give the teaching signal. Firstly the insect walk randomly, then usually the insect dies at 100 steps without getting the bait, but in many trials there is a case that he happen to be able to get the bait. We adopt this behavior as a teaching signals, and if there is not a teaching signal for some step, the action for this step is taken at random.

As a result of the experiment, the insect can take the bait in some cases, but in many cases he cannot get the bait, he just goes around the bait in vain. The cause of this failure is that the teaching signal in this case is taken from the random walk happen to be able to get the bait and the missing part of the teaching signal is supplied by random selection.

To improve the situation, we use the genetic algorithm[4] to determine the teaching signals. A string of the algorithm has 39 elements which represent action signals. Each element takes three values; 0,1, and 2 indicating the action.

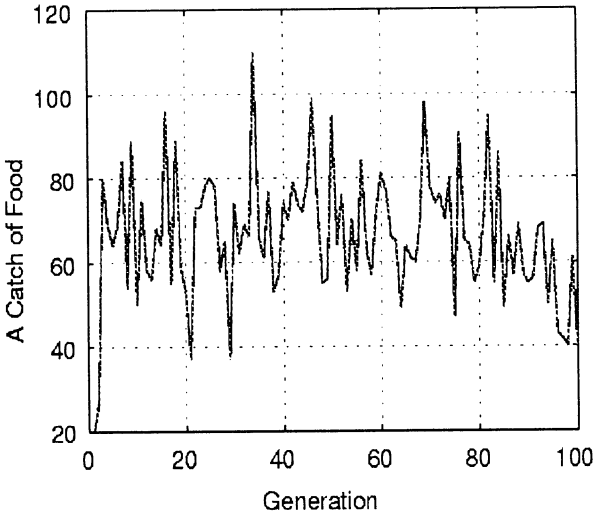


Fig.5 Bait capture frequency versus generation

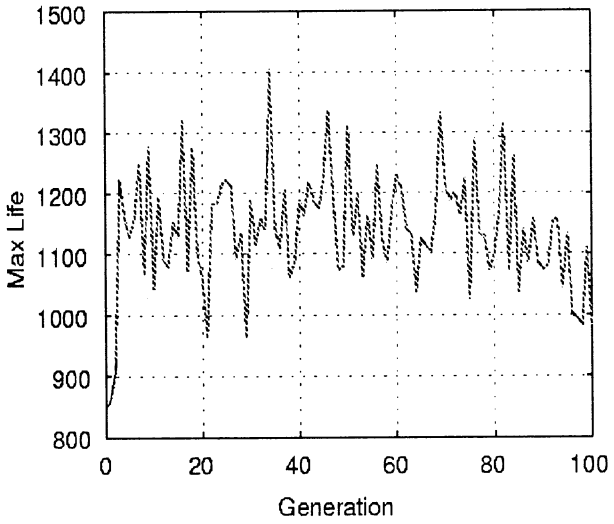


Fig.6 Life span versus generation

5 Optimization of number of neurons

The number of the neuron of each every layer is decided using the genetic algorithm. The algorithm is as follows:

I. The neural network of the four layer is made.

II. The population is 8 and the number of string element is 12, each of three element represents the number of the neuron of each layer.

III. The initial values for the string is given at random.

IV. The fitness is the life span till the insect dies.

The experimental results are shown in Fig.5, Fig.6. Both bait capture frequency and life span record the highest value at near 40 generation, after that these values are the descent tendency.

6 Capture by the coordination action of two enemies

As pointed out in section 3.2, one enemy and one insect tend to cause endless run and chase. Then we consider more complicated situation where two enemies and one insect exist. We point out the problems to be considered for this case as follows:

1. The both enemies have to grasp the position of the each other.
2. It is set so that the evaluation score may rise, when both sides attack is chosen.
3. The role is separated; one is specialized as a chaser and the other is specialize as a ambush.
4. Different strategies are given to the chase and the ambush.
5. To oppose to two enemies, the insect visual field has to be enlarged, then the insect can have strategy to get out the situation confronting two enemies.

7 Summary

The simulation of the vital phenomenon of the nature is considered by using neural networks and genetic algorithm. In learning with teacher and in the environment only with insect and bait, the insect surely catches the bait and continually extends his life; he can possibly survive forever and if there are one enemy and one insect they tend to cause endless run and chase. In tripartite environment where the insect, the enemy and the bait exist, the insect can take a bait and escapes from the enemy, but sometimes the insect is captured by the enemy during the turn to escape which gives the insect one step behind. The cause of this failure is dependent on the teaching signal. In the case of learning without teacher, the insect can take the bait in some cases, but in many cases he cannot get the bait, he just goes around the bait in vain. The cause of this failure is that the teaching signal in this case is taken from the random walk happen to be able to get the bait and the missing part of the teaching signal is supplied by random selection. To improve the situation, we use the genetic algorithm to determine the teaching signals. Using the genetic algorithm, the optimal number of the neuron of each every layer is searched. Both bait capture frequency and life span take the highest value at near 40 generation. We consider more complicated situation where two enemies and one insect exist and point out the problems to be considered for this case.

References

[1] Kazuo Kyuma, "Modern Control Technologies.Tread and Application of Neural Network Technology.," *J. of the Institute of Electrical Installation Engineers of Japan* , Vol. 17, No.5, pp. 60-65, 1995.

[2] Hitoshi Iba, "from Genetic Algorithm to Artificial Life. ," 「*Computer Today*」 , Vol.11, pp.42-65, 2001.

[3] Kanji Ueda, "Development of Artificial Life Study. ," *System Control Information*, Vol.38, pp.31-38, 1996.

[4] , "Using genetic algorithms for optimization. ," 「*Analytical Chemistry*」 , Vol. 16, No.5 pp. 678A-679A, 1996.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Mutual tests among agents in distributed intrusion detection systems using immunity-based diagnosis

Yuji Watanabe and Yoshiteru Ishida
Dept. of Knowledge-based Information Eng.
Toyohashi University of Technology
Toyohashi, Aichi 441-8580 JAPAN
watanabe@tutkie.tut.ac.jp and ishida@tutkie.tut.ac.jp

Abstract

Distributed intrusion detection systems have some advantages over centralized systems, such as scalability, resist subversion, and graceful degradation. With respect to resist subversion, however, self-monitoring is a difficult problem. One possibility is that each intrusion detection system is checked periodically by others.

In this paper, we propose mutual tests between intrusion detection system and mobile agent using immunity-based diagnosis. Some simulation results show that the credibility of normal intrusion detection system remains stable near 1, otherwise decreases to 0, and then corrupted ones are identified. Furthermore, we make sure that the diagnostic capability depends on some parameters.

Keywords: Intrusion detection system, Immunity-based diagnosis, Mobile agent, Self-monitoring

1 Introduction

The goal of intrusion detection is to identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators [1]. In the last few years, the need of intrusion detection system clearly increases with the growing number of network services, and then a large number of intrusion detection systems have been proposed (e.g., [2])

In order to design and build intrusion detection system, some researchers have drawn inspiration from the biological immune system. Forrest et al. have incorporated many properties of natural immune systems (distributed computation, error tolerance, adaptation and so on) into intrusion detection system [3, 4]. Spafford et al. also have developed the distributed intrusion detection system using autonomous agents regarded as immune cells [5, 6]. Mobile agents have been employed in intrusion detection system just as immune cells can circulate through the body [7, 8].

The intrusion detection systems inspired by the immune system are categorized into distributed system. Distributed intrusion detection systems have some advantages over centralized systems, such as *scalability, resist subversion*, and *graceful degradation* [6]. With respect to *resist subversion*, however, *self-monitoring* is a difficult problem. In other words, corrupted intrusion detection system cannot identify illegitimate use correctly; therefore it is necessary to discern which ones can be faulty. Although Spafford et al. suggested one possibility with each monitoring agent being checked periodically by several others, they achieved no detailed examination [6].

In this paper, we propose mutual tests using an *immunity-based diagnosis* in distributed intrusion detection system. The original immunity-based diagnostic model has been proposed by Ishida who is one of authors [9, 10]. The immune system can be considered as fully distributed diagnosis, where a large number of immune cells detect and eliminate non-self by stimulating and suppressing other cells. The diagnosis is performed by mutual tests among units and dynamic propagation of active state.

Furthermore, mobile agents contribute to the mutual tests in the diagnosis. In conventional approaches [7, 8], mobile agents directly monitor host computers, while mobile agents in our method observe intrusion detection systems. Our mobile agent acts as an additional module for existing intrusion detection systems.

To verify the feasibility of our diagnosis, we carry out some simulations. The result shows that the credibility of normal intrusion detection system remains stable near 1, otherwise decreases to 0, and then corrupted ones are determined. In addition, we execute the diagnosis changing two parameters: number of mobile agents and number of intrusion detection system's rules. We confirm what effects these parameters have on the diagnostic capability.

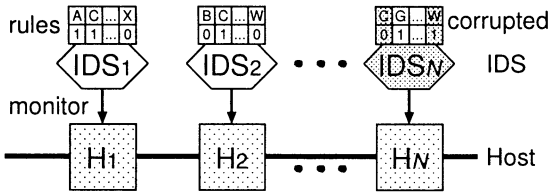


Figure 1: Simulated distributed intrusion detection system.

2 Distributed intrusion detection system with mobile agents

2.1 Simulated intrusion detection system

For easy performance analyses, the immunity-based diagnosis is applied to a simulated distributed intrusion detection system. Figure 1 illustrates the simulated system, where N intrusion detection systems can monitor the corresponding host computer using some rules. Real intrusion detection systems as show in Fig. 2 possess a lot of complicated rules, which are collected in some files according to service (for example, *dns.rules* file corresponding to DNS service). The rules are simplified by a pair of label (A, B, C, $\cdots$) and data (0 or 1). The total number of rules is defined by L, and each intrusion detection system has $L_h(\leq L)$ rules on average because each host provides different service. If $L_h = L$, then distributed intrusion detection system become homogeneous, otherwise heterogeneous.

We suppose that corrupted intrusion detection system includes some wrong rules represented by inversion of data (0 or 1). In Fig.1, although C:1 and W:0 are correct pairs, the corrupted intrusion detection system IDS_N has two inverted rules. Each simulation starts with the condition where there are $N_f(\leq N)$ corrupted intrusion detection systems.

2.2 Mobile agent

Each mobile agent with a piece of rules can migrate from host to host in order to check intrusion detection system mutually as depicted in Fig.3. The average number of agent's rules is represented by $L_a(\leq L)$. We assume that there are checks among agents on the same host (e.g., Ma_1 and Ma_2), while there is no test among hosts (e.g., IDS_1 and IDS_2) because mobile agent exists on behalf of communication between hosts. We will explain a concrete test outcome in **3.2**.

At the beginning of simulation, each intrusion detection system creates some mobile agents by duplicating a part of the rules. As a result, normal intrusion detection system bears fault-free mobile agents, while corrupted intrusion detection system has faulty mobile

```
[root@wata root]# ls /etc/snort/
RCS                         local.rules         snmp.rules
attack-responses.rules      misc.rules          snort.conf
backdoor.rules              multimedia.rules    snort_tutkie.conf
bad-traffic.rules           mysql.rules         sql.rules
chat.rules                  netbios.rules       telnet.rules
classification.config       nntp.rules          tftp.rules
ddos.rules                  oracle.rules        virus.rules
deleted.rules               other-ids.rules     web-attacks.rules
dns.rules                   p2p.rules           web-cgi.rules
dos.rules                   policy.rules        web-client.rules
experimental.rules          pop3.rules          web-coldfusion.rules
exploit.rules               porn.rules          web-frontpage.rules
finger.rules                reference.config    web-iis.rules
ftp.rules                   rpc.rules           web-misc.rules
icmp-info.rules             rservices.rules     web-php.rules
icmp.rules                  scan.rules          x11.rules
imap.rules                  shellcode.rules
info.rules                  smtp.rules
[root@wata root]# head /etc/snort/web-cgi.rules
# (C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
#    All rights reserved.
# $Id: web-cgi.rules,v 1.56 2002/08/18 20:28:43 cazz Exp $
#--------------
# WEB-CGI RULES
#--------------
#

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI
HyperSeek hsx.cgi directory traversal attempt"; uricontent:"/hsx.cgi";
content:"../../"; content:"%00"; flow:to_server,established;
reference:bugtraq,2314; reference:cve,CAN-2001-0253;
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI
HyperSeek hsx.cgi access"; uricontent:"/hsx.cgi"; flow:to_server,
established; reference:bugtraq,2314; reference:cve,CAN-2001-0253;
classtype:web-application-activity; sid:1607;  rev:3;)
[root@wata root]#
```

Figure 2: Example of snort rules [2].

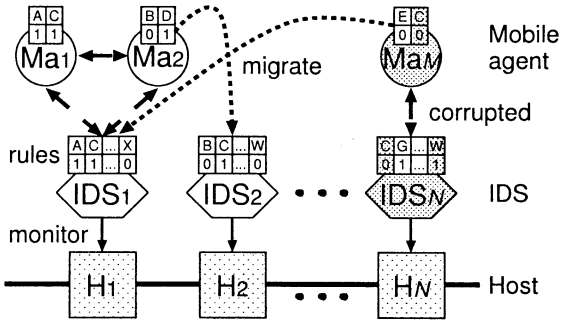


Figure 3: Mutual tests using mobile agents.

agents with some inverted rules (for example, Ma_M in Fig.3). Note that false mobile agents misdiagnose normal intrusion detection system.

3 Immunity-based diagnosis

3.1 Distributed diagnosis model

The distributed diagnosis models inspired by the biological immune system have been proposed by Ishida [9, 10]. The distributed diagnosis is performed by mutual tests among units and dynamic propagation of active states. In the model, each unit has the capability of testing other units, and being tested by the others as well. A state variable R_i indicating the *credibility of unit* is assigned to each unit and calculated as follows:

$$\frac{dr_i(t)}{dt} = \sum_j T_{ji}R_j + \sum_j T_{ij}R_j - \frac{1}{2}\sum_{j\in\{k:T_{ik}\neq 0\}}(T_{ij}+1), \tag{1}$$

$$R_i(t) = \frac{1}{1 + \exp(-r_i(t))}, \tag{2}$$

where the credibility $R_i \in [0, 1]$ is a normalization of $r_i \in (-\infty, \infty)$ using a sigmoid function. In equation (1), T_{ij} denotes binary test outcome from unit i to j as defined in **3.2**.

The diagnosis represented by the differential equation (1) has two characteristics. First, the credibility of tested unit i is updated by the sum of *the test value weighted* by the credibility of testing unit j. The weighted test value leads to neglect the test outcome of false unit with low credibility. Secondly, the credibility of unit i is evaluated not only from the opinions of other testing units, but also from the opinions of what the unit said to the other units. The former corresponds to the first term of the right-hand side of equation (1), and the latter to the second and third term. We call the latter *reflection effect*. The reflection effect is somewhat similar to the situation that if you criticize a highly respected person, it affects your own credit.

3.2 Test outcome

We explain how units, namely, both IDS and mobile agent, can produce their test outputs. The test outcome is assigned to -1, 0 or 1 according as whether or not rules are the same:

$$T_{ji} = \begin{cases} 1 & \text{if all rules match} \\ -1 & \text{if one or more mismatches exist} \\ 0 & \text{if all rules are not comparable} \\ -1/0/1 & \text{if unit } j \text{ is abnormal} \end{cases} \tag{3}$$

For example, in Fig. 3, the test outcome between Ma_1 and IDS_1 becomes 1 with the agreement of both rules A and C, the output between Ma_2 and IDS_1 is 0 because Ma_2 rules are not comparable with IDS_1 ones, and corrupted Ma_M and IDS_N replay unstably.

4 Simulation

4.1 False positive and false negative

The feasibility of the immunity-based diagnosis is verified by some simulations. In each simulation step, we record not only the credibility of intrusion detection system but also two evaluation indexes, that is, *false positive rate* α and *false negative rate* β as follows:

$$\alpha = \frac{N_t^{low}}{N - N_f}, \tag{4}$$

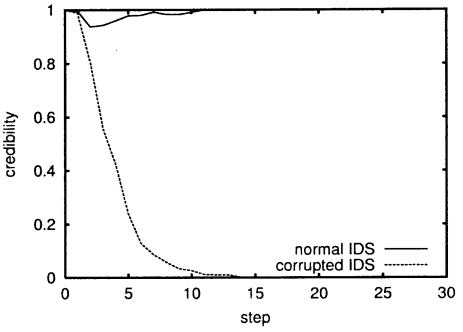


Figure 4: Transition of the average credibility R_i for normal and corrupted IDS over 50 trials.

Table 1: Parameters list.

	Description of variable
$R_i(0)$	Initial value of credibility
$r_i(0)$	Initial value of intermediate variable
N	Number of IDSs (hosts)
N_f	Number of corrupted IDSs
M	Number of mobile agents
L	Total number of rules
L_h	Average number of IDS's rules
L_a	Average number of agent's rules

$$\beta = \frac{N_f^{high}}{N_f}, \tag{5}$$

where $N - N_f$ denotes the number of normal intrusion detection systems, and N_f the number of corrupted ones. N_t^{low} is the number of normal intrusion detection systems with the credibility of not more than 0.8 ($R_i \leq 0.8$), while N_f^{high} is the number of corrupted intrusion detection systems with the credibility of not less than 0.2 ($R_i \geq 0.2$). The false positive means that the diagnosis regards normal as abnormal, while the false negative results from identifying abnormal as normal.

4.2 Results

Figure 4 illustrates transition of the average credibility for normal and corrupted intrusion detection system over 50 trials. In this simulation, the parameters listed in Table 1 are fixed: $R_i(0) = 1.0$, $r_i(0) = 1.0$, $N = 50$, $N_f = 50$, $M = 300$, $L = 5000$, $L_h = 500$ and $L_a = 50$. The result shows that the credibility of normal intrusion detection system remains stable near 1, otherwise decreases to 0, and then corrupted one is determined.

Other simulations are carried out with changes of two parameters, that is, the number of mobile agents

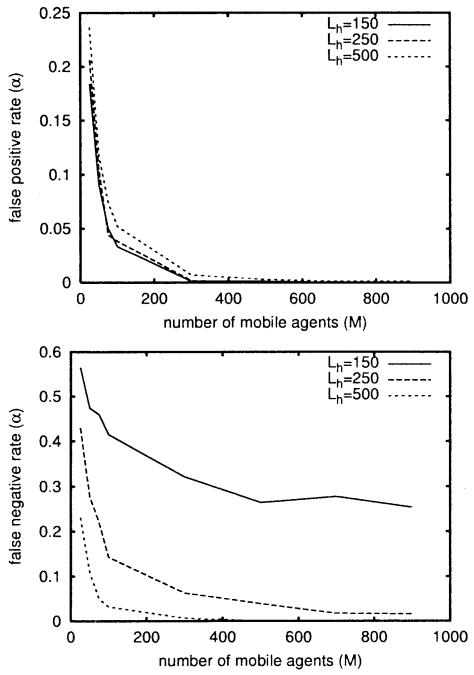


Figure 5: Average false positive/negative rate (α and β) vs. number of mobile agents (M).

(M) and the average number of intrusion detection system's rules (L_h). Figure 5 presents average false positive rate and false negative rate after 30 steps over 50 trials, changing M and L_h. From these results, by the more mobile agents an intrusion detection system is mutually tested, the more precisely its credibility can be calculated. In terms of L_h, the more common rules all intrusion detection systems have, namely, the more homogeneous all intrusion detection systems become, the more easily corrupted ones will be detectable.

These results also demonstrate that the false negative rate β tends to be inferior to the false positive rate α. The reason is probably that some corrupted agent that a corrupted intrusion detection system produces as alter ego at the beginning of simulation can increase the credibility of the parental corrupted intrusion detection system. We conclude that these parameters have important effects on the diagnostic capability.

5 Conclusions and further work

In this paper, we proposed mutual tests between intrusion detection system and mobile agent using the immunity-based diagnosis. The result shows that the credibility of normal intrusion detection system remains stable near 1, otherwise decreases to 0, and then corrupted ones are identified. Furthermore, we confirm that the diagnostic capability depends on both the number of mobile agents and the number of intrusion detection system's rules.

In further work, we will examine the diagnostic capability in more detail and incorporate the immunity-based diagnosis to a real distributed intrusion detection system.

Acknowledgements

This research has been supported in part by International Communications Foundation.

References

[1] B. Mukherjee, L. T. Heberlein, and K. N. Levitt (1994), Network intrusion detection. *IEEE Network*, 8(3), pp. 26–41

[2] Snort.org. http://www.snort.org/.

[3] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff (1996), A sense of self for unix process. In *Proc. of 1996 IEEE Symposium on Security and Privacy*, pp. 120–128

[4] S. Hofmeyr and S. Forrest (2000), Architecture for an artificial immune system. *Evolutionary Computation Journal*, 7(1), pp. 45–68

[5] M. Crosbie and E. Spafford (1995), Defending a computer system using autonomous agents. In *Proc. of the 18th National Information Systems Security Conference*

[6] E. Spafford and D. Zamboni (2000), Intrusion detection using autonomous agents. *Computer Networks*, 34, pp. 547–570

[7] G. Helmer, J. Wong, V. Honavar, and L. Miller (1998), Intelligent agents for intrusion detection. In *Proc. of the IEEE Information Technology Conference*, pp. 121–124

[8] D. Dasgupta (1999), Immunity-based intrusion detection systems: a general framework. In *Proc. of the 22nd National Information Systems Security Conference*, pp. 18–21

[9] Y. Ishida (1990), Fully distributed diagnosis by PDP learning algorithm: towards immune network PDP model. In *Proc. International Joint Conference on Neural Networks*, pp. 777–782

[10] Y. Ishida (1996), An immune network approach to sensor-based diagnosis by self-organization. In *Complex Systems*, 10, pp. 73–90

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Intrusion Detection Algorithm based on Artificial Immune System

Jae-Won Yang, Dong-Wook Lee, and Kwee-Bo Sim

School of Electrical and Electronic Engineering, Chung-Ang University, Korea
221, Huksuk-Dong Dongjak-Ku, Seoul 156-756, Korea
emfvnf@jupiter.cie.cau.ac.kr, kbsim@cau.ac.kr

Abstract

The trial and success of malicious cyber attacks has been increased rapidly with spreading of Internet and the activation of a internet shopping mall and the supply of an online, or an offline internet, so it is expected to make a problem more and more. The goal of intrusion detection is to identify unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators in real time. In fact, the general security system based on Internet couldn't cope with the attack properly, if ever, other regular systems have depended on common vaccine softwares to cope with the attack. But in this paper, we will use the positive selection and negative selection mechanism of T-cell, which is the biologically distributed autonomous system, to develop the self/nonself recognition algorithm and AIS (Artificial Immune System) that is easy to be concrete on the artificial system. For making it come true, we will apply AIS to the network environment, which is a computer security system. This is NID (Network Intrusion Detection) algorithm.

Keywords: positive selection, negative selection, positive detector, negative detector, NIDS, AIS

1. Introduction

The reason which traditional techniques to attack a network system are gradually transformed into new ones is found in the universality of security systems. Also, the struggling process of invaders for overcoming it is the starting point of the change. The attack ability of invaders is superior to the defense one of protectors in the current security models and the protection has the cycle of method that can deal with the existing attack one. Also, that cyber crimes are concrete and organized caused to be an important part of our life by Internet affects the transformation of the traditional attack models much. However the main motive of the pattern transformation is the rise of protector's security level. The universality of firewall and intrusion detection system supplies effective countermeasures to the traditional attack pattern. But technologies and tools for invading a system successfully and overcoming the firewall have been developed continuously for some years [1-3].

In this paper, we propose the algorithm for dealing with SYN flooding attack after modeling positive selection and negative selection using the production principle of T-cell of the biological immune system.

We refer to the method for dealing with SYN flooding attack using positive selection in the Section 2 and using negative selection in the section 3. Finally, we propose the hybrid intrusion detection algorithm using both these selection methods in the section 4.

2. Detection of SYN Flooding Attack by Positive Selection Algorithm

2.1 TCP SYN Flooding Attack

When a client system asks TCP connection for the service that supplies to all client systems of server system, the client and server system exchange a chain of messages each other. The client asks a connection of the server by sending a SYNx message and then, the server recognizes the receipt of SYNx message by sending SYNy and ACKx+1 messages to the client. As the client now sends ACKy+1 message to the server, the TCP connection asked by the client is completed. Fig. 1 shows the 3-way handshake that is a connection technique being applied to all TCP connections. However if the client doesn't do it, the server's status becomes the "Half-Open State" that must wait for arrival of the message. Although capacity of backlog queue at each operating system of the system is various, rapid SYN flooding attacks cause the queue to be full, the service of server results in quit [1].

There are general alternatives for SYN flooding attack. For example, though each OS has a different size of backlog queue, there is a method that controls the size of it to protect the attack. And there is a method that monitors packets inputted into a server system to detect intrusion [1].

In this paper, we made use of the second one. When the packet asking TCP connection is inputted into a server system, we use a packet monitoring method to judge the purpose of packet asked by a client system whether it is intrusion or not after capturing and analysing the packet monitoring the packet inputted by

way of gateway and router. Of course, the packet said above is a virtual one. We used data to simulate a detection algorithm, which is made of integers and randomly produced by us. The environment is also a virtual one, which was composed artificially of simulators.

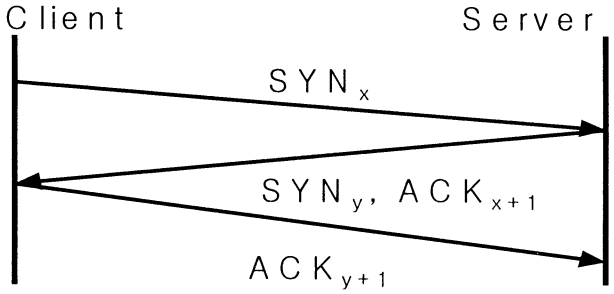


Fig. 1. 3 way-handshake

2.2 Positive Selection Algorithm against SYN Flooding Attack

As said above, the data that is set up as MHC protein out of the packet captured and analysed by using packet-monitoring method is SYN, RST, and sequence number bits [1-3].

After matching RST bit, SYN bit, and sequence number bit of new packet inputted with Positive Detector (PD), if both of them would accord, the data of the new packet is used to update the PD, otherwise discarded.

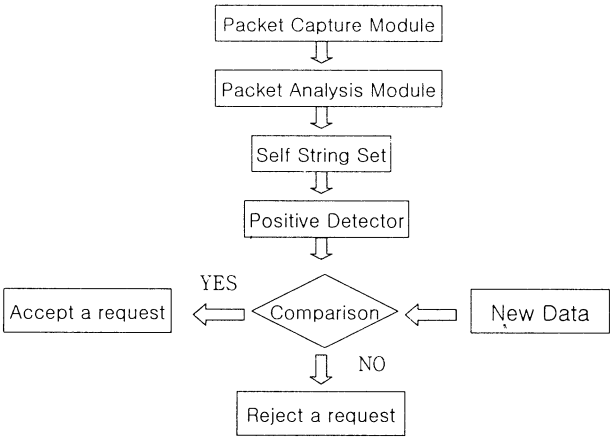


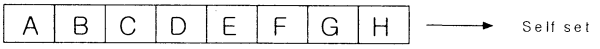
Fig. 2. The flowchart of Positive Detection algorithm

The algorithm is as follows and the fig. 2 shows the flowchart of the algorithm.

1. Capture virtual packets inputted into a server system.
2. Analyse the packets.
3. Set up the data extracted out of a normal packet as self string set.
4. PD based on self-set is initialized by using the positive selection method.
5. PD is matched with the data of the new packet?
   5.1 If yes
   The new data becomes the content of PD
   5.2 else
   Reject the current connection request [4].

2.3 Simulation

Fig. 3 shows the composition of self-set. The elements A, C, D are set up as the MHC protein of biological immune system. They are located in the new packet. Each element of self-set is as follows.



A - Sequence number bits
B - Ack number bit
C - RST bit
D - SYN bit
E - FIN bit
F - Window bit
G - Check sum bits
H - Index bits

Fig. 3. The composition of self-string

We set up the simulation environment. Each number of detectors was 50 and was composed entirely of integers. The packets are virtual ones and mean new data being to be matched with PD, which were artificially made by us, also its number is 50. PD is a detector set, which is composed of data matched with new data1 produced randomly out of self-string. After setting up a system as self-set, only some amount of the self-set is used to match itself with new data. After new data2 being matched with the PD, the new data2 is disposed whether it is self-things or not. At this time, the new data2 mean nonself-things, whose number is 50. That is, the next fig. 4 represents the number of nonself-strings detected by PD.

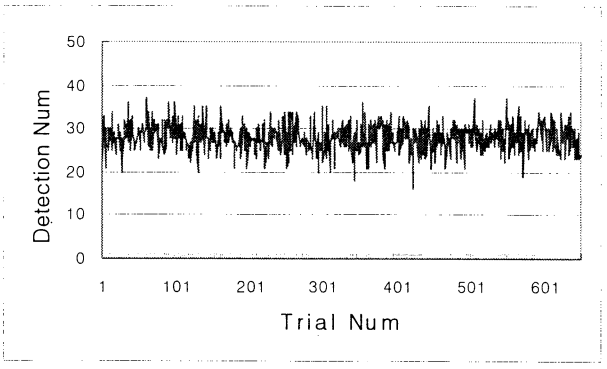


Fig. 4 The result window of Positive Detection

3. Negative Selection Algorithm

Detector strings are produced randomly as similar as T-cytotoxic cells in the thymus. The strings matched with self-strings are eliminated. The only strings not matched with self-strings are used to compose detector set, R. This selection method of composition of R is negative selection. This procedure is repeated until the protection level reaches the demand one. The procedure is that matches S already stored with R_0 produced randomly. And then if the R_0 would be judged as same as the pattern of S, it is rejected, otherwise after accepted, it is used to update the existing R. That is, detector set, R being different from S is set up using the negative selection.

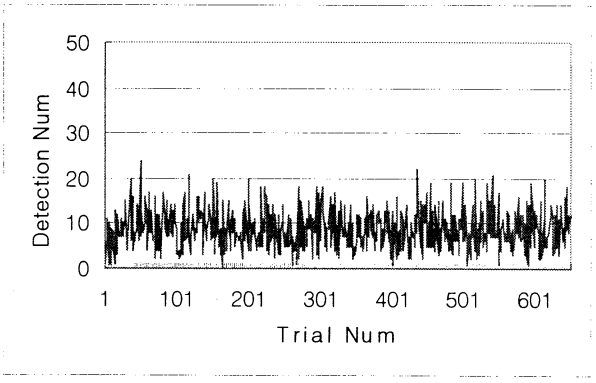


Fig. 5 The result window of Negative Detection

As same as the above positive detection method, the number of random-2 data that should be matched with negative detector (ND) is 50 and that of ND is also 50. The detection result is showed in the fig. 5. It means that the number of nonself-things having been detected by ND is lower than that of PD.

Consequently, this method is used to detect anomolous strings unlike the positive selection that selects by means of specific signatures. Furthermore, It means that the only use of negative detector making use of the negative selection has the limit of protection method against an anomalous intrusion [3][5-6].

4. Hybrid Algorithm of Positive selection and Negative selection

Intrusion detection system is divided into two parts that are misuse detection way and anomaly detection way by means of detection method using the data out of a target for analysis. After misuse detection way extracts specific signatures from existing attack behaviors and examines whether such signature exist or not in the object for analysis, if specific thing exists, intrusion is found out. Therefore, this way should maintain the index of the signature about the existing attack. The detection rate is various according to whether the index is maintained in the late version or not.

The false positive error rate of the misuse detection way is lower than that of the anomaly detection way, but the false negative error rate of the former, which means that protector can't detect attack patterns being not in the index, is higher than that of the latter. On the contrary, the anomaly detection way defines the range of normal behaviors based on existing conditions of network use, regards all behaviors being against these normal ones as abnormal things, and starts to detect the system.

What defines the range of normal behaviors is important and ambiguous in the anomaly detection method. The simplest method to approach the problem is based on the statistical method. With monitoring the condition of network for some times, it is the method that detects the occasion of abnormal condition emergence being relative to such statistics.

In this paper, we propose the algorithm of the fig. 6 to mix the two methods. After passing by the matching rule part, self-data would pass by two selection processes.

The algorithm is as follows.

1. Make Self strings
2. Make PD and ND
3. Matching PD with new data?
   - 3.1 If yes
     The new data is sent to the next step
   - 3.2 If no
     Reject the request
4. Matching the ND with the new data passed above?
   - 4.1 If yes
     Reject the request
   - 4.2 If no
     Goto the next step

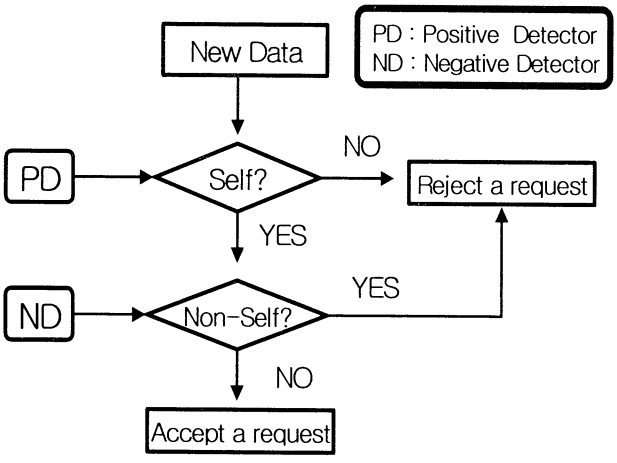


Fig. 6 The flowchart of Hybrid NIDS algorithm

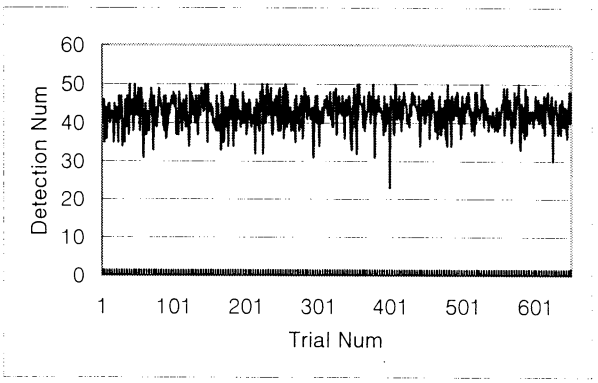


Fig. 7 The result windows of Hybrid Detection

For matching positive detectors and negative detectors with new data, we produced detectors and the data, each 25, 25, and 50. The PD and ND were produced by the method that was said as above. And the new data were produced randomly only for simulation. The matching result is shows in the fig. 7. Fig. 7 shows the detected numbers of nonself-thing. Consequently, the detecting ability of hybrid method against nonself-things is superior to another ones. We could find out that the Hybrid algorithm contributes the confidence increase of detection.

5. Conclusion

Biological immune system (BIS) is an autonomous, distributed one structurally. Especially, each independent cell protects a living body from pathogens or external materials using mutual communication and cooperation organically and operates the second immune response to detect anomalous things using learning cells and memory cells.

In this paper, we proposed the algorithm that deals with the intrusion trial in the computer environment as modeling immune cells of BIS. The algorithm using the production process of T-cell, i.e. positive selection and negative selection, and hybrid selection, improves the confidence of intrusion detection because it should pass by a recognition process for normal access and a double recognition process for abnormal intrusion.

Also, we simulated each algorithm which had modeled on the above three methods of the production process of immune cells. The best detection is a hybrid one, the second one is a positive thing, and the last one is a negative thing. Through this simulation result, the hybrid detection is good at detecting a virtual SYN flooding attack. This suggests the possibility that the hybrid detection can detect another DDoS attacks.

Acknowledgments

This research was supported by the spin-off program from Ministry of Commerce, Industry and Energy of Korea.

References

[1] Computer Emergency Response Team, "TCP SYN Flooding and IP Spoofing Attacks," *CERT Advisory: CA*, pp. 96-21, 1996.

[2] P.D' haeseleer, S, Forrest, and P. Helman. "An immunological approach to change detection: Algorithms, analysis and implication," *Proceeding of the 1996 IEEE Symposium on Research in Security and Privacy*, Los Alami. 1996.

[3] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principles of a Computer Immune System," *New Security Paradigms Workshop*, pp. 75-82, 1998.

[4] W. Stevens, *TCP/IP Illustrated*, vol. 1, Addison Wesley Publishing, Company, 1994.

[5] J. B. Gu, D. W. Lee, K. B. Sim, and S. H. Park, "An Immunity-based Security Layer against Internet Antigens," *Transactions on IEICE*, vol. E83-B, no.11, pp. 2570-2575, 2000.

[6] D. Dasgupta, and S. Forrest, "An Anomaly Detection Algorithm Inspired by the Immune System," *Artificial Immune Systems and Their Applications*, pp. 262-276, 1999.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A Mathematical Analysis for Effectiveness of a Honeypot against Internet Worms

Takeshi Okamoto
Department of Information Network Engineering
Kanagawa Institute of Technology
1030, Shimo-ogino, Atsugi, Kanagawa 243-0292, Japan.
take4@nw.kanagawa-it.ac.jp

Abstract

A honeypot can be used as a deception system to capture worms. Actually, network administrators have been using some of honeypots such as LaBrea. In this paper, we analyze the effectiveness of the honeypot against worms by modeling the spread of a worm in the network with the honeypot mathematically. In addition, we show the condition under which epidemics are unlikely to occur.

Keywords: Internet worm, population dynamics, honeypot, epidemics

1 Introduction

An Internet worm spreads on the Internet autonomously as if it were alive. The worm is different from a computer virus, because the virus needs a user to run an infected program for spreading, but the worm does not need him/her. The first worm on Internet is Morris Internet worm[1], which began to spread in 1988 by exploiting a vulnerability of `sendmail` program. Incidents of Code-Red[2] and Nimda[3] exploiting vulnerabilities (e.g.[4, 5, 6]) of Microsoft® Internet Information Service (IIS) web server software are fresh in our memory. A common feature of those worms is exploitation of vulnerabilities in widely used services on Internet.

Since the incidents of Code-Red and Nimda, many network administrators have installed an anti-virus software or simply configured the access list of IP packets (i.e. ingress filtering and egress filtering[7]) to filter out those worms. These methods have effect on protection of a local network, but are difficult to reduce the ability of those worms to spread. Another method is the use of a honeypot, which is a kind of security tools[8]. Some honeypots can capture specific activities, such as worms and scanning activities. LaBrea[9], which is one of honeypots, not only captures worms, but also slows or disables worm attack. LaBrea monitors external attempts to access all the unused IP addresses within a local network. Detecting a attempt, LaBrea establishes a connection as if the IP address was used (the computer was on-line), and it holds the connection until either the worm or LaBrea is shutdown. At the time, the worm cannot spread anywhere. In this way, LaBrea slows down the spread of worms, and makes a dent in the spread. Therefore, it would be considered that epidemics have not occurred if many network administrators have used the honeypot. However, there have been few efforts to analyze the effectiveness of the honeypot against worms theoretically.

In this paper, we analyze the effectiveness of the honeypot against worms by modeling the spread of a worm in the network with the honeypot. In Section 2, we model the spread of a simple worm based on Code-Red I worm mathematically, and show that our solution matches actual observed data[10] . In Section 3, we model the spread of the worm in the network with the honeypot, using previous epidemiological models (e.g.[11, 12]), and analyze the effectiveness of the honeypot. Moreover, we show the condition under which epidemics are unlikely to occur. In Section 4, we summarize the effectiveness and future works.

2 Modeling the spread of a worm

We model a simple worm such as Code-Red I, which is well investigated by many organizations (e.g.[2]). Note that Code-Red I, which began to spread on July 19, 2001, is different from Code-Red II, which began to spread on August 4, 2001.

Based on Code-Red I, we assume that a worm attempts to infect a randomly chosen computer within all the computers, and the attempt succeeds only if the computer is vulnerable, but it is not infected. After infecting, the worm creates multiple threads to spread to other computers. For example, Code-Red I creates 99 threads on the infected computer. For the sake of simplicity, we assume that the rate of threads at which an infected computer can process per unit time is the same constant a for all the computers, ignoring network throughput and CPU load. Hence, the assumption would not be valid for the large a. As well as epidemiological models, we refer to the rate of threads as the infection rate.

In order to evaluate the effectiveness using the honeypot without human factors (patching a security hole, installing a vulnerable program or turning off a computer, etc.), we assume that an infected computer is not cured forever. That means this model belongs to the SI model ($Susceptible \rightarrow Infected$)[12].

We assume that a network consists of susceptible nodes, infected nodes and immune nodes. The susceptible nodes are vulnerable computers, which can

be infected with worms. The infected nodes are infected computers with worms. The immune nodes are invulnerable computers, which are immunized against worms, or unused (off-line) computers. The fraction of vulnerable computers in all the computers is a constant b, ignoring both the decrease by patching a security hole and turning off a computer temporally, and the increase by installing such a vulnerable program.

Let $X(t)$ denote the total sum of worms in all the infected computers at time t. Note that the $X(t)$ is different from the number of infected computers. Let N denote the number of all the computers. If aN is sufficiently large, stochastic effects can be ignored, and the problem can be analyzed deterministically. In this case, we can convert $X(t)$ to $x(t) \equiv \frac{X(t)}{aN}$, a continuous quantity representing the fraction of infected computers in all the computers.

Consider a particular worm on an infected computer. The fraction of susceptible computers at time t is $b-x$, so the expected number of worms which can infect other susceptible computers is $X(b-x)$. Since the infected computer creates multiple threads after the infection, the expected number of newly active worms is $aX(b-x)$. Hence, the differential equation can be expressed as

$$\frac{dx}{dt} = ax(b-x). \tag{1}$$

Eq. 1 is the logistic growth equation. The solution to Eq. 1 is

$$x(t) = \frac{x_0 b}{x_0 + (b-x_0)e^{-abt}}, \tag{2}$$

where $x_0 \equiv x(0) < b$ is the initial fraction of infected computers in all the computers. Figure 1 is a typical example of dynamics of the fraction $x(t)$ in Eq. 1. Staniford et al. have proposed the following differential equation[10]:

$$\frac{da}{dt} = Ka(1-a). \tag{3}$$

They showed that their solution matched observed data in the beginning phase of the spreading of Code-Red I[1] . Substituting $a(t) \equiv \frac{x(t)}{b}$ and $K \equiv ab$ into Eq. 3, Eq. 1 can be obtained. Therefore, the solution of Eq. 1 matches the observed data in the beginning phase of the spreading as well as their solution.

3 Analysis for effectiveness of a honeypot

Inspired by LaBrea, we assume a following simple honeypot:

- The honeypot monitors only one IP address such as LaBrea@Home[13] for home use, although LaBrea can monitor all the unused IP addresses within a local network.
- The honeypot can permanently capture connections from worms, ignoring both network throughput and CPU load, although LaBrea requires approximately 1215 bytes per hour for keeping a connection.
- The computer with the honeypot is not vulnerable.

Let $Y(t)$ denote the number of worms captured by the honeypot in all the computers at time t. As well as $X(t)$, we convert $Y(t)$ to $y(t) \equiv \frac{Y(t)}{aN}$, a continuous quantity representing the fraction of infected computers with captured worms in all the computers. Let c denote the fraction of computers with the honeypot in all the computers. The interval c is $c \leq 1-b$ because the computers with the honeypot are not vulnerable.

Now consider a particular worm on a infected computer. Since the total fraction of infected computers at time t is $x+y$, the fraction of susceptible computers is $b-x-y$. Then, the expected number of worms which can infect other susceptible computers is $X(b-x-y)$, so the expected number of newly active worms is expressed as $aX(b-x-y)$, whereas the expected number of captured worms is cX. Hence, a pair of nonlinear differential equations can be expressed as

$$\begin{cases} \dfrac{dx}{dt} = ax(b-x-y) - cx, \\ \dfrac{dy}{dt} = cx. \end{cases} \tag{4}$$

At $c = 0$, the first equation in Eq. 4 corresponds to Eq. 1, because $y(t) = 0$ for all t. Hence, we consider the model for $c > 0$. The both fractions $x(t)$ and $y(t)$ in Eq. 4 can be solved numerically. Figure 1 shows the dynamics of fractions $x(t)$ and $y(t)$ given by Eq. 4, and $x(t)$ given by Eq. 1 for comparison. After a short-lived growth spurt, the fraction $x(t)$ in Eq. 4 decays exponentially, and eventually approaches 0.0. The sum of fractions $x(t)$ and $y(t)$ in Eq. 4 is less than $x(t)$ in Eq. 1 for all t. In these parameters, the honeypot is effective to make a dent in the spread of worms.

Consider the maximum of $x(t)$ and $y(t)$ in Eq. 4. Let x_{max} and y_{max} denote the maximum of $x(t)$ and $y(t)$ for $t > 0$, respectively. Dividing the first equation in Eq. 4 by the second equation in Eq. 4 yields the following differential equation:

$$\frac{dx}{dy} = \frac{1}{c'}(b-x-y) - 1, \tag{5}$$

where $c' \equiv \frac{c}{a}$ is the effective capture rate. Assuming $a \geq 1$, we have $0 < c' \leq 1-b$.

The solution of Eq. 5 with the initial conditions $x(t=0) = x_0$ and $y(t=0) = 0$ is

$$x = (x_0 - b)e^{-\frac{1}{c'}y} - y + b. \tag{6}$$

[1] However, their solution did not correspond to the observed data after the worm population reaches a peak. This difference would be attributable to the lack of modeling network congestion and overload of CPU, etc.

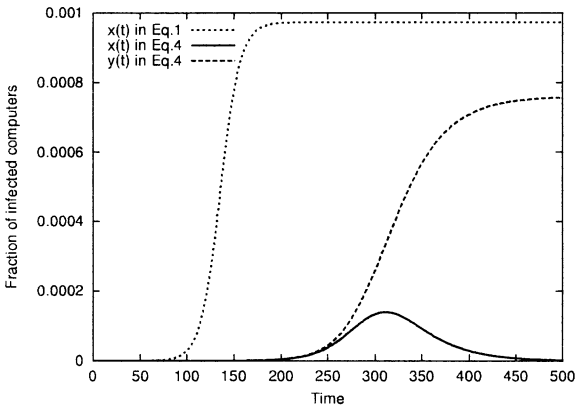


Figure 1: Dynamics of fractions $x(t)$ and $y(t)$ for $a = 100.0, b = 4176048/2^{32}$, $c = 0.05$ and $x_0 = 1/2^{32}$, where 4,176,048 is the number of sites using IIS web server software on Internet, which was estimated by Netcraft in July, 2001[14]. The curves of $x(t)$ and $y(t)$ in Eq. 4 are the numerical solution of Eq. 4 given by *MATHEMATICA®*. The curve of $x(t)$ in Eq. 1 is the analytical solution of Eq. 1 for comparison.

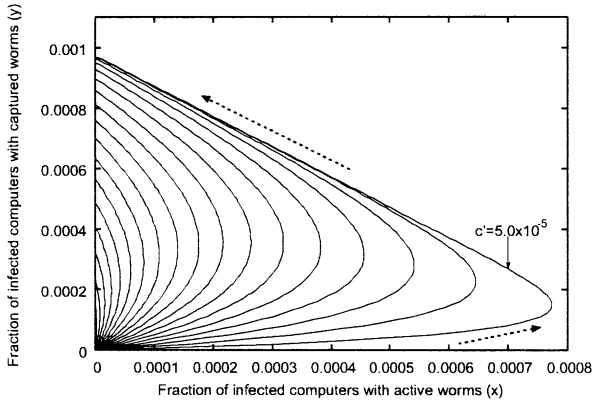


Figure 2: Trajectories given by Eq. 6. All the trajectories are obtained by varying c' from 5.0×10^{-5} to 1.1×10^{-3} every 5.0×10^{-5} from the right trajectory. Other parameters are $x_0 = 1/2^{32}$ and $b = 4176048/2^{32}$. The dashed line arrows indicates direction of time flow.

Figure 2 shows trajectories given by Eq. 6. The fraction x reaches a peak, and then decreases to 0.0 for all trajectories. In short, all worms are eventually captured by honeypots.

Since x_{max} corresponds to x^* at the peak point (x^*, y^*) of trajectories as shown in Fig. 2, the derivative of Eq. 6 with respect to y at the peak point equals to 0 as follows:

$$\frac{dx(y^*)}{dy} = -\frac{1}{c'}(x_0 - b)e^{-\frac{1}{c'}y^*} - 1 = 0. \quad (7)$$

The solution of Eq. 7 is

$$y^* = -c' \log \frac{c'}{b - x_0}. \quad (8)$$

Since $y(t) > 0$ for $t > 0$, we must satisfy $c' < b - x_0$ in Eq. 8 for $b < \frac{1-x_0}{2}$. Substituting Eq. 8 into Eq. 6, we obtain

$$x_{max} \equiv x^* = c' \log \frac{c'}{b - x_0} + b - c', \quad (9)$$

where c' must satisfy the condition $c' < b - x_0$ for $b < \frac{1-x_0}{2}$.

Now, let x_{max} be a function of b and c'. Figure 3 shows the fraction x_{max} as a function of c'. As shown in Fig. 3, the fraction x_{max} decreases with increasing c' for all curves. Notice that x_{max} approaches x_0 as $c' \to b - x_0$ for $b < \frac{1-x_0}{2}$, which implies that the initial worm is immediately captured before it infects with other computers.

The fraction y_{max} exists on the vertical axis as shown in Fig. 2. Substituting $x = 0$ into Eq. 6, we obtain

$$y_{max} = b + c' W(\frac{1}{c'}(x_0 - b)e^{-\frac{b}{c'}}), \quad (10)$$

where the function W is Lambert function which satisfies $z = W(z)e^{W(z)}$, and can be solved numerically. Let y_{max} be a function of b and c' as well as x_{max}. Figure 4 shows the fraction y_{max} as a function of c'. As shown in Fig. 4, the fraction y_{max} decreases with increasing c' for all curves. For $b \leq 0.5$, the fraction y_{max} drops suddenly in the vicinity of $c' = b$, and then approaches x_0 as $c' \to \infty$ as shown in the small inset of Fig. 4. Figure 3 and Figure 4 show that epidemics are unlikely to occur if approximately $c' > b$ for $b \leq 0.5$ at a rough estimate. Assuming $a > 0$, epidemics are unlikely to occur under the condition $c' > b$ for all b.

Consider the effective capture rate c' at which epidemics do not occur. Although we have indicated that the condition is approximately $c' > b$, the lower bound of c' could decrease less than b because LaBrea can cover all the unused IP addresses within a local network. However, the effective capture rate c' decreases with increasing the infection rate a, because c' is dependent on a. Worm authors can easily increase the infection rate a by increasing the number of threads. In fact, the number of threads has increased from 99 to 300 or 600, comparing Code-Red I with Code-Red II. At the present moment, the infection rate would be fortunately suppressed owing to the limit of network throughput and CPU load, even if worm authors increased the number of threads. Today, network throughput and CPU performance have improved considerably. Hence, we might have to run more honeypots than ever for keeping the effective capture rate in the future.

4 Summary

In this paper, a mathematical model for the spread of a worm in the network with a honeypot has been developed, and the result has indicated the condition un-

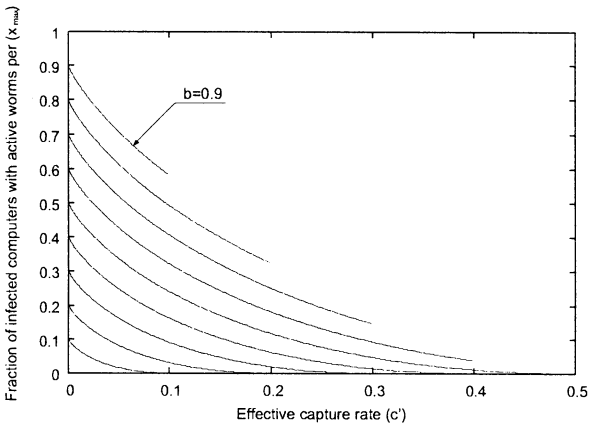


Figure 3: Fraction x_{max} as a function of c' for $x_0 = 1/2^{32}$. For $b < \frac{1-x_0}{2}$, the interval c' is $0 < c' < b - x_0$. For $b \geq \frac{1-x_0}{2}$, $0 < c' \leq 1 - b$ because of $c \leq 1 - b$ and $a \geq 1$. All the curves are obtained by varying b from 0.1 to 0.9 every 0.1 from the lowest curve.

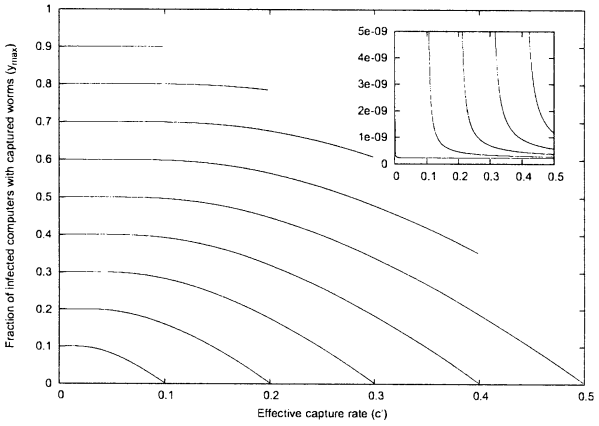


Figure 4: Fraction y_{max} as a function of c' for $x_0 = 1/2^{32}$. The interval c' is $0 < c' \leq 1-b$ because of $a \geq 1$. All the curves are obtained by varying b from 0.1 to 0.9 every 0.1 from the lowest curve. The small inset displays scaling up the range from 0.0 to 5.0×10^{-9}.

der which epidemics are unlikely to occur. At a rough estimate, epidemics are unlikely to occur if the effective capture rate c' exceeds approximately the fraction b of vulnerable computers.

We have assumed the infection rate is constant. However, the infection rate probably depends on worm population size because there is a limit to network throughput and CPU load. In order to determine the more accurate condition under which epidemics do not occur, it is necessary to analyze a model dealing with network congestion and overload of CPU.

References

[1] E. H. Spafford(1989), The Internet worm: crisis and aftermath, Communications of the ACM 32, pp. 678–687.

[2] Cooperative Association for Internet Data Analysis(2001), CAIDA analysis of Code-Red. Code-Red worms: a global threat. http://www.caida.org/analysis/security/code-red/

[3] A. Mackie, J. Roculan, R. Russell et al.(2001), Nimda worm analysis, http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf

[4] SecurityFocus(2000), Microsoft IIS and PWS extended unicode directory traversal vulnerability. http://online.securityfocus.com/bid/1806

[5] SecurityFocus(2001), Microsoft IIS/PWS escaped characters decoding command execution vulnerability. http://online.securityfocus.com/bid/2708

[6] SecurityFocus(2001), MS Index Server and Indexing Service ISAPI extension buffer overflow vulnerability. http://online.securityfocus.com/bid/2880

[7] P. Ferguson and D. Senie(2000), Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing, RFC 2827.

[8] L. Spitzner, M. Roesch and D. Dittrich(2002), Honeypots: definitions and value of honeypots. http://www.spitzner.net/honeypot.html

[9] T. Liston(2001), Welcome to my tarpit: the tactical and strategic use of LaBrea. http://www.hackbusters.net/LaBrea/LaBrea.txt

[10] S. Staniford, V. Paxson and N. Weaver(2002), How to own the Internet in your spare time, Proceedings of the 11th Usenix Security Symposium. http://www.icir.org/vern/papers/cdc-usenix-sec02/cdc.pdf

[11] R. M. Anderson and R. M. May(1991), Infectious diseases of humans: dynamics and control, *Oxford University Press.*

[12] J. C. Frauenthal(1980), Mathematical modeling in epidemiology, *Springer-Verlag.*

[13] LaBrea@Home. http://www.hackbusters.net/LaBrea/lbathome.html

[14] Netcraft(2001), Netcraft web server survey. http://www.netcraft.com/Survey/index-200106.html

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

A Study on the position control of an SMA actuator using time delay control

H. J. Lee
Dept. of Mechanical Engineering
Korea Advanced Institute of Science and Technology, Taejon, Korea 305-701
hjlee@imhp.kaist.ac.kr

J. J. Lee
Dept. of Mechanical Engineering
Korea Advanced Institute of Science and Technology, Taejon, Korea 305-701
jjlee@mail.kaist.ac.kr

Abstract

Shape Memory Alloy (SMA) actuators have hard nonlinearities including backlash-like hysteresis and saturation. These nonlinearities result in steady-state error and limit cycle problem when conventional controllers such as proportional integral derivative (PID) are used for trajectory control. In this study, the SMA actuator dynamics of the type of rotary motion is derived and thereafter time delay control (TDC) is applied to the SMA actuator. As well, PID control based on the TDC is applied to the SMA actuator. The gain tuning method is explained concretely, and TDC results are compared with PID control based on the TDC and PID tuned by Ziegler-Nichols method.

1. Introduction

Great attention has been focused on SMAs in recent years, especially in the development of innovative engineering systems such as smart materials, active absorbers, micro actuators, etc. Compared with other alloys, SMAs have unique characteristics such as pseudo-elasticity and shape memory effects (SME), so consequently many application areas have arisen since the discovery of SMAs. Pseudo-elasticity has been successfully applied to medical and non-medical applications; for example, dental arches, catheter guide wires, eyeglass frames, PCS antennas, brassier wires, etc. In comparison with pseudo elasticity, the SME application area has been more or less limited. SME has been applied to vascular stents, pipe couplings or fasteners, headings of thermostats, valves, and actuators of micro robots; however, applications to precise engineering systems are somewhat problematic due to the highly nonlinear dynamic behavior of this alloy.

Examples of hard nonlinearities include backlash-like hysteresis and saturation due to SME making precise control difficult. Thus, numerous approaches for SMA actuator control have been proposed, but some problems remain. Ordinary PID control schemes often show steady-state error and limit cycle problem. Majima *et al.* [1] proposed a control system composed of a PID feedback loop and a feedforward loop. They showed that the limit cycle oscillations were greatly rejected and the tracking control performance was improved. Grant and Hayward [2] applied a variable structure control scheme to a pair of antagonist actuators and realized smooth and robust control. Kumagai *et al.* [3] applied neuro-fuzzy logic based control to SMA actuators, but tracking was not so accurate.

Hasegawa and Majima [4] tried to compensate for the hysteresis of SMAs using an inverse hysteresis model for the SMA plant; however, a generalized model that explains the behavior of SMA well is hard to obtain, and perfect compensation is also hard to achieve even if a generalized model is obtained. On the contrary, TDC does not require a mathematical model and also has robustness against system parameter variations and disturbances. The key concept of TDC is the use of time delay information to estimate the total plant nonlinearities.

TDC has been applied in many important real plants such as a DC servo motor by Chang and Lee [5], robot systems by Hsia and Gao [6], a magnetic bearing by Youcef-Toumi and Reddy [7], an electro hydraulic servo-motor by Chin *et al.* [8] In these applications, TDC provided highly satisfactory results even under large system parameter variations and disturbances.

2. Dynamics of an SMA actuator

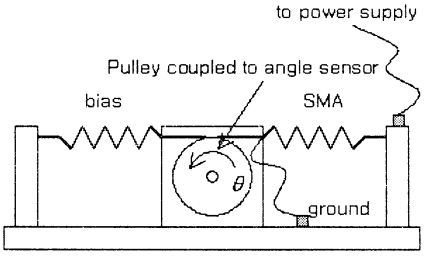


Fig. 1 SMA actuator of the rotary motion type

SMA actuators can be classified into two types of configurations on the basis of restoration force: bias or differential types. In a bias type actuator, the SMA actuator can be shrunk by thermal excitation and can be restored to its original shape by a bias spring force. A differential type actuator consists of two SMA actuators, and restoration can occur only by thermal excitation of the other SMA actuator. On the other hand, SMA actuators can be also classified by two types of configurations on the basis of motion: linear translational and rotary motion types.

We consider the dynamics of the rotary motion type actuator as shown in Fig. 1. Here, θ denotes the rotary angle of moment of inertia, J; R is the radius of the pulley coupled with the angle sensor; ξ is the martensite fraction. Since thermal term is relatively small in comparison to rotary angle and martensite fraction terms, the SMA constitutive equation can be expressed as follows:

$$\dot{P} = AR\dot{\theta} + B\dot{\xi} \tag{1}$$

where P is the tensile load, which SMA has developed and coefficients, A and B are determined by Majima *et al.* [1] The load of bias spring is just a function of rotary angle, θ and can be expressed as follows:

$$\dot{P}_b = -KR\dot{\theta} \tag{2}$$

where K is the stiffness of the bias spring. Considering equilibrium of SMA actuator of the rotary motion type, the following equation can be obtained:

$$J\ddot{\theta} + \rho\ddot{\theta} + R^2(K + A)\dot{\theta} = -RB\dot{\xi} \tag{3}$$

An empirically based cosine model of the transformation kinetics has been developed by Liang and Rogers [10] as follows:

$$\xi = \frac{1-\xi_0}{2}\cos\left[a_M(T - M_f) + b_M\sigma\right] + \frac{1+\xi_0}{2} \quad \text{if A} \rightarrow \text{M}$$
$$\xi = \frac{\xi_0}{2}\left[\cos\left\{a_A(T - A_s) + b_A\sigma\right\} + 1\right] \quad \text{if M} \rightarrow A \tag{4}$$

where ξ_0 means the martensite fraction at transformation starting point. Coefficients a_M, b_M, a_A and b_A are defined by Liang and Rogers [9] using four characteristic temperatures M_f, M_s, A_s and A_f, which are martensite finish, martensite start, austenite start and austenite finish temperatures under stress free conditions, respectively. Four characteristic temperatures, which are listed in Table I, can be obtained by differential scanning calorimetry (DSC). From the simple expression of kinetics, the martensite fraction can be written as follows

$$\dot{\xi} = C\dot{T} + D\dot{P} = \frac{1}{1 - BD}(C\dot{T} + ADR\dot{\theta}) \tag{5}$$

If the above equation is substituted for (3), dynamics can be expressed as follows:

$$J\ddot{\theta} + \rho\ddot{\theta} + R^2(K + A + \frac{ABD}{1 - BD})\dot{\theta} = -R\frac{BC}{1 - BD}\dot{T} \tag{6}$$

Since C and D are dependent on the phase transformation state, (6) can be integrated as follows:

$$J(\ddot{\theta} - \ddot{\theta}_i) + \rho(\dot{\theta} - \dot{\theta}_i) + R^2(K + A + \frac{ABD}{1 - BD})(\theta - \theta_i) = -R\frac{BC}{1 - BD}(T - T_i) \tag{7}$$

where subscript "i" represents the starting point of the phase transformation state, and thus it must be updated whenever the phase transformation state changes. Here, coefficients of each term are assumed to be constant for integration. Coefficients, C and D are expressed according to phase transformation state as follows:

If A → M

$$C = -\frac{1-\xi_0}{2}\sin\left[a_M\left(T - M_f\right) + b_M\sigma\right]a_M,$$
$$D = -\frac{1-\xi_0}{2S}\sin\left[a_M\left(T - M_f\right) + b_M\sigma\right]b_M$$

If M → A

$$C = -\frac{\xi_0}{2}\sin\left[a_A\left(T - A_s\right) + b_A\sigma\right]a_A,$$
$$D = -\frac{\xi_0}{2S}\sin\left[a_A\left(T - A_s\right) + b_A\sigma\right]b_A \tag{8}$$

If no phase transformation

$$C = D = 0$$

The above equation has been driven by Lee and Lee [10], and S is the equivalent sectional area of SMA; so, for the coil spring type actuator, the inverse of S is equal to $8\sqrt{3}D/\pi d^3$. One reason why (7) is divided into three equations in the form of (8) is TDC theory begins on the assumption that plant dynamics and gain matrix are uniformly continuous [11]. The uniform continuity of C and D guarantees that plant dynamics and gain matrix are uniformly continuous in (8). The final equation of the rotary type actuator can be expressed in the companion form as follows:

$$\begin{pmatrix}\Delta\dot{\theta}\\ \Delta\ddot{\theta}\end{pmatrix} = \begin{pmatrix}0 & 1\\ -\frac{R^2}{J}(K + A + \frac{ABD}{1 - BD}) & -\frac{\rho}{J}\end{pmatrix}\begin{pmatrix}\Delta\theta\\ \Delta\dot{\theta}\end{pmatrix} + \begin{pmatrix}0\\ -\frac{R}{J}\frac{BC}{1 - BD}\end{pmatrix}\Delta T \tag{9}$$

where $\Delta\theta, \Delta T$ are $\theta - \theta_i, T\text{-}T_i$ and R is the radius of pulley in Fig. 1.

3. TDC application to an SMA actuator

Control input of TDC in a SISO system is expressed as follows:

$$u(t) = u(t - L) + \hat{b}^+[\Delta\ddot{\theta}_d - \Delta\ddot{\theta}(t - L) + 2\zeta\omega_n(\Delta\dot{\theta}_d - \Delta\dot{\theta}) + \omega_n^2(\Delta\theta_d - \Delta\theta)] \tag{10}$$

where $\Delta\theta_d$ is desired trajectory; $\hat{b}^+$, ω_n and ζ are TDC gains to be selected; Constant scalar, $\hat{b}^+$ is a nonzero element of a pseudo-inverse of $\hat{\mathbf{b}}$ which is a constant nominal matrix of $\mathbf{b}(\Delta\boldsymbol{\theta}) = (0 \quad -\frac{R}{J}\frac{BC}{1 - BD})^T$; L a sufficiently small time delay. By Youcef-Toumi and Wu [11], TDC is stable only if $\hat{b}$ is chosen such that

$$\left|b_{(k)}\hat{b}^+ - 1\right| \le \alpha < 1 \text{ for } k > N \ge 1 \tag{11}$$

where (k) denotes the time at the kth sampling instant,

i.e., $t = kL$.

Table I. Specification of the selected SMA and the bias spring

Ni-Ti SMA coil spring		
Young's modulus	Austenite	Approx. 83 *GPa*
	Martensite	Approx. 28 *GPa*
Poisson's ratio		0.33
Coil diameter		1.58 *mm*
Wire diameter		0.38 *mm*
Number of turns		30
Transformation Temperature (from DSC)	M_f	33 ^oC
	M_s	49 ^oC
	A_s	44 ^oC
	A_f	60 ^oC
C_M		Approx. 7 $MPa\cdot{}^oC^{-1}$
C_A		Approx. 10 $MPa\cdot{}^oC^{-1}$
Bias coil spring		
Young's modulus		200 *GPa*
Coil diameter		6 *mm*
Wire diameter		0.6 *mm*
Number of turns		40

From (9), if we know A, B, C and D then input gain, $b_{(k)}$ and natural frequency of the SMA actuator, ω_n can be found out. Thus, the selection of $\hat{b}$ and ω_n among three TDC gains becomes very easy. Using properties and geometric data of the SMA actuator in Table I, the bounds of A are $0.23 \le A \le 0.69 (N/mm)$. The bounds of B are $-6.9 \le B \le -2.3\ (N)$ where x_L is about 10mm, which is the maximum stroke SMA spring can have. Roughly estimating, the bounds of C and D are $-0.10 \le C \le 0\ (^oC^{-1})$ and $0 \le D \le 1.83\ (N^{-1})$. In (9), $-\frac{R^2}{J}(K + A + \frac{ABD}{1-BD})$ means ω_n^2. From the above results, we know that natural frequency of the SMA actuator is within the following bounds:

$$0 \le \omega_n^2 \le 8.34 \times 10^3 \frac{R^2}{J} \tag{12}$$

From the stability condition, $\hat{b}$ must be selected within the following bounds:

$$\hat{b} \le \frac{b_{(k)}}{2} = -0.35\frac{R}{J} \tag{13}$$

Finally, the damping ratio, ζ can be determined for overshoot not to occur. TDC gains are much more meaningful and intuitive than PID gains. PID gains are difficult to select for a closed loop system to have good performance, and so many tuning methods have been developed. Ziegler-Nichols tuning rules have been widely used to tune PID controllers in control systems where the plant dynamics are not precisely known. Consequently, these rules can be a good solution of SMA actuators. Among the list of PID tuning methods, the systematic gain tuning method that utilizes PID control based on TDC proposed by J. Park and P. H. Jang [12] is worthy of notice. In the experimental verification, above remarked three control methods are compared.

4. Experimental verifications

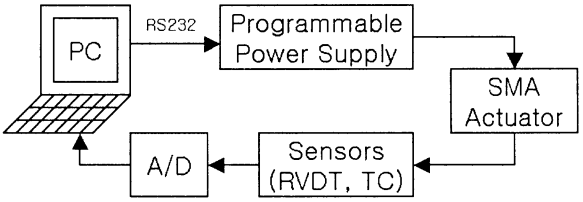


Fig. 2 Experimental setup: Main components

For the verifications of the above TDC, we conducted experiments with the rotary motion and bias-type SMA actuators as shown in Fig. 1. The range of the rotary angle of the SMA actuator is from 93.4° (when the SMA coil spring relaxes maximally) to 58.8° (when the SMA coil spring shrinks maximally). A block diagram of the main experimental components is shown in Fig. 2.

The above-mentioned TDC scheme and PID control scheme based on TDC were implemented by LabVIEW programming. All the control algorithms ran with a sampling frequency of 20Hz and included an anti-windup scheme to prevent abrupt controller output changes.

Table II. Gains tuned by Ziegler-Nichols first method and TDC gains

Controllers	Gains
Ziegler-Nichols	K_p=0.247, T_i=5.45
TDC	*bhat*=−10000, ω_n=100, *zeta*=50

First, PID control was conducted using the first Ziegler-Nichols tuning method, which is based on a decay ratio of approximately 0.25; a PI controller is used in the study and the gains are shown in Table 2. PID control tuned by the Ziegler-Nichols method shows very slow step response and steady-state error; however, TDC and PID control based on TDC show a rapid rise time and no steady-state error as shown in Fig. 3. Their results show nearly similar performance, but PID control based on TDC is a little better than TDC. This is because of noise due to numerical differentiation during the calculation of the TDC control input.

Fig. 4 shows experimental results according to payload variations. As the payload increases, the tracking accuracy decreases in the case of PID control tuned by the Ziegler-Nichols method, but does not become worse in other cases. This experiment proves that TDC and PID control based on TDC are robust to payload variations.

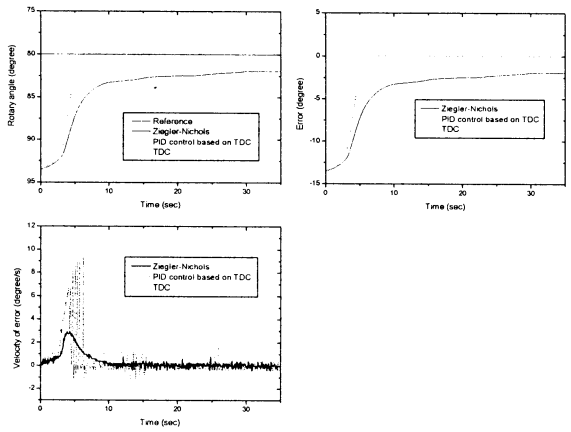


Fig. 3 Experimental results when step input is applied (without payload variation). Three controllers are compared: PID tuned by Ziegler-Nichols method, PID control based on TDC and TDC.

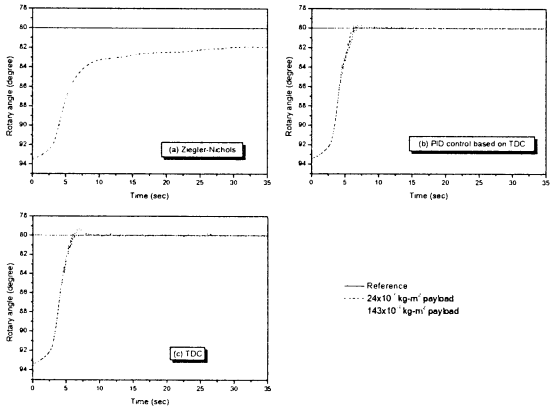


Fig. 4 Experimental results according to payload variations: $24\text{x}10^{-3}$ and $143\text{x}10^{-3}$ $kg \cdot m^2$ (a) PID control tuned by Ziegler-Nichols method when step input is applied. (b) PID control based on TDC when step input is applied. (c) TDC when step input is applied.

5. Summary

In this paper, we derived the SMA' dynamics applicable to TDC. The dynamics was divided into three parts according to phase transformation state, and that was implemented numerically. TDC gain bounds satisfying stability were determined by experimental information of the selected SMA. Systematic gain tuning of PID control based on TDC and PID control tuned by the Ziegler-Nichols method were also applied to the SMA actuator. In experiments, TDC and PID control based on TDC exhibited high tracking accuracy and robustness to moment inertia variations in contrast with PID control tuned by the Ziegler-Nichols method.

Acknowledgements

This study was supported by the HWRS-ERC project of KOSEF (Korea Science and Engineering Foundation). We would like to express our gratitude for this support.

References

[1] S. Majima, K. Kodama and T. Hasegawa, "Modeling of Shape Memory Alloy Actuator and Tracking Control System with the Model," *IEEE T. Contr. Syst. T.*, Vol. 9, No. 1, pp. 54-59, 2001.

[2] D. Grant and V. Hayward, "Variable Structure Control of Shape Memory Alloy Actuators," *IEEE Contr. Syst. Mag.*, Vol. 17, No. 3, pp. 80-88, 1997.

[3] A. Kumagai, P. Hozian and M. Kirkland, "Neuro-Fuzzy Based Feedback Controller for Shape Memory Alloy Actuators," *Proceedings of SPIE-The International Society for Optical Engineering*, 3984, pp. 291-299, 2000.

[4] T. Hasegawa and S. Majima "A control system to compensate the hysteresis by Preisach Model on SMA actuator," *Proceedings of the 1998 9th International Symposium on Micromechatronics and Human Science*, pp. 171-176, 1998.

[5] P. H. Chang and J. W. Lee, "An Observer Design for Time Delay Control and Its Application to DC Servo Motor," *Control Eng. Pract.*, Vol. 2, No. 2, pp. 263-270, 1994.

[6] T. C. Hsia and L. S. Gao, "Robot Manipulator Control using Decentralized Time-Invariant Time-Delayed Controller," *Proc. IEEE Int. Conf. on Robotics and Automation* pp. 2070-2075, 1990.

[7] K. Youcef-Toumi and S. Reddy, "Dynamic Analysis and Control of High Speed and High Precision Active Magnetic Bearing," *J. Dyn. Syst.-T. ASME*, Vol. 114, pp. 623-632, 1992.

[8] S. M. Chin, C. O. Lee and P. H. Chang, "An Experimental Study on the Position Control of Electrohydraulic Servo System using TDC," *Control Eng. Pract.*, Vol. 2, No. 1, pp. 41-48, 1994.

[9] C. Liang and C. A. Rogers, "Design of Shape Memory Alloy Actuators," *J. Intel. Mat. Syst. Str.*, Vol. 8, pp. 303-313, 1997.

[10] H. J. Lee and J. J. Lee, "A numerical analysis of the buckling and postbuckling behavior of laminated composite shells with embedded shape memory alloy wire actuators," *Smart Mater. Struct.*, Vol. 9, pp. 780-787, 2000.

[11] K. Youcef-Toumi and S. T. Wu, "Input/Output Linearization Using Time Delay Control," *J. Dyn. Syst.-T. ASME*, Vol. 114, pp. 10-19, 1992.

[12] J. Park and P. H. Chang, "A Systematic Gain Tuning of PID control Based on the Concept of Time Delay Control (TDC)," *Proceedings of the 1997 Korea Automatic Control Conference*, Vol. 1, pp. 1093-1096, 1997.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Control of a Nonholonomic Mobile Robot Using RBF Network

Changmok Oh and Ju-Jang Lee
Department of Electrical Engineering & Computer Science
Korea Advanced Institute of Science and Technology
Daejon 305-701 Rep. of Korea

Abstract

A control structure that consists of kinematic controller and RBF network is presented. A proposed control law is developed by general learning rule of RBF network and its stability is guaranteed by Lyapunov theory. This control scheme can be applied to the tracking a reference trajectory problems. Moreover, there is no require off-line learning and the control law guarantee the small tracking errors.

1 Introduction

There are so many works about solving the problem of motion under nonholonomic constraints using kinematic model of a mobile robot. Recently, there are a few works to solve above problem considering the dynamics of mobile robot[1].

Another intensive area of research has been neural-network(NN) applications in the closed-loop control. In contrast to classification applications, in feedback control the NN becomes part of the closed-loop systems. Therefore, it is desirable to have a NN control with on-line learning algorithms that do no require preliminary off-line turning.

In this paper, we present the application of Radial-Basis-Function (RBF) network, which is a kind of the NN, to a mobile robot system. Because of the existence of NN in the control loop, it is needed to prove the stability of entire control system. In addition, the weights in the RBF network should stay bounded.

Traditionally the learning capability of a NN has been applied to the navigation problem in mobile robots. In these approaches the NN is trained in a preliminary off-line learning phase with navigation pattern behaviors.

Mobile robot navigation can be classified into three basic problems : tracking a reference trajectory, following a path, and point stabilization. Some nonlinear feedback controllers have been proposed for solving these problems. The main idea behind these algorithms is to find suitable velocity control inputs which stabilize the closed-loop system.

This paper is organized as follows. In Section 2, we present some basics of nonholonomic systems and RBF networks. In Section 3, we present the RBF network controller as applied to the tracking problems. Stability is proved by Lyapunov theory. Section 4 gives some simulation results and Section 5 presents some conclusion.

2 Preliminary

2.1 Mathematical Preliminary

Let R^n and $R^{m\times n}$ denote the space of real *n-dimensional* vector and of real *m-by-n dimensional* matrix, respectively. The *norm* of a vector $x \in R^n$ is defined by

$$\|x\| = \sqrt{x^T x} \tag{1}$$

The norm of a matrix $A \in R^{m\times n}$ is defined by

$$\|A\| = \sqrt{\lambda_{max}(A^T A)} \tag{2}$$

where $\lambda_{max}(\cdot)$ is the maximum *eigenvalue* of $A^T A$. The *Frobenius norm* of a matrix is defined by

$$\|A\|_F^2 = tr(A^T A) = \sum_{i,j} a_{ij}^2 \tag{3}$$

Definition 1: *Uniformly Ultimate Boundedness(UUB)* [3]

Consider the dynamic system $\dot{x} = f(x)$ with $x \in R^n$. Let the initial time be t_0, and the initial condition be $x_0 \equiv x(t_0)$. Then the solution $x(t)$ is *uniformly ultimately bounded(UUB)* if there exists a compact set $U_x \subset R^n$ such that for all x_0, there exists a $\delta > 0$ and a number $T(\delta, x_0)$ such that $\|x(t)\| < \delta$ for all $t > t_0 + T$.

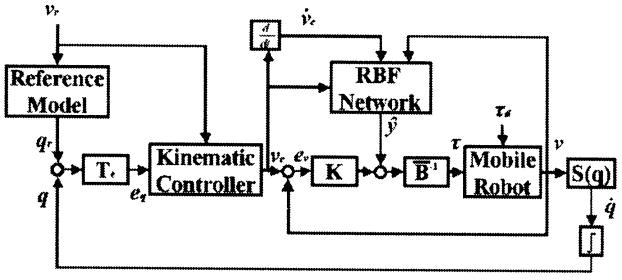


Figure 1: A Nonholonomic mobile robot

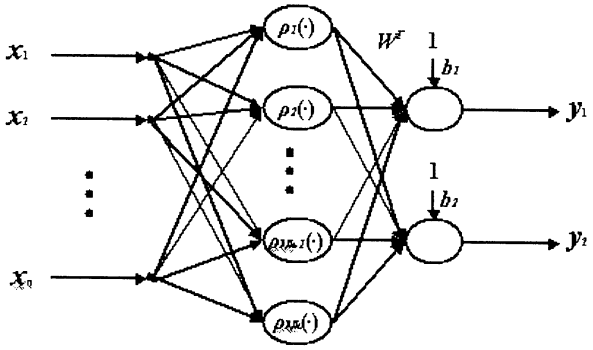


Figure 2: the RBF network

2.2 Nonholonomic Mobile Robot

Fig. 1 shows the simple figure of ERC Robot mobile base. ERC Robot was developed to assist old or disabled people in walk training[4]. As one user leaning on ERC Robot moves in one direction or rotates, it is controlled to follow the trajectory of user. Ordinary, people used fixed structure in walk training. In that case, some problem occurs, e.g. there exist a restriction of the training place and fixed circumstance make user easily be tired of traing.Using ERC robot in walk traing, we can reduce this problem.

The mobile robot in Fig. 1 is the target system for the proposed RBF network controller. It consists of a vehicle with two driving wheels mounted on the same axis, and two casters. Independent actuators achieve the motion and orientation of the above mobile robot.

According to [1], the complete equations of motion of the nonholonomic mobile platform are given by

$$\dot{q} = S(q)v(t) \tag{4}$$

where $v(t) \in \mathbf{R}^{n-m}$ is a velocity vector, $S(q)$ is a full rank matrix (r= n-m) formed by a set of vectors spanning the null space of constraint matrix, and

$$\bar{M}(q)\dot{v} + \bar{V}_m v + \bar{F} + \bar{\tau}_d = \bar{B}\tau \tag{5}$$

where $\bar{M}(q) \in R^{r\times r}$ is a symmetric positive definite inertia matrix, $\bar{V}_m(q,\dot{q}) \in R^{r\times r}$ is the centripedal and coriolis matrix, $\bar{F}(v) \in R^{r\times 1}$ represents surface friction, $\bar{\tau}_d$ denotes bounded unknown disturbances, $\bar{B} \in R^{r\times 1}$ is the input vector. And following properties are hold.

Property 1. $\bar{M}(q)$, the norm of the $\bar{V}_m(q,\dot{q})$, are bounded.

Property 2. The matrix $\dot{\bar{M}} - 2\bar{V}_m$ is skew-symmetric.

2.3 RBF Network

Fig. 2 shows one of the typical RBF network. The RBF network has been shown to have universal approximation ability[2]. So, unknown function $f(x)$ can be approximated by RBF network by

$$f(x) = W^T\rho(x) + \varepsilon \tag{6}$$

with ideal and unknown weight W, the Gaussian functions $\rho(x)$ and the estimation error bounded by

$$||\varepsilon|| < \varepsilon_N \tag{7}$$

for a given constant ε_N.

3 Control Design

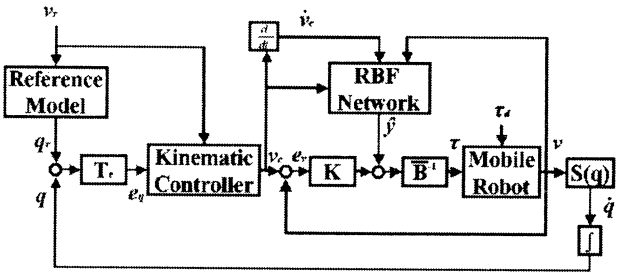


Figure 3: the proposed controller

The structure for the tracking control system is presented in Fig. 3. We use conventional kinematic controller to control the mobile robot[5]. And the function of RBF network is to approximate the dynamics of mobile robot.

We can transform the Eg.(5) using velocity error $e_v = v_c - v$ as

$$\bar{M}\dot{e}_v + \bar{V}_m e_v - f + \bar{\tau}_d = \bar{B}\tau \tag{8}$$

where $f = \bar{M}\dot{v}_c + \bar{V}_m v_c + \bar{F}$. This unknown dynamic term can make the trajectory tracking fail. To solve this problem, RBF network learns function f and compensate it. And the learning rule is as follows.

$$\dot{\hat{W}} = \beta \Psi e_v^T - \kappa\beta||e_v||\hat{W} \tag{9}$$

Then, it is guaranteed that the given mobile robot system follows the reference trajectory, i.e., the velocity tracking error, the position error, the NN weight estimates are UUB. It is easily derived using Lyapunov equation as

$$V = k_1(e_1^2 + e_2^2) + 2k_3 v_r(1 - \cos e_3) + V_1 \tag{10}$$

where

$$V_1 = \frac{1}{2}e_v^T \bar{M} e_v + \frac{1}{2\beta} tr(\tilde{W}^T \tilde{W}) \tag{11}$$

where $tr(\cdot)$ means the trace and $\tilde{W} = W - \hat{W}$. The proof is omitted.

4 Simulation Result

In this section, we show the RBF Network scheme and compare its performance with conventional approaches. For this purpose two controller have been implemented and simulated in MATLAB.

Fig. 4 and Fig. 5 show that there exists the steady state error in the case that bounded unknown disturbances and friction are included.

The response of proposed controller is shown in Fig. 6 and Fig. 7. It is clear that the performance of the system has been improved with respect to conventional controller. The validity of the RBF network controller has been evidently verified.

5 Conclusion

A stable controller is proposed and the stability of the whole system is shown by simulation results. This scheme is valid as long as the velocity control inputs are smooth and bounded, and the disturbances are also bounded.

It is usually impossible to get a perfect knowledge of the mobile robot parameters. To solve this problem,

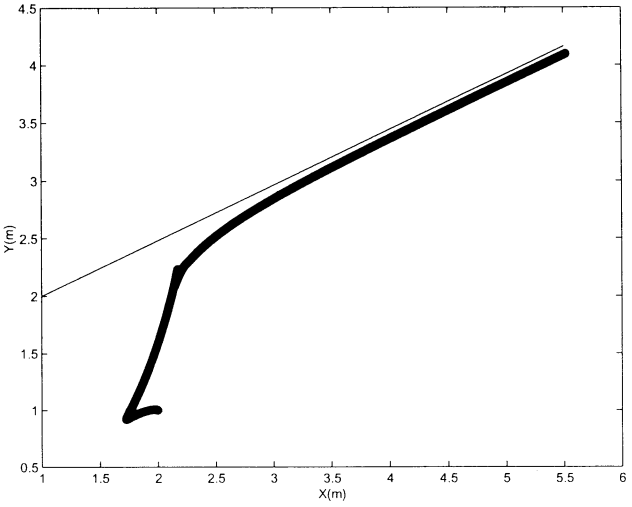


Figure 4: the conventional controller

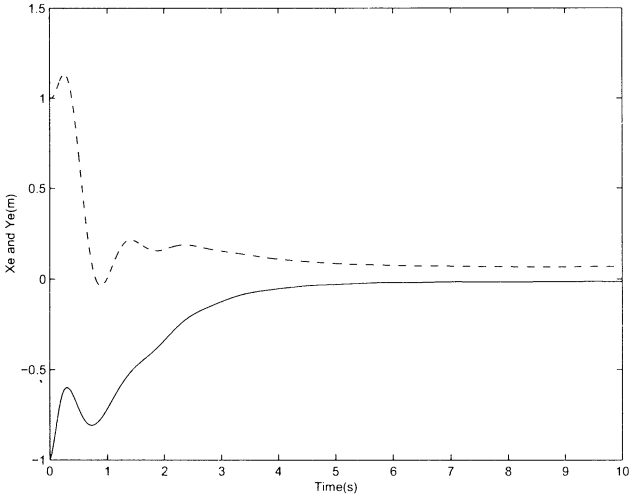


Figure 5: the conventional controller

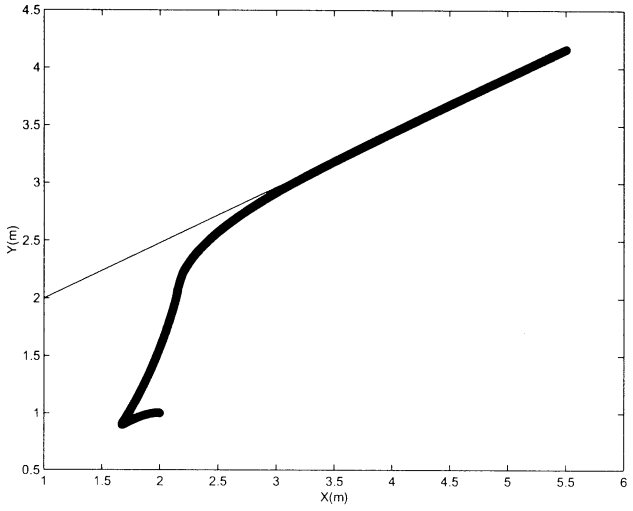


Figure 6: the proposed controller

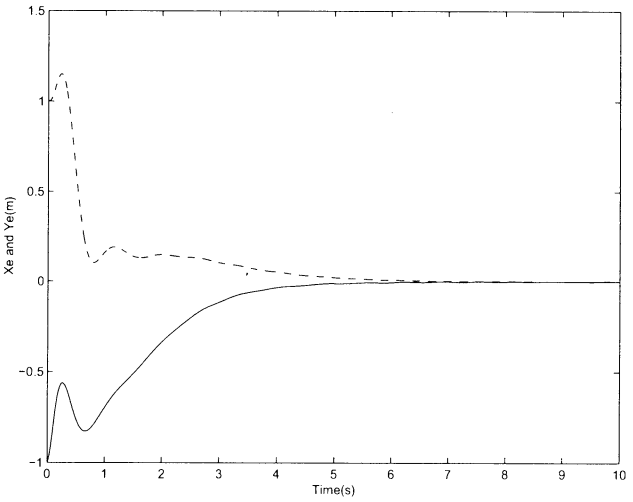


Figure 7: the proposed controller

RBF network with guaranteed performance has been derived. By using RBF network, there is not need of a priori information of the dynamic parameters of the mobile robot, because the NN learns them.

References

[1] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks",*IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp.589-600, 1998.

[2] R.M. Sammer and J.-J.E. Slotine, "Stable adaptive control and recursive identification using radial Gaussian networks", in *Proc. IEEE. Decision Contr.*, Brighton, U.K., 1991.

[3] F.L. Lewis, S. Jaganmathan, and A.Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems.* London, U.K.:Tayler and Francis, 1999.

[4] C.Y. Lee, K.H. Seo, C.M. Oh and J.J Lee, "A system for gait rehabilitation with body weight support: mobile manipulator approach",*Journal of HWRS-ERC*, 2001.

[5] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp.384-389.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Qualitative and Quantitative Information-based Level Control of the PWR Steam Generator of the Nuclear Power Generation

Hyeon Bae, Kee-Soo Jung*, Young Kwang Woo, Jae Ryong Jung, Sungshin Kim, and Bo-Hyeun Wang**

School of Electrical and Computer Engineering, Pusan National University, Busan, Korea
*Ulsan Office, Korea Plant Service & Engineering Co., Ltd., Ulsan, Korea
**Division of Electronic Engineering, Kangnung National University, Gangwon-do, Korea

Pusan National University 30 Changjeon-dong, Keumjeong-ku, Busan 609-735, Korea
E-mail: baehyeon@pusan.ac.kr, jungks1@khnp.co.kr, yokwoo@pusan.ac.kr, jrjung@pusan.ac.kr, sskim@pusan.ac.kr, bhw@kangnung.ac.kr,

Abstract – Level control of steam generators is very important factor for stable operation of nuclear power generations is safety. It is not easy way to control nuclear power generations because nuclear power generations have uncertainty and nonlinearity. In this paper, the automatic method for the steam generator is applied with respect to simulator data. This approach consists of two steps. Firstly, we check which a controller is in the unstable condition with the signals of steam flow, feed water, and level. Neural network model is employed for classification of the pattern. And then fuzzy gain tuner is used for the gain tuning. In design of fuzzy gain-tuner, the experienced knowledge is employed for making fuzzy rules. Human's knowledge is very important information to make decision in the field. Especially, if the system has no exact mathematical model, linguistic information is useful knowledge to design the system models.

Keywords: Nuclear, self-tuner, neural networks

1. Introduction

1950s, 440 sites of nuclear power generation are taking charge about 16% of whole electric-power production worldwide present. Because present 16 numbers of domestic nuclear power generation are taking charge more than one third portion for stable electric-power production and cost reduction, etc., the weight of nuclear power generation is very high. Accordingly, stable operation of nuclear power generation and improvement of utilization ratio is big target for country electric-power production. Improving reliability of measuring and control equipments is required to achieve this goal.

Automatic control of the water level of steam generator is known as that it is difficult because there are two specific phenomena that are shrink and swell effects are contrary to change of the flow rate.

We wished to apply neural network and fuzzy in this paper and control gain of two PI controller that is used in the nuclear power generation. Because two controllers for water level control are coupled each other, there is difficulty that both of gains should be considered together.

Basic information used in tuning of control gains is got from water level, amount of feed water, and steam flow rate. We wish to apply the proposed method that can analyze pattern of signals and select controller from analyzed results using the characteristic of neural network in this paper. After choosing one controller between a water level controller and a feed water controller by applying the neural network, the PI controller gain is tuned by fuzzy self-tuner suitably.

2. Nuclear Power Generation

2.1 Nuclear Generator

Used data in this experiment are got from the power plant model which uses the pressurized light water reactor (PWR) is being selected widely in the world.

Steam power generation generates stream by heating water in boilers using oil, coal, gas and so on, but in PWR nuclear generation, water of the first part system is heated by the nuclear fission of uranium and then the heated water of the first part system heats feed water of the second part system for generating steam in the steam generator [1].

2.2 Water Level Control of Steam Generator

The water at the lower part and the steam at the upper part exist in steam generator; height of lower water is called as water level of the steam generator. The water level of the steam generator is divided by wide range level measuring most part of steam generator interior and narrow range level considering the important operating part for level.

If the steam pressure is decreased by increase of sudden stream flow rate at operating the power generation, the water level could be increased by increment of the steam flow rate of the water. This phenomenon is called as swelling. On the contrary, if the steam pressure is declined by decrement of the steam flow rate, the water level is decreased by decrement of the amount of air bubble. This phenomenon is known as shrinking [2]. Both phenomena cause the non-controllable situation in the steam boiler, because these phenomena are non-linear and uncertain conditions.

2.3 Modeling of Water Level of Steam Generator

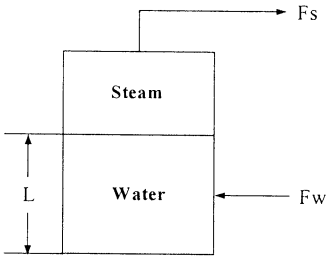


Fig. 1. Scheme diagram of the steam generator level.

The mass variation is as follows that is calculated by the difference between the feed water flow rate *Fw* and the steam flow rate *Fs* [1].

$$\frac{dM}{dt} = Fw - Fs \tag{1}$$

Let *Fer=Fw-Fs* and the volume of the water at the lower part of the steam generator is changed according to the mass *M* and amount of air bubble *Vs*.

$$\frac{dV}{dt} = \frac{1}{\rho}\frac{dM}{dt} + \frac{dVs}{dt} \tag{2}$$

Total variation of steam generator is calculated by the following equation.

$$L(S) = Ls(S) + Ld(S) \tag{3}$$

The changed water level *Ld* by low temperature feed water and bubble is expressed by the below equation ($dVs/dt \neq 0$, $dLd/dt \neq 0$).

$$Ld(S) = -\frac{G_2}{1+\tau_2 S} \cdot Fer(S) \tag{4}$$

The water level change can appear as follows because it is proportional in volume change.

$$Ls(S) = \frac{G_1}{S} \cdot Fer(S) \tag{5}$$

And the wide water level of the steam generator is changed corresponding to the mass change

$$\frac{dLw}{dt} = G_3 \cdot Fer \tag{6}$$

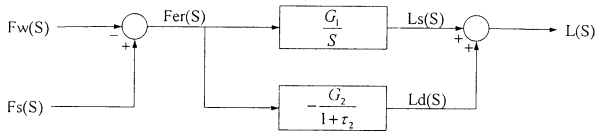


Fig. 2. Block diagram of the steam generator level.

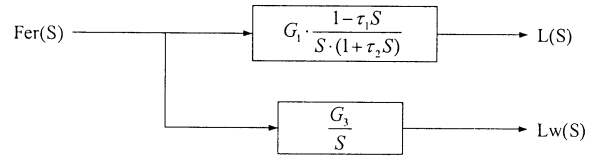


Fig. 3. Model of the steam generator level.

3. Neural Network and Fuzzy Logic

3.1 Neural Network

Neural network had been studied more than 30 years since Rosenblatt applies perceptron to pattern classification learning in late 1950s. Adjustable weights are updated using error between the target value and the output value in this paper. This network learning is called as supervised learning, and it is input/output mapping of the network of given training data [3].

At early, perceptron has been started on attempt that uses simple component and is able to design the intelligent and self-learning system.

Backpropagation MLP is adaptive network that has node applied in the same function according to input signal. This network is tried to access in various fields and is studied lively in pattern recognition, signal processing, data compression, and automatic control specially. Input of node net is defined as summation of weight sum of input signals that include bias term.

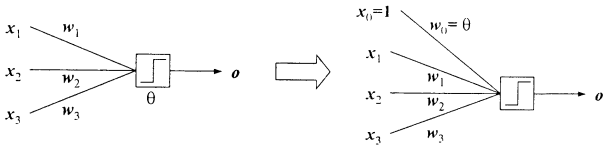


Fig. 4. Bias connection as the weight.

3.2 Fuzzy Logic

It is very difficult to describe the natural phenomenon that happens in real world exactly. Therefore, approximation or fuzziness is required to get the suitable model. For the example, it is trying to study most of the nonlinear system in the domain of the liner system.

The point of fuzzy system is that it is based on knowledge including *IF-THEN* rule. Fuzzy *IF-THEN* rule can be extracted by expert's knowledge.

In fuzzy logic, the membership function is applied to express the linguistic value mathematically. In this paper, Gaussian function that can consider nonlinear feature usually was employed for fuzzy membership function.

Fuzzy *IF-THEN* rule can be expressed by conditional presentation as follows.

$$\text{If } x \text{ is } A, \text{ Then } y \text{ is } B \tag{7}$$

where *A* and *B* are linguistic values defined by fuzzy set that exist on the universal set *X* and *Y* respectively. "*X* is *A*" and "*Y* is *B*" are called as the antecedent part and the consequent part respectively in general.

4. Design of Fuzzy Self-Tuner

4.1 Implementation of Steam Generator Simulator

Fig. 5 shows the steam generator simulator. The simulator is consisted of a controller and a model, and the controller and the model are designed to subroutine within each module.

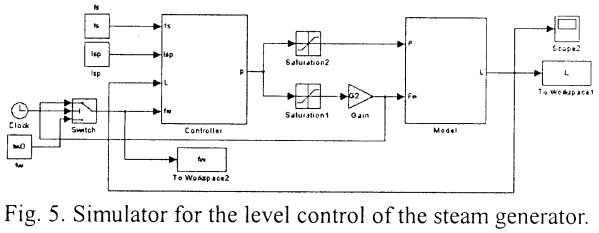


Fig. 5. Simulator for the level control of the steam generator.

Fig. 6 is the implemented steam generator model that is constructed for the simulator.

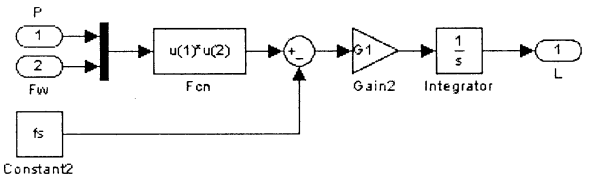


Fig. 6. Level model of the steam generator.

The simulator model achieves implementation of the level controller for the steam generator. Fig. 7 and 8 show the general water level controller and implemented model by Matlab/Simulink.

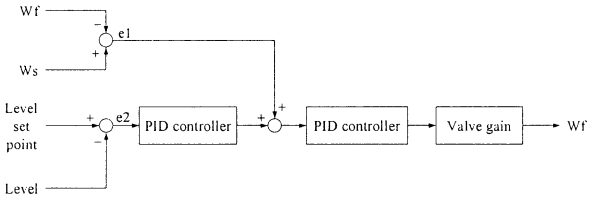


Fig. 7. Structure of the level feed water controller.

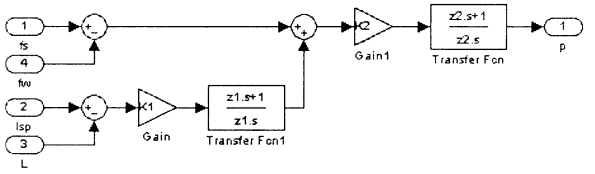


Fig. 8. Implementation of the controllers.

4.2 Pattern Recognition using Neural Network

Input/output data of the neural network that are applied in this paper are collected in the variation conditions of the data.

Fig. 9 shows the data of level variation conditions. Feed water data that are changed as handling the level are applied for the pattern recognition. Reference level is adjusted at 50. In other words, level 50 is the set point. Fig. 9 shows the trend of level and feed water of the steam generator with changing the water level from 50 to 45, 46, 47, and 48.

Fig. 10 is the graphs with respect to the variation conditions of feed water variation. The graph of Fig. 10 is trend of the data variation as retaining the level and changing the feed water. The figures show the level and feed water of the steam generator when the general set point of feed water is changed from 1.8 to 1.3, 1.4, 1.5, and 1.6.

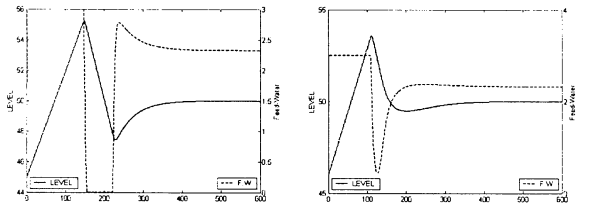


Fig. 9. Graphs of 45 and 46 level of the steam generator.

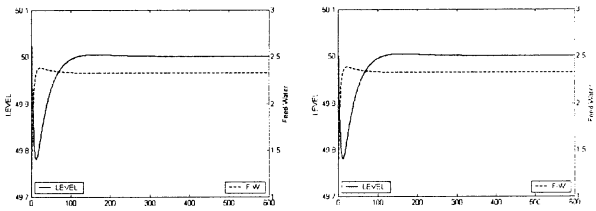


Fig. 10. Graphs of 1.3 and 1.4 feed water of the steam generator.

Table 1 shows the input data structure of the neural network that contains the level and feed water data. The numbers of sampling are 600 samples and simulation conditions are 4 different conditions, respectively.

Table 2 is the target data structure of the neural networks for the training of the model. In general, the binary number is used for the pattern classification. In this paper, the binary numbers are applied for training the model.

The neural network model that is employed in this paper is a kind of the learning model called as MLP (multilayer perceptron) usually that is a feedforward model. Input node consisted of 1200 numbers of input data and 7 hidden nodes as referred before.

Table 1. Input data of the neural network.

	Data (1200)	value / number	Conditions of input data							
			Level tuning (initial: 50)				F.W tuning (initial: 1.8)			
			48	47	46	45	1.6	1.5	1.4	1.3
Input data	Level data	600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600
	F.W data	600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600	1 ↓ 600

Table 2. Structure of target data for the neural network.

Contents \ Output		Target output y							
		Level data				F.W data			
Target variable		y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
Target value	Level tuning	**1**	**1**	**1**	**1**	0	0	0	0
	F.W tuning	0	0	0	0	**1**	**1**	**1**	**1**

4.3 Results of the neural network model

The weight update is achieved after training all input data set. And the results are calculated by repeating 10000 epochs. As shown in the final result, the error is fitted at 0.0025 after training.

Table 3. Results of the neural network.

Contents	Output / Input	Level data				F.W. data			
Training result	Change level	y_1 (48)	y_2 (47)	y_3 (46)	y_4 (45)	y_5	y_6	y_7	y_8
		0.947	**0.932**	**0.927**	**0.894**	-0.008	0.0028	0.0119	0.019
	Change F.W	y_1	y_2	y_3	y_4	y_5 (1.6)	y_6 (1.5)	y_7 (1.4)	y_8 (1.3)
		-0.0003	-0.001	-0.022	0.0446	**0.942**	**0.942**	**0.942**	**0.942**
Training error	Change level	0.0522	0.0680	0.0727	0.1057	0.0086	-0.003	-0.012	-0.019
	Change F.W	0.0003	0.0010	0.0227	-0.044	0.0579	0.0579	0.0576	0.0572

4.4 Design of Fuzzy Self-tuner using Knowledge

In this section, the control performance for gain tuning of the water level controller is tested. As shown in Fig. 11, the control performance is progressed fast because it means that the water level of the steam generator is changed rapidly if the water level is lowered to 45 and 46.

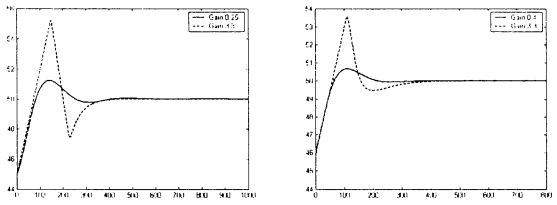


Fig. 11. Graphs of 45 and 46 level with respect to the gain.

Next, the simulation of the gain tuning of feed water controller is achieved. Fig. 12 is the result to seek the better gain that gives good performance after dropping the feed water flow rate to 1.3 and 1.4, and tuning the gain.

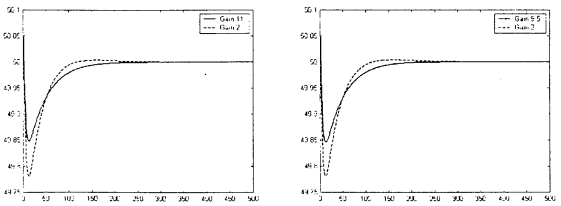


Fig. 12. Graphs of 1.3 and 1.4 feed water corresponding to gain.

The fuzzy logic is implemented to find the fine gain of controller automatically in this paper. Table 4 expresses each overshoot and undershoot value. As shown in Table 4, the good performance gain is different according to the difference between maximum and minimum. The inference rules are generated by the empirical knowledge of the experts.

From these basic rules, the implemented fuzzy inference is developed. For the implementation of the fuzzy logics, the membership functions are defined as based on the operator's knowledge.

Table 4. Overshoot and undershoot in the several conditions.

	Level				Feed water			
	48	47	46	45	1.6	1.5	1.4	1.3
Max	50.7574	52.0467	53.5823	55.2559	50.0042	50.0041	50.0040	50.0039
Min	49.9590	49.8086	49.4671	47.4162	49.7796	49.7802	49.7810	49.7812
Difference	**0.7984**	**2.2381**	**4.1152**	**7.8397**	**0.2246**	**0.2239**	**0.223**	**0.2227**
Gain	**1.05**	**0.6**	**0.4**	**0.25**	**5**	**7**	**9.5**	**11**
Rules	1. If Difference is very small then Gain is very big 2. If Difference is small then Gain is big 3. If Difference is big then Gain is small 4. If Difference is very big then Gain is very small							

4.5 Simulation Results of Fuzzy Self-tuner

The fuzzy self-tuner was designed using Matlab/Fuzzy Toolbox firstly and then this designed model was connected to existent Matlab/Simulink model for composing the system.

Fig. 13 shows the control structure that was composed by applying the fuzzy self-tuner that uses the fuzzy logic together. And Fig. 14 shows the control performance when the water level is lowered to 45 and 46.

If the water level becomes lower, overshoot and undershoot increase greatly. In this condition, the performance drops as the result in an experiment of front, but the fuzzy self-tuner could reduce the most undershoot.

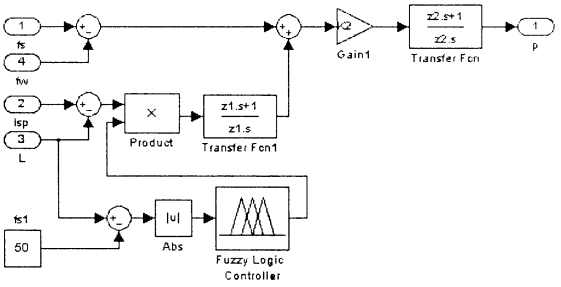


Fig. 13. Level controller containing a fuzzy self-tuner.

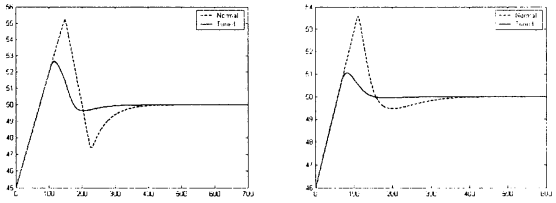


Fig. 14. Graphs of 45 and 46 level with respect to the gain.

5. Conclusion

The water level control of the steam generator is important for operating that is efficient in the operation of the nuclear power generation. In this experiment, we wished to design the self-tuner of the controller gain using the measured data in the steam generator.

Actually, the performance evaluation of the water level controller and the feed water flow rate controller is depending on the operator's knowledge in the field. The operator analogizes the status of the controllers using the difference of the time. As considering this feature, the neural network model can recognize two cases. After this classifying step, the process of the tuning gain is performed. In this experiment the fuzzy self-tuner was designed and applied to tuner the controller gain automatically.

If this experiment could be judged that increases the performance in the actual field, the labor of the operators should be reduced and it will help to decide operating job as removing the cause of the unexpected configuration that is started from the momentary misjudgment of the inexperienced operator.

References

[1] "The Development and Application of Steam Generator Level Total Digital Control System," Korea Electric Power Co., Ltd, Korea Electric Power Research Institute, Technical Report, 1998.

[2] Jin Wook Han, "Development of a Real-Time Self-Tuning Fuzzy-PID Controller for Water Level of Steam Generator," Dept. of Electrical Engineering, Chonbuk National University, M.S. thesis, 1999.

[3] Young-Kwan Lee and Byung-Hak Cho, "The Development and Application of Localization of Power Generation Simulator for Training Operator." Korea Electric Power Co., Electric Power Research Institute, Technical report, 1998.

Proc. of The Eighth Int. Symp. on Artificial Life and Robotics (AROB8th '03)
Beppu, Oita, Japan, 24-26 January, 2003

Automatic Moving Object Detection Algorithm For Region-based Tracking

Eun-Young Song, Changmok Oh and Ju-Jang Lee
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, 305-701

Abstract

We propose an automatic moving object detection algorithm for region-based object tracking. Our algorithm consists of three stages. At first stage, initial partition of each frame is obtained by the segmentation using mean shift density estimation. The motion of each initial partition is estimated by a region matching, at the second step. In the region matching stage, the occlusion problem can be occurred by the movement of the object. The occlusion results in a inaccurate motion estimation. To overcome this occlusion problem, we propose a new motion validation scheme. At final stage, the initial partition segments moving same direction merge into the moving objects. The experimental results demonstrate the robustness to the occlusion problem of the proposed approach.

1 introduction

Object tracking in video sequence is an important task in vision research area. It has a variety of application such as vision base control system, security system, vehicle tracking system. To implement this system, the automatic moving object detection is required. There are several moving object detection methods such as feature-based method, change detection methods using inter-frame difference, and motion-based method.

Most of the moving object detection methods rely on change detection[6],[7]. This method is motivated by the assumption that moving objects usually entail intensity changes between successive frames. Thus, the motion is detected statistical hypothesis testing on the difference image between two consecutive frames. This method has an advantage in computational point of view. However, it has two major drawbacks. First, if objects are not fully textured, only the boundary of the objects can be checked, while the interior of the objects remain unchanged. And the background appearing abruptly can be checked as the moving objects.

Motion-based method[2] is motivated by the assumption that the motion vector of the moving object is can be distinguished from the motion vector of the background. Since this method use the real motion vector estimated in the image sequence, it is an efficient method for the single object detection and multiple object detection.

In this paper, we have proposed a new algorithm for automatic detection of moving objects. Our method consists of three stages. At first stage, each frame is segmented in several initial partitions. The motion estimation of each partition is obtained by the region matching procedure, in the second stage. Finally the initial segment moving the same direction can be merged into one moving object.

The paper is organized as follows. In Section II, we discuss the proposed moving object detection algorithm. Experimental results are reported n Section III, and the conclusion follows in Section IV.

2 Automatic Moving Object Detection

2.1 Overview of the moving object detection scheme

Fig.1 shows the procedure of the proposed moving object detection procedure. First the initial segmentation is applied in a input image. From the segmentation, we can obtain the initial partition. Then, the motion vector of the each partition from the first stage is estimated at the second stage. The motion vector is obtained by the region matching, however the occlusion problem can be occurred by the change of the region shape caused by the moving object. To solve this problem, our method has proposed the occlusion check stage and motion validation stage. Finally, from the estimated motion, we can get the moving object

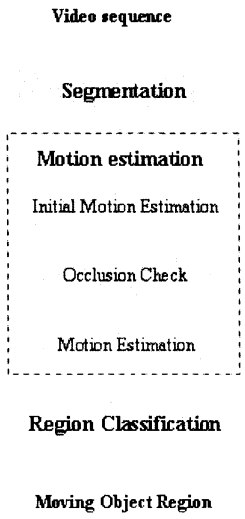


Figure 1: Moving object detection process

region.

2.2 Initial segmentation

Initial segmentation is accomplished by the mean shift clustering method. The basic theory for estimation of the density gradient was proposed in 1975 by Fukunaga and Hosteler[1]. The mean shift algorithm has been proposed as a method for cluster analysis[3]. The method has been proved to provide robust and reliable solutions for many vision tasks. From the segmentation we can obtain the initial partitions.

2.3 Motion estimation using the region matching

In order to estimate the object motion, each object region of the current frame is compared against all region of a previous frame. The motion vector is all region of a previous frame. The motion vector is finally obtained by finding minimum position on error surface which contains all matching distortion over all candidate motion vectors. In general, the region matching techniques[4] employ the mean absolute difference(MAD) to measure the region matching distortion. Let be the kth frame, for each region , the region matching can be defined as

$$MAD(i) = \frac{\sum_{(x,y)\subseteq R_i} \|I_{k+1}(x,y) - I_k(x+u, y+v)\|}{N_{total}} \quad (1)$$

where N_{R_i} is the total pixel number in the region R_i and is the motion vector (u, v). The motion vector can be estimated by finding the point having the minimum $MAD(i)$ value.

2.4 Occlusion check and motion estimation scheme

Most motion estimation techniques based on region suffer from the occlusion problem[5]. When the object moves to other position, the shape of the objects except the moving object is changed. From the occlusion problem can cause serious problem in both motion estimation stage and moving object classification stage. In order to overcome this problem, we have proposed a new motion estimation method.

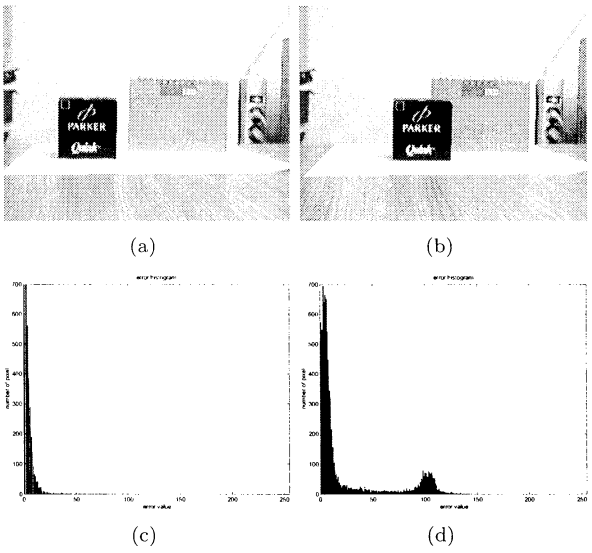


Figure 2: The error histogram by the occlusion problem (a)k-th frame image (b)(k+1)th frame image (c) The error histogram of the moving object region (d) The error histogram of the occlusion region

Fig2(c) shows the error histogram of non-occlusion region and Fig.2(d) that of occlusion one. Fig.2(c) and Fig.2(d) shows the result that is similar to the prediction above. As illustrated in Fig.2(c), errors of non-occlusion region are distributed around zero with the Gaussian distribution. From the Fig.2(c), it can be easily seen that the variance is small because the distribution is placed around the mean value. In Fig.2(d), it is shown that the errors of occlusion region are added to the error distribution, which is observed in Fig.2(c), by Gaussian noise. From the figures, the mean value of occlusion region's errors is larger than that of non-occlusion region's errors, because occlusion region has more errors. From this, the following observation is

obtained:
Observation 1: The mean value of errors of occlusion region is larger than that of non-occlusion region.
Observation 2: The variance of errors of occlusion region is larger than that of non-occlusion region.
Based on these two observation result, we proposed the occlusion detection algorithm using the distribution of the error histogram.
The occlusion detection algorithm is as follow. We assume that the estimated motion vector (u, v) for the region R_i is valid.

1. The follow procedure is executed for all region $R_i, i = 0, .N$.
   1) The error of each pixel in regionR_i, $e_i(x, y)$ is computed.
   $$e_i(x, y) = \|I_{k+1}(x, y) - I_k(x+u, y+v)\| \quad (2)$$
   2) The error mean value of the region R_i, m_i is computed.
   $$m_i = \frac{\sum_{(x,y)\subseteq R_i} e(x, y)}{N_{R_i}} \quad (3)$$
   where N_{R_i} is the pixel number of the region R_i.
   3) The error variance of R_i, σ^2 is.
   $$\sigma_i^2 = \sum_{(x,y)\subseteq R_i} (e(x, y) - m_i)^2 \quad (4)$$
2. Using the result of (1), the average region mean, m_{ave} and average error variance,σ_{ave} are computed.
   $$m_{ave} = \frac{\sum_{i=1}^{N} m_i}{N} \quad (5)$$
   $$\sigma_{ave} = \frac{\sum_{i=1}^{N} \sigma_i}{N} \quad (6)$$
3. For all regions, the following decision test classifies occlusion and non-occlusion regions
   $$\text{Decide occlusion region } \sigma_i^2 \geq \sigma_{ave}^2 \quad (7)$$
   $$\text{Decide non-occlusion region } \sigma_i^2 \geq \sigma_{ave}^2 \quad (8)$$

By the decision rule above, it can be distinguished whether the given region is occlusion region or not. The result that the region is occlusion one implies that motion estimation procedure has mistakes. Therefore, the occlusion region needs another motion estimation.Yet, the mean absolute error using region matching brings out same results. In this paper, therefore, new matching error is presented for the motion estimation of occlusion region.

The proposed matching error function is as follow:

$$e_{occ} = \frac{e}{N_R} \cdot w_1 + N_{occ} \cdot w_2 \quad (9)$$

where e and N_R notes the error and the pixel number of remaining part except occlusion pixels and N_{occ} notes the number of pixel in the occluded area. $w1$ and $w2$ are the weighting factors and they can be determined based on the size of non-occlusion and occlusion area. The first term of function 9 produces the errors of non-occlusion part in the region R_i. By excluding errors of occlusion region, the wrong motion estimation by occlusion region's errors can be avoided. The second term of the error function 9 is the size of occlusion region in the area where motion vectors are moving. When the object moves to the wrong direction, N_{occ} value is getting larger. Therefore, the N_{occ} value prevents the object region from moving to wrong direction.

3 Simulation Result

The moving object detection algorithm is applied to the two consecutive frames. Fig.3 shows the procedure of the moving object detection scheme with the successive frame image with the occlusion problem. From the two consecutive frames, the motion vector can be estimated. Fig.3(c)and (e) is the result of the motion estimation and moving object detection using the conventional motion estimation method and Fig.3(d) and (e) is the result of motion estimation and the moving object detection using the proposed motion estimation result. Fig.3(e) shows the false moving object detection by the occlusion, the object, which didn't move, is detected also as moving object. Fig.3(e)and(f), show that the proposed algorithm is effective moving object detection algorithm to the presence of the occlusion problem.

4 Conclusion

Moving object detection based on the region motion suffers from the occlusion problem. In this paper, in order to the occlusion problem, we proposed the occlusion detection algorithm. Proposed moving object detection algorithm consists of three stages. At first, the frame is segmented into the initial partitions

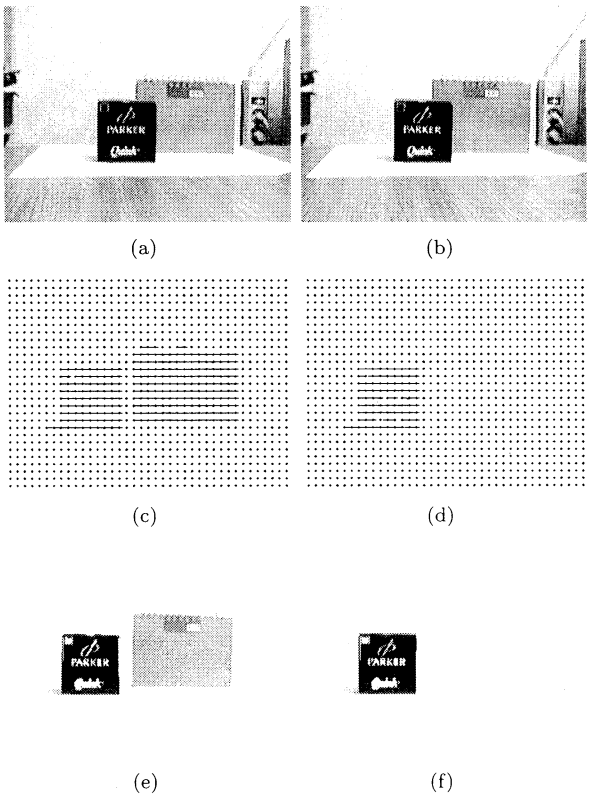


Figure 3: The experimental result of the proposed moving object detection algorithm (a),(b) test images (c)Estimated motion vector using conventional method (d)Estimated motion vector using the proposed method (e)Detected moving object using the convention method (f)Detected moving object using the proposed method

and in the next step, the motion vector of each initial partition is estimated. At the occlusion check and motion validation stage, the false detection of the moving object is avoiding. Experimental results demonstrate that our proposed algorithm can successfully extract moving object from the image sequence and track the moving object obtained by the proposed automatic moving object detection algorithm.

References

[1] K. Fukunaga, L.D Hostetler, "The Estimation of the Gradientof Density Function, with Applications in Pattern Recognition," *IEEE Trans. Info. Theory*,Vol. 1 IT-21, pp. 32-40, 1975.

[2] J. Badenas, J. M. Sanchiz, et. Al , "TMotion-based segmentation and region tracking in image sequences," *Patten Recognition*,Vol. 34, pp. 661-670, 2001.

[3] Dorin Comaniciu, Peter Meer, "Mean Shift: A Robust AproachTowrd Feature Space Analysis," *IEEE Trans. PAMI*,Vol. 24, pp. 603-619, 2002.

[4] L. K. Liu and E. Feig, "A block-based motio estimation invideocoding," *IEEE Trans. Circuits Syst. Video Techno*,Vol. 16, pp. 419-421, 1994.

[5] Y. Tsaig, A. Averbuch, "Automatic Segmentation of Moving Objects in Video Sequences : A Region Labeling Approach," *IEEE Trans. Circuits Syst. Video Technol*,Vol. 12, pp. 597-612, 2002.

[6] N. Paragois and G. Tziritas, "Adaptive detection and localization of moving objects in image sequences," *IEEE Trans. Image Processing*,Vol. 9, pp. 497-500, 2000.

[7] A. Neri, S. Colonneses, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Processing*,Vol. 66, pp. 219-232, 1998.

SHUBUNDO INSATSU Co. Ltd.
2 — 1 — 2 1 Hagiwara, Oita, 870-0921, Japan
Tel: 097-551-8148, Fax 097-552-0360
E-mai: info@shubundo-p.co.jp
http://www.shubundo-p.co.jp