PROCEEDINGS OF INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS

(AROB '97)

February18-20, 1997
B-Con Plaza, Beppu Oita, JAPAN

Editors : Sadao Fujimura & Masanori Sugisaka
ISBN4-9900462-7-7

Proceedings of International Symposium on

ARTIFICIAL LIFE AND ROBOTICS

(AROB '97)

February 18–20, 1997

B-Con Plaza, Beppu, Oita, JAPAN

Editors: Sadao Fujimura & Masanori Sugisaka

ISBN4-9900462-7-7

INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS (AROB '97)

ORGANIZED BY

International Organizing Committee of AROB

CO-SPONSORED BY

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

CO-OPERATED BY

The Institute of Electrical and Electronics Engineers, Tokyo Section (IEEE)
The Institute of Electrical Engineers of Japan (IEEJ)
The Institute of Electronics, Information and Communication Engineers (IEICE)
The Institute of Systems, Control and Information Engineers (ISCIE)
Japan Robot Association (JARA)
Robotics Society of Japan (RSJ)

SUPPORTED BY

Oita Prefectural Government
Oita Municipal Government
Oita Chamber of Commerce and Industry
Beppu Municipal Government
Oita Industrial Group Society
Oita System Control Society
NHK Oita Station
OBS Broadcast Company
TOS Broadcast Company
OAB Broadcast Company
Asahi Shimbun
Nikkan Kogyo Shimbun
Oitagodo Shimbun
Science and Technology Agency

Following companies financially support AROB'97

Chiyoda Corporation
Fanuc Ltd.
Kyushu Electric Power Company Incorporated Oita Branch Office
Matsushita Research Institute Tokyo, Inc.
Mitsubishi Electric Corporation
Mitsubishi Heavy Industries, Ltd.
NEC Corporation
Nishikawa Keisoku
Nishi Nippon Electric Wire Cable Co., Ltd.
Nisshinbo Mec Inc.
Ricoh Company, Ltd.
Shintsurukai Kosan Co., Ltd.
Sugahara Industry Co., Ltd.
The Oita Chamber of Commerce and Industry
Yanai Denki Kogyo Co., Ltd.
Yasukawa Electric Co., Ltd.
Yatsushika Sake-Brewing Co., Ltd.

HONORARY PRESIDENT

M. Hiramatsu (Governor, Oita Prefecture)

INTERNATIONAL ORGANIZING COMMITTEE

Chairman: M. Sugisaka (Oita Univ., Japan)
K. Abe (Tohoku Univ., Japan)
S. Arimoto (The Univ. of Tokyo, Japan)
W.B. Arthur (Santa Fe Institute, USA)
Z. Bubunicki (Tech. Univ. of Wroclaw, Poland)
J.L. Casti (Santa Fe Institute, USA)
T. Christaller (Univ. of Bielefeld, Germany)
J. Epstein (Brookings Institution, USA)
T. Fujii (RIKEN, Japan)
S. Fujimura (The Univ. of Tokyo, Japan)
T. Fukuda (Nagoya Univ., Japan)
T. Gomi (AAI, Canada)
H. Hagiwara (Kyoto School of Computer Science, Japan)
I. Harvey (Univ. of Sussex, UK)
P. Husbands (Univ. of Sussex, UK)
D.J.G. James (Coventry Univ., UK)
T. Jinzenji (Sanyodenki Co. Ltd., Japan)
R.E. Kalaba (Univ. of Southern California, USA)
H. Kashiwagi (Kumamoto Univ., Japan)
S. Kitamura (Kobe Univ., Japan)
K. Kyuma (Mitsubishi Electric Co., Japan)
C.G. Langton (Santa Fe Institute, USA)
J.J. Lee (KAIST, Korea)
C. Looney (Univ. of Nevada-Reno, USA)
G.I. Marchuk (Russian Academy of Sciences, Russia)
K. Matsuno (MITI, AIST, Japan)
H. Miura (The Univ. of Tokyo, Japan)
T. Mushya (Brain Function Lab., Inc., Japan)
T. Nagata (Inst. of Systems & Information Tech./Kyushu, Japan)
M. Nakamura (Saga Univ., Japan)
H.H. Natsuyama (California State Univ., USA)
Y. Nishikawa (Osaka Inst. of Tech., Japan)
R. Pfeifer (Univ. of Zurich-Irchel, Switzerland)
M. Raibert (MIT, USA)
S. Rasmussen (Santa Fe Institute, USA)
T.S. Ray (ATR, Japan)
T. Shibata (MITI, AIST, Japan)
K. Shimohara (ATR, Japan)
L. Steels (VUB AI Lab., Belgium)
K. Tamura (MITI, AIST, Japan)
H. Tanaka (Tokyo Medical & Dental Univ., Japan)
Y. Tokura (ATR, Japan)
K. Tsuchiya (Kyoto Univ., Japan)

S. Ueno (Kyoto School of Computer Science, Japan)
A.P. Wang (Arizona State Univ., USA)
W. Wells (Univ. of Nevada-Las Vegas, USA)
T. Yamakawa (Kyushu Inst. of Tech., Japan)
Y.G. Zhang (Academia Sinica, China)

STEERING COMMITTEE

Chairman: M. Sugisaka (Oita Univ., Japan)
J.L. Casti (Santa Fe Institute, USA)
S. Fujimura (The Univ. of Tokyo, Japan)
H. Kashiwagi (Kumamoto Univ., Japan)
M. Nakamura (Saga Univ., Japan)
S. Ueno (Kyoto School of Computer Science, Japan)

PROGRAM COMMITTEE

Chairman: S. Fujimura (The Univ. of Tokyo, Japan)
Vice Chairman: K. Aihara (The Univ. of Tokyo, Japan)
S. Arimoto (The Univ. of Tokyo, Japan)
T. Fukuda (Nagoya Univ., Japan)
Y. Hori (The Univ. of Tokyo, Japan)
S. Kitamura (Kobe Univ., Japan)
H. Kobatake (Tokyo Univ. of Agriculture & Tech., Japan)
J.-J. Lee (Korea Advanced Inst. of Science and Tech., Korea)
T. Nagata (Inst. of Systems & Information Tech./Kyushu, Japan)
M. Nakamura (Saga Univ., Japan)
Y. Nishikawa (Osaka Inst. of Tech., Japan)
T. Omori (Tokyo Univ. of Agriculture & Tech., Japan)
T.S. Ray (ATR Human Information Processing Research Labs., Japan)
K. Shimohara (ATR Human Information Processing Research Labs., Japan)
S. Shin (The Univ. of Tokyo, Japan)
M. Sugisaka (Oita Univ., Japan)

LOCAL ARRANGEMENT COMMITTEE

Chairman: T. Nakano (Oita Univ., Japan)
T. Endo (Oita Univ., Japan)
M. Nanjo (Oita Industrial Research Inst., Japan)
M. Sugisaka (Oita Univ., Japan)
A. Tominaga (Oita Univ., Japan)

TECHNICAL PAPER INDEX

Special lecture (Invited talk)

Genetic algorithm

Evolution

Information and intelligence

Complex systems

Robot and its control

Behavior analysis

Special session "Chaos"

From sensory substitute technology to virtual reality research

Tohru Ifukube

Laboratory of Sensory Information Engineering, Research Institute for Electronic Science, Hokkaido University, Sapporo 060, JAPAN

1.Introduction

The purpose of sensory substitute studies is to create images which are similar to original sensations by transmitting information to central nervous system through the residual senses or the nervous systems in order to substitute the lost or damaged senses. On the other hand, virtual reality is one of technologies to display information which is similar to real images, and its research approach is almost the same as the sensory substitute studies except that the sensory substitute devices are used for only by the disabled.

We have been carrying out the sensory substitute studies for about 25 years, and have designed several substitute devices which are in practical use or will be put into use. And also, we have got many findings such as sensory integration, concept formation, sensory-motor association in the human brain. I will refer to our researches regarding the sensory substitutes and mention how the researches have been related to the virtual reality researches.

2. Tactile Voice Coder

Through fundamental research on auditory and tactile information processing, we developed a tactile voice coder for the deaf about 20 years ago which has been manufactured in Japan. Our research was broadcast as a documentary program titled "my finger can hear letters". The tactile voice coder is a device which produces sound spectral patterns that are analyzed in 16 frequency components to an index fingertip by using a piezo-electric vibrator array consisting of 16 rows by three columns as shown in figure 1.

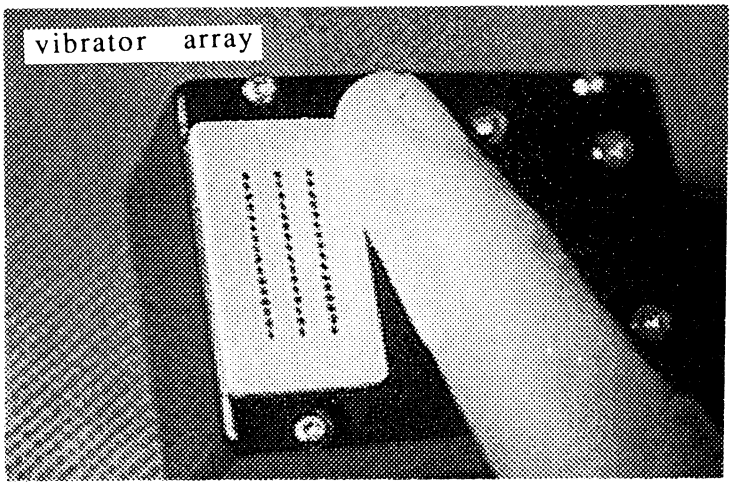


Fig.1. A vibrator array for tactile voice coder

The device makes it possible to discriminate the first and the second formants of vowels, and some consonants such as fricatives. It assists lip reading and hearing aids and also helps to obtain feedback for speech training. It is already in practical use, and studies on its evaluation are being done at some institutes for the deaf in Japan. From identification tests of Japanese monosyllables consisting of /ku/, /su/, /nu/, /fu/, /ru/, /tsu/ and /yu/, the identification rate increased from 23% (lip-reading only) to 68% (lip-reading plus tactile voice coder) after one week of training. We are now improving the tactile voice coder using a digital signal processor (DSP) to make it smaller and cheaper. The findings and technologies regarding tactile voice coder will be applied to a tactile display for virtual reality systems.

3. Voice Typewriter

About 15 years ago, the tactile voice coder was applied to a voice typewriter for acquired deaf people for whom it is difficult to learn lip-reading and finger language. This device can convert monosyllabic voice sounds such as /a//ka//sa//ta//na//ha/ into Japanese letters almost in real time and shows them on a display. There are only 5 vowels and 14 consonants in Japanese. Every Japanese word is pronounced as a series of monosyllables. For example, Tokyo is pronounced as /to//u//ki//yo/ and /u/. Every monosyllable corresponds to one Japanese letter. This makes it easy to design a voice typewriter.

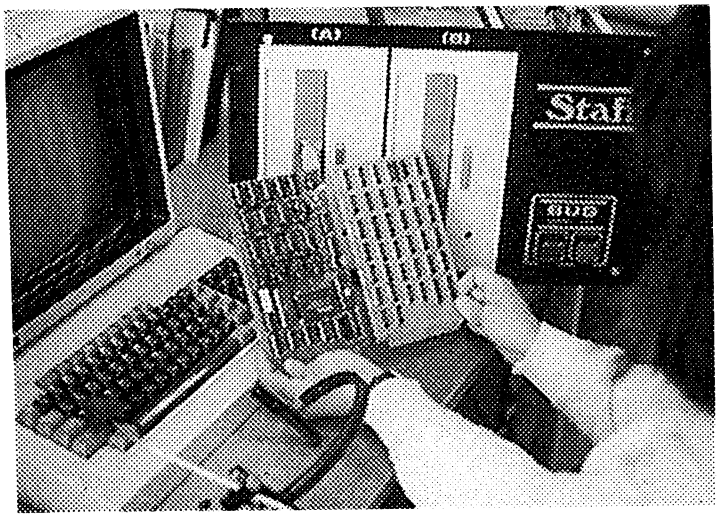


Fig.2. Voice typewriter

Figure 2 shows the voice typewriter in which a microprocessor was used. Some algorithms were performed by the hardware to decrease the conversion time. Its recognition rate is 96% and its response time is about 0.2, which changes 50 monosyllabic voice sounds per minute to Japanese letters. It is a substitute for finger language or conversation writing. It can also be used as a typewriter for the upper limb disabled. The voice typewriter was manufactured and applied to the input device of a Japanese word processor. This technology will play an important role in interactive virtual reality systems.

4. Cochlear Implant

I have been studying auditory prosthesis which stimulates surviving auditory nerves of the deaf who have lost the function of receptors called hair cells inside the cochlea. However, it was almost impossible to continue the study in Japan because of a lack of coresearchers. Therefore, I went to Stanford University to continue the study of auditory prosthesis about 12 years ago. In our auditory prosthesis, auditory nerve cells are electrically stimulated by 8 electrodes placed in the cochlea. So, spectral patterns were analyzed in 8 steps. The 8 signals were transmitted electro-magnetically from a transmitter outside the body to a receiver inside the body. However, when my voice stimulated auditory nerves of patients, they said that the perceived signals sounded very different from voices which they remember. Furthermore, their ears were infected twice during my stay at Stanford. So, I gave up that research.

The figure 3 shows configuration of our cochlear implant. A hole has to be drilled into the skull in order to implant an electrical circuit. It was impossible to use for young people whose skulls were

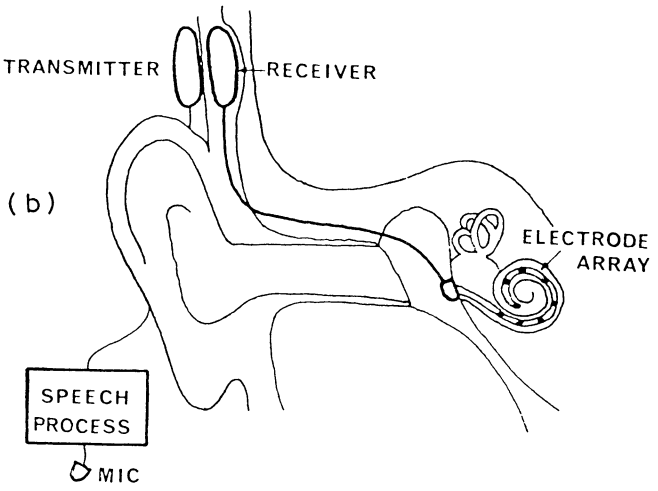


Fig.3. Cochlear implant

growing. It seems important at this time to develop auditory stimulators outside of the cochlea that will improve hearing. I have decided to change the method of the auditory prosthesis after returning to Japan. In our new method, only one electrode is contacted to the cochlear membrane instead of inserting 8 electrodes. This is a less invasive method compared with the conventional ones. But in this single channel method, the amount of transmitted information is much less than for the 8 channel method. So, we have designed a device which can convert the speech spectrum patterns into time sequential patterns in order to increase the amount of information.

Although our method was not so effective for recognition of speech especially consonants, we have acquired some findings regarding sensory association in the human brain. These findings will be related to create an association between virtual reality and the real world.

5. Tinnitus Suppresser

It has been known that fifty or sixty percent of hearing impaired patients suffer from tinnitus which causes them stress and sleepless. The effect that electrical stimulation to the cochlea suppresses tinnitus to some extent was observed in the course of developing our cochlear prosthesis for the deaf. We have treated tinnitus patients using a tinnitus suppresser which we had developed. With this treatment about 30 % of the patients answered that the tinnitus was disappeared and any side effects were not observed. This method, however, suppresses tinnitus for only a short period of time, so many tinnitus suffers have long been waiting for an implant system which can suppress tinnitus whenever it appears.

The implantable tinnitus suppresser which we developed consists of a wave generator, a primary coil outside the ear, a secondary chip coil implanted under the skin of an external ear and a Pt-Ir electrode placed on the promontory of the cochlea as shown in figure 4. The secondary coil connected with an electrode was

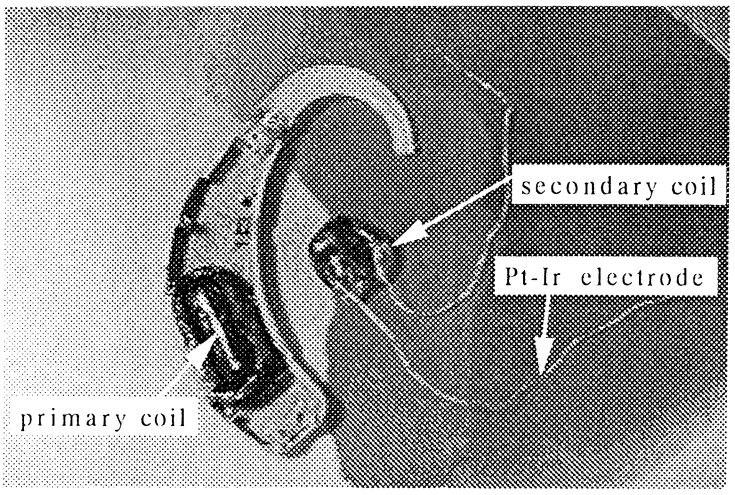


Fig.4. Tinnitus suppresser

implanted for 7 patients. Using this implant system, the tinnitus can be suppressed whenever it appears. It is also important to investigate the mechanism of tinnitus (virtual sound produced in the brain) in the field of virtual reality research.

6. Digital Hearing Aid

In general, elderly people who have suffered from a hearing impairment have less ability to understand spoken language even though they can hear the speech sounds. This phenomenon seems to be due to a decrease in the recognition of auditory time patterns in the speech area of the cortex. So, in cooperation with the Central Research Laboratory of the Hitachi company, we have designed a hearing aid which can slow down speech without any pitch frequency change by using a digital signal processor as shown in figure 5.

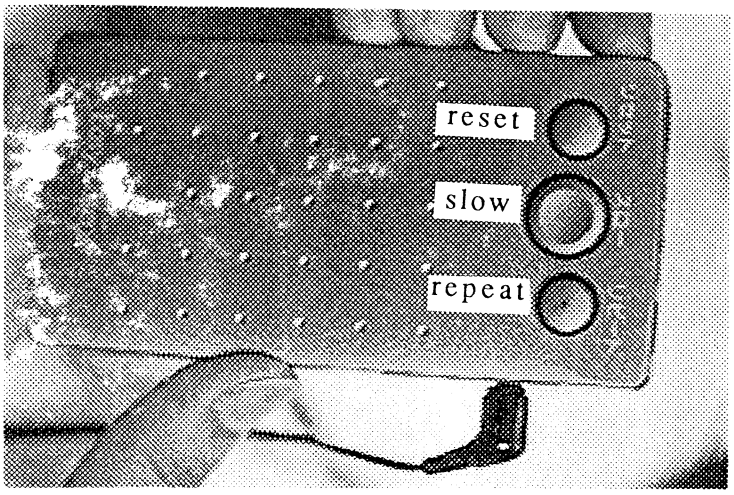


Fig.5. Digital hearing aid

This hearing aid was evaluated and improved by our university. In our hearing aid, first, the pitch frequency of a vowel part is extracted, and then the pitch wave is repeated so that the duration of the vowel part can be slowed down. We have proved that the device is effective in catching the meaning of rapidly spoken sentences for the elderly sensorineural hearing impaired. The hearing aid has been manufactured by the Hitachi LTD. Facilitating speech comprehension will become important in order to construct virtual reality systems for the elderly.

7. Electric Artificial Larynx

Various methods for vocal rehabilitation have been applied for laryngectomees who have lost their speech function. We have studied the vocalization mechanism of a mynah bird, which can imitate human voices, in order to apply it to a synthetic sound generator for laryngectomees. An electric artificial larynx is one of the artificial larynges for the people who have failed to master the other substitute speech. However, this prosthesis makes it difficult to produce natural voices. From the analysis of the mynah's vocalization mechanism, we found the mynah can clearly imitate phonetic information such as intonation and accent. This is the reason why mynah's voices can be heard as natural voices by human. In order to improve a conventional electric artificial larynx, we have proposed a new method that can allow laryngectomees to control intonation by using their respiration.

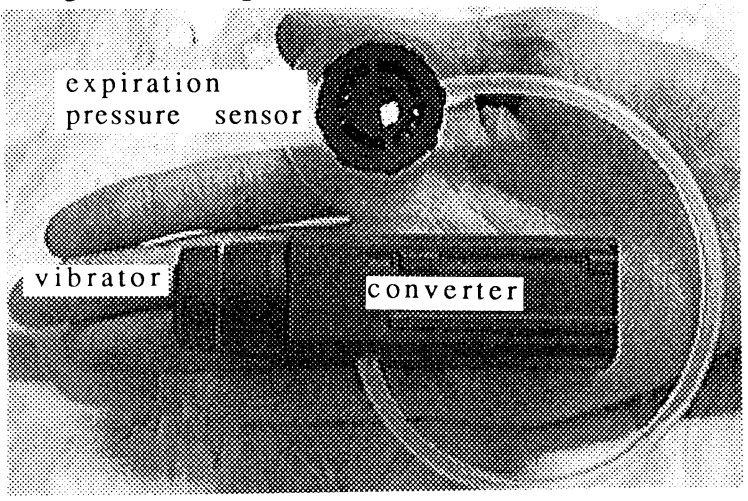


Fig.6. Electric artificial larynx

The device consists of three parts as shown in figure 6. The first part is a pressure sensor that can detect expiration air pressure produced from a stoma made by a surgical incision into the neck. The second part is an electromechanical vibrator that can be attached to the neck. By using the optimal parameters, the pitch pattern of the electric larynx voice became clearly similar to the pattern produced from a normal subject after one day training. This type of electric artificial larynx has just manufactured by a company in Japan. It has been proven that intonation is very important to make an artificial larynx voice natural. This finding is concerned with speech synthesis used in virtual reality systems.

8. Ultrasonic Eyeglasses

A new model of a mobility aid for the blind has been developed using a microprocessor and ultrasonic devices. In this model, a down swept frequency modulated ultrasound is emitted

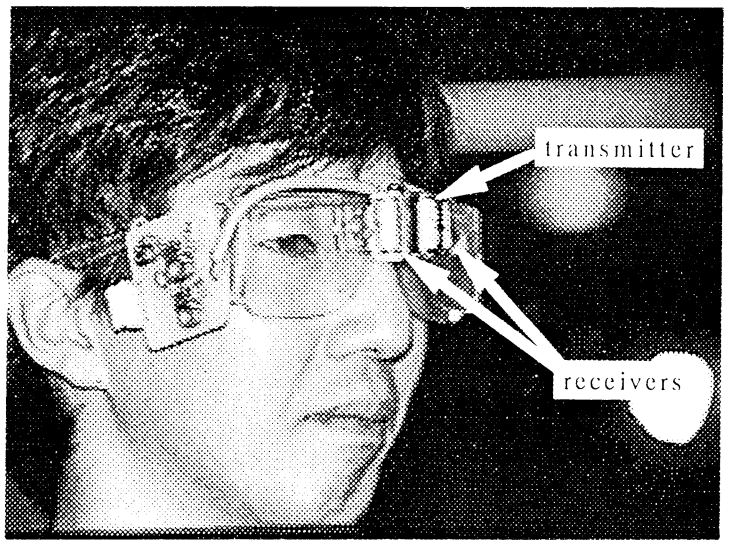


Fig.7. Ultrasound eyeglasses

from a transmitter with broad directional characteristics in order to detect obstacles as shown in figure 7. Ultrasound reflections from the obstacles are picked up by a two-channel receiver. The frequency of the emitted ultrasound is swept from 70 to 40 kHz within 1 ms, so it has almost the same characteristics as the ultrasound which a FM-bat produces for echo location. The frequency of the reflected ultrasound wave is reduced by about 50 : 1 by using a microprocessor with A/D and D/A converters. These audible waves are then presented binaurally through earphones. In this method, obstacles may be perceived as localized sound images corresponding to the direction and the distance of the obstacles. With it a blind person can recognize a 1-mm-diameter wire. It was also proved that the blind could discriminate between several obstacles at the same time without any virtual images. This mobility aid, modeled after the bats' echolocation system, is very effective at detecting small obstacles placed in front of the head.

However, most of the blind people can detect obstacles without these devices. This ability is called "obstacle sense". The figure 8 shows two steps of obstacle sense, first perception and final

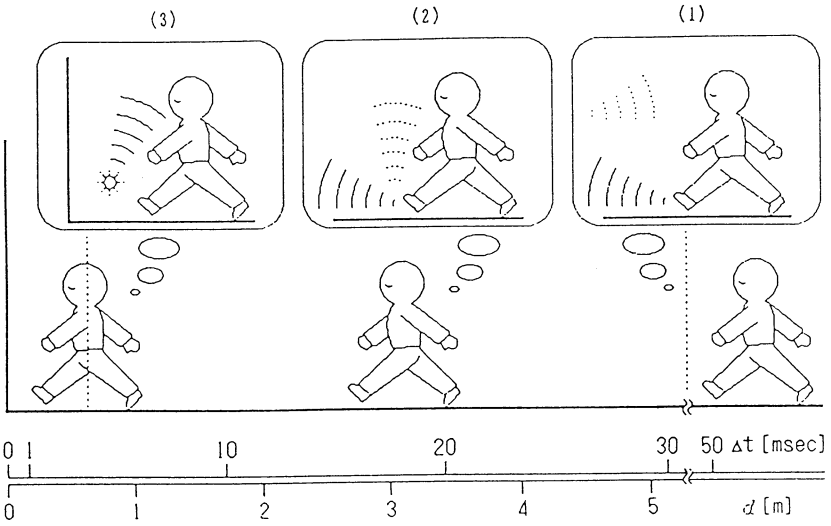


Fig.8. Two steps of obstacles sense ((3) final appraisal (2)first perception)

appraisal. We have been investigating the mechanism of the obstacle sense based on psychophysical experiments using blind students. We have found the reason why the blind can detect the obstacle is that they can discriminate the tiny difference of sound field between with obstacle and without obstacle. We are intending to design a mobility aid device which uses the ability of the obstacle sense. Furthermore, we could make the blind hear "virtual obstacles" by controlling the sound field produced from a speaker array. This study has also been related to virtual reality research.

9. Force Display

We have designed a force display which will be able to be used for virtual reality system by using metal hydride actuator. A metal hydride (MH) actuator, which uses the reversible reaction between heat energy and mechanical energy of a hydrogen absorbing alloy, has recently attracted much attention. Hydrogen absorbing alloys are capable of storing as much hydrogen gas as approximately 1000 times of their own volume. By heating the alloy, hydrogen equilibrium pressure increases and hydrogen desorbed, whereas by cooling the alloy, hydrogen equilibrium pressure decreases and hydrogen absorbed.

In this way, it is possible to utilize the mechanical energy of hydrogen gas pressure by manipulating heat. A Peltier element is used as a heat source, and by changing the direction of electric current to the element, the alloy is heated or cooled. The construction of the functioning part uses metal bellows, which insulate the hydrogen. Figure 9 shows an MH actuator. The drive

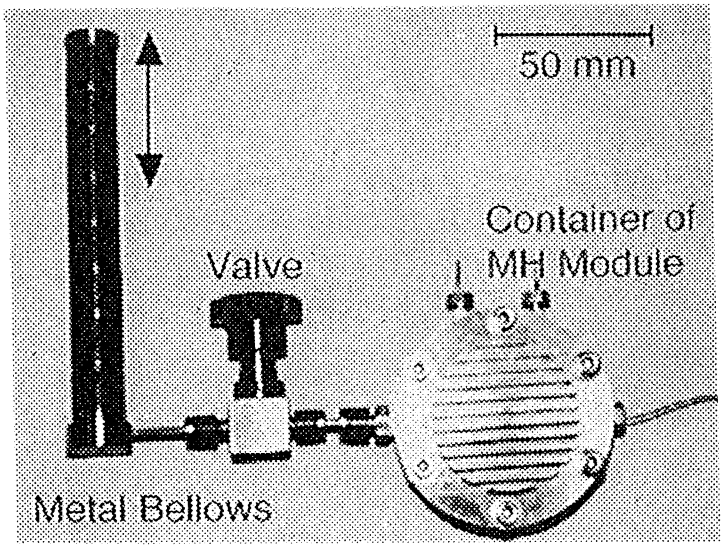


Fig.9. An MH actuator

function using hydrogen absorption and desorption has a buffer effect and it prevents extreme power changes or shock. Thus, this MH actuator is tender to humans and so suitable for use in equipment which is attached to human. The figure 10 shows a force

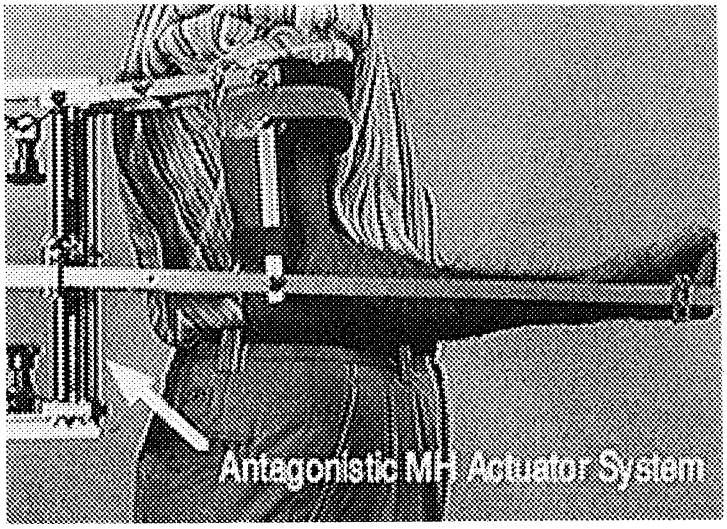


Fig.10. Force display

display in which two MH actuators are used. We have also developed a a robot hand with sensory feedback and a wheelchair with a lifter using an MH actuator with 40 g alloy in cooperation with a company as shown in figure 11 and 12.

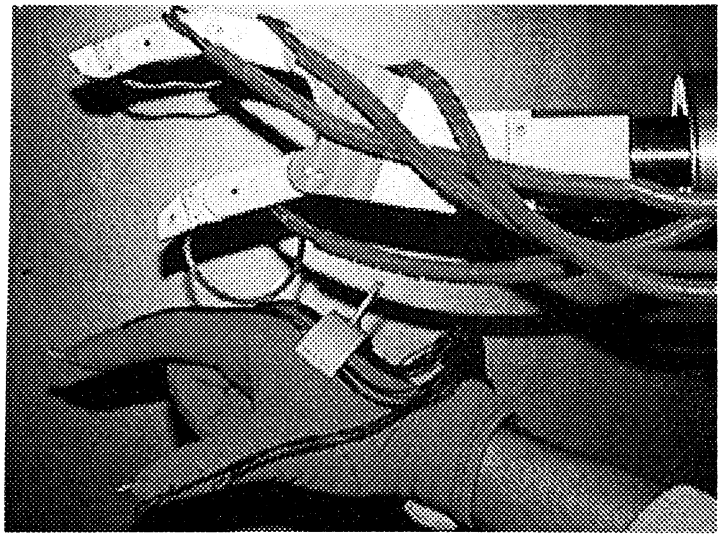


Fig.11. A robot hand with sensory feedback

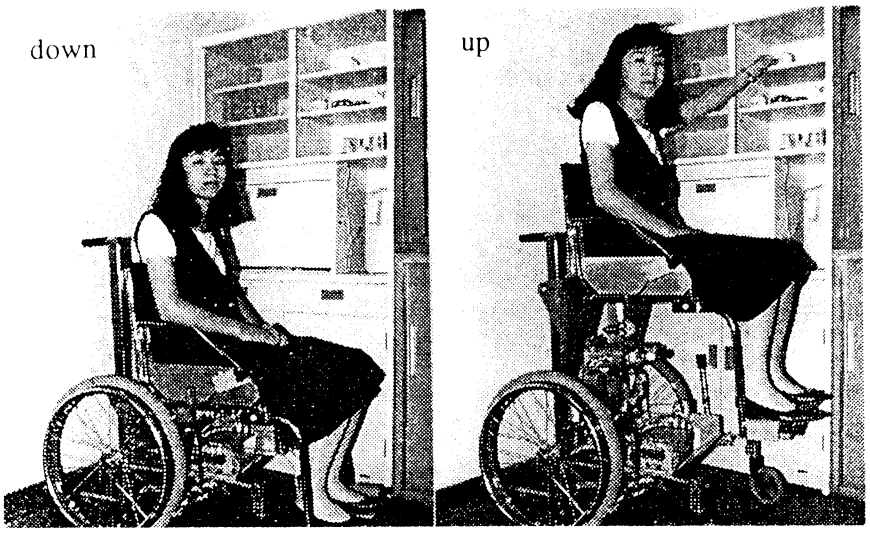


Fig.12. Wheelchair with a lifter

10. Conclusion

We have constructed a virtual reality system which consists of a head mount display, a speaker array, a rotational chair, and a sound proof room as shown in figure 13. The temperature of the room can be changed in the range from -4 degree to plus 40 degree. We are now investigating a mechanism of sensory integration. For example, we have investigated how the rotational stimulation influences the visual or auditory sensations. This kind of findings will be useful to design both better model of virtual reality system and rehabilitation devices.

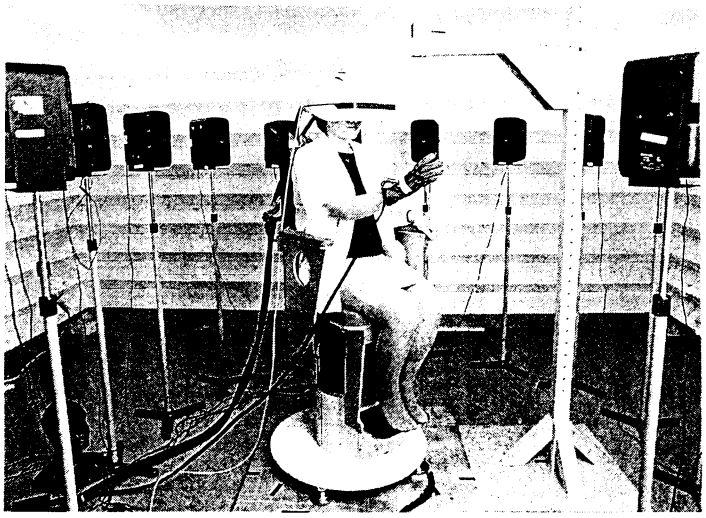


Fig.13. Virtual reality system

Present robots and computers are also disabled from the view point of their sensory functions. The basic findings concerning human senses and the substitute devices for the disabled will be useful for designing virtual reality systems and artificial senses for robots. These technologies of the virtual reality system and the artificial senses will be applied to design better models of communication and mobility aid systems for people with sensory disorder in the near future. This is my approach of sensory aid and virtual reality researches (figure 14).

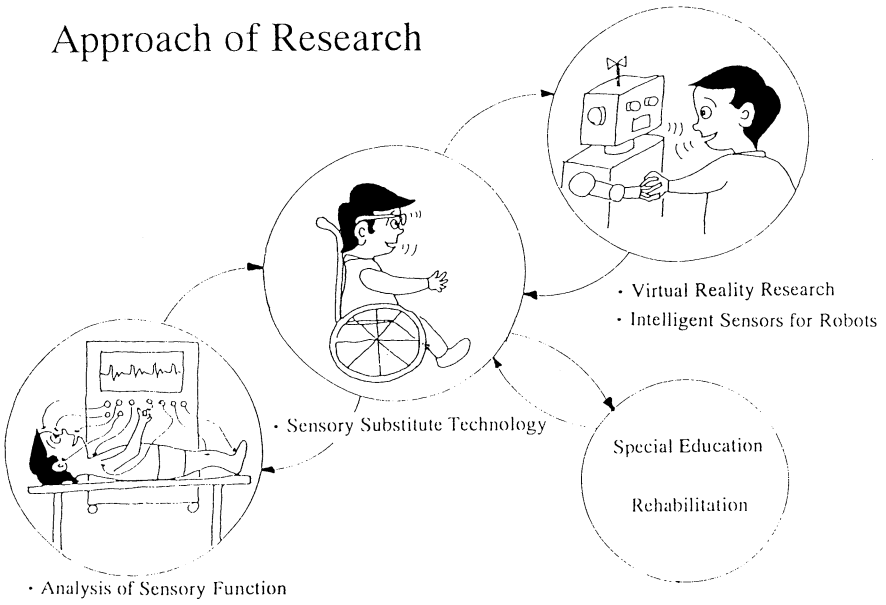


Fig.14. Approach of research

"The application of artificial life to interactive computer installations"

by Christa Sommerer (1) **& Laurent Mignonneau** (2)

(1) & (2) ATR Advanced Telecommunications Research
2-2 Amity Hikaridai, Seika-cho Soraku-gun, 61902 Kyoto, Japan
Tel: 81-77495-1426, Fax: 81-77495-1408 and 43-7612-47278 (Austria)
christa@mic.atr.co.jp, laurent@mic.atr.co.jp, http://www.mic.atr.co.jp/~christa

Abstract

Christa Sommerer and Laurent Mignonneau are working on the creation of interactive computer installation that combine artificial life and real life by means of human-computer interaction.
Focusing on real-time interaction and evolutionary image processes, visitors of their interactive computer installations become essential parts of the systems, by transmitting their individual behavior, emotion and personality to the image processes
of the work. Images in these installations are not any more static, prefixed and predictable, but become 'living systems' themselves, representing the minute changes of the viewers interactions with the evolutionary image processes.
Natural evolution has brought about a vast variety of forms and structures in nature. Sommerer and Mignonneau are interested in how artificial evolution can function as a tool of visual creation process: design should not any more be done by a designer or artist, but should emerge through the evolutionary image process itself.

1. Introduction:

Artificial Life has become a topic that is not only interesting to the scientific community but has influenced the art world as well [1].
Traditionally art was considered to be the sublime creation of the artist, who in Immanuel Kant's [2] terms was the genius, through who's inherited 'ingenium' nature decided the laws of art.
However, since the early 1920 and 30 artists such as Kurt Schwitters, Man Ray, Marcel DuChamps and others introduced the idea of 'art as a process' in a new art form called DADAism. Random processes and automatisms were implied for the first time, to create such then controversial works as for example Schwitters 'Ursonate' or Marcel DuChamps famous 'Ready Mades'.
In the 1950s artists like John Cage, Robert Rauschenberg, Nam June Paik and others started to also included the audience in the creation process, thus widening the concept of the art work to a further dimension. 'Happenings', 'Performances' and 'Video Art' were part of a new movement called FLUXUS [3]. It considered art as a communication process between the artist, the art work and the audience. Artists such as Steina and Woody Vasulka [4] investigated the new technical and conceptual possibilities of Video Art since the 1960s. Other developments since then explored the idea of the 'art as a process' further and new art forms such as 'Land Art", "Installation Art" and "Conceptual Art" were invented. With the appearance of the new computer technologies another dimension was introduced to the artistic creation process:
time and virtual space [5].

2. Artificial Life and Art:

Principles of Artificial Life and the emergence of advanced computer technologies in the early 1990 allowed the artists for the first time to study the visual creation process itself. One of the first artists to implement artificial life principles in the visual creation process is Karl Sims [6]. In his 'Genetic Images' he allowed visitors to choose images that then would develop by means of genetic cross-over and mutation. The resulting images represented a mixture of human selection and preferences as well as artificial genetics. Other artists and designers have been dealing with artificial life in a more commercial and game-like fashion, creating for example CD Roms like 'Sim Life' and others.

In 1993/94 Christa Sommerer and Laurent Mignonneau introduced their interactive computer installation "A-Volve", one of the first systems, where visitors could actually create artificial creatures, interact with them and watch them evolving [7].
In their collaboration with Tom Ray [8], a biologist and the creator of 'Tierra', they developed this interactive system by implementing principles and methods of artificial life into the artistic creation process. One year later in 1995 they developed "Phototropy" another interactive installation, that allows visitors to interact with virtual insects [12] by nurturing them and reproducing them.
In 1996 Sommerer and Mignonneau became interested in the building blocks of visual creation and investigated how simple structures can create complex looking shapes and forms by using genetic manipulations: they developed "GENMA - Genetic Manipulator" an interactive installation, which allows visitors to create, manipulate and explore the design and form of artificial insects.
These three interactive computer installations shall be described here in more details.

3. Interactive Computer Installations

3.1. A-Volve [9]:

In the interactive real-time environment "A-Volve" visitors interact with virtual creatures in the space of a water filled glass pool. These virtual creatures are products of evolutionary rules and influenced by human creation and decision. Designing any kind of shape and profile with their finger on a touch screen, visitors will "bear" virtual three dimensional creatures, that are automatically "alive" and swim in the real water of the pool.
Algorithms calculate the creatures form and their movement in space. "A-Volve" provides a novel system, where movement and fitness is linked to the design and shape of a creature.

3.1.1.) Creation:

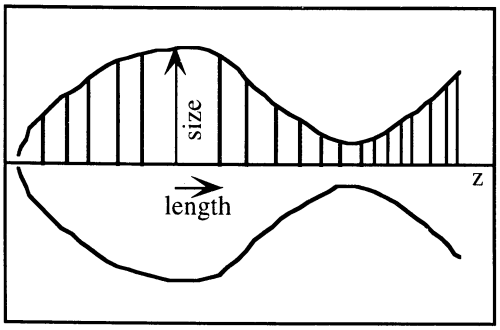


Fig. 1: Side view

A special touch screen editor created by L. Mignonneau, allows the visitors to draw any form, as shown in Figure 1. As he draws a side view of a creature, the outline of this drawing will be mirrored around the z-axis. Our software now subdivides this drawing into 20 points or vertices, that provide the length and size for each parameter point. We then add this size and length information to the genetic code of the creatures genetic string, like shown in Figure 3.

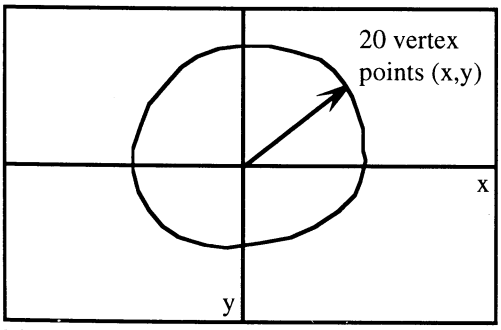


Fig. 2: Section

In order to get a three-dimensional form, we not only need a side view but also depth information. This time the visitor draws a form that represents the section through the creatures body along the z-axis (Fig. 2).

The same process of acquiring vertex points is applied to the creatures section. We again acquire 20 vertex point for x and 20 vertex points for y, all of which are added to the creatures genetic code, as shown in Figure 3.

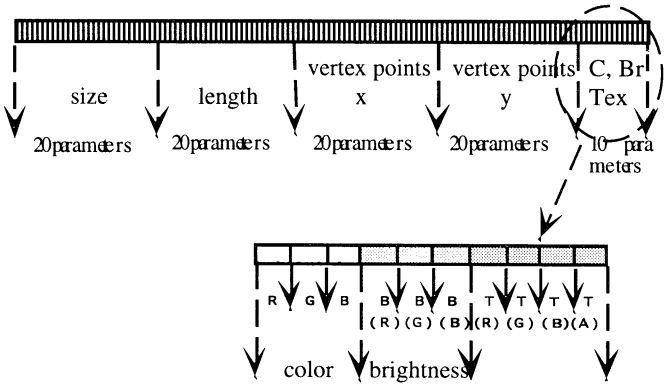


Fig. 3: Genetic String

Figure 3 shows the genetic string of the creature with its 90 parameters. Each creature in the pool has such a genetic string, the creatures are haploid and asexual.

3.1.2.) Fitness and Speed:

The movement and behavior of the virtual creature is decided by its form, how the viewer was designing it on the touch screen. Behavior in space is, so to speak an expression of form. Form is an expression of adaptation to the environment. Form and movement are closely connected, the creatures capability to move will decide its fitness in the pool.
The distance in which a creature pushes itself forward trough one muscle contraction, is its speed.
This speed is equivalent to the creature's fitness.
They more distance a creature can make with one muscle contraction, they faster its is, hence the fitter its is.
(for details see [9]).

3.1.3.) Predator - Prey Behavior:

The fittest creature will survive longest and will be able to mate and reproduce. The creatures will compete by trying to get as much energy as possible. Thus predator creatures will hunt for prey creatures, trying to kill them.
A creature is not born as a predator or a prey, but will decide whether to attack or to flee, depending on the other creatures fitness and energy at the current moment.
A creature can be a predator first, but finally become a prey when a fitter creature with more energy comes into the pool. The creatures therefore have to change their strategy constantly whom to attack and whom to flee. As the energy level (E) also decreases by each movement the creature does, it can become hungry again, even if it has already eaten. As soon as the critical level of E<1 is reached, the creature becomes a potential predator.
If it is then fit and fast enough to catch and kill other creatures, it will look for a suitable prey to attack:

through his vision system the predator always chooses the prey that is nearest of it and will provide the most energy with the smallest effort of movement.

3.1.4.) Mating and Genetic Exchange:

If two strong creatures with E >1 meet, they can create an offspring and a new creature can be born. It carries the genetic code of its parents. Mutation and cross-over provide a nature-like reproduction mechanism, that follows the genetic rules of Mendel. (for details see [9]).

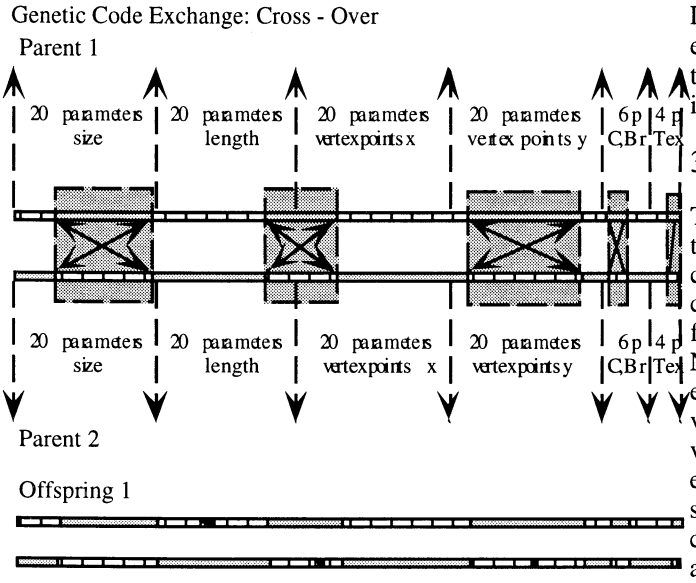


Fig. 4: Cross-Over and Mutations

Figure 4 shows an example of how the Cross-Over operation is performed between the parents.
This newly born offspring will now also react and live in the pool, interacting with visitors and other creatures. Algorithms, developed by Mignonneau and Sommerer ensure smooth and natural movements and "animal-like" behavior of the creatures.

3.1.5.) Birth - Death:

The "A-Volve" creatures can be born in two different ways:
1. Being created by the visitors on the touch screen
2. Being born by mating and genetic exchange of two parents creatures

They can die in 3 different ways:
1. Starving - they couldn't get enough energy by killing other creatures
2. Natural Death- the maximum life time was reached
3. Being Killed - a prey gets killed by a predator

3.1.6.) Evolution:

As the genetic code of the offspring is transported from generation to generation and the emphasis of the system is based upon selection for fitter creatures, the system is able to evolve over time towards fitter creatures. Although the evolution could take place by itself and without influence from outside, the system is designed in a way, the visitor and his interaction and creation of forms will significantly influence the evolutionary process. We can consider the visitors a form of an external selection mechanism.
The three main internal parameters, Fitness, Energy and Life Time, regulate the interaction, reproduction and evolution among the creatures. The external parameters are the visitors drawings on the touch screen as well as his interaction with the creatures.

3.1.7.) Interaction Creature - Visitor:

The creatures also interact with the visitors, by reacting to their hands movement in the water. If a visitor tries to catch a creature, it will try to flee or stays still, if it gets caught. The visitor so is able to influence the evolution by for example protection preys against predators.
None of the creatures is pre calculated, they are all born exclusively in real time, through the interaction of the visitors and the interaction of the creatures. Thus a large variety of forms will be possible, representing human and evolutionary rules. By closely connecting the real natural space of the water to the unreal virtual living space of the creatures, "A-Volve" minimizes the borders between "real" and "unreal", creating a further step, after "Interactive Plant Growing"[10], [11], in the search of Natural Interfaces and Real-Time Interaction.
"A-Volve" is installed permanently at NHK building, NTT Nagoya from July 96 to July 98.
More information about "A-Volve" is available at: http://www.mic.atr.co.jp/~christa and [9].

3.2. Phototropy [12]:

"Phototropy" is a biological expression describing the force, that makes organisms like for example bacteria or plants, follow the light, in order to get nutrition and hence, to survive.
"Phototropy" is an interactive computer installation where visitors interact with virtual insects through a normal flash light. By lightening onto parts of a 3x4 meters screen, the viewer will awake virtual insects, which are born in cocoon-like growth forms. The insects soon will start to fly and follow the real physical beam of the viewers flash light, seeking for their energy source, light. If the insects reach enough light, they can live longer and reproduce; reaching not enough light, they will die fast. Since all insects want to gain as much energy as possible, a big swarm of insects will follow all changes and movements of the visitors lamp.

The visitor has to be careful: lightening too much onto the insects, will burn them; but if carefully guided, they will mate more frequently and increase their population.
The genetic code of the single insect is transferred from the parents to the children: thus always new generations and individuals of virtual insects will be born, all following and fighting for light. The visitor in "Phototropy" thus supports, develops and enhances the live of artificially living insects populations.

3.3. GENMA - Genetic Manipulator [13]:

GENMA is a machine, that enables us to manipulate artificial nature of a micro scale: abstract amoeboid artificial three-dimensional forms and shapes.
Principles of artificial life and genetic programming are implemented in the creatures construction, allowing the visitor to manipulate their virtual genes in real time.
Looking into a mirrored glass box the visitor sees those creatures as stereo projections in front of him. By putting his hands into the glass box, he will try to grab the creatures that are virtually floating in the space of the box.
The genetic code of each creature is schematically displayed on a touch screen. By using his finger on the touch screen the visitor can manipulate the genetic code of the creature and thus in real time change and modifying its look and appearance in the glass box.
Selecting and merging different parts of the genetic string, and recombining them, he can engaging in more intense experiments and learn how to create complex forms out of seemingly simple structures at the very beginning.
Taking parts of the genetic strings, cutting, pasting or multiplying them, adding mutations and variations, GENMA allows the visitor to explore the tools of genetic manipulation.
On a visual level GENMA furthermore explores the concept of "natural design" or "auto design", a design that is not any more prefixed and controlled by the artists, but represents the degree of interest and interaction of each single visitor. Each visitor will be creating the forms he wants to see, aided by artificial genetics, mutation and manipulation. To push it further, the audience thus will become 'creators' or 'artists' themselves, using the power and possibilities of such tools.

4. Natural Interfaces:

Sommerer and Mignonneau have been most interested in the invention of Natural Interfaces, as they transport the concept of life, variation and personality. Using for example living plants as interface, does not only provide an interesting new connection between computers and a living being, but also poses the question of what a plant is, how we perceive it and how we interact with it.
Natural Interfaces allow us to transcend our personality into the virtual space. They also avoid the fear one has, when entering the virtual space. Sommerer and Mignonneau so far investigated the use of plants, water, unencumbered 3D full body detection ("3D Video Key" [14]) and light.

5. Multi-Layer Interaction and Artificial Life:

Interaction is interesting to the visitors if it is not linear or predictable but like a journey. The more one engages in interaction the
more one can learn about it and explore it. Non Linear Interaction and Multi-Layered Interaction should be easy to understand at the very beginning and should be rich, so one is able to discover continuously different levels of experiences. This is where Artificial Life can provide a new form of creation process, that is not pre-designed and predictable but varies according to the parameters employed in the systems.
In "A-Volve" for example artificial evolution and selection mechanisms create a system of virtual creatures, that are semi-autonomous and provide a variety and complexity of forms and interactions, that are not created by the artists or designer but by the interaction between the visitors and the artificial creatures themselves.

7. Conclusion:

Interactivity and Artificial Life will teach us to rethink our definition of art and will broaden our view, as it will allow us to integrate personality, variety, processes of nature and a new reflection on art and life itself.
The artist who creates such installations only provides the frame work: the visitors themselves now create the art work through their interaction with each other, with the system and with the image processes of the work. As the images in the installations are not any more static, prefixed and predictable, they become like living processes themselves, representing the influence of the viewers interactions onto them and the internal principles of variation, mutation and evolution. The images processes are not any more reproducible, but continuously changing and
evolving. The art work therefore could be metaphorically considered to be a living system itself [1], that represents the relation and interaction between life and artificial life.
And as already Friedrich Nietsche pointed out "all essential things appear *in spite of everything*" [15].

References:

[1] M. Kusahara, C. Sommerer, L. Mignonneau, "Art as Living System," *Systems, Control and Information*, Vol 40, No. 8, pp. 16-23, 1996.

[2] I. Kant, "Kritik der Aesthetischen Urteilskraft," in *Kritik der Urteilskraft,* B180,181, A 178, 179, pp. 241 ff, Suhrkamp Taschenbuch Wissenschaft, 1996.

[3] C. Goodman, "The Electronic Frontier: From Video to Virtual Reality", in *Info Art '95,* Kwangju Biennale Foundation, pp. 23 - 42, 1995.

[4] St. & W. Vasulka, *Machine Media,* San Francisco Museum of Modern Art, 1996.

[5] K. Stiles, "Art and Technology," in *Theories and Documents of Contemporary Art*, University of California Press, pp. 384-396, 1996.

[6] K.Sims, "Artificial Evolution for Computer Graphics," in *Computer Graphics*, Vol.25, 4, pp. 319-328, 1991.

[7] C. Sommerer, L. Mignonneau, "A-Volve: a real-time interactive environment," in *ACM Siggraph Visual Proceedings*, pp. 172-173, 1994.

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

[9] C. Sommerer, L. Mignonneau, "A-Volve - an evolutionary artificial life environment," in *Artificial Life V*, C. Langton et K. Shimohara, eds., MIT Press, in print, 1997.

[10] C. Sommerer, L. Mignonneau, "Interactive Plant Growing," in ACM Siggraph Visual Proceedings, pp. 164-165, 1993.

[11] C. Sommerer, L. Mignonneau, "Interactive Plant Growing," in *Ars Electronica 93* - Genetic Art Artificial Life, P.Weibel, ed., pp. 408-414, 1993.

[12] C. Sommerer, L. Mignonneau, "Phototropy," in *Oltre il villaggio globale - Beyond the Global Village*, M.G. Mattei, ed., Electra Edition Milano,pp. 134 ff, 1995.

[13] C. Sommerer, L. Mignonneau, "GENMA-Genetic Manipulator," in *Ars Electronica'96*- Memesis The Future of Evolution, Springer Wien-New York, pp. 294-295, 1996

[14] C. Sommerer, L. Mignonneau, "Trans Plant," in *Imagination*, Tokyo Metropolitan Museum of Photography, Chapter 2 ff, 1995.

[15] G. Bachelard, *Der neue wissenschaftliche Geist*, Suhrkamp Edition, pp.12, 1988.

Advances in brain research and future of artificially assisted neural functions

Masanori Otsuka
Institute of Bio-active Science
Nippon Zoki Pharmaceutical Co., Ltd.
Yashiro-cho, Hyogo-ken 673-14, Japan

In this lecture I will try to help the audience to understand how the brain works. I also wish to present my ideas about how experts of artificial life and robotics can contribute to brain science.

The human brain contains about 10^{11} neurons. It contains also glia cells and other kinds of cells, but neurons are the most important components.

A typical **neuron** looks like what is shown in Fig.1. Dendrites and the cell body receive inputs (either excitatory or inhibitory) from other neurons. The cell body integrates the inputs the neuron receives and as a result, produces (or does not produce) an electrical signal called **action potential**. The action potential is sent very rapidly along a thin long cable, called axon. When the action potential arrives at the ending of the axon, a chemical messenger (or messengers), called a **neurotransmitter**, is liberated and transmits a message to the next neuron through **receptors**. Neurons are connected by contacts, which are called **synapses**. A neuron is connected to many neurons so that the human brain contains about 10^{14} synapses .

The brain function is carried out by many neurons, and each neuron works as mentioned above. Mainly, during the former half of the 20th century, essential basic elements for brain functions, such as neurons, action potential, synapses, neurotransmitters, receptors, were clarified. In the 1950s, however, despite a good knowledge about these elements, brain scientists remained uncertain whether they could soon have a clear concept of how brain functions were achieved by these elements. The works of D.H.Hubel and T.N.Wiesel in the 1960s clearly showed that this was now possible.

When you see a lion, you run away. This is typical human behavior. In response to input from the outside world, you exert your action with your muscles. What happens in your brain during this behavior ? The scheme shown in Fig.2 is similar to that which Dr. Hubel often uses to explain brain functions.

A light is shone on receptors (photoreceptors, which should not be confused with neurotransmitter receptors) in the retina of your eye. Each receptor transmits the light signal to the next stage cells (bipolar cells). Each bipolar cell receives information from many receptors and sends signals to the next stage cells (ganglion cells). Each ganglion cell again receives information from many bipolar cells. Such processes continue for many stages. Finally motoneurons in the spinal cord are activated and send signals to muscles to cause movements. It should be noted 1) that during the information processing, only necessary information

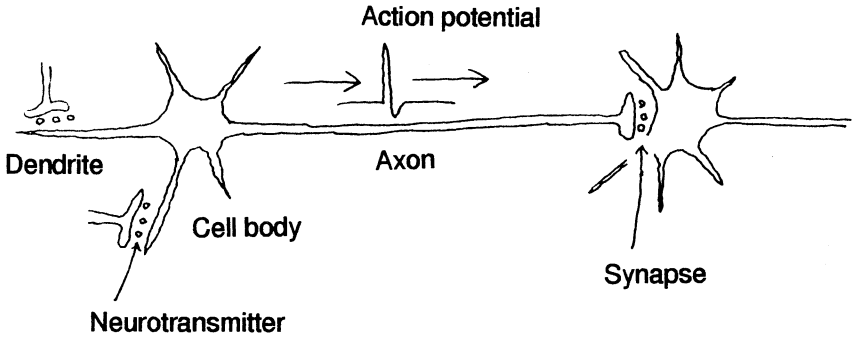


Fig. 1

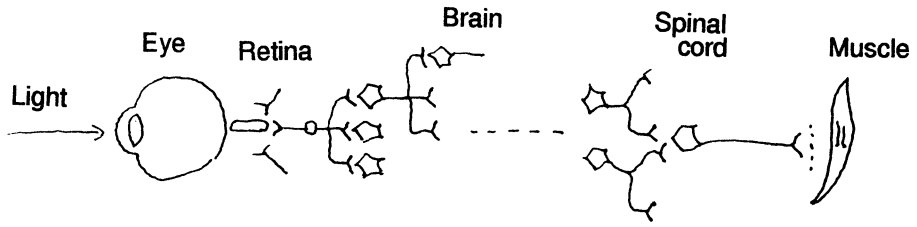


Fig. 2

is transmitted to next stages whereas unnecessary information is discarded and 2) that intense message is conveyed by high frequency action potentials, but each action potential is of the same size.

Now I wish to describe in more detail the advances of brain science in terms of several important items.

(1) Neuron

Toward the end of the 19th century, it has become gradually clear that the nervous system consists of neurons, which are not continuous but contact with other neurons. In other words, a neuron is separated from other neurons but not fused with them. This theory is known as the neuron theory, and a Spanish anatomist Ramón y Cajal was one of the major scientists to promote this theory. The contact between neurons was named **synapse** by C.S.Sherrington in 1897.

(2) Action potential

The first exact recording of an action potential was made by A.L.Hodgkin and A.F.Huxley in 1939 by inserting an capillary electrode into a giant axon of a squid. The diameter of this axon is about 1mm which is exceptionally large. The amplitude of the action potential is about 0.11V, and its duration 1.5 msec. The mechanism of the action potential was thoroughly clarified by Hodgkin and Huxley in the 1950s. Action potential is produced by initial Na^+ inflow through the surface membrane of the axon and later K^+ outflow (Fig.3).

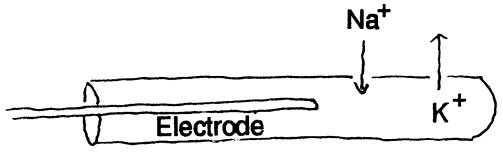


Fig. 3

(3) Synapse and neurotransmitter

At the contact between neurons (or between neuron and muscle), i.e. synapse, the signal (either excitatory or inhibitory) is transmitted from a neuron to the next by a neurotransmitter or transmitters. This process is called chemical transmission, a concept established in the 1920s. Many time-consuming studies revealed neurotransmitters one after another. Among these neurotransmitters, amines (acetylcholine and noradrenaline) are important in the peripheral nervous system whereas amino acids (glutamic acid, γ-aminobutyric acid, glycine) are important in the brain; some amines (dopamine, serotonin) are also important in the central nervous system, and some peptides play neurotransmitter roles in both peripheral and central nervous systems.

(4) Molecular neurobiology

Ions such as Na^+ and K^+ flow through specific ion channels in the surface membrane and neurotransmitters bind receptors to produce excitatory or inhibitory effect. Recent progress in electrical recording techniques made it possible to reveal the activity of a single ion channel or a single receptor. Furthermore, by the use of recent molecular biological techniques the mRNAs encoding the ion channels and the receptors were cloned so that the structures of channels and receptors as proteins were revealed.

(5) Cortex

As already stated, the works of Hubel and Wiesel in the 1960s showed that we can now have a clear concept of how neurons in the cerebral cortex (visual cortex) work together to process information conveyed from the outside world. They recorded activity of a single neuron in the visual cortex of a cat or monkey, let them see various visual images on the screen (Fig.4), and thus made clear the modes of activity of neurons in the visual cortex. For example, a typical neuron in the visual cortex is activated when a line is projected onto

a specific part of the retina in a specific orientation.

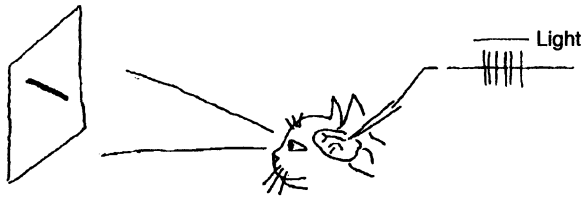


Fig. 4

Besides making clear the modes of activities of cortiocal cells during the processing of visual information, the works of Hubel & Wiesel gave answers to our several naive questions about the brain. **Is our brain blank when we are born and inexperienced ?** In the visual cortex of a very young and visually inexperienced cat, the activities of many cells were shown to be similar to those of adult cats, suggesting that many neural connections in the brain are already fairly well established at the time of birth. **Is there any clear difference between brains of young and adult animals ?** Hubel and Wiesel showed that a closure of one eye for a short period leaves a serious and irreversible damage for vision in young cats but not in adult cats, suggesting that proper uses of our brain at proper times (critical periods) are important for the full development of our innate capability. **Are brains of individuals the same at birth ?** The central visual pathway of Siamese cats was shown to be subtly but distinctly different from that of usual cats, suggesting that each individual has a genetically different, though similar, brain. These notions are easily acceptable ones but they have now neurophysiological bases.

We know that our brain works not only in response to the outside stimuli, but also when we have apparently no stimuli. In this respect, recent SPECT (Single Photon Emission Computed Tomography) study by H.Matsuda et al. revealed a surprising but very easily acceptable fact. Schizophrenic patients often tell us that they have clear auditory or visual hallucination and doctors are puzzled whether they are really experiencing sounds or visions. It was shown by SPECT that auditory center in the brain of a schizophrenic patient shows an activity which is similar to that of normal human hearing sounds when the patient appeals that he has an auditory hallucination.

Finally I wish to present briefly my thoughts as to how experts of artificial life and robotics can contribute to the brain science. I am sorry I have almost no knowledge about artificial life so that my proposals might be just nonsense. In modern life, we know that many blind people walk with white cane. No doubt, a few million years ago, when human beings started their existence, blind people walked in a similar way. What a pity in view of the vast progress of science and technology! To convey visual information into the brain of a blind person, we need to develop some means to stimulate cells in retina or visual cortex or to substitute for visual information other sensory stimuli, e.g. auditory or tactile. In this connection, it is important to know that the brain of a child is very different from the brain of an adult, and that the function of visual cortex is seriously damaged if a person becomes blind at birth but relatively intact if he becomes blind in adulthood. I wish to encourage experts of artificial life to have basic knowledge of brain science and collaborate closely with brain scientists. Our auditory, motor, or language ability may be likewise artificially assisted.

Parallel Computation of Distributed Genetic Algorithm on Loosely-coupled Multiprocessor Systems

T. Matsumura[1], M. Nakamura[1], J. Okech[2], and K. Onaga[1]
[1]Department of Information Engineering, University of the Ryukyus, Okinawa, Japan
[2]Department of Computer Sciences, Jomo Kenyatta University, Nairobi, Kenya

Abstract

In this paper, we propose and evaluate a distributed and parallel processing method of genetic algorithms on the loosely-coupled multiprocessor systems. In the experimental evaluation, we observe that the solution quality depends on the network topology of the system and the communication frequency. The solution generated in the hypercube system and torus system was uniformly better than the completely-connected system and the ring topology system has the possibility to generate higher quality solutions even if its convergence speed is slow.

Keyword: genetic algorithm, parallel processing, loosely-coupled multiprocessor system, network topology, optimization problem

1 Introduction

Genetic algorithms [1] are focused as one of meta schemes to solve combinatorial-optimization-type problems. Many papers reported its effectiveness and usefulness [1]. However, genetic algorithms generally take expensive computation cost. Thus, there are some proposals on parallel execution of genetic algorithms to reduce the computation costs [2]. By the way, the distributed computation of GA contributes to keeping variety of chromosomes to avoid immature convergence. Therefore, the parallel and distributed execution of genetic algorithms has the possibility to obtain the high speed computation and the high quality solution.

In our previous work [2], we proposed and evaluated parallel execution schemes of a genetic algorithm in distributed environments in which the chromosome set was divided (distributed) into the number of processors and each processor operated on the assigned chromosome set. We call this method "distributed genetic algorithms."We observed that the distributed computation of GA contributes to keeping the variety of the chromosomes to avoid convergence at the local minimum points.

This paper considers parallel and distributed computation of genetic algorithms on loosely-coupled multiprocessor systems. We propose a parallel execution method of distributed genetic algorithms in which each processor element executes genetic operations on its own chromosome set and each processor element communicate with only the neighbors in order to reduce communication overhead. From the previous work [2], we can guess that the solution quality of the parallel genetic algorithms depends on how the processor elements communicate each other. Therefore, the solution quality generated by our neighboring communication method seems to depend on the network topology of the multiprocessor system.

We in Section 2 describe the parallel machine model considered in this paper and in Section 3 propose a parallel execution method. Section 4 shows some results of the experimental evaluation. We conclude this paper in Section 5.

2 Parallel Machine Model

In this paper, we consider distributed and parallel computation of GA on loosely-coupled multiprocessor systems that are more suitable for massively parallel computation and easier VLSI implementation than tightly-coupled one. However, the communication overhead problem is serious for loosely-coupled system but not for tightly-coupled one.

Figure 1 shows the processor element of the loosely-coupled multiprocessor systems considered in this paper. Each processor element is composed of a computation processor for genetic operations and communication processors for communication handling. The processor elements are connected by communication link based on the network topology of the multiprocessor system.

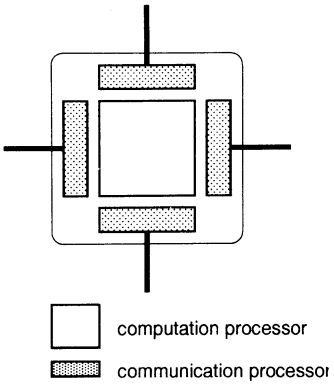


Figure 1: Processor Element

Genetic operations are executed on the computation processor and communication between two processor elements are managed by the communication processors. By function of the communication processor, genetic operations and communications can be performed simultaneously and asynchronous communication between processor elements can be realized.

3 Parallel Computation of Distributed Genetic Algorithm

In this subsection, we describe the detail of parallel execution of distributed genetic algorithms on loosely-coupled multiprocessor systems.

At the beginning of the execution, each processor generates its own chromosome set. Then as the ordinary genetic algorithm, each processor carries out genetic operations, crossover, mutation, and selection, on its population at each generation.

Since the isolated execution of genetic algorithms is not effective [2], each processor exchanges the copies of some chromosomes with the neighbors, called the chromosome migration. In our method, they exchange the copies of the chromosomes with the high fitness value to obtain high-quality solutions.

Their communications are restricted among the neighbors (we say two processors are neighbors when there is a directed communication link between them) to reduce the communication overhead. On the parallel processing of the searching-type program, the more communication among processors in general results in better quality solutions. However, when it comes to the parallel execution of genetic algorithms, it is not always true. We will show in Section 4 as experimental results that not so much communication avoids immature convergence of the solutions, that is, the migration interval and its size (the number of exchanged chromosomes) should affect the solution quality. Moreover, we will also report that the solution quality of our method depends on the network topology that determines how chromosomes spread in the multiprocessor system.

The procedure for each processor element PE(i) is shown as follows:

1. Receive GA parameters and a signal to start executing from the host processor.
2. Initialize a population POP(i).
3. Evaluate each chromosome in POP(i).
4. If it is time for migration, then send the copies of the best k chromosomes in POP(i) to each neighbor processor element, and receive those from each neighbor, and append all the received chromosomes to POP(i).
5. Select the chromosomes for the next generation by the roulette wheel way with the elite strategy. POP(i) is now renewal.
6. Create new chromosomes by applying crossover and mutation to chromosomes in POP(i) according to the crossover rate and the mutation rate.
7. If the termination condition holds, stop and send the best chromosome among POP(i) to the host processor, otherwise, go to 3.

The condition of the *if*-statement in Step 4 depends on the migration interval and k means the migration size.

4 Experimental Evaluation

4.1 The Multiple Knapsack Problem and Genetic algorithm

The zero/one multiple knapsack problem [3] consists of m knapsacks of capacities $c_1, \cdots, c_m$ and n objects, each with a profit p_i. The i^{th} object weighs w_{ij} when it is considered for inclusion in the j^{th} knapsack of capacity c_j. The problem is to find a vector $x = x_1, \cdots, x_n$ such that

$$\sum_{i=1}^{n} w_{ij} x_i \leq c_j \qquad j = 1, \cdots, n$$

and

$$P(x) = \max \sum_{i=1}^{n} x_i p_i$$

This is a combinatorial optimization problem known to be NP-Complete and many solutions rely on heuristic-based methods.

In our experiment, we applied our method to the multiple knapsack problem, since we can get problem instances known the optimum solutions.

4.2 Virtual Multiprocessor Systems by PVM

Using the real multiprocessor systems with several topologies is difficult because of the finance problems. Therefore, in the experimental evaluation we used network-connected workstations with PVM (Parallel Virtual Machine) [4] that is a free software providing communication and synchronization functions instead of the real multiprocessor systems. By the combination of workstations and PVM, we can construct virtual multiprocessor systems with arbitrarily network topologies, such as ring, torus, hypercube, and so on. Figure 2 explains the idea of the virtual multiprocessor system using PVM. Note that to use the virtual multiprocessor systems is enough to evaluate the solution quality generated by our parallel methods but not to estimate its speedup effects. Thus, we need to evaluate, by another way, the speedup ratio of our method compared with the sequential genetic algorithm execution.

For the experimental evaluation, the virtual multiprocessor systems were established on IBM RS6000 Power Stations connected by the Ethernet. The parallel genetic algorithm was implemented by C language including PVM synchronization and communication primitives. As communication network topology, we employed completely-connected one, hypercube, torus, ring.

4.3 Evaluation Results

We evaluate the solution quality generated by the proposed method to investigate the influence of the chromosome migration that depends on the migration interval, its size, and the network topology. In this experiment, we applied the method to the multiple knapsack problem and used the chromosome representation and the genetic operations proposed in [5].

We experimented on 48 problem instances of the multiple knapsack problem known their optimum values. These problem instances are available from the OR Library [6] where problem instances with the optimum solutions are collected for several kinds of the combinatorial optimization problems.

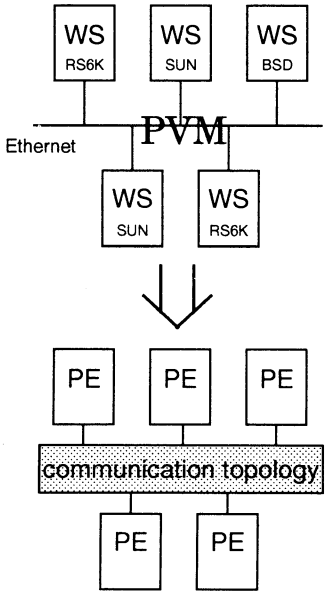


Figure 2: Parallel Virtual Machine

We picked up two classes of problem instances, Class1 and Class2, from the 48 problem instances according to the results of pre-experiments. That is, by the ordinary simple GA [7], for the problem instances in Class1 we could always obtain the optimum values, for those in Class2 we hardly obtain them.

Our parallel GA was run on the virtual multiprocessor systems described in Section 4.2. Through the experiment, the following parameters are used:

- total population size = 50
- maximum generation = 80
- crossover rate = 1.0
- mutation rate = 0.0
- number of processor elements = 16
- migration rate = 1

The other parameters, migration interval, migration size, and network topology, are varied at each case.

We ran our parallel GA at several cases. We observed the followings:

(1) For problem instances in Class1, we found no difference of the solution quality among the topologies, migration intervals, and migration sizes.

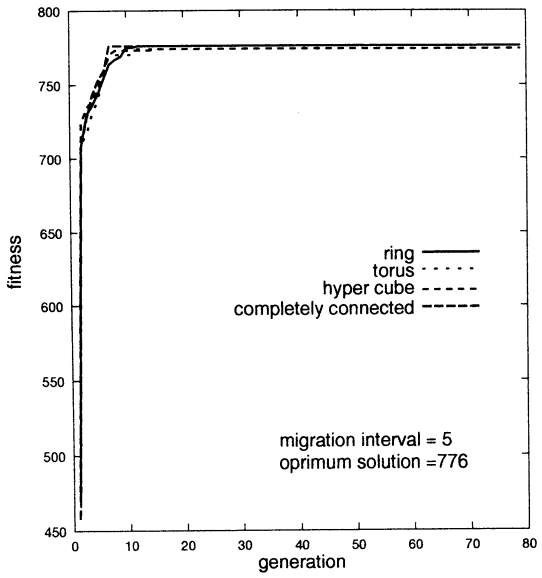


Figure 3: Result for problem instance pb6

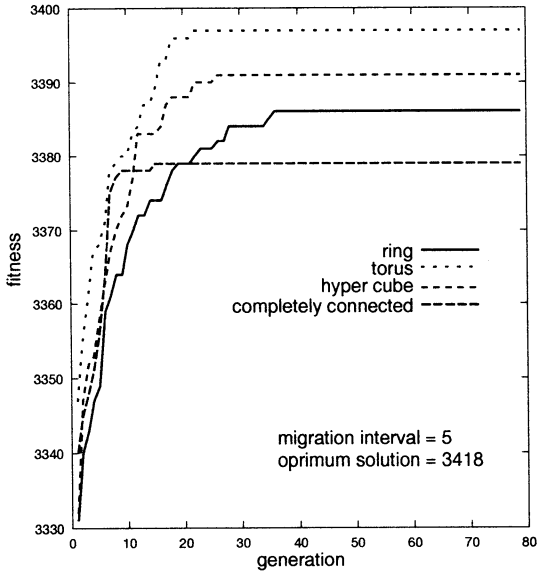


Figure 4: Result for problem instance hp1

That is, any combinations of the parameters are enough to obtain the optimum solution for the instances in Class1. Figure 3 shows the results of the execution for a problem instance(pb6) in Class1.

(2) For problem instances in Class2, when the migration interval is shorter, the solution was more often converged at local minimum points. When the migration interval is longer, the solution was converged more slowly. The solution by hypercube and torus were uniformly better than the others. That by completely connected network was worse than any others. That by ring has possibility to become better when we expand the generation steps. Figure 4 depicts the results of the execution for a problem instance(hp1) in Class2 with the migration interval 5.

5 Concluding Remarks

In this paper, we considered the parallel and distributed genetic algorithm on the multiprocessor systems. By experimental evaluation, we observed that the solution quality depends on the network topologies and the communication frequency. As future works, we evaluate the speedup ratio of our method and investigate parallel execution of genetic algorithms on the *massively* parallel multiprocessor systems.

References

[1] Srivinas, S. Patnaik, "Genetic Algorithms: A Survey," *IEEE Computer Society*, Computer, Vol. 27 No 6, pp. 17-26, 1994.

[2] J. Okech, et al, "A Distributed Genetic Algorithm for the Multiple Knapsack Problem using PVM, " *Proc. of ITC-CSCC'97, pp. 899-902*, 1996.

[3] Khuri, S. Back, T. Heitkotter, "The Zero/One Multiple Knapsack Problem and Genetic Algorithms," *Proceedings of the 1994 ACM Symposium of Applied Computation*, ACM Press 1994.

[4] Geist Al, et al, *PVM 3 User's Guide and Reference Manual*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1994.

[5] O. Miyagi, "Improvement of a Genetic Algorithm for the Knapsack Problems," *ME Thesis*, University of the Ryukyus, Japan, 1995.

[6] URL http://mscmga.ms.ic.ac.uk/

[7] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991

A New Hybrid GA Solution to the combinatorial optimization problems - An application to the multiprocessor scheduling Problem-

Morikazu Nakamura Beatrice M. Ombuki Koji Shimabukuro Onaga Kenji
Department of Information Engineering
University of the Ryukyus, Okinawa, Japan
morikazu@ie.u-ryukyu.ac.jp

Abstract

The multiprocessor scheduling problem is one of the classic examples of NP-hard combinatorial optimization problems. Developing exact optimization for such problems is a difficult process, rather researchers have tried to develop approximation algorithms and some of the most effective methods have been shown based on the list scheduling scheme. This paper presents a hybrid genetic algorithm approach to the combinatorial optimization problems.

Keywords: Hybrid Genetic Algorithms, Priority List, Multiprocessor Scheduling Problem

1 Introduction

Although past research indicates that some of the most effective methods to the combinatorial optimization problems is based on the list scheduling scheme[4], the quality of a scheme greatly depends on the quality of the priority list to be used. Recently Genetic algorithms, GAs have been shown to have potential for solving combinatorial optimization problems. To enhance the performance of GAs, the idea of a reinforcement of GAs with a heuristic technique tuned to local search effectiveness came up. In this paper we propose a genetized knowledge genetic algorithm, gkGA based on four reinforced GAs from the ordinary idea of scheduling research.

We first describe the performance of each of the four heuristics. Through experimental evaluation we find that each of these heuristics is problem dependent and thus non is uniformly superior than the others. Then we employ gkGA which automatically selects a scheme that is best suited for a given problem instance. By applying the gkGA to the multiprocessor scheduling problem we find that the proposed gkGA generally generates better schedules compared with the individual schedules and is not problem dependant.

2 The Multiprocessor Scheduling Problem

Let a task graph, TG = (V,E) be a directed graph with the vertex set V = $\{T_1, T_2, \ldots, T_k\}$ and the edge set E = $\{e_{ij}\}$. Suppose TG represents a process, where T_i = task to be executed and its execution time t_i, is given. An edge indicates a precedence relation between its two incident vertices. For example an edge from T_i to T_j implies T_i must be completed before T_j is initiated. The largest sum of execution time for a task T_i , for the path from T_i to the sink T_z is called "the time depth of T_i", denoted by $td(T_i)$.

The multiprocessor scheduling is to assign each task of the graph to a free processor from a pool of p processors such that each processor executes only one task at a time while maintaining the precedence relation a and then construct the schedule of the earliest finish time i.e, time to the last task completed.

2.1 Task Assignment Scheme **TAS**

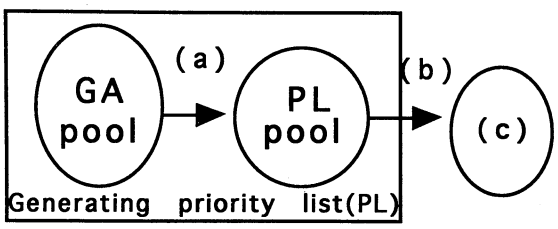
As pointed out in [4] there are some weak points of the conventional list scheduling scheme. we use a simple scheme denoted by " TAS " in our method.

** TAS **

1. Let t = top of the priority list PL and remove it from the PL.
2. If all the predecessors of t are already assigned, t is assigned to an idle processor available. The initiating time of t is defined by: $\max(\min_i\{idle\_t(PE_i)\}, \max_j\{c\_t(T_j)/T_j$ is an immediate predecessor task of t$\})$ where idle_t(PE_i) is the time that PE_i becomes idle and C_t(T_j) is the time that T_j is completed, otherwise, exit with failure.
3. If PL is empty, we have a schedule, otherwise return 1.

3 Reinforced Genetic Algorithms

In this section we introduce four heuristic reinforcements of GA based on well known heuristics as a prelude to the gkGA method.



(a) Conversion scheme
(b) Task Allocation Scheme (TAS)
(c) Searching space (schedules)

Fig 1. Outline the proposed Method

Figure 1 shows the outline of our method. The strings in the GA pool are converted and reshaped by a conversion scheme into a priority list and then a schedule is constructed by TAS.

3.1 Conversion of Strings into Priority Lists by Heuristics

We show here the four schemes, O, A, B and C to convert chromosomes into priority lists.

** Scheme O **

1. Let PL = s, i.e, a string is simply a priority list.

** Scheme A **

1. A string s is decomposed into sub strings $s= s_1 \mid s_2 \mid ....\mid s_k$ such that the genes in s_i have the depth i. The genes of s_i are put into a queue q_i from the left to right, for $1 \leq i \leq k$.
2. PL is made empty.
3. The top of each q_i, $1 \leq i \leq k$, whose predecessors are already assigned is chosen.
4. Then append the task T_j, that has the longest execution time, $t(T_j)$, among the tops chosen at 3 to the tail of PL. T_j is removed from the associated queue.
5. Return to 3 until all the queues are empty.

** Scheme B **

The steps for Scheme B are similar to those of scheme A except for step 4.

4. Append the task T_j, that has the most immediate successor tasks, $suc(T_j)$, among the tops chosen at 3 to the tail of P. Task Tj is removed from the associated queue.

** Scheme C **

The steps for Scheme C is similar to those of scheme A except for step 4.

4. Append the task T_j, that has the largest time depth, $td(T_j)$, among the tops chosen at 3 to the tail of P. T_j is removed from the associated queue.

3.2 String Representation and Genetic Operations.

(1) String representation and initial population

A sequence of task identifiers is used as a task string. In a given task graph, the strings in the initial population are randomly generated such that the task identifier is arranged monotonically according to the depth of the task, the largest first. An example of a task string from a task graph is 1 3 2 | 5 4 | 7 6 1 8 where a vertical bar differentiates between two adjacent different depths.

(2) Genetic operations

(a) Crossover of two strings

Two strings are randomly chosen and also a 0, 1 mask pattern with the same length as the strings is generated randomly. By applying uniform crossover new strings of the same depth monotonicity are generated from given two initial strings. For example P1 and P2, are parents producing children C1 and C2 and M represents a randomly generated mask pattern.

P1: 1 2 3 4 5 6 7 8
P2: 2 1 3 5 4 7 6 8
M: 0 0 0 1 0 0 1 1
C1: 2 1 3 4 5 6 7 8
C2: 1 2 3 5 4 7 6 8

(b) Mutation

A simple mutation operation is applied whereby two genes T_i and Tj at the same depth are randomly chosen from a randomly chosen string and then they are interchanged.

3.3 Description of the Reinforced GA

1. Initialization

The initial population is randomly generated from strings created as described in Section 3.2 above.

2. Evaluation of the fitness function
The fitness function of the GA is defined by the length of the schedule as follows;

f(s) = Large_num - max tpj

Where s is a string, tpj denotes the complete time of a processor PE_j in the schedule generated from s and Large_ num is a large positive number. Apparently a larger value of the fitness corresponds to a better schedule.

3. Genetic operations
For randomly selected strings, the uniform crossover and/ or the mutation in 3.2 is/are applied.

4. Reproduction
The roulette wheel selection is used to create new generation.

5. If time is up, stop and return the best string, otherwise go to 2.

4 Genetized Knowledge Genetic Algorithm , gkGA

In section 3 we have discussed how in each reinforced GA, one string conversion scheme based on a heuristic is applied to the strings for converting into a priority list. Table 1 shows that performance of each of the four schemes is problem dependent. We now introduce gkGA based on automatic selection of a string conversion scheme best suited for a given problem instance. Information on heuristics is also represented as genes on the strings. Three binary genes are prepared and attached to strings described in Section 3 to represent the four heuristic schemes O, A, B, and C. The first three genes of the strings are called "heuristic substrings" and the rest "task substrings." The genes in a heuristic substring corresponds to scheme A, B or C from left to right, respectively. For example, 100-xxxx..xxx corresponds to the priority list converted by applying scheme A to the string xxxx..xx. The substring 000- corresponds to the scheme O. Besides, hybridizing of heuristics is introduced here. The string 110- means applying hybrid heuristics of scheme A and B, say scheme AB. We hope that such hybridized heuristics may create a more appropriate priority list than a simple one.

** Conversion of Strings into a Priority List by Hybridized Heuristics **

Let the heuristic substring = $x_a x_b x_c$.

1 Decompose a task substring s into $s = s_1 \mid s_2 \mid \ldots \mid s_k$, such that the genes in s_i have the depth i. Then put the genes of s_i into a queue q_i from the left to right, for $1 \le i \le k$.
2 make PL empty
3 Choose the top of each q_i, $1 \le i \le k$ whose predecessors are already assigned.
4 By descending order of the execution time, sort the chosen tasks and for each T_j, define the score, $St(T_j)$, as the position of Tj in the above sort. Then sort the chosen tasks by descending order of the number of the immediate successor tasks and define the score, $Ssuc(T_j)$, as the position of T_j in this sort. Lastly sort the chosen tasks by descending order of the number of the time depth and define the score, $Std(T_j)$, as the position of T_j in this sort.
Let $SCORE(T_j) = x_a * St(T_j) + x_b * Ssuc(T_j) + x_c * Std(T_j)$.
5 Append the smaller $SCORE(T_j)$ task, T_j, among the tasks chosen at Step 3 to the tail of PL. Remove T_j from the associated queue. In case of a tie, the deeper task comes first.
6 Unless every substring is empty, return Step 3.

4.2 Genetic Operations and Initial Population for gkGA

For task substrings, the same operations as used in Section 3 are used in gkGA.
(1) Initial Population
Initial population is randomly generated such that heuristic substrings consists of 000, 100, 010 and 001.

(2) Crossover for heuristic substring
The two-points crossover is used for heuristic substring.

** gkGA **
1. Initialization:
An initial population is generated randomly as in Section 4.2.

2. Evaluation of each string: The fitness value of each

string is calculated according to the fitness function in Section 3.3 (A string is converted into a priority list t by the scheme in Section 4.1).

3. Genetic Operations for a task substring: For randomly selected strings, the crossover in Section 3.2 and/or the mutation(4) in Section 3.1 is applied.

4 Genetic Operations for Heuristic Substrings: For randomly selected strings the crossover (2) in Section 4.2 is applied and/or the mutation (3) in Section 4.2 is applied.

5. Reproduction: The population for new generation is generated by the roulette wheel selection

6. If time is up, stop and return the best string, if not, go to 2.

5 Experimental Results

Table 1 Experimental Results: data1 - data 5, task size of 50 is used and data6 - data8, task size of 100 is used.

	Scheme O	Scheme A	Scheme B	Scheme C	gkGA
data1	102	102	101	101	101
data2	86	87	86	86	86
data3	100	102	102	101	100
data4	97	98	97	96	96
data5	89	88	87	87	87
data6	206	204	201	203	200
data7	185	182	185	185	181
data8	179	173	179	180	172

The shaded parts show the shortest schedules in each problem instance.

The experimental evaluations were carried on a Sun SS5(75MHz) workstation using C language. Task graphs of sizes 50 and 100 were used with population size 100 and generation span of size 50. The mutation rates were 0.03 and 0.06 while the crossover rate 1.0 for the task sub strings. The heuristic substrings crossover/mutations were set at 0.0.

We observe from the Table 1 above that:

(1) performance of our four methods depends on the problem instance.

(2) gkGA generally produces better schedules at all problem instances.

6 Conclusion

Through computer evaluation,we have shown the effectiveness of gkGA by applying to the multiprocessor scheduling problem. Preliminary results also show that crossover of heuristic substrings leading to hybridized heuristics would improve the results of gkGA. As a future problem, we will do further experimental evaluation of gkGA on larger problem instances. We also think that gkGA can be applied to other scheduling problems like the Job shop problem.

References

1] Davis, L.ed. Handbook of Genetic Algorithms, Van Nostrand Reinhold,1991.

[2]Kasahara, H. and Narita, S., "Practical multiprocessing scheduling algorithms for efficient parallel processing," IEEE Trans.computers,vol. C-33, no.11, pp. 1023-1029,1984.

[3] Nakasumi, M., " A process scheduling by parallel genetic algorithm," Proc. of Fifth meeting of Special Interest Group on Parallel Processing for Artificial intelligence, pp. 1-4, 1994 (in Japanese).

[4] Kasahara, H., Parallel Processing Technology, Corona Publishing, 1991 (in Japanese).

Evolution of Vision System by Genetic Algorithm

Nian Wen Kozo Okazaki
FUKUI UNIVERSITY
Bunkyo 3-9-1. Fukui.910.Japan

Shinichi Tamura
OSAKA UNIVERSITY
Yamadaoka 2-2,Suita,Osaka,565,Japan

Abstract

This paper proposes a framework that a genetic algorithm is applied to determine and construct an organ,especially a neural network of a creature. The vision system of a creature is a result of genetic evolution. Thus. we are tring to realize it on the computer.

By using the method. we search how the visual organ of animal is evolved under a special environment(ex. specialized visual organ of animal to catch the moving insect),and how many variations of neural networks exist. Besides, we think it is possible to generalize the method to an automatic generation of various kinds of visual recognizing system by adding various kinds of evolution directions.

1. Introduction

Pattern recognition is one of the important tasks in intelligent information processing. Neural network is very attractive algorithm to carry out the pattern secognition . This paper proposes a framework that a genetic algorithm is applied to determine and construct a neural network as an organ of a creature. The vision system of a creature is a result of genetic evolution. Thus. we are trying to realize it on the computer. By using the method. we search how the visual organ of animal is evolved under a special environment(ex. specialized visual organ of animal to catch the moving insect),and how many variations of neural networks does exist. Here. we assume that the visual organ of a target creature is composed of neural network with arbitrary structure and connection weights. Input images to the vision system are composed of two kinds - various snake images and food images. The target creature will be eaten by the snake if misrecognize (when the input image is the snake), and it will be multiplied if correctly recognize the food when it meets with the food (when the input image is the food). In these environments, we try to search how the vision system of the creature is evolved by using genetic algorithm.

We implemented that the code representing the neural network of the vision system is embedded in the virtual creature's gene. The number of the input patterns are set as 40,100,407 and 814 . The probability of self_copy all is set as 0.4 when the creature is multiplied. The crossover rate with another individual is set as 0.4. We did the experiments by using four types of the neural networks and obversed their evolution results.

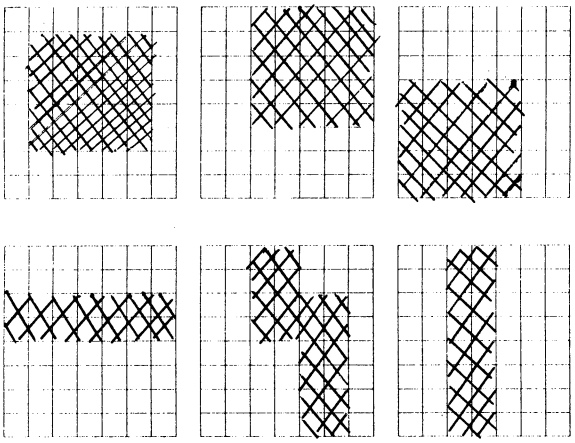


Fig. 1 input patterns of snake and food

2. Discerning network

Fig.1 shows input patterns which are snakes' and foods' shape by 8×7 pixels. The numbers of

input units corresponding with these are 56. We assume processing units are connected with eath other. Each connection between two units is biderectional with the same weight.For example, Fig.2 shows the connection network where there are 8 processing units. Calculation is done by asynchronously. The state of an unit to excite or suppress is determined by threshold value.The arrangements and the connection weights among the units are determined by genetics. The neural network of the vision system is evolved by genetic algorithm.We compared and examined four types of network shown below.

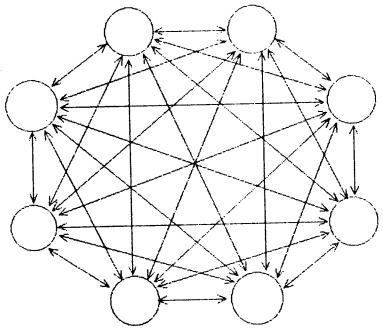


Fig.2 connection network structure

Type 1: It is shown in Fig.3 where there is one unit output layer which gives decision based on the global state of the processing units.We assume that the processing units of neuron are eight, and take connections with 56 input units. The value of connection weights can select -1 or +1.

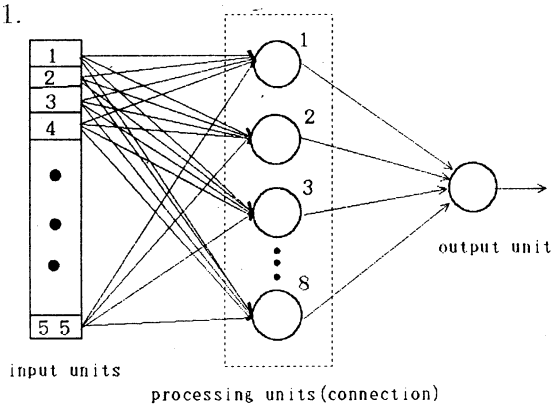


Fig.3 network structure of type 1

Type 2: We have no distinction between processing and output layer,and set processing units of neuron as nine.Each of the 56 input units can be connected with any processing units at the same time. The connection weights between two units can take -1 or 0 or +1 (the probability in which 0 is selected is set as 0.5). The ninth processing unit is set as output unit. It is shown in Fig.4.

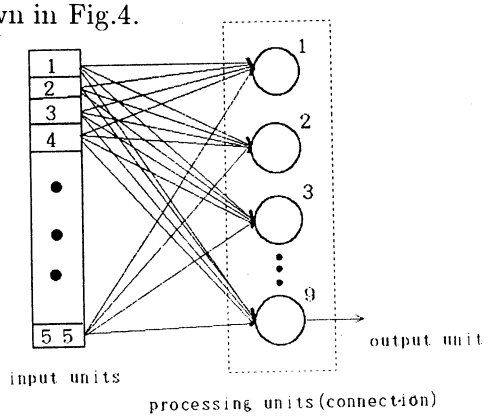


Fig.4 network structure of type 2

Type 3: It is assumed that processing units in the neuron is nine,and the number of connections from the input units to one processing unit is 20.The connection weights between two units can take -1 or 0 or +1 at the same probability.The ninth processing unit is regarded as output unit.It is shown in Fig.5.

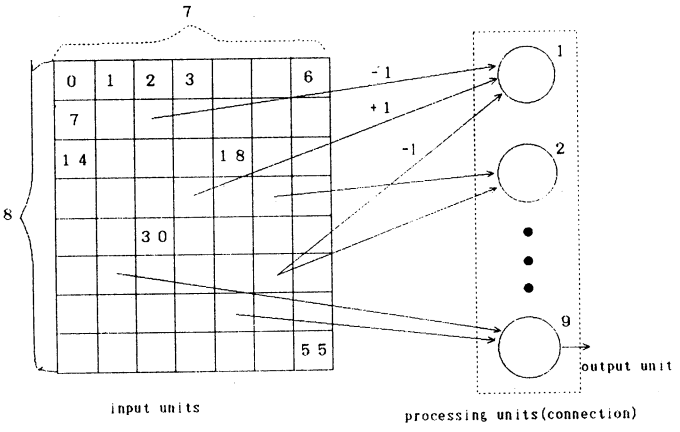


Fig.5 network structure of type 3

Type 4: There is one unit in output layer.We assume that the number of processing units is eight. And the number of connections from input units to one processing unit is set as 20.The connection weights between two units can take -1 or 0 or +1 at the same probability.

3. Model of genetic algorithm

3.1 Individual genotype

We must set form of creature's gene when we apply genetic algorithm. We assume that the code representing the neural network of the vision system is regarded as the virtual creature'gene. The creature's gene in the network of type 1 and 2 is composed of weights among all pairs of units . But,the creature's gene in the type 3 and 4 contains location of input units which are connected to the processing units, besides all union weights among units. For example, Fig.6 shows the form of the creature's gene of type 3. It is made of two parts. The first is formed by nine segments in which every segment expresses the input units address and connection weights from the input units to the processing unit. The other shows connection weights among nine processing units.

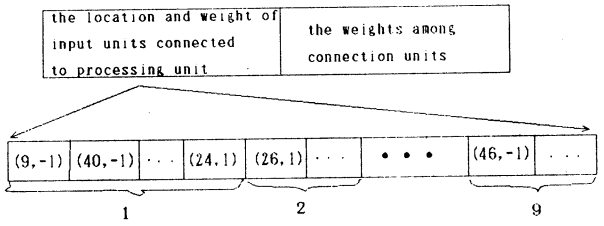


Fig.6 structure of creature genotype

3.2 generation replacing

The creature's generation replacing is done by conditions of life or death shown in table 1 as a result of action (recognition result) for the input pattern. The creature will die if approaches and eats the snake, and it will survive if runs away when the creature meets with the snake.When the creature meets with the food, it will continue alive even it escapes from the food . But it will multiply if can eat the food (shown in Table 1). In the multiplication it crosses with another creature which is chosen from living creatures randomly so that it gives birth to next generation. But, it is possible that the condition of surviving(e.g.there are forms of snakes and foods which are difficult to be recognized) is so strict that all of the creatures will die out. Even the creature which has excellent gene may make a mistake. Actually, it was not easy for creature to evolve when we ran the program.

Here, the number of creature is set small and the multiplication or death is made under the condition of food probability of 50% and every time it was forced to judge whether the target is snake or food. The number of indivividuals is more large and eates so many times in the real world. Therefore, we adopt an experience degree and the criterion function as shown in Table 2 and eq. (1) and (2) to stabilize the evolution by a kind of averaging.

One generation is set as one year. The creature's ages increase according to the number of generations after original birth. The purpose of introducing the experience is for the relief of an accidental mistakes and the value of it is increased or decreased as long as the creature survives. Initial value of all creature are set 0. If the value is inherited when the creature is copied or crossovered, it has a possibility of increasing the number rapidly and make worse the efficiency. Therefore, experience degree values of newly born creatures are set 0. The creature's experience degree is added 2 according to one generation if it can escapes from the snake when it meets with the snake,and if it can eat the food when it meets with the food. Or else, it is decreased by 2 if it can not correctly recognize them.The creature die when the experience degree becomes minus. If the number of creatures become two times that of the first stage,The individuals with least experience degree and priority-order are removed.

Table 1 condition of life or death

input	snake	snake	food	food
output	eat	escape	eat	escape
result	death	survive	multiplication	survive

Table 2 experience degree and condition of life or death

input	snake	snake	food	food
output	eat	escape	eat	escape
experience degree	-2	+2	+2(multiplication)	-2

$$priority\_order = f(experience\_degree) + \frac{experience\_degree}{creature\_ages)} \quad (1)$$

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

4. Result of experiment

We did the experiments in four type networks that we proposed in chapter 2 for the next four input pattern cases.

case 1: Number of patterns is 40.the form of patterns is complicated.

case 2: Number of patterns is 100. the form of patterns is complicated.

case 3: Number of patterns is 407. the form of patterns is not as complicated as case 1 and 2.

case 4: Number of patterns is 814. the form of patterns is a little more complicated than case 3.

Here, the first number of creatures was set as 50.In the multiplication. the probability of self_copy was set as 0.4.and the probability crossing with another creatures was set as 0.6.The crossover was done by uniform crossover where crossover rate was set as 0.65. The mutation rate was set as 0.05.

The tables(3-6) show the results we tested whether the above 50 creatures could recognize the snake and the food for the four input pattern cases. The first recognition rate was 50% approximately. The best in the tables was the recognition rate of the creature which has the highest recognition ability. The average value was the averaged recognition rate of 50 creatures.

From the tables, we can understand that the best and the averaged recognition rate of type 1, 3 and 4 was above 90%. Here,the result of type 4 is the most excellent and type 2 is the worst.

From the result of the type 2 . we can understand that when the 56 input units can connect any of the 9 processing units in the same time, the searching space becomes large, and the evolution becomes slow. Besides, the bigger the number of patterns is,the more unstable the evolution is.

The reason that the result of type 4 is the best may be as follows: Number of the processing units are 8 (smaller than type 3),and each processing unit can connect to as much as 20 of the 56 input units. Therefore, searching space becomes smaller and the evolution becomes faster.

Besides,the patterns' number of case 1 is fewer, the recognition rate is certainly higher. However,in case 2,the patterns' number is fewer than case 3 and 4.therefore, the form of patterns is rather complicated and the recognition rate is rather lower than case 3 and 4(shown in Table 3 and 5).

5. Conclusion

We had examined how the vision system composed of the neural network of a creature which can recognize the snake and the food will be evolved under the influence of circumstances.The results are :

(1) The evolution becomes better when we introduced the experience degree and gave chance of survival. Because even the creature which has excellent gene may make a mistake.

(2) It is better to recognize based on all of neuron's states rather than one's neuron set output units.

(3) Every processing units connected by only parts ofv the 56 input units is better than by all parts of the 56 input units.

References

[1] Hiromi Hirano: "Making mrural network using C-language," Personal media, 1991.

[2] Hopfield.J.J.: "Neurons with graded response have collective computational properties like those of two-state neurons," Proceedings of the National Academy of Sciences,81, pp.3088-3092, 1984.

[3] Yoh-Han Pao: "Adaptive Pattern Recognition and Neural Networks," Addison Wesley, 1989.

[4] J.H.Holland: "Adaptation in Natural and Artificial Systems," The Univ. Michigan Press (1975), MIT Press (1992).

[5] C.G.Langton et. al(Eds.): "Artificial Life 2, Proceedings of the Worhshop on Artificial Life," Feb.1990, Santa Fe, New Mexico, Addison-Wesley, 1992.

[6] T.Nagao,T.Agui and H.Nagahashi: Structural evolution of neural networks having arbitrary connections by a genetic method, IEICE Trans. INF. & SYST. Vol. E76-D, No.6, pp.689-

Table 3 the result of type 1 network

No. of input patterns		generation number												
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	12000	14000	16000
case 1	the best	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500
(40)	average	.96800	.97500	.97500	.97500	.97500	.97500	.97500	.97350	.97300	.97250	.97500	.97100	.97200
case 2	the best	.85000	.88000	.89000	.90000	.90000	.90000	.90000	.90000	.90000	.93000	.93000	.93000	.93000
(100)	average	.77900	.83840	.85800	.88340	.86560	.87120	.88140	.86720	.87660	.91540	.91820	.99260	.91580
case 3	the best	.92383	.92383	.94595	.94595	.94595	.95577	.95823	.95823	.97297	.97297	.97297	.97297	.97297
(407)	average	.91204	.89263	.92059	.94029	.94044	.93096	.94329	.93936	.93081	.96693	.96880	.97297	.96128
case 4	the best	.89435	.92629	.92629	.95455	.95455	.95455	.95455	.95455	.95455	.95455	.95455	.95455	.95455
(814)	average	.85115	.91405	.91332	.89133	.93032	.93445	.92636	.93012	.92676	.92111	.92840	.93803	.92479

Table 4 the result of type 2 network

No. of input patterns		generation number												
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	12000	14000	16000
case 1	the best	.92500	.92500	.92500	.92500	.92500	.92500	.92500	.92500	.92500	.92500	.92500	.92500	.90000
(40)	average	.90850	.90150	.90450	.89200	.90350	.90450	.90600	.90750	.90850	.90450	.90250	.88250	.84750
case 2	the best	.81000	.79000	.85000	.88000	.88000	.88000	.89000	.89000	.89000	.90000	.90000	.90000	.90000
(100)	average	.69600	.64740	.76140	.81420	.78860	.82520	.82380	.82320	.84160	.85440	.83640	.85400	.85940
case 3	the best	.85747	.85012	.89435	.91155	.91155	.92383	.92383	.92383	.92383	.88619	.89543	.89720	.89228
(407)	average	.79214	.74575	.77641	.87921	.87464	.86713	.90398	.89258	.88943	.88619	.89543	.89720	.89228
case 4	the best	.82064	.80221	.81450	.80467	.79730	.82555	.83661	.85995	.79853	.85504	.82924	.81941	.79853
(814)	average	.73381	.69197	.73489	.70440	.70840	.76651	.76371	.76796	.68700	.75860	.76228	.73206	.74636

Table 5 the result of type 3 network

No. of input patterns		generation number												
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	12000	14000	16000
case 1	the best	.92500	.92500	.97500	1.0000									
(40)	average	.90200	.89850	.97150	1.0000									
case 2	the best	.89000	.90000	.90000	.90000	.91000	.93000	.94000	.94000	.94000	.94000	.94000	.94000	.94000
(100)	average	.84300	.86280	.85340	.83740	.87520	.90500	.92080	.92560	.92020	.90920	.92180	.92400	.92560
case 3	the best	.89926	.94349	.94595	.94595	.95086	.95086	.95086	.95332	.95332	.95332	.95332	.95332	.95332
(407)	average	.82742	.92108	.92855	.92108	.93381	.93238	.93268	.93430	.93843	.93666	.93813	.94482	.93086
case 4	the best	.85504	.83661	.86118	.88698	.91400	.91646	.92138	.92260	.92260	.92138	.92629	.94349	.95700
(814)	average	.81189	.77135	.78818	.80108	.88430	.89233	.89278	.88452	.88943	.90017	.91140	.91179	.94361

Table 6 the result of type 4 network

No. of input patterns		generation number												
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	12000	14000	16000
case 1	the best	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500	.97500
(40)	average	.96350	.97500	.97500	.97500	.97150	.97500	.97500	.97500	.97500	.97300	.97500	.97500	.97300
case 2	the best	.97000	.98000	.98000	.98000	.98000	.98000	.98000	.98000	.98000	.99000	.99000	1.0000	
(100)	average	.95580	.96580	.97940	.98000	.98000	.98000	.97960	.98000	.98000	.98300	.99000	1.0000	
case 3	the best	.93857	.95332	.96806	.97297	.97297	.98526	.98526	.98526	.98526	.98526	.98526	.98526	.98526
(407)	average	.91430	.90595	.95292	.95450	.97017	.96860	.97091	.97258	.96560	.96948	.98521	.98526	.98526
case 4	the best	.86732	.96192	.97052	.98034	.98034	.98034	.98280	.98280	.98280	.98280	.98526	.98649	.98649
(814)	average	.79184	.92332	.94961	.97818	.98015	.97978	.98039	.97885	.98047	.98052	.98081	.98270	.98548

Emergence of Gaits of a Legged Robot by Collaboration through Evolution

Takashi Gomi and Koichi Ide
Applied AI Systems, Inc.
340 March Road, Suite 600
Kanata, Ontario, Canada K2K 2E4
Tel: +1 613 592 3030; Fax: +1 613 592 2333
E-Mail: gomi@Applied-AI.com
ide@Applied-AI.com

Abstract

Gaits of a legged robot were generated using evolutionary robotic techniques. A physical robot was shared by 50 software invoked control processes in each generation. Each program competed for a better score for its gait. A genotype that describes leg motions of each robot was defined and used in behavior generation and reproduction of offsprings. An evaluation function that favors early generation of collaborative movements for efficient forward motion of the robot was defined and tested in the experiment.

1. Introduction

In 1990, Pattie Maes conducted an experiment to generate gaits for a legged robot using learning [1]. She used reinforcement learning to encourage coordination among otherwise arbitrary and random motions of the legs of the six-legged Genghis robot. In order to avoid inflicting 'pain' generated by touch sensors installed on the belly of the insect robot, and to maximize the 'pleasure' implied by "distance being travelled" signal from an add-on wheel to the body of the robot, the robot adjusted the timing of leg motions. Within a minute or so, all six legs were coordinated in synchrony, and a nice `alternating tripod gait' emerged. Of course, there was no intentionality on the part of the artificial creature and the phenomenon was purely a physical one. In a subsequent experiment, we have confirmed the emergence of the gait using Genghis-II, a sister six-legged robot to Genghis.

Our attempt now is to conduct a similar experimental study on emergence of the physical phenomenon (a gait) using an evolutionary method. The eight-legged OCT-1b walking robot we had developed (See Figure 1 photo) is a revised version of an artificial creature developed for a museum exhibit called OCT-I [2]. It is now loaded with a sequence of 50 sets of control process (i.e., population size = 50) and run one at a time for the evolution of a proper gait. Software which represents each individual is given a fixed amount of time (or lifespan) in a time-shared fashion on the physical robot. The performance of the gait of each individual is evaluated for its fitness and fit individuals and their offsprings are incrementally selected through generations in the population. A number of evolution runs has been conducted during the past several months with modified parameters.

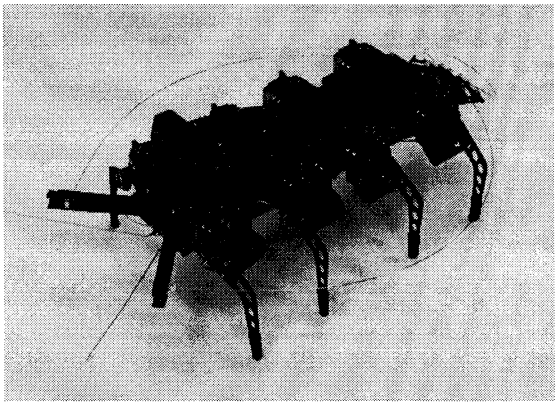


Figure 1. OCT-Ib Robot

2. The genotype

The genotype, shown diagrammatically in Figure 2, contains eight sets of genomes, each set describing characteristics of the motion of a leg. The characteristics are described by the amount of delay (up to 5 seconds) after which the leg begins to move, the current status of the leg (two bit status of the leg, one for the vertical motion and one for the horizontal motion indicating the direction of the leg's motion), end positions of both vertical and horizontal swings of the leg (-64 to +64, zero being center for both

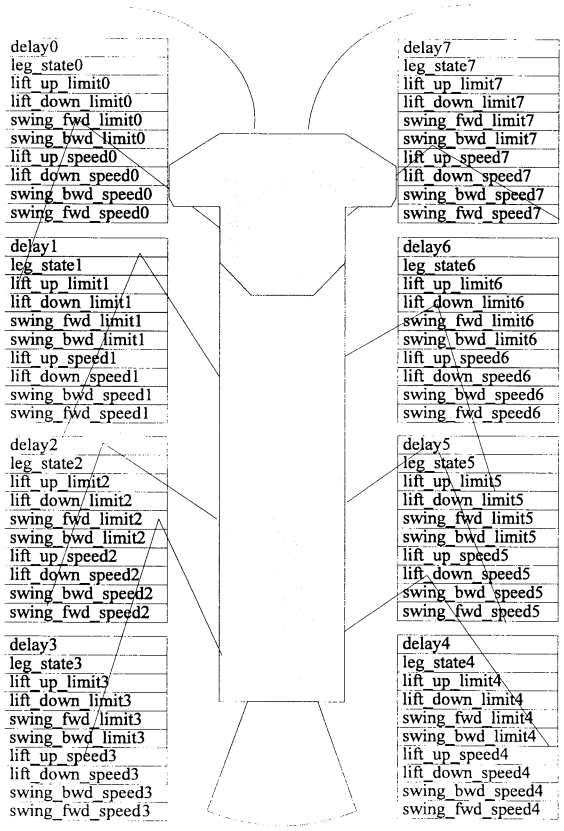


Figure 2. Genotype

down/up and back/front maxima), and vertical and horizontal angular speed of the leg (-10 to 10, negative being upward and forward speed). The parameter sets or the genome are repeated for each

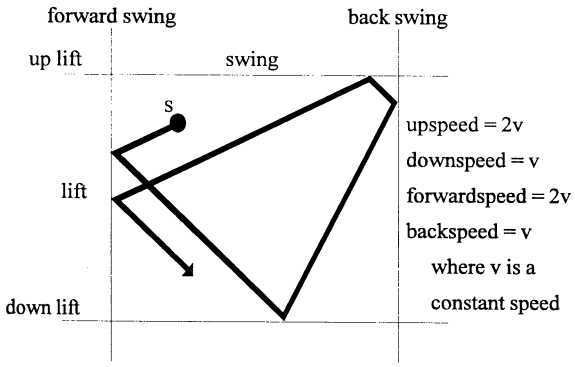


Figure 3. Sphere of leg's motion

robot. Each individual in the population is a process which enacts leg motions using a common software and in accordance with the genotype.

Figure 3 depicts the sphere of the leg's motion. The limits of the swings, the leg's angular speed, starting position determined by the delay, and direction of the motion decided by the status bits are shown. Since the physical robot is fully embodied and situated in the real world, actual motions of the robot's legs and those of the body also depend on the state of the creature's immediate environment and interaction between the environment and the robot.

3. The evaluation function

The evaluation function is set in favor of a robot which stands up, evolves coordination among its legs motions, and has a tendency to move forward. Positive feedback is given to an individual when a backward stroke (not forward motion) of a leg occurs while there is sufficient but not excessive loading on it. Current/over current sensor outputs for both vertical and horizontal axis of the leg are used to identify the "proper loading" condition on each leg. Similarly, if a leg swings back to a starting position (forward or negative horizontal movement with no load on the leg), then positive feedback is given. Other leg movements result in negative feedback. Also, if any of the touch sensors located on the belly of the robot is activated (meaning the robot is hitting the belly), then a demerit point is added.

The first term of the fitness function implies the collection of the leg's trajectories. When the leg lifts up and swings forward with no load, ***lift_swing***

$$Fitness = strides \times (1 - overcurrents) \times differences \times (1 - hits) \times 1000$$

where

$$strides = \frac{\sum_{i=1}^{lifespan} lift\_swing_i}{64 \times lifespan}$$

when the leg lifts up and swings back, lift_swing = 1.
when the leg lifts down and swings forward, lift_swing = 1.
otherwise lift_swing = -1.

$$overcurrents = \frac{\sum_{i=1}^{lifespan} \sum_{j=1}^{legs} (lift\_current_{ij} + swing\_current_{ij})}{lifespan}$$

$$differences = \frac{\sum_{i=1}^{lifespan} (up\_lift_i - down\_lift_i) + (forward\_swing_i - back\_swing_i)}{lifespan}$$

$$hits = \frac{\sum_{i=1}^{lifespan} belly\_hit_i}{lifespan}$$

becomes 1. When the leg pushes down and swings backward with a proper load, the term also becomes 1. Otherwise lift_swing is set to -1. This term represents how a forward motion would develop.

The second term describes the smoothness of legs' movement. When a leg swings or lifts, the driving

electric current is monitored at the corresponding servo motor. If the current is over the threshold set separately for swing and lift, an over current flag is counted. The sum of over current readings for all motors during the lifespan is tallied.

The third term is the possible range of leg motion which is calculated by summing up the possible range of motion for each axis over all legs. This term indicates how well the motions are in general during the lifespan.

The final term is the number of belly hits detected by two belly sensors. This term demonstrates how the walker is lifting the body.

At the end of each generation, an individual with the highest fitness score is allowed to continue in the following generation. The next five best fit individuals are selected as parents to produce 20 offsprings. A random selection of further parents is made from the rest of the population.

4. The experiment

While OCT-Ib robot has a relatively large number of sensors (active infrared sensors with the ability to receive modulated signals, contact sensors or whiskers, light sensors, motor current/over current sensors), only the motor current/over current sensors are used for this experiment. An additional set of belly contact sensors were added. This would detect and eventually eliminate individuals that are not capable of standing up. In order to allow lengthy evolution runs, an external DC power source is used instead of on-board NiCd batteries.

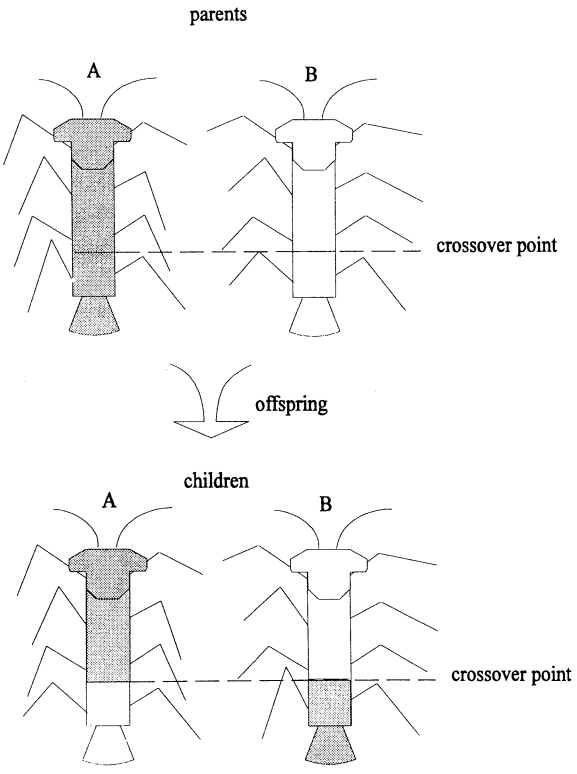


Figure 4. Crossover scheme

The manufacturer of the leg motors guarantees a minimum 100 hours of continuous operation before any electrical or mechanical failure. An individual in this experiment is given 40 seconds of lifespan. Thus the tethered robot can sustain at least 180 generations of experiments run at population size of 50.

In order to avoid costly loss of time by having to restart an evolution from scratch halfway through the experiment, an evolution backup system was implemented. It stores crucial data from each generation (i.e., individuals' genotype and the parameters for the generation itself). This allowed us to go back to any point in the history of an evolution run to resume the process from there, often with a modified evaluation scheme or evolutionary operators.

5. Gaits emerged

A good half of the population stops hitting the belly after only 5 generations, and after generation 10, most individuals, except for sporadic few, succeed in achieving stand-and-walk status with a faint gait. After several to a few dozen generations in a typical evolution run, the robot begins to show some noticeable forward movements. The results vary from run to run, and some specialization across the entire species often emerges throughout a run.

Some runs had to be "reset", as they obviously went astray, typically leaving a creature not capable of completely standing up, or they began walking in circles in the test field. Obviously, in these cases early convergence took place on an undesirable local peak in the fitness landscape.

Most dominant among the emergent gaits, however, was a mixture between alternating tetrapod and wave gaits. Two sets of legs, the left four and the right four, alternate in the former, while sequential wave-like propagation of leg motions backwards emanating the front legs signify the latter. In some of the runs, generation 21 or 22 became a watershed. Steady improvements in walking patterns were observed by then, but some deteriorated afterwards. Another common trend found was the way the robot swings its legs. It typically swings the leg backward slowly to evade over current detection and at the same time avoiding slipping or loss of friction with the walking surface. The leg motor draws between 100 and 200mA in such a cautious move.

In other runs, generation 35 became a peak, and

gaits diverged afterwards. In some of the generations, an odd individual learned to move backward. Observing such individuals, we discovered that typically one or two legs began developing a strong backward swing, likely by mutation, and this in effect overtook other the relatively quiet forward swing of other legs. Another oddity that happened once in several generation is a crawler like a sea turtle during an egg-laying expedition on the beach. In such an individual, the hind (last two) legs give the impression of being paralysed, while the forward legs push backward in unison to give the body a forward move. In one generation, about 20 per cent of the population was dominated by crawlers.

Using the physical robot one generation takes a minimum 33.3 minutes at a life span of 40 seconds per individual, if run continuously. In actual experiments, it typically takes one and one-half hours since the run is halted frequently for observations, recording, and modification of parameters. The longest evolutionary run so far is 57 generations.

6. Discussions

The emergence of specialization of gaits throughout an evolution run possibly implies less than optimal selection of evolutionary parameters (e.g., the method and rate of crossover, the rate of mutation, the percentage of individuals selected by Darwinian selection, etc.). More analysis is needed in this area.

The quick manner in which the belly-hitting is eliminated seems positive. However, in the eyes of an observer, this also appears as the emergence of cautiousness or even timidity very early in evolution, which sets the course for the rest of evolution. Most larger swings of the legs were eliminated as the creature tries to avoid 'falls' or accidental squatting that activates one of two belly contact sensors.

An experiment which took about 35 generations created a population of creatures most members of which abandoned the use of the mid-legs. We speculated that this is because of the rigid body structure which is unlike most natural creatures: it was not necessary to activate other than the four legs which supported the robot in the front and back. In many other runs, the middle four legs often bore the majority of the burden, possibly for a similar reason.

Although the guaranteed life of the motors was 100 hours, all motors lasted considerably longer than that allowing sufficient experimentation. The over current sensor was used to avoid shortening of life, as well as the 'proper contact' sensor. Application of the drive current is cut if it exceeds a threshold set at about 750mA. The impromptu stopping of a leg often resulted in achieving a synchronization of motion between that leg and others. This resulted in a form of gait coordination since some of other legs unloaded the load carried by the stopped leg. The leg resumes its contribution to bearing and walking only after the load is sufficiently reduced.

Despite having them in the genotype, the angular velocity did not play a major role in evolving gaits. The speed with which the legs are swung more or less converged to a relatively low average. Along with the timidity observed in swing ranges, we concluded that this was a cautious manner acquired by the robot through evolution. Similar precautionary slowing down in speed of an evolving robot was observed elsewhere [4].

In some of the runs, evolutionary parameters were changed during the course of the experiment, as often as every few generations. The backup system was effective in allowing frequent and impromptu halting, modification, and resumption of the run. In one such modified experiment, a system of feeding back subjective evaluation of robots' performance was implemented. This is to directly inject the subjective judgement of the observer of the experiment to the selection process. A keyboard is used to simply tag an individual if its performance is good. This results in a substantial hike in the value of its evaluation function, in most cases enough to have it selected for parenthood. In another run, the evolution was halted to introduce a copy function which copied high scoring leg parameters on one side of the body to legs on the other side across the center line of the body. Desirable results were obtained in both cases.

At this stage of the project we are not systematically chasing the evolutionary parameters. However, once a reasonable rough tuning of the parameters of the experiment is done, we would conduct a more systematic investigation of the runs.

References

[1] Pattie Maes and Rodney Brooks, "Learning to Coordinate Behaviors", Proc. AAAI'90, Boston

[2] Genjirou Nishi, Shigeru Shinbori, Takashi Gomi, Koichi Ide, and Patrick Maheral., "Mechanimals for Mechquarium". AAI Technical Report 96-03"

[3] Lawrence Davis ed., "Handbook of Genetic Algorithm", Van Nostrand Reinhold, NY, 1991.

[4] Dario Floreano and Francesca Mondada, "Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot", In From Animals to Animats 3: Proceedings of Third Conference on Simulation of Adaptive Behavior, edited by D. Cliff, P. Husbands, J. Meyer, S.W. Wilson. MIT Press.

Progressive Evolution Model Using A Hardware Evolution System

Tomofumi Hikage†, Hitoshi Hemmi††, and Katsunori Shimohara†
†NTT Human Interface Laboratories
1-1 Hikarinooka Yokosuka-Shi
Kanagawa 239, JAPAN
††Evolutionary Systems Department,
ATR Human Information Processing Research Laboratories,
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-02, JAPAN

Abstract

A progressive evolution model is proposed in which evolution takes place stepwise to match environmental changes. A verification system of the model is constructed on a hardware evolution system called AdAM (Adaptive Architecture Methodology) in which each individual circuit takes parallel input sequences and operates accordingly. A measurement is designed to express "environmental complexity" which is suitable for such a parallel simultaneous operations scheme.

Simulations using an artificial ant problem, a modified John Muir Trail, show that in progressive evolution model circuits can evolve such that complex behaviors can be obtained easily.

1 Introduction

The genetic method has been already used in various fields and its effectiveness is known. It is important for the genetic method to be able to solve difficult and large scale problems with the spread of application fields. There are several methods for dealing with such problems. For example, one method is to use a strong primitive. This method involves the use of GP (Genetic Programming) to extend GAs (Genetic Algorithms) to treat structured data. Another is a method to shorten the time required for processing one generation and thus enable many generations to be processed. This method is called hardware evolution. We propose another approach, a model to deal with difficult and large scale problems by using environmental changes as a source of evolution, i.e. a progressive evolution model. In the evolution process, conventional functions are maintained and new functions are acquired.

2 Hardware Evolutionary System: AdAM

The effectiveness of the progressive evolution model was verified on a hardware evolutionary system named AdAM[1]. We will briefly review this system. Figure 1 shows a system block diagram of the simulator version of AdAM.

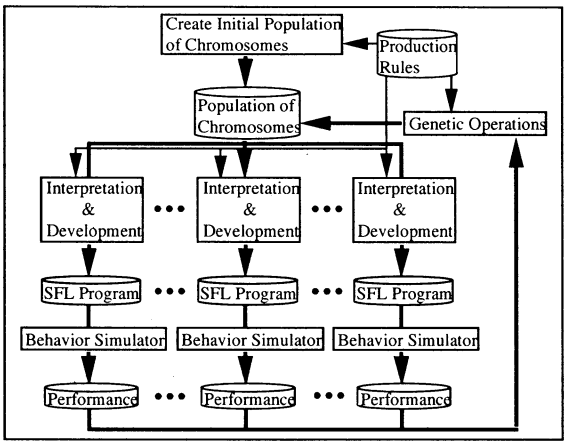


Figure 1: System Block Diagram of AdAM

The operation of the system is as follows. In Fig. 1, first, an initial chromosome population is randomly created according to a production rules set. These production rules are the Backus-Naur form of HDL (Hardware Description Language) named SFL, and chromosomes correspond to the parse tree of this language. Each chromosome generates an individual SFL program through interpretation and development. Individual SFL programs, i.e., individual hardware behaviors, are simulated and evaluated by a certain fitness

criterion. Selection and a set of genetic operations are performed on the chromosome population and produce next the generation of population. Such evolutionary processes eventually evolve individuals that encode for desirable hardware behaviors.

3 A Progressive Evolution Model

3.1 Basic idea

The purpose of the progressive evolution model is to accelerate evolution by exchanging environments. In the model, the population evolves in environments that are simpler than the target environment (the evolutionary system works in practice) and that require some functions that are useful in the target environment. We call such an environment a progressive environment. Evolution takes place stepwise to match the changes in the environment.

In conventional GAs (Genetic Algorithms) and GP (Genetic Programing), the population evolves in only the target environment to solve a problem. We can imagine that the time to evolve depends on the complexity of the target environment. If the complexity is high, the evolution system spends a lot of time solving the problem. If it is too high, the evolutionary system will cease to exist, because it can not produce an individual that is effective for the target environment.

3.2 The artificial ant problem

The artificial ant problem is a problem in which artificial ants adapt their behavior in a certain environment so that they gather food faster and more effectively; the best artificial ant is an ant that collects the most food in the fewest steps. Usually the environment is an arangement of food in a toroidal space of 32×32 cells (see Fig. 3). In our case, each artificial ant takes five input pins showing food existance and performs one of four actions.(Fig. 2)

3.3 A Measurement to express environmental complexity

As shown in Fig. 2, each ant has five inputs and two outputs. In the ant's inner mechanism, not only each input needs to be taken into account, but also arbitration among them is indispensable; an ant may encounter two different foods.

There are 96 combinations of inputs and actions. We call each such combination an action primitive.

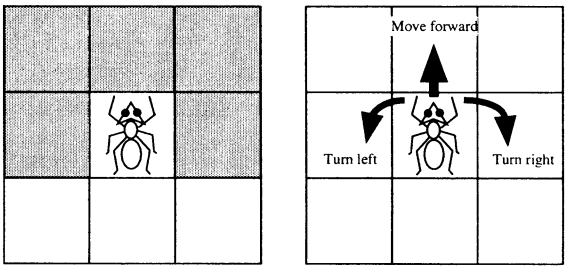


Figure 2: Artificial Ant

Action primitives can be used to express the difficulty in adapting to an environment,i.e. the more action primitives the best ant must have, the harder the environment is to adapto to. We call this number the complexity of the environment. [1]
For instance,
{input from front : "move forward"}
{input from front, input from right : "turn right"}
are two examples of action primitives.

4 Progressivelly changing environments

To test the progressive evolution model, we designed a series of environments for artificial ants. Figure 7 shows the final target environment, and Figs. 3–6 show intermediate environments leading to the target environment. To take all the food in each environment, ants must obtain several action primitives. The target and the intermediate environments are designed so that the obtainment occurs incrementally.

The first intermediate environment in Fig. 3 requires four action primitives:
{input from front:"move forward"}
{input from right front:"move forward"}
{nothing:"move forward"}
{input from right:"turn right"}
So the complexity of this environment is 4.

The second intermediate environment in Fig. 4 requires three action primitives in addition to those of the first one:
{input from left front:"move forward"}
{input from front and left front:"move forward"}

[1] Of course, in sequential circuit phenomena, action primitive should be context sensitive; that is, action to the same inputs may vary according to situation. But the general idea of environment complexity should be useful for sequential circuits.

{input from left:"turn left"}
So the complexity of the environment is 7.

In the same way, the third intermediate environment in Fig. 5 requires two more action primitives in addition to those of the previous ones:
{input from front and right front:"move forward"}
{input from right and left front:"turn right"}
So the complexity of the environment is 9.

The fourth environment in Fig. 6 adds three action primitives:
{input from front and left front and left:"move forward"}
{input from front and right front and right:"turn right"}
{input from left and left front:"turn left"}
Thus, the complexity is 12.

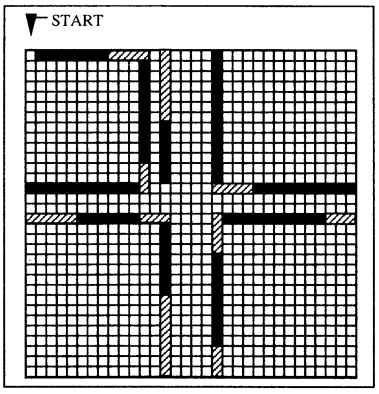


Figure 3: Progressive Environment 1

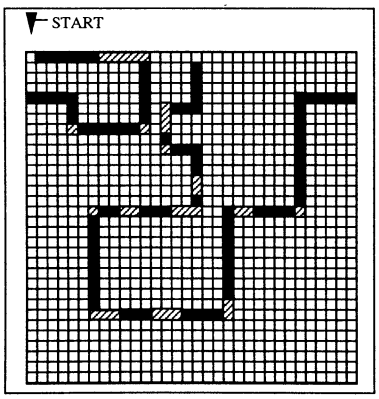


Figure 4: Progressive Environment 2

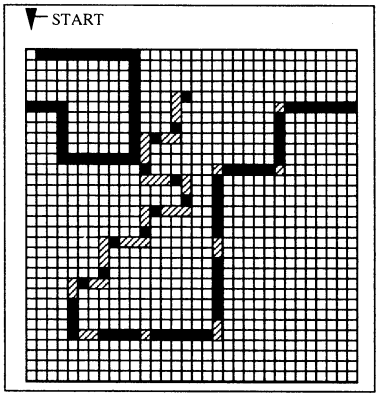


Figure 5: Progressive Environment 3

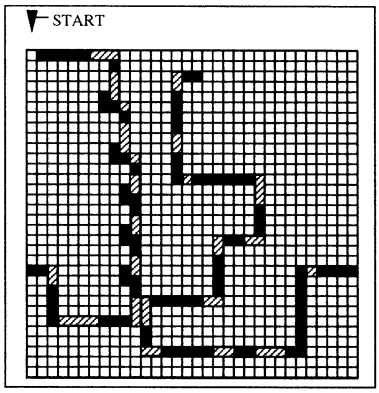


Figure 6: Progressive Environment 4

Finally, the target einvironment requires two more action primitives in addition to those of the previous ones:
{input from front and right front and left front : "move forward"}
{input from right and left : "turn right"}
And thus the complexity measure is 14.

The timings to exchange the environments are set five generations after the best individual appears in each environment. In addition, the experiments use an elite strategy so that the best individuals spread through the population and most of the circuits adapt to the environment during the five generations.

5 Result

The experimental result obtained in early generations with the progressive evolution model is shown in Fig. 8. In this figure, the best individuals in the first, second, and third intermediate environments appear in the lst, 11th, and 18th generations respectively. For

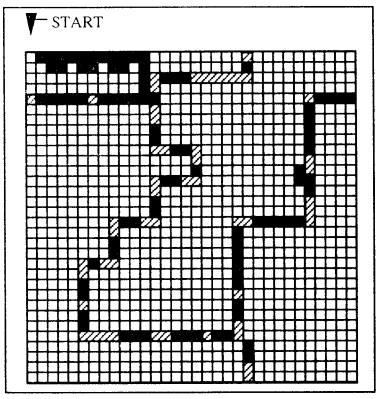


Figure 7: Target Environment

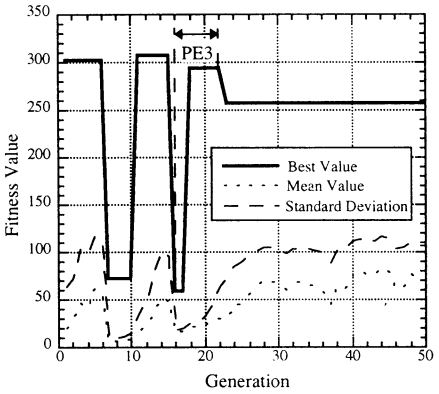


Figure 8: Result of Progressive Evolution Model (early generation)

comparison, Fig. 9 shows the result obtained for evolution in the third environment with an ordinary evolution model. In this figure, the best individual appears in the 77th generation, indicating that a progressive model is more efficient than an ordinary model.

Figure 10 shows the fitness transition for the generations as a whole. It takes about 7,900 generations to produce the best individual in the fourth environment. The reason for this result is considered to be that the action primitives which the environment requires are much more difficult to obtain than the previous ones. But once the primitives are obtained, only one generation is needed to obtain another two (equally difficult) action primitives and adapt to the final environment.

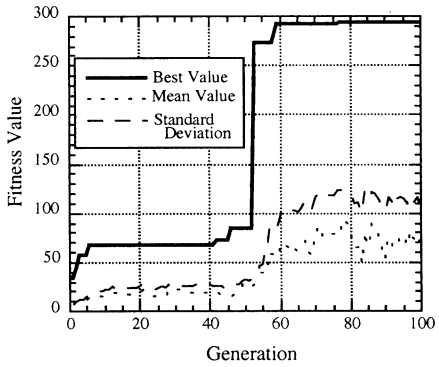


Figure 9: Result of evolution in only PE 3

6 Conclusion

This paper has proposed a progressive evolution model, and a complexity measurement suitable for hardware evolution. Experiments with artificial ant problems have demonstrated the effectiveness of the progressive model.

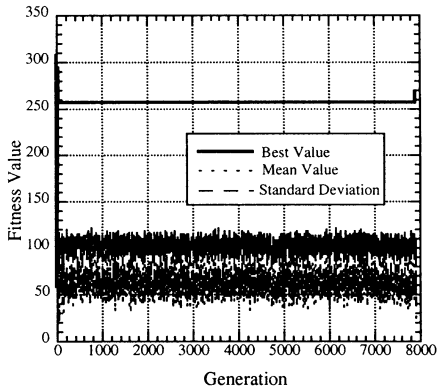


Figure 10: Result of Progressive Evolution Model

References

[1] H.Hemmi, M.Jun'ichi, K.Shimohara, "Development and evolution of hardware behaviors.", In *Artificial Life IV*,1994

Artificial Life Model Describing Developmental Processes of Plants: L-system Model Reflecting on Cell-Differentiation and Cell-Death

Y. Arima, M. Okamoto*

Dept. of Biochemical Engineering and Science,
Kyushu Institute of Technology,
Iizuka-city, Fukuoka 820 Japan
*(fax): +81-948-29-7801, (e-mail): okahon@bse.kyutech.ac.jp

Abstract

We have proposed new L-system model reflecting on cell-differentiation and cell-death by introducing Kaneko's dynamic clustering model. In addition to the rule of L-system, we have included the following main factors reflecting on cell-growth and cell-differentiation of each cell: (i) feeding schedule of nutrient transported from root; (ii) threshold value causing cell-differentiation (switching to next process in the rule of L-system); (iii) threshold value for cell-death or starvation; (iv) decay rate constant of absorbed nutrient. Based on the L-system rule describing the development of scale leaves of Japanese Cypress, we have simulated the temporal morphogenetic development of the tree and examined the effects of proposed new factors on the developmental process (shape-change) of the tree. We found that the temporal developmental phase and shape in steady state are drastically changed by these factors even if the rule of L-system is fixed.

KEYWORD : Lindenmayer-system, morphogenesis, cell-defferentiation, cell-death, dynamic clustering.

1. Introduction

L-systems [1,2] were introduced by Lindenmayer in order to model morphogenetic processes in growing multicellular organisms. Focused on the development of scale leaves of Japanese Cypress (*Chamaecyparis Obtusa*, Fig. 1), Nishida [3] has proposed the rule of L-system model producing similar tree-like shapes to observed Cypress. On the other hand, Kaneko [4] has proposed a novel mathematical model explaining the mechanism for cell differentiation based on the dynamic clustering in a globally coupled nonlinear system. His model of cell growth consists of (i) nonlinear dynamics in each cell; (ii) nonlinear and global interaction among cells through a medium; (iii) growth and death of a cell depending on its internal state.

By introducing these factors, he found that disparity in activities of cells emerges, that is, cells are classified into two completely distinct groups; the population of cells in one group is very few but they contain large concentration of chemicals, while the population of the other group is large but the concentration of the chemicals are much smaller. Indeed the observed plants seem to have a similar glowing mechanism to this; every cell in plant has a different nutrient condition with time and it will be destined or programmed to develop or differentiate or die in accordance with amounts of transported nutrient.

Fig. 1. Photo of the observed scale leaves of Japanese Cypress.

In this study we have proposed new L-system model reflecting on cell-differentiation and cell-death by introducing Kaneko's dynamic clustering model. In addition to the rule of L-system, we have included the following predominant factors reflecting on cell-growth and cell-differentiation of each cell: (i) feeding schedule of nutrient transported from root; (ii) threshold value causing cell-differentiation (switching to next process in the rule of L-system); (iii) threshold value for cell-death or starvation; (iv) decay rate constant of absorbed nutrient. We also considered differentiation noise value; when a cell differentiates, child-cells having almost identical nutrient are formed according to the rule of L-system. "Almost" here means that amount of nutrient in each child-cell is imbalanced or distributed within a range of small noise [4]. Based on the L-system rule describing the development of scale leaves of Japanese Cypress, we have simulated the temporal morphogenetic development of the tree and examined the effects of proposed new factors on the developmental process (shape-change) of the tree. We found that the temporal developmental phase and shape in the steady state are drastically changed by these factors even if the rule of L-system is fixed.

2. L-system model of Japanese Cypress

Our L-system is basically the same as one proposed by Nishida [3], and can be represented by as follows:

$$\begin{array}{llll} A \longrightarrow C\,D\,B & B \longrightarrow C\,E\,A & C \longrightarrow C & \\ D \longrightarrow F[-B] & E \longrightarrow F[+A] & F \longrightarrow F & (1) \end{array}$$

where A, B and C represent the different kinds of vegetative apex, and D, E and F shows resulting scale leaf from vegetative apex as shown in Fig. 2-(a). A bracket shows the branching, which belongs to "turtle interpretation [2]", and the signs + and - represent turn left by angle δ and turn right by angle δ, respectively; in this study δ was fixed at 22.5°. The development of shape change in accordance with rule (1) is shown in Fig. 2-(b).

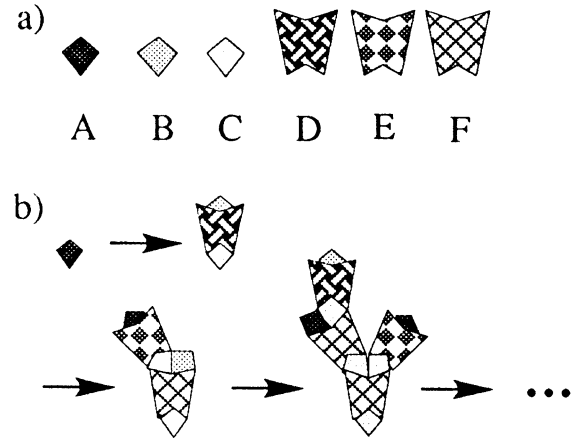


Fig. 2. Constituents and developmental sketch of L-system model of Japanese Cypress. (a), constituents; (b), initial developmental sketch according to the rule (1).

In addition to this rule (1), we have included the following predominant factors reflecting on cell-growth and cell-differentiation of each cell (cell is supposed to be individual state A to F in (1)): (i) feeding schedule of nutrient transported from root; (ii) threshold value causing cell-differentiation (switching to next process in the rule of L-system); (iii) threshold value for cell-death or starvation; (iv) decay rate constant of absorbed nutrient. Combined these four factors, a differential equation of nutrient (X_i) involved in each cell (A to F in (1)) can be represented by,

$$dX_i(t)/dt = U_i(t) - kX_i(t) \quad (i= 1,2,..., n) \qquad (2)$$

where $U_i(t)$ is absorbed nutrient from at time t coming from its parent-cell ("parent-cell" here means directly combined one beneath the cell itself), and k represents the first-ordered decay rate constant, and n is the total number of cells belonging to the state A to F, which will increase with developmental time. The time t is assumed to be the discrete time step of L-system. Furthermore it was supposed the $U_i(t)$ can be represented by as follows:

$$U_i(t) = (1 - \alpha) * U_{p,i}(t) \quad (i= 2,..,n) \qquad (3\text{-a})$$

$$U_1(t) = U_{p,1}(t) \qquad (3\text{-b})$$

where $U_{p,i}(t)$ shows the nutrient of parent-cell of the child-cell i at time t, and α is the relative percentage of nutrient that parent-cell of the child-cell i absorbs ($0 < \alpha < 1.0$). The $U_{p,1}(t)$ corresponds to the amount of nutrient coming from the root (underground) at time t. When a parent-cell differentiates to m child-cells, the following equation is used instead of Eq. (3-a):

$$U_i(t) = (1 - \alpha) * U_{p,i}(t) / m \pm \varepsilon \quad (i= 1,2,..,n) \qquad (4)$$

where ε shows the differentiation noise; when a parent-cell differentiates, child-cells having almost identical nutrient are formed. "Almost" here means that amount of nutrient in each child-cell is imbalanced or distributed within a range of small noise.

The Eq. (2) can be analytically solved as follows under the initial condition $X_i(0) = 0$:

$$X_i(t) = U_i(t) / k * (1 - \exp(-kt)) \qquad (5)$$

It was supposed that cell-differentiation and cell-death are caused according to the following rule as for the value of $X_i(t)$:

$$\begin{array}{ll} \text{if } X_i(t) > \theta_d, & \text{cell-differentiation (switching to next process in the rule of L-system)} \\ \text{if } X_i(t) < \theta_f, & \text{cell-death (no longer survive)} \end{array} \qquad (6)$$

In case of cell-death, the remaining nutrient of destined cell is supposed to be succeeded by its parent-cell. Since the cell size is growing with $X_i(t)$, the profile of relative cell size ($S(t)$) versus $X_i(t)$ was assumed by the following growth function:

$$S(t) = 1 / [1 + \exp(1-X_i(t))] \qquad (7)$$

By using the rule and equations (1) through (7), we have simulated the temporal morphogenetic development of scale leaves of Japanese Cypress.

3. Effects of feeding schedule of nutrient on the temporal development

The standard parameter values in (1) to (7) was set up as follows:

$$\begin{array}{l} U_{p,1}(t) = 10 \text{ (constant)}, \quad k = 0.05, \quad \alpha = 0.2, \\ \varepsilon = 0.001, \quad \theta_d = 2.0, \quad \theta_f = 0.6 \end{array} \qquad (8)$$

The temporal development under the standard condition (8) is shown in Fig. 3, where time-step of (a), (b), (c) and (d) is 5, 20, 60 and 80, respectively. Fig. 4 shows the magnifying figure on the temporal development of a certain apex part of the leaves during $20 < t < 50$; each branching causes cell-differentiation (see the branchings perpendicularly extending) or cell-death (see the most left-sided branching). The effects of the amount of nutrient ($U_{p,1}(t)$, constant-flow) on the

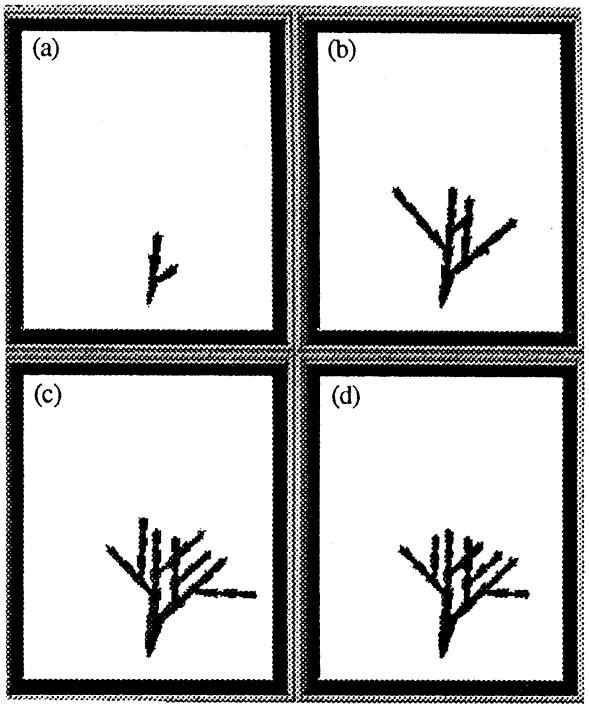


Fig. 3. Temporal development under the standard condition (8). Time-step (arbitrary unit): (a), 5; (b), 20; (c), 60; (d), 80.

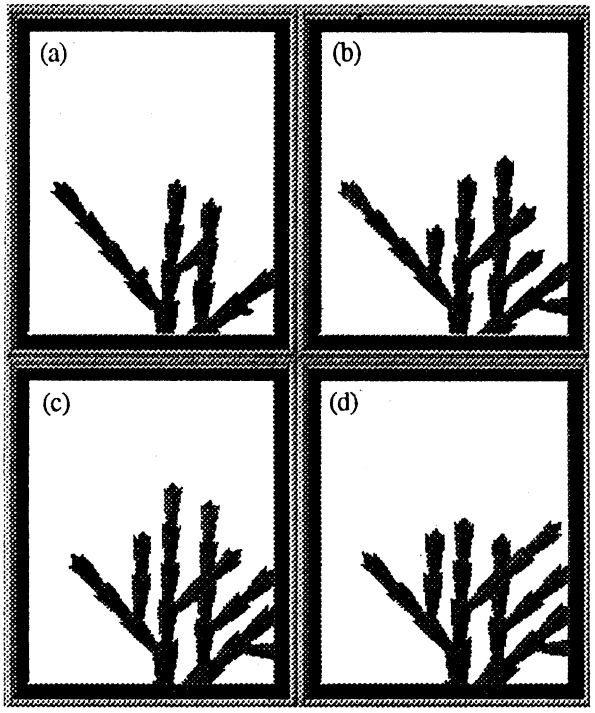


Fig. 4. Magnifying figure on the temporal development of a certain apex part of the leaves during $20 < t < 50$. Time-step (arbitrary unit): (a), 20; (b), 30; (c), 40; (d), 50.

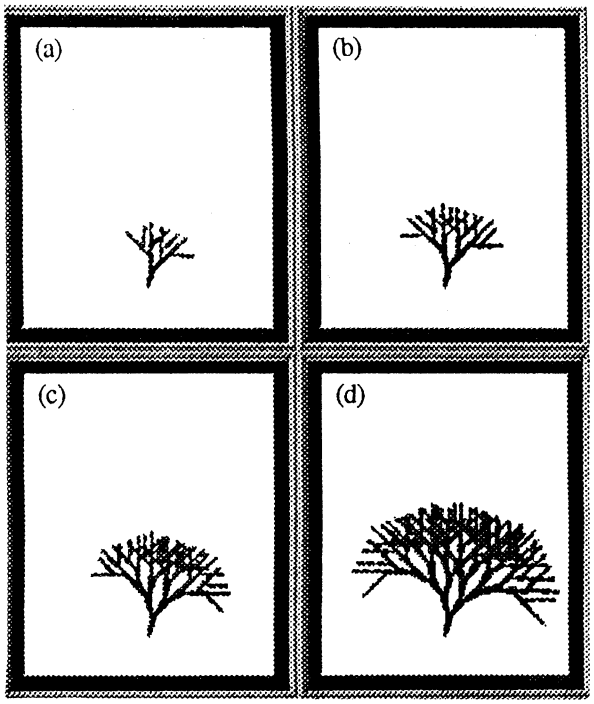


Fig. 5. The effects of the amount of nutrient ($U_{p,1}(t)$ constant-flow) on the developmental size at the steady state. $U_{p,1}(t)$ (constant): (a), 10; (b), 30; (c), 100; (d), 500.

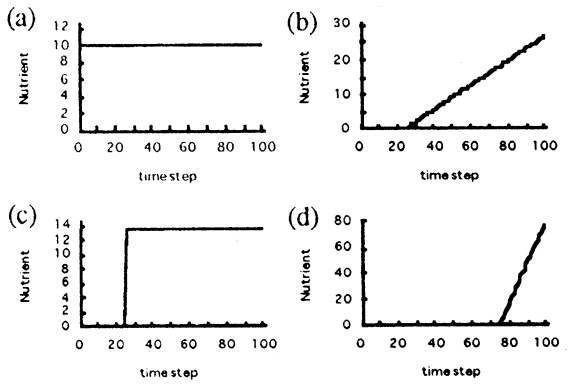


Fig. 6. Various kinds of feeding schedule of nutrient $U_{p,1}(t)$ during $0 < t < 100$ under the condition of that the total amount of $U_{p,1}(t)$ is the same. (a), $U_{p,1}(t)=10$ (constant); (b), $U_{p,1}(t)=0.04$ (constant) during $0 < t < 25$, $U_{p,1}(t) = 0.349t - 8.697$ during $25 < t < 100$; (c), $U_{p,1}(t)=0.04$ (constant) during $0 < t < 25$, $U_{p,1}(t)=13.32$ (constant) during $25 < t < 100$; (d), $U_{p,1}(t)=0.04$ during $0 < t < 75$, $U_{p,1}(t) = 3.0644t - 229.79$ during $75 < t < 100$.

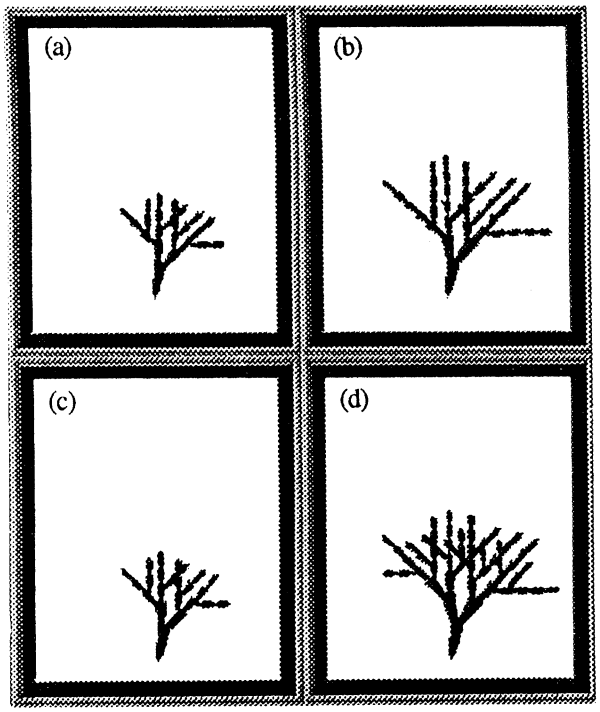


Fig. 7. Developmental profile at t=100 under the feeding schedule show in Fig. 6-(a) to (d). Feeding schedule: (a), Fig. 6-(a); (b), 6-(b); (c), 6-(c); (d), 6-(d).

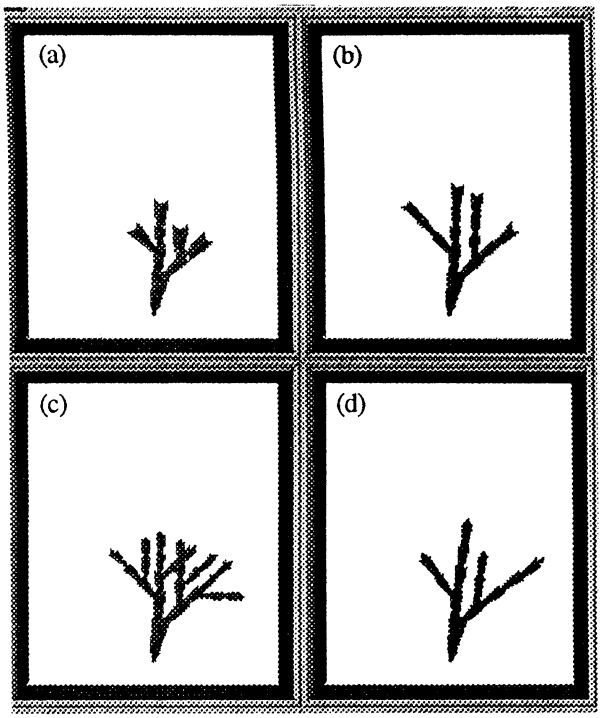


Fig. 8. The effects of θ_d on the developmental size at steady state. θ_d-value: (a), 1.0; (b), 1.5; (c), 2.0; (d), 9.0.

developmental size at the steady state is shown in Fig. 5. Fixed the total amount of $U_{p,1}(t)$ from t= 0 to 100 at 1000, we have changed the feeding schedule of nutrient as shown in Fig. 6. The developmental profile at t=100 under the condition of each schedule is represented in Fig. 7; it was revealed that the developmental size at steady state is different by feeding schedule even if the total amount of nutrient is fixed (linearly increasing feeding (Fig. 6-(b) and (d)) is efficient for large-growing).

4. Effects of the threshold value causing cell-differentiation or cell-death

As shown in the condition (6), each cell at time stage t is destined either to grow as it is, or to differentiate, or to die in accordance with the value of nutrient ($X_i(t)$). Figs. 8 and 9 show the effects of θ_d and of θ_f on the developmental size at steady state, respectively.

In Fig. 8, since the developmental size of Fig. 8-(c) is the largest among them, there exists optimal θ_d value which determines the largest-development. Compared of Fig. 8-(a) with (d), developmental size is almost same while θ_d value is different by 9 times in both cases. What is the mechanism that both developmental sizes are almost same? According to the rule of L-system shown in (1), in case of cell-differentiation, most of the cells are divided by two therefore, nutrient in parent-cell will be dived by three or two depending on α-value (with considering nutrient in parent-cell after cell-differentiation). Supposed that nutrient is divided by three, in the case of Fig. 8-(a), since θ_d is small, most of the vegetative apex cells will be differentiated and nutrient

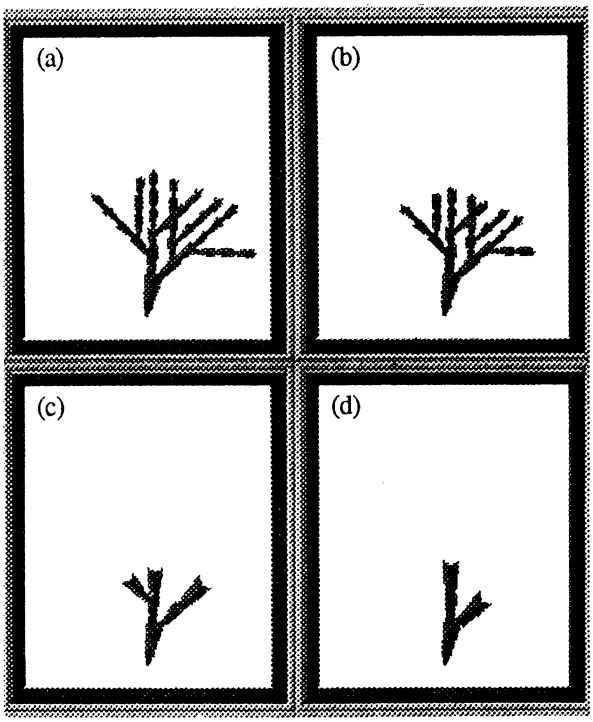


Fig. 9. The effects of θ_f on the developmental size at steady state. θ_f-value: (a), 0.06; (b), 0.6; (c), 1.0; (d), 1.5.

in parent-cell (in this case, the value of nutrient is around 1.0) is divided by three (= 0.3). This value of nutrient in each child-cell is less than the threshold value θ_f, which causes the death destination for child-

cell. Namely vegetative apex cells are actively differentiated, however, most of the child-cells can not survive. On the contrary to this, in the case of Fig. 8-(d), since θ_d is large, the possibility that the nutrient in parent-cells beyonds to θ_d is very small, which will lead to keep growing as it is without child-cells. Namely vegetative apex cells are sleeping (growing without child-cells). While these two situations as for cell-differentiation are quite different each other, resulting developmental size at steady state is almost same in both cases. As shown in Fig. 9, the decrease in the value of θ_f will cause to the larger developmental size at steady state. The θ_f shows the nonlinear effects on the size; compared of Fig. 8-(b) with (c), and of (c) with (d), θ_f is magnified by almost same ratio (x1.67 in the former case and x1.5 in the latter case), however, the former case results to the drastical change in developmental size.

5. Effects of relative percentage of nutrient parent-cell absorbs

In Eq. (3-a) or (4), α is the controlling factor to determine the amount of nutrient in parent-cell; the remaining relative percentage of nutrient, $(1-\alpha)$, is distributed among child-cells. Fig. 10 shows the effects of the α value on the developmental profile at steady state.

By decreasing in α value, only vegetative apex in each branching is repeatedly differentiated, which leads to the developmental structure with long straight extension (see Fig. 10-(a)).

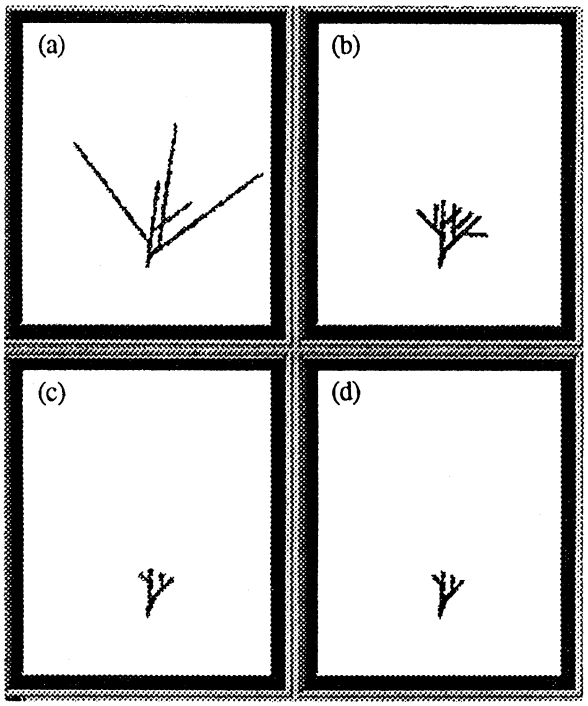


Fig. 10. The effects of the α value on the developmental profile at steady state. α-value: (a), 0.05; (b), 0.2; (c), 0.3; (d), 0.5.

6. Effects of decay rate constant

As shown in Eq. (2), first-ordered decay rate constant of nutrient, k, is the key factor to control the transient time until the amount of nutrient is saturated at a certain level. In the case of that this value is large, system will soon reach to steady state before developmental stage is not so matured. Thus this value is also predominant factor to control the developmental size at steady state. Fig. 11 shows the effect of k-value in Eq. (2) on the developmental size. As shown in Fig. 11-(a), the decrease in k-value leads not only to the large growing at steady state but also to the frequent cell-differentiation of vegetative apex.

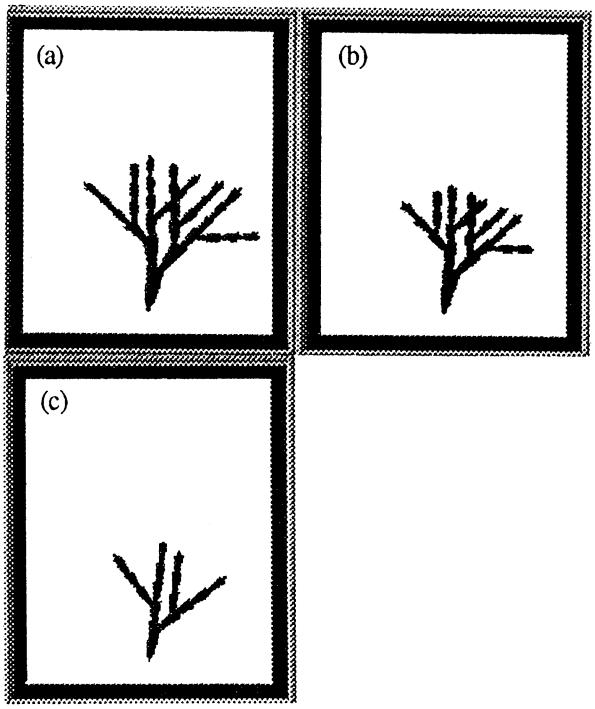


Fig. 11. The effect of k-value in Eq. (2) on the developmental size at steady state. k-value: (a), 0.03; (b), 0.05; (c), 0.1.

7. Discussion

In this study we have proposed new L-system model with considering cell-differentiation, cell-death and feeding schedule of nutrient. We found that temporal developmental phase and size (also shape) at steady state are drastically influenced by the factors such as $U_{p,1}(t)$, θ_d, θ_f, α and k even if the rule of L-system is fixed. Supposed that the rule of L-system is genetic code or inherent time-table for producing functional proteins, the factors such as θ_d, θ_f, α and k will be internal signals controlling the switching time to proceed to the next developmental stage, and feeding schedule $U_{p,1}(t)$ can be called environmental (= external) perturbation. The child-cells (succession) will inherit genetic code from parent-cells (in this case, every child-cell inherits the same rule of L-system), however, this inheritance is not sure of the

same developmental phase as parent-cell has. The temporal phase will considerably be changed by the internal and external signals or factors.

Reference

[1] A. Lindenmayer, "Mathematical models for cellular interactions in development, I, II", J. Theor. Biol., Vol. 18, pp. 280-315, 1968.
[2] P. Prusinkiewicz, A. Lindenmayer, "The algorithmic beauty of plants", Springer-Verlag, New York, 1990.
[3] T. Nishida, "K0L-system simulating almost but not exactly the same development; case of Japanese Cypress", Memoirs of the Fac. of Sci., Kyoto Univ., Series of Biol., Vol. 13, pp. 97-122, 1980.
[4] K. Kaneko, T. Yomo, "Cell division, differentiation and dynamic clustering", Physica D, Vol. 75, pp. 89-102, 1994.

Symmetry and Symmetry-Breaking in Multi-Agent Behavior

Franco di Primio

GMD - German National Research Center for Information Technology
Artificial Intelligence Research Division
D-53754 Sankt Augustin
E-mail: franco.diprimio@gmd.de

Abstract

In this paper we present some results about properties of the spatial behavior of systems consisting of many artificial agents (robots). The general goal is to understand how the complexity of group behavior is related to individual behavior and how differences arise and can be grounded. We argue that symmetry and symmetry-breaking are essential factors for the organization of spatial group behavior. They can serve as guiding principles for the construction of agents for real world environments.

Keywords

TEAM ROBOTICS - SYMMETRY - SPATIAL BEHAVIOR

1 Introduction

The general motivation of the work described in this paper is to find out some fundamental guiding principle for the construction of agents for real world environments. To this purpose, our basic method of investigation is to study primitive forms of spatial behaviors concentrating on simple rather than complex agents and on group rather than on individual behavior. While the focus on physical space allows to consider more situated, i. e. embodied models of computational mechanisms [7], the study of 'groups of related single and simple agents' (that we call 'multi-agents') opens up, in our opinion, a new perspective on robot design methodology. Obviously, when constructing real world agents, relevant design decision are to be taken concerning the relative complexity of the agents and their environment. The most frequent starting point in this respect consists in considering one single agent operating in different environments that can, if needed, also be adapted (i. e. modified with respect) to the capabilities of the agent. In particular, the modus operandi of development is normally (so to speak) *vertically* directed, meaning that it leads to a growing complexity of the capabilities of the agent under consideration and a multi-layered, often tower like architecture of increasing sophistication. Our approach is different. We have reason to believe that a reduction of the complexity of both the design process and the resulting system can be achieved when - in relation to given environments - basic capabilities are distributed *horizontally* over more than one agent. The group of agents as a whole (the multi-agent) is then expected to show the same (or even a greater problem solving) capability as a single, complicated agent.

2 Lessons from biology: Elephants don't develop by aggregation

In our approach, the unity of analysis and synthesis is the group, not the single agent. From a biological point of view, this means that we focus on 'aggregation' rather than 'growth' processes. Biological complexity or 'multicellularity' comes into being in two quite different ways: „In one a cell divides and two daughter cells remain stuck together. As they in turn divide and remain in contact, the colony becomes larger. We call this simply size increase by *growth*. The other method of becoming multicellular is by *aggregation*, of which slime molds are a prime example." [4, p. 40]. Growth, i.e. morphogenesis (shape-generation) processes are very complicated. Only general principles like differentiation (cells become specialized in dividing labor) and morphogenetic movement (the movement of cells from one part of the embryo to another) are understood or, to be more precise, well documented experimentally. The crucial step of developmental biology, however, i.e. the linking of development with genetics, is only partially fulfilled. We believe that engineers and AI researchers have no chance to technically reconstruct complex individual agents by aggregation. Multicellular organisms like elephants not only do not play chess [6], they also do not develop by aggregation („an absurd idea from any point of view" [4, p. 65]). Complex individuals are the result of growth on the basis of complex genetic programs. The single-robot centered design methodology should, in our opinion, mimic development by growth to be successful. But in this respect, we lack not only the technology but also the necessary knowledge of the processes involved in the *wet* ware. Considering groups of simple agents intended to perform modestly simple tasks is a better technical strategy and might give the opportunity to understand how small 'embodied' programs 'aggregate' to complex ones. We can also – hopefully at least for a while – dispense completely with the problem of genetically controlled growth and concentrate on the other aspects (movement of cells, specialization, regulation, communication or so-called induction processes like chemical signaling) that also play a crucial role in aggregation.

3 Related work

Current work on robot groups (team or collective robotics) is concentrating on abstract, graph theroretical aspects [2], or high level, distributed intelligence (model building, planning and communication) [3]. Next to our view is the research on very primitive forms of coordination and cooperation (as in the cellular robotics [11], referring to the social insects or swarm intelligence metaphor), or the design of emergent behaviors on the basis of purely local interactions between the individual agents rather than the intentional negotiation between them [14]. Research based on the social intelligence hypothesis (s., for instance, [8]) is also on a similar track. Up to now, these works seem, however, to lack a comprehensive theoretical foundation and do not aim at finding unifying principles. In contrast, our ultimate goal is to develop a theory of behavior in which (possibly) all (embodied) computational mechanisms are based on symmetry principles that have firm mathematical foundations in group theory.

4 The role of symmetry

Besides the cases where symmetry has been brought in relation to aspects of perception and art [1], up to now, as a topic of natural sciences, symmetry problems have been studied mainly in the context of physics ([12] [15]). Investigations related to living organisms are also available (s., for instance, the explanations in [17, p. 189 ff] concerning the gaits of different animals). As for technical systems, however, there are – to our knowledge – only sporadic examinations resulting mostly in very general statements. [13, p. 23] points out, for instance, that all actively moving animals have an external bilateral symmetry. Analogously, all our means of transport are bilaterally symmetric.

In this paper, we want to be more specific and show the relevance of symmetry aspects for robot behavior. Basically, we reconsider on the basis of previous work [10] typically single-robot, body-centered spatial contrasts like left/right, front/back, up/down in the context of small groups of robots and ask for the meaning and role of symmetry and symmetry-breaking as a factor for the organization of spatial group behavior. The term behavior is used (in a restricted sense) as a synonym for movement (of robots) in space. A behavior sequence is called symmetric if each state the system under consideration (robot or group of robots) passes through is symmetric [12, p. 39].

Notice that like in physics where the symmetries of laws are really important and not the symmetries of things, for us the main focus is on symmetries of behavior and how they can be expressed computationally and not (only) on symmetries of body. In the following sections, we discuss the basic contrasts and the group behaviors based on them. As for the computational aspects, we confine ourselves – for obvious reasons of space – to very rough hints.

5 Front-back

Biologically, a very, perhaps the most primitive kind of behavior is where an organism responds to something in its environment. It might be movement away from or towards light or some other property of the environment (like food, chemicals, etc.) [4, p. 123]. At first sight, target-oriented movement seems to be a necessary and sufficient basis for the distinction between the front and back part of an organism. Forward movement could be considered as attraction towards a stimulus, backward movement as repulsion from a stimulus. From a physical point of view, however, target oriented movement can be interpreted as the simple movement of a sphere in an unstable equilibrium. A sphere has no front and no back. It is by definition perfectly symmetric. The orientation is then given by the asymmetry of the particular environment (landscape), i.e. by its shape and the distribution of mass that causes the attractive force of gravity. Notice that in this physical example we have removed biology, but the essential constraints for moving behavior remain. The movement of a bacterium along a chemical gradient is structurally almost the same as the movement of a sphere in physical space if the mass landscape is substituted (or complemented) by a chemical one. The particular distribution of chemicals and concentration thereof determines the direction and perhaps also the speed of movement.

In order to get the distinction between front and back, it doesn't even suffice to break the symmetry of the sphere in a simple way. If, for instance, a sphere is deformed to become a cylinder, then the capability to move is conserved, even if only along one direction. A cylinder rolling down an inclined plane has, however, no more a front or back part than a sphere. We claim that a difference between front and back can be grounded only if one considers a group of moving objects, for instance, (at least) two spheres or two cylinders, that are different in some respect (for instance, dimensions). Suppose that in a given landscape two spheres that are close to each other and in an unstable equilibrium roll in the same direction and along the same path. Notice that moving along the same path could mean that the two spheres are symmetric with respect to some translation (if they have equal dimensions). The sphere that reaches first a point of stability (the minimum of a valley) can then be considered to be the first one, the other the last one. But 'first' and 'last' do not mean 'anterior' (front) and 'posterior' (back). This can be *extrinsically* right, i. e. in relation to an observer (an observer could say that the spheres are rolling one after (behind) the other), but not *intrinsically* (with respect to the objects themselves) [16]. Front and back require an asymmetry. In most living organism the difference depends on the distribution of the sensors that is asymmetric (the front part is where the 'eyes' are). This asymmetry is, by the way, the reason why a child rarely confuses front with back, while the dif-

ficulties with the distinction between left and right, for which no (external) asymmetry is given, are well known. In the system of two spheres rolling along the same path in the same direction one can speak of front and back only if the spheres have, for instance, different dimensions (or colors). The smaller one could then be considered, as a convention, to be the anterior (front) part of the whole system, the other the posterior one. Depending on the direction one would then have either a forward or a backward movement.

It might be difficult for the reader to follow this argumentation because in language we use to describe positions and movements of objects in a mixed way. Sometimes we refer to our coordinate system as observers (extrinsic view) sometimes to a coordinate system related to the objects under consideration. A perhaps more realistic example of a group movement *not* allowing the distinction between front and back is a running liquid like water. A river can be viewed as a big collective movement (an aggregation) of molecules. We speak, however, of the 'mouth' of a river, suggesting that the river has a 'front'. The source could then also be called the back of the river. But this is an extrinsic description based on the asymmetry of the environment and not of the river itself. I do not know if in biology it is possible to give a consistent explanation of a bacteria movement as a (molecular) group movement. This cannot, however, be excluded. But if a distinction can be made between front and back, then an asymmetry must be there. For technical systems, the distinction between front and back is in its most basic form (only) possible when two objects (at least one of them should be capable of moving alone) are unequal and connected together.

6 Following behavior

The following behavior of animals (and robots) is an instance of the more basic imitation behavior (*allelomimesis, do what my neighbour is doing*) [9, p. 296], which is an expression of positive feedback. At the level of the single animal or robot, the important property for this behavior is not the bilateral symmetry (to be discussed below), but the asymmetry between front and back. Following occurs when the bodies of at least two animals or robots move forward along the same (horizontal) path. This means that they are directed to each other in such a way that no mirror symmetry is given. If these conditions are given except for the fact that the movement is not forward but backward, then we have a (rather uncommon) behavior for the description of which we have no single word. Notice that for systems with perfect symmetry (two 'spherical' robots), it wouldn't be reasonable to speak of following behavior. One couldn't distinguish if they move together forward or backward.

7 Up-Down

For the up and down contrast, physically the difference is grounded on the force of gravity. For a standing object, symmetric or not, down is the part where the force is greater. Biologically, the distinction presupposes, for both development by growth and aggregation, differentiation processes at the cell level. For higher animals, it is the asymmetry in the distribution of sensors that – as for the front and back contrast – at last justifies the distinction. The 'head', where the most sensors are concentrated, is the upper part (for a snake this description is however not obvious). For lower organisms, the basis is a differentiation of the constituent parts. For a slime mold, for instance, when the slug rights itself, the amoebae that are up remain alive and form the spore, the ones that are down die and build the stalk [4, p. 3ff and 124ff]. For technical systems too, the contrast does only make sense when (at least) two (functionally) different subsystems are connected 'vertically'.

8 Scaling behavior

At the individual level, the (animal) behavior based on the distinction between up and down is the pronk or hopping or climbing (for a wheeled vehicle, this is, of course, rather difficult). Climbing over an obstacle (for instance, scaling a wall) can also be a resulting behavior at the group level. Scaling in this sense takes place when the bodies of at least two animals or robots move forward along the same (almost) vertical path, i.e. are oriented towards each other in such a way that no mirror symmetry is given. Thus, as for the following behavior, a basic precondition is the asymmetry between up and down. The difference is that in the following behavior the movement takes place on an horizontal path. Notice that a successful scaling technique is possible when the robot (animal) which is down helps the other one by pushing it up, and the robot that is up by pulling the one that is down. In contrast with the following behavior, the opposite behavior of scaling (moving backward along the same vertical path) is more usual (think of two people descending backward a ladder or a rope simultaneously).

9 Bilateral symmetry

Bilateral symmetry is ubiquitous in the animal kingdom and it seems almost impossible to imagine a (terrestrial) animal that does without. To understand our contribution with respect to robots, it is important to distinguish between external and internal bilateral symmetry. The external bilateral symmetry of animals is in general almost perfect. The internal organs, however, are almost always asymmetrically distributed (the kidneys and lungs are an exception). Now, we believe that it is possible to imagine (and construct) robots that have both internal and external bilateral symmetry. Such a (multi-)robot can then be considered to consist of two 'half robots' which are for themselves not bilaterally symmetric (in the functional sense). Take, for instance, the primitive single robot proposed

by V. Braitenberg [5] consisting of one sensory unit, one actuator and one processing unit, mounted on a platform (Fig. 1):

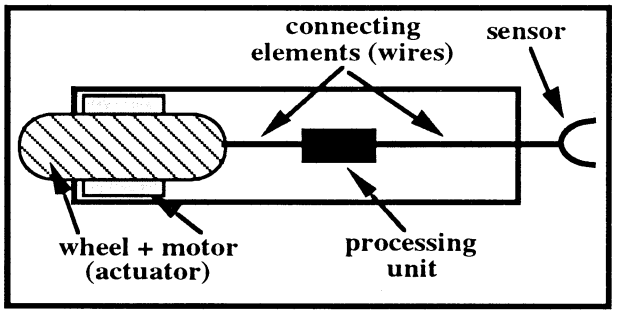


Fig. 1: Structural view on a single-agent

This 'vehicle' has a purely geometric bilateral symmetry along the axis connecting the mid of the sensor with the wheel, but this symmetry has no functional meaning. Such a primitive robot is only able to move straight forward in one direction. The (real world) forward movement can, of course, be subject to perturbations. Now, if two such vehicles are structurally coupled together through a rigid element (dark hatched rectangle in the graphic below), one obtains a functionally relevant bilateral symmetry. The relevance can be seen in the fact that the 'multi-robot' is now capable to turn right (or left), when the stimulus (perhaps a light source) is on the left (or right). We suppose that the intensity of the stimulus is converted into pulses for the motors in a direct proportional way.

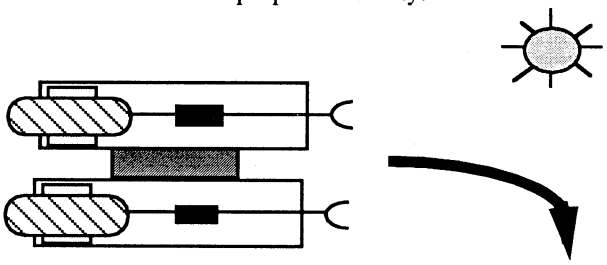


Fig. 2: Two rigidly coupled single-agents build a multi-agent that avoids a stimulus

From a behavioral point of view, one obtains the capability of a 'spherical' robot, so to speak, which is able to move away from whatever direction the stimulus (the lower part of the gradient of the gravity field) is. Left and right can now be defined arbitrarily in relation to the axis of the multi-robot. Again, notice that a necessary and not arbitrary distinction between left and right is only possible when the two component robots are in some respect different, i. e. when an asymmetry is given. The following physical example in which biology is, once more, completely removed, makes it clear. If two equal wheels are properly connected together through a rigid axis, one gets a system capable of moving (rolling) straight 'forward', for instance, down a smooth slope. But it is no use to speak of left and right wheel in an intrinsic manner. But if the wheels are different and have, for instance, a different diameter, you can ground the distinction. Notice that the coupled system is then able or constrained to move in a curve (turn). It could, however, also move (roll) straight forward depending on the form and nature of the slope. The rigidly coupled multi-robot system described above is also able to move straight forward (when the light source is exactly in front of it). 'Sensing' the force of gravity and sensing light in an asymmetric environment can thus produce similar effects. There is, apparently, a trade-off between the (a)symmetry of a system and the (a)symmetry of its environment.

10 Meeting and avoidance behavior

By avoidance behavior we understand here the (group) behavior of two robots that meet at some place and, instead of colliding, avoid each other continuing their movement in their original direction. In this case, as for the following behavior, the asymmetry between front and back part of the body is essential. The bilateral symmetry is also relevant. Avoidance behavior in the above sense presupposes another group behavior, frontally directed movement (confrontation or meeting). This is given when the two robots both move forward along the same path and when they face each other, i. e., their orientation is such that a mirror symmetry is given, the front parts of their bodies being nearer than their backs. Avoidance is only possible if this symmetry is broken, i. e. the two robots must change their direction to the left resp. to the right in the sense of their intrinsic reference frame. If this does not happen (this is, for instance, the case when two people meet in a narrow corridor or at a door, and the first one tries (according to his view) to go left, and the other to go right, or vice versa), then the mirror symmetry is conserved and we get a 'behavior conflict'. The total effect is an oscillation (hopefully not lasting very long) of the (two-body) system.

11 Circumventing (avoiding) obstacles

We are here not primarily interested in how a single robot can avoid an obstacle. We consider the situation where two robots have to perform such a task. Two robots can circumvent a convex obstacle if first they are able to start and keep going forward together (side by side) in the same direction; second, when they hit an obstacle, they go in opposite direction, remaining, however, in 'contact' with the obstacle. When they meet at the 'other' side of the obstacle, then they start again going together side by side (for more details. s. [10]). In our present framework, we can view this process as a partly symmetric behavioral sequence. 'Partly symmetric' means that a series of changes in symmetry occurs. To understand this, it is convenient to idealize the situation using a model where the robots

as well as the obstacle are spheres of equal dimensions (s. Fig. 3).

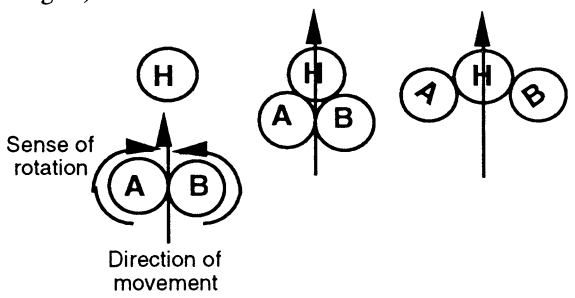


Fig. 3.1 **Fig. 3.2** **Fig. 3.3**

The obstacle (H) has – as stationary sphere – a complete rotational and mirror symmetry. The left robot (A) is supposed to move forward by rotating clockwise along its vertical axis. The right robot (B) moves in the same direction as A, but rotating counterclockwise. Between the two robots there is then a mirror symmetry (Fig. 3.1). At the moment in which the two robots touch the obstacle, the whole system acquires the symmetry of an equilateral triangle (Fig. 3.2). The circumvention starts when this symmetry is broken (Fig. 3.3). The basic mirror symmetry is, however, conserved (the obstacle acts, so to speak, as a mirror). From a computational point of view, this group behavior could be (for real robots) implemented on the basis of a wall following strategy. Indeed, for each robot the other robot together with the obstacle could be seen as a (partly dynamically arising) wall it has to follow (s. Fig 4, where the left robot is outlined in the graphic as a rectangle with a line indicating the front part). Here, again, the symmetry aspect is evident.

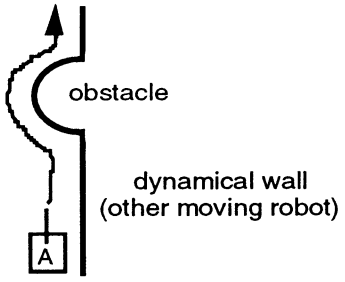


Fig. 4

Finally, please note that this avoidance group behavior is related to the example given in § 9. As argued in [dPM 94], the basic difference consists in the form of coupling (rigid and not rigid or 'soft'). The common property is that in both cases the multi-robot shows capabilities that are not present at the level of the single 'components' of the systems.

12 Territory partition

By territory partition we mean the group behavior of at least two robots which start from common or different home bases and divide up an area in order, for instance, to explore it. We consider a special case (two robots in a particular labyrinth) and show that initial conditions in form of symmetry properties suffice to control the partition process. The labyrinth is sketched in Fig. 5.1 (the inner, brick-like rectangles delimit the corridors). The robots are outlined as (named) oriented rectangles (A and B, the lines indicating the front parts). The entry and common home base is symbolized through hatching.

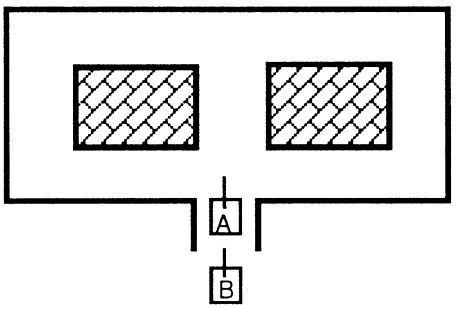


Fig. 5.1

If the robots are constructed (programmed) in such a way that both move approximately at the same speed, and that the first one (A) keeps going on the left wall (left-hand rule) and the second one (B) on the right wall (right-hand rule), then the two robots will meet (Fig. 5.2).

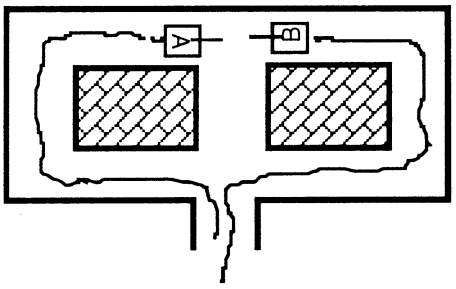


Fig. 5.2

This is evident by the given structure of the labyrinth. It can, however, be proved that, if the labyrinth has an outward limiting and connected wall (this is always the case when the entry of the labyrinth is also the only exit), the robots must meet. This is a result of the mirror symmetry at the control level. If the meeting place is within a corridor (we suppose that corridors are so narrow that robots cannot move along them side by side), one of the robots must move back to the nearest branching point. This means a symmetry-breaking in the (frontally directed) movement of the two robots. Once both robots have reached the branching point, they *only* need to reverse the wall following strategy (inversion of the mirror symmetry). The result is then

a partition of the labyrinth.

Clearly, in this problem other aspects related to higher, memory-based capabilities play also a role. For instance, how is the decision of moving back to the previous branching point taken? Furthermore, the partition obtained as a result of this simple symmetric exploration strategy is in general not optimal (the overlapping part of the two territories is not necessarily minimal) and even not complete (the sum of the routes of the two robots yields no route through all corridors, which means that parts of the whole territory are left out). The question arises if after all and how the robots can solve locally these global problems. Doubtless, further capabilities (like communication) are to be taken into account.

13 Conclusions and outlook

We have collected enough evidence for the hypothesis that symmetries in body, i. e. both sensory and effector space might be considered as being (chrono)logically posterior to symmetries in group space. Bilateral or mirror symmetry are, for instance, related to partner symmetry (two coupled 'half-robots' moving side by side or facing each other). As a further result, we have shown that besides the rather obvious cases of following and scaling behavior also some form of obstacle avoidance and the exploration of restricted environments can be described and - from an implementation point of view - realized, as group behaviors based on symmetry properties. Symmetry-breaking plays the indispensable role of a transition and an escape mechanism for stalemate situations, like confrontations (dead-locks) and oscillations. Of course, much remains to be done, for instance, an investigation of the contrast inside-outside that seems to require the 'interaction' of at least three components.

As an application area, we are going to investigate 'distributed cognitive mapping'. The basic idea is to let different agents build concurrently (partly overlapping) maps of some area (for instance, networks of pipes) and to study how they can cooperate in order to solve problems the spatial localization of which is not restricted to the overlapping part of the maps. Basic research questions are if and how common 'concepts' develop and the role of imitation and communication. Specific work will include simulations and real world experiments with minirobots.

Acknowledgments

I am very grateful to the colleagues of the cognitive robotics group (particularly H.-J. Grimm, U. Licht, B.S. Müller and E. Rome) who stimulated me with constructive remarks in many discussions about the ideas presented in this paper. I thank T. Christaller (head of the group and AI research division) for creating the conditions and the opportunity to pursue this research. Last but not least, I thank Ursula Bernhard for reading the English text

References

[1] Caglioti, G., Symmetriebrechung und Wahrnehmung, Vieweg, 1993

[2] Bender, M. A./Slonim, D. K., The Power of Team Exploration: Two Robots Can Learn Unlabeled Directed Graphs, in: Proc. of 35th Annual Symposium on Foundations of Computer Science, IEEE CS Press, 1994, pp. 75-85

[3] Bond, A. H./Gasser, L. (eds), Readings in Distributed Artificial Intelligence, Morgan Kauffmann, 1988

[4] Bonner, J. T., Life Cycles, Princeton University press, Princeton, 1995

[5] Braitenberg, V., Vehicles - Experiments in Synthetic Psychology, MIT Press, 1987

[6] Brooks, R. A., Elephants don't play chess, in: Maes, P. (ed), Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, MIT Press, 1990

[7] Brooks, R. A./Stein, L. A., Building Brains for Bodies, A.I. Memo 1439, MIT, 1993.

[8] Dautenhahn, K./Christaller, T., Remembering, Rehearsal and Empathy – Towards a Social and Embodied Cognitive Psychology for Artifacts, Arbeitspapiere der GMD, Nr. 956, GMD, November 1995

[9] Denebourg, J.-L./Goss, S., Collective patterns and decision making, in: Ethology, Ecology and Evolution, 1, 1989, 295-311

[10] di Primio, F./Müller, B.S., Problem Solving Abilities of Systems fo Artificial Agents, in: Proc. From Perception to Action Conference, Lausanne, 1994, IEEE CS Press, 322-325.

[11] Fukuda, T./Ueyama, T., Cellular Robotics and Micro Robotic Systems, World Scientific, London, 1994.

[12] Genz, H./Decker, R., Symmetrie und Symmetriebrechung in der Physik, Vieweg,, 1991

[13] Ivanov, V. V., Gerade und Ungerade, Hirzel, Stuttgart, 1983

[14] Mataric, M. J., Designing Emergent Behaviors: From Local Interactions to Collective Intelligence, in: Meyer, J.-A./Roitblat, H. L./Wilson, S. W. (Eds), From animals to animats 2, MIT Press, 1993, 432-441.

[15] Rosen, J., Symmetry in Science, Springer-Verlag, Berlin, 1995

[16] Retz-Schmidt, G., Various Views on Spatial Prepositions, in: AI Magazine, Summer 1988, 95-105

[17] Stewart, I./Golubitsky, M., Fearful Symmetry, Penguin Books, London, 1992

Some New Look at Metabolism-Repair System

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

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

ABSTRACT------In this paper we proposed a modified linear Metabolism---Repair system, discussed the mechanism of repair machinery of cells which is different from [2,3], some new ideas are shown.

Key words-----Metabolism--Repair systems; transcription; replication; system identification.

I. Introduction

In the recent years many works devoted into the problem that to describe the bio-chemical process by using mathematical models. A basic problem is the metabolism in cells. A represent of it is that by R. Rosen[1]. He introduced the notion of metabolism-repair (M,R)-network in order to show formally how the features of repair and replication could be naturally induced solely from a cell metabolism machinery. In [2,3] J. Casti presented a linear (M,R) -systems by using the concept of state space and theory of realization based on Kalman work and B.L. Ho algorithm. He showed some theorems and examples to explain how a linear (M,R) -systems to operate. In this paper we try to give some new idea to the linear (M,R)- systems introduced by J.Casti. In this section we will summary the main notion and results briefly in [2,3].

The metabolism-Repair system is a such system that containing an internal model of itself. Basically, the cell has two components: one is a metabolic part representing the basal chemical process in a cell, another one is a maintenance or repair component that ensures continued viability of the cell at existence of external disturbances. The Fig.1 is a sample (M,R)-systems introduced by R. Rosen and J. Casti.

One can represent the metabolic activity of the cell as a map: $f : \Omega \to \Gamma$, $\omega \to \gamma$. The fundamental role of the repair component is to restore the metabolism f, the job of the repair function is to employ some of the cellular output to produce a new metabolic map that is then used to process the next input sequence. Thus, one suppose that the repair function can be abstractly represented by the map: $P_f : \Gamma \to H(\Omega,\Gamma)$, $\gamma \to f$. The P_f is called as the repair machinery or an error-correcting mechanism.

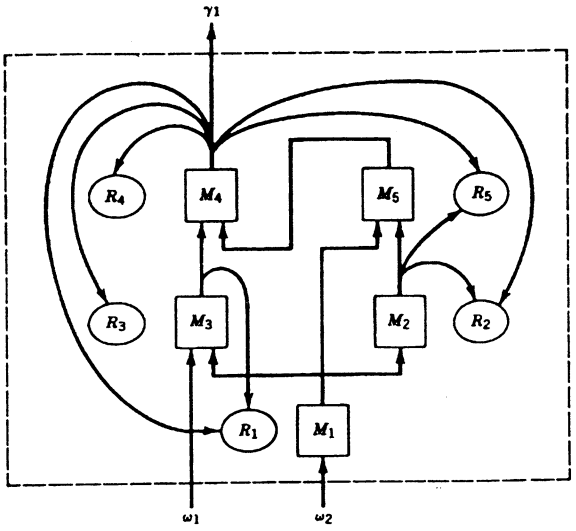


Fig.1 A metabolism network

However, how to deal with the error made by P_f? One can represent the reproductive machinery of the cell by the map: $\beta_f : H(\Omega,\Gamma) \to H(\Gamma,H(\Omega,\Gamma))$, $f \to P_f$. One can summarize the discussion so far by the diagram shown in the following:

$$\begin{array}{ccccccc} & f & & P_f & & \beta_f & \\ \Omega & \longrightarrow & \Gamma & \longrightarrow & H(\Omega,\Gamma) & \longrightarrow & H(\Gamma,H(\Omega,\Gamma)) \\ \text{(metabolism)} & & \text{(translation)} & & \text{(transcription)} & & \\ & & \text{(repair)} & & \text{(replication)} & & \end{array}$$

Fig.2 The diagram of (M,R)-systems

This is a diagram of (M,R)- systems. If the maps in a (M-R) system are all liner, we call it as a linear (M-R) system. Some notion and results on linear (M-R) system showed in [2,3] are listed below.

Let $\omega = (u_0, u_1, u_2, ...)' \in \Omega$, $u_i \in \Re^m$,
$\gamma = (y_1, y_2, y_3, ...)' \in \Gamma$, $y_i \in \Re^p$. (1.1)
In view of linearity assumption on f, it can be expressed as a sequence of matrices:
$f \approx \{ A_1, A_2, A_3, ...\}$, $A_i \in \Re^{p \times m}$, (1.2)
such that the action $f : \Omega \to \Gamma$, $\omega \to \gamma$ can be represented as

$$y_t = \sum_{i=0}^{t-1} A_{t-i}\, u_i \,, \qquad i = 1,2,3, \ldots \qquad (1.3)$$

The above structure of input/output relation can also be rewritten using a block Toeplitz matrix **F** as $\gamma = \mathbf{F}\omega$, or,

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} A_1 & 0 & 0 & \cdots \\ A_2 & A_1 & 0 & \cdots \\ A_3 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \end{pmatrix} \qquad (1.4)$$

This formulation could also be rearranged in the block Hankel form if one want to solve the realization problem.

The repair structure considered in [2,3] is also linear. the repair map $P_f : \Gamma \to H(\Omega,\Gamma)$, $\gamma \to f$ must abstractly produce f, given the output $\gamma \in \Gamma$ produced by f from the input $\omega \in \Omega$. Let

$$H(\Omega,\Gamma) = \{ \text{all } f \mid f : \Omega \to \Gamma,\ \omega \to \gamma \}\,. \qquad (1.5)$$

One can represent the repair action as

$$w_\tau = \sum_{i=0}^{\tau-1} R_{\tau-i} v_i \,, \qquad \tau = 1, 2, 3, \ldots \qquad (1.6)$$

where (w_i, v_i) are the output and input to repair system respectively, with elements R_j being linear map determined by γ and f. Just as the metabolism f was represented by the sequence $f \approx \{ A_1, A_2, A_3, \ldots\}$, $A_i \in \Re^{p \times m}$, one can now see that the repair system P_f can be represented as

$$P_f = \{ R_1, R_2, R_3, \ldots\}\,, \quad R_j \in \Re^{p \times pm} \qquad (1.7)$$

Similarly to (1.4), one can also identify P_f with the Toeplitz Matrix to solve the equation (1.6).

In the almost same way, one can solve the replication mechanism. Since β_f is a linear map accepting inputs of the form $f \approx \{ A_1, A_2, A_3, \ldots\}$ and producing outputs $P_f = \{ R_1, R_2, R_3, \ldots\}$, one must have a representation of the action of β_f as

$$C_\sigma = \sum_{i=0}^{\sigma-1} U_{\sigma-i} e_i \,, \qquad \sigma = 1, 2, 3, \ldots \qquad (1.8)$$

So, just as with f and P_f, one have the representation of β_f as

$$\beta_f = \{ U_1, U_2, U_3, \ldots\}, \quad U_i \in \Re^{p \times p2m2} \qquad (1.9)$$

together with the associated Toeplitz identification.

After finishing the modeling and parameters estimation of linear (M,R)-systems, the author of [2,3] considered the problem of stabilization of the linear (M,R) systems by repair machinery. Main function of repair mechanism is to restore the correct input/output behavior (ω, γ) in the face of changes in either the environmental input ω or the metabolism machinery f. Some sufficient and necessary conditions are shown in [2,3].

In this paper we try to give some modification for the modeling of (M,R)-systems in order to make the model of linear (M,R)-systems more reasonable.

II. Metabolic machinery

Essentially, the metabolic process in a cell is a biochemical reaction process. It should have a transient state and a ultimate stationary state for a alive cell. Especially, the life of a cell is finite, so y_i, the output of metabolic machinery, should be a finite bounded and stationary sequence. If y_i is a vector, $y_i \in \Re^p$, then

$$\| y_i \| \le C, \quad i = 1, 2, 3, \ldots, T$$

and $$\| y_i \| = 0, \quad i > T, \qquad (2.1)$$

where C is some a constant and T is the life of the cell. (2.1) is a necessary constrain condition to characterize the metabolic process. If output do not satisfies the condition (2.1), we can not think of that this is a real cell with a finite life. Based on this view, the metabolic process can be thought as a map:

$$y_t = f(u_i, i < t) \qquad (2.2)$$

where f may either linear or nonlinear and the input u_i may be deterministic or stochastic. In linear case, (2.2) has a similar form to (1.3) and also condition (2.1) should be satisfied. Usually, f is genotypic and implicit, we can not describe it exactly so far. What we could do is to describe and identify it by using some mathematical models. Especially, if we consider a linear (M,R)-system, it can be described as $\gamma = \boldsymbol{F}\omega$ subject to (2.1), or,

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \\ \vdots \\ y_t \end{pmatrix} = \begin{pmatrix} A_1 & 0 & 0 & \cdots & & 0 & 0 \\ A_2 & A_1 & 0 & 0 & & \ddots & 0 \\ A_3 & A_2 & A_1 & 0 & 0 & \ddots & \vdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ A_N & & \cdots & & \cdots & A_2 & A_1 \\ A_1 & & & & & A_3 & A_2 \\ \vdots & & \vdots & & \vdots & \vdots & \vdots \\ A_{L-1} & \cdots & A_2 & A_1 & A_N & \cdots & A_L \end{pmatrix}_{t \times N} \begin{pmatrix} u_{k-N} \\ u_{k-N-1} \\ u_{k-N-2} \\ \vdots \\ u_{k-1} \\ \\ \end{pmatrix}$$

or

$$y_k = \sum_{i=1}^{N} A_{i+L}\, u_{k-i} \,, \quad \text{if } k > N, \text{ where } L = k \bmod(N)$$

or

$$\sum_{i=1}^{N} A_i\, u_{k-i} \,, \quad \text{if } k \le N, \text{ (here } u_j = 0, \text{ if } j < 0),$$

subject to (2.1).

(2.3)

Where $\boldsymbol{F}$ is different from **F** in (1.4), $\boldsymbol{F}$ is finite Toeplitz matrix and it has the size of T x N, T is the life time and N is the length of the finite map f. In this model, $f = \{A_1, A_2, A_3, \ldots, A_N\}$. Usually, T>>N and N>>1. The reason is that the biochemical

process in a cell, i.e. its life time, will keep a longer time compare with its transient time N, although the model is linear but not means the process is very simple, so N >>1.

Strictly speaking, this kind of discrete model is for the convenience of computation.. If T>>N and N>>1, however, obviously there are some difficulties in computation and analysis to the discrete model. Generally speaking, a continuous model that is a convolution of input and impulse response, which means the map f, would be more reasonable. Unfortunately, there are some other difficulty for solving this kind of convolution model unless the input would be a impulse.

In [2,3] the author presented a example for linear (M,R)-system, there

$\omega = (1, 1, 0, 0, ...)$,

and $\gamma = (1, 2, 3, 4, ...)$ = natural numbers sequence. One easily obtain the metabolic map as follows

$f=\{A_1, A_2,...\}=\{1, 1, 2, 2, 3, 3, 4, 4, ...\}$.

Obviously, it does not satisfy the condition (2.1), so it is not a metabolic system in the sense of this paper.

If we want to describe a metabolism as a essential nonlinear (M,R) systems according to (2.2), the Artificial Neural Networks model is a selection. The structure of the nonlinear map f can be parameterized and f consists of the parameters including the numbers of layers, the numbers of nodes in each layer and the all weights.

III. Making a copy of metabolic map f

The repair subsystem introduced in [2,3] has a basic function that to restore the metabolic subsystem while a cell is failure. It means that to make a copy of the map f when the metabolic subsystem (cell) is alive and substitute it when the cell go to dead. No doubt, this is the basic meaning for metabolism. However, the real process is much more complicated. In fact, from the observations of biologists some cells go to dead in the metabolic process and some others are split into two almost same parts. So, any model to describe the biochemical process in a cell, whatever linear or nonlinear, should have two changes: one is to characterize the process to go to death, another one is to fit the function of replication by split process. The model (2.3) can be used to explain this repair function.

Here we emphasize that to make a copy of the map f means the translation and restore process of genetic information, but replication means duplicate a real cell. Some biological explanations have given by using the experiments of synthesis of "wet" protein, but what does it means in mathematical model? We think of that it means the **system identification.** We explain the repair map P_f in [2,3] as a process of the system identification, there the function of P_f is to make f^*, which is a copy of the information included in f, by using the output γ of metabolic machinery (in fact, including the input ω also!). If we accept (2.3) as the model of linear (M,R)-systems, then there are several method to identify $(A_1, A_2, ... A_N)$, that is the map f. Of cause, some mathematical conditions should be satisfied, for example, the condition of permanent exciting. (2.3) can be rewritten as

$$\gamma = Uf$$

and $$\hat{f} = U^+ \gamma, \qquad (2.4)$$

where U^+ is the pseudo-inverse of U. we can take $\hat{f}$ as f^*. We say map f^* is a copy of f that means $\|f^*(\omega) - f(\omega)\| < \varepsilon$, for any given $\varepsilon > 0$. If input u_i would be stochastic, then we have to use the density of power spectrum to identify f. If this system can not be identified, for instance, the repair machinery Ri use the output of the correspondent metabolic machinery as its input, then it is impossible to make the copy of the map f, because the condition of permanent exciting not be satisfied. We call this metabolic machinery (cell) **non-repairable.** Otherwise, it is **repairable.** The **stabilization** problem of repair machinery P_f discussed in [2,3] become the **consistency** problem here. If a cell is replicatable but not stabilized or not consistence, then this cell would be "**Cancer cell".**

IV. Failure and replication

The failure of a metabolic machinery. In the model (2.3) if $y_t = 0$, as $t \geq T$ and $y_t \neq 0$, as $t < T$, then we call the metabolic machinery is **dead at time T**. It implies that $\|A_j\|$, i = 1, 2, ... N. become smaller and smaller since certain time $t^* < T$ and then $\|A_j\| = 0$, i = 1, 2, ... N. ultimately at time T. More exactly speaking, the failure of a metabolic machinery means that some change is happened in map f or in inputs.

Based on our opinion, in, the total number of the cells in a mature organization keeps as a constant, occur of replication is with death of a cell. At time T a cell dead and send a control signal to start a replication process. In real biological process the replication is the process of cell's split, in our mathematical model the replication means the split of the matrix $\boldsymbol{F}$ into two parts: upper and lower. For example, if the cell received a replication start signal then the follows:

$$F = \begin{pmatrix} A_1 & 0 & 0 & \cdots & & 0 & 0 \\ A_2 & A_1 & 0 & 0 & & \ddots & 0 \\ A_3 & A_2 & A_1 & 0 & 0 & \ddots & \vdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ A_N & & \cdots & & \cdots & A_2 & A_1 \\ A_1 & & & & & A_3 & A_2 \\ \vdots & & \vdots & & \vdots & \vdots & \vdots \\ A_{L-1} & \cdots & A_2 & A_1 & A_N & \cdots & A_L \end{pmatrix}_{t \times N} \Rightarrow$$

$$FU = \begin{pmatrix} A_1 & 0 & 0 & \cdots & & 0 & 0 \\ A_2 & A_1 & 0 & 0 & & \ddots & 0 \\ A_3 & A_2 & A_1 & 0 & 0 & \ddots & \vdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ A_N & & \cdots & & \cdots & A_2 & A_1 \\ \vdots & & & & & \vdots & \vdots \\ A_{r-1} & \cdots & \cdots & \cdots & \cdots & A_{r+1} & A_r \end{pmatrix}_{m \times N}$$

and

$$FL = \begin{pmatrix} A_r & \cdots & A_1 & A_N & \cdots & A_{r+2} & A_{r+1} \\ \vdots & & \vdots & & \vdots & & \vdots \\ A_{L-1} & \cdots & A_2 & A_1 & A_N & \cdots & A_L \end{pmatrix}_{(t-m) \times N}$$

We call the both as ***FU*** and ***FL***. Obviously, the two metabolic machinery consisted of ***FU*** and ***FL*** have the same behaviors with ***F***, so the replication process finished.

V. Mechanism of repair

Based on our opinion and understanding to metabolism- repair systems the repair function itself includes two stages: one is to make a copy of the metabolic map f , and then to split itself into two parts which have the same behaviors with the original map when it received a control signal from a cell while which is failing . In fact, we think that there is replication but not map β_f to make a copy of map P_f , it is unnecessary to consider the question that how to repair the repairer? Otherwise, there would be some infinite circle for repairing. The metabolism-repair systems we proposed has the following scheme:

f — ***no*** → alive and go to dead
$\Omega \longrightarrow \Gamma$ [Is this repairable?]
(metabolism) ***yes***
↪ $\xrightarrow{P_f} f^*$ $[\in H(\Omega,\Gamma)]$
(translation)
(repair machinery) ← control signal to start
↓ split process
$F = FU + FL$
(transcription and replication)

Fig. 3 The scheme of repair mechanism

VI. Conclusion

In this paper we showed some new look at Metabolism- Repair system, especially to linear one. The main opinions are as the following:

1) Due to the life of a cell is finite and limited, the discrete model to metabolic machinery should be finite degree but not infinite. Of cause, it may be a big finite. some special identification method should be developed. More reasonably a continuous model is better.

2) To make a copy of metabolic map f means to restore the information hidden in genes, it similar to that process from DNA to mRNA. Mathematically, it is Systems Identification.

3) There are some cells which are non-repairable, they can not be repaired while they failure or dead.

4) The repair process to those repairable cells is, at first, to make a copy of metabolic map, and then to start the replication or transcription by splitting itself into two cells which have the same behavior when it received control signal from a cell being dead.

5) The map β_f to repair the repairer is unnecessary.

Of cause, we consider one cell here only, the situation for (M,R)-network would be much more complicated. There are still many problems and questions to open.

References

[1] Rosen.R, "Some relational cell models: the metabolism-repair systems." In: Foundations of Mathematical Biology, Vol.2, edited by R.Rosen, Academic Press, New York, 1972, pp.217-253.

[2] Casti, J. L, "Linear Metabolism-Repair Systems", Int. J. General Systems, Vol.14, pp.143-167, 1988

[3] Casti, J. L, REALITY RULES: Picturing the World in Mathematics, Chapter 7, JOHN WILEY & SONS. INC, 1992,

Information Processing Using Chaos With Application to Mobile Robot Navigation Problem

Changkyu Choi and Ju-Jang Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-Dong Yusong-Gu, Taejon 305-701, KOREA
FAX : +82-42-869-3410 Email : jjlee@ee.kaist.ac.kr *

Abstract

What we pursue throughout this study of information processing using artificial neural network and chaos is to devise a memory model that resembles human behavioral characteristics. For that purpose we construct a framework of the macroscopic model of responding process in biological systems. Incoming stimulus is applied to the sensory receptors and preprocessed in the pattern matching block. The matched pattern output which is the vector composed of parameter set and/or initial conditions switches to one of the chaotic memory which basically wanders chaotically. After the chaotic memory is stabilized to one of the stable equilibrium points and the limit cycles, the performance of it is estimated. If the performance is not good enough to execute a certain task, a rearranging and reshaping process of the attractors are followed. As a chaotic memory we use the famous Lozi map because it has rich dynamics with simple equations. Simulations are performed for the mobile robot regulation problem in case of slight modification of the pre-planned path which is the specific ordered series of points of the Lozi attractor.

1 Introduction

A salient aspect of neurons in biological systems is their incredibly large number. The human cerebral cortex has more than 10^{10} neurons, and there are probably ten times that number in the nervous system as a whole. Each neuron receives 1,000 to 100,000 synapses, and each neuron transmits to a correspondingly large number of neurons in its neighborhood [7]. Therefore feedforward convergence and divergence of connections with continuous distributions are found in many areas of cortex. Neural populations do not exist merely because of large numbers of neurons driven by common input in parallel. They arise by virtue of the feedback interactions among cortical neurons that yield cooperative activity. In ANN, autoexcitation is commonly used as a nonbiological way to compensate an ANN for the use of the first order ODE and the omission of explicit terms for synaptic and dendritic cable delays. As a result of investigating and modeling of biological neurons, the artificial neural network (ANN) was born and has been popular due to its function approximation capability.

Related to control theory, many researchers have tried to find the efficient way of using the ANN for control algorithm. Despite the generic defects ANN bears, flourishing of this field stems from a certain belief that the perfect mimicry of human behaviors will come true someday. Though the modern technical improvement enables us to precisely control the systems of severe nonlinearity, no one doubts there are kinds of works in which man is superior to the automatic controller, such as car driving and parking problems.

What we pursue throughout this study of information processing using ANN and chaos is to devise a memory model that resembles human behavioral characteristics. It is believed that the chaos is a basal state in biological systems. When a pretrained known stimulus is applied to the sensory

*This research is partially supported by the Korea Science and Engineering Foundation under grant 956-0900-002-2.

receptors, the chaotically wandering states converges to the one of their limit cycles to represent the meaning of the stimulus for the subject instead of the properties intrinsic to the stimulus. This means that the output pattern is more dependent on past experience with the stimulus stored in the modified synapses than it is on the stimulus attributes. Based on these investigations we proposed a macroscopic model of biological information processing systems [8].

The organization of this paper is as follows. The proposed memory structure is elucidated in section 2. And the dynamics of the famous Lozi attractor is briefly discussed. Section 3 gives a path planning algorithm for the holonomic mobile robot. To construct a feasible path of the nonholonomic mobile robot we will rearrange and reshape the attractor trajectories. Finally, we draw some concluding remarks and further works in section 4.

2 The Proposed Memory Structure of the Biological System

Figure 1 illustrates the macroscopic model of the responding process in biological systems. In our model we assume that chaos is the basal state of the recognition process, and adds some perturbation to the reminded data when none of the prestored data results in good performance. An ultimate aim of the chaotic memory is to design a strange attractor, which is an union of the closures of unstable periodic orbits. Each of the unstable as well as stable periodic orbits contains its own information correspondingly. Since the periodic points are dense in the strange attractor of the bounded phase space and we use both of the stable equilibrium points and the limit cycles as the memory agents, the capacity of the memory is greatly improved.

As things now stand, however, we don't know how to design such a versatile memory. Andreyev and his colleagues proposed one challenging way to store and retrieve informations on the stable limit cycles of one–dimensional maps [1]. They showed that their memory model can be realized by neural networks based on piecewise–linear maps [2] [3]. But, by their method generally $2n + 1$ linear segments are needed to store n symbols.

While many theoretical researchers pay their attention to chaos because of its strangeness, a lot of practical engineers noticed chaos by their rich dynamics because the system with very simple structure shows a variety of dynamic regimes by varying their parameters and initial conditions. Thus their problems are how to use it and what is the application if we incorporate it into the system of interest. Our unsolved problems is that how to design such a strange attractor, which contains the prescribed points as an attractor in it, more effectively and more simply. At this stage this remains as the further work and we focus our attention to the attractor being already in existence. For example, the Lozi system which is a simplified version of the Henon map is represented by the following iterations.

$$x(n+1) = 1 + y(n) - a|x(n)| \tag{1}$$

$$y(n+1) = bx(n) \tag{2}$$

As shown in Fig. 2, this simple deterministic system reveals to be asymptotically stable, periodic and even chaotic.

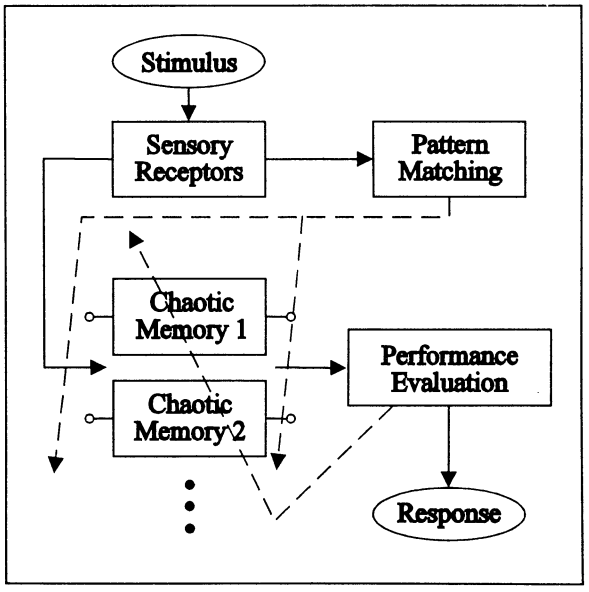


Fig. 1 The macroscopic model of the responding process in biological systems.

To elucidate the function of each block in Fig. 1, we consider the regulation problem of the holonomic mobile robot. For this purpose we take only the position of the mobile robot into account. The stimulus is the bipolar code depending on the regulating situation that the robot is confronted. This

encoded code is passed to the pattern matching block which is an associative memory and the most similar parameters are obtained through adaptive resonance.

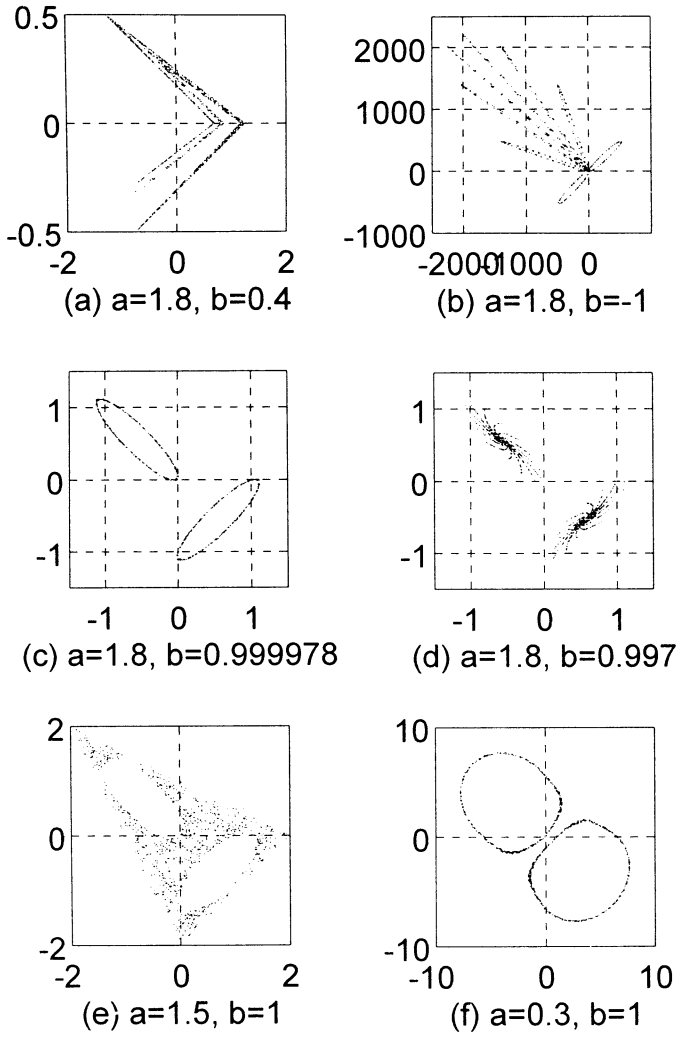


Fig. 2 A variety of dynamic regimes in Lozi map.

Then, the classified parameters which are judged to proper to the given situation are applied to the chaotic memory. The existence of many chaotic memories are possible because the parking, the driving inside the city and the driving on the highway call for different strategies. Since we consider the regulation problem alone, the only chaotic memory is needed, which is the Lozi system depicted in Fig. 2(d). While all the figures show their steady state responses, the trajectory of Fig. 2(d) is not multi-segments but one segment that gyrates to the one of the equilibrium points. If the extracted path is not suitable to the situation to some degree, then a rearrangement of the paths and the retrial of the pattern matching are performed. Simply speaking, the role of the pattern matching and chaotic memory blocks is to generate a feasible geometric path that the robot can easily follow. When none of the 14 geometric paths are not suitable to follow, a kind of deformations such as twist and stretch take places to fit in one of the geometric paths for the regulating situation of the vehicle.

3 Regulation Problem of a Mobile Robot

As a simple illustration, we consider a holonomic mobile robot regulation problem. The problem is to find a feasible path from the randomly starting point to the origin. Six parameter sets (a, b) are given initially which are $(1.8, 0.4)$, $(1.8, -1)$, $(1.8, 0.999978)$, $(1.8, 0.997)$, $(1.5, 1)$ and $(0.3, 1)$. The default parameter set is $(1.5, 1)$ and the $(1.8, 0.997)$ is always selected because we consider the regulation problem. As a given candidate path, left 14 spiral trajectories of Fig. 2(d) are chosen. They are translated to the origin and deformed in order to form a circle (Fig. 3). To be clearly the original gyrating trajectory of Fig. 2(d) is rotated 45^o clockwise, translated down vertically and then stretched 4 times that size horizontally. Assuming that the asymptotically stable equilibrium point be (x_e, bx_e), the combination of linear transformations reads

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 4c\theta & 4s\theta \\ -s\theta & c\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_e \\ bx_e \end{bmatrix}, \quad (3)$$

where $c\theta$ and $s\theta$ mean $cos(\pi/4)$ and $sin(\pi/4)$, respectively.

The numbers from 1 to 28 represent the evolution history of the system, so that the points $(x(1), y(1)), (x(29), y(29)), (x(57), y(57))$ and so on form a trajectory to the origin. Quadruple vectors encoded as the bipolar codes are the representatives of the corresponding trajectory. A code $c = (-1, 1, -1, 1)$ is a binary number 0101 which is 5 as a decimal number. In Fig. 3 the most significant bit, -1s are omitted. The 14 bipolar encoded patterns are stored in connection matrix

using outer product construction below.

$$T = \sum_{\alpha=1}^{14} (c_\alpha c_\alpha^T - I_4) \tag{4}$$

The most similar fundamental pattern is retrieved through the adaptive resonance asynchronously.

$$c_i' = sgn\left(\sum_{j=1}^{4} T_{ij} c_j\right) \tag{5}$$

If the performance of the resultant path is bad, it seeks a new path by using the methods such as in [4][5][6]. Because four unreserved codings are present, the newly generated pattern should be reflected to the connection matrix by

$$T_{new} = T_{old} + (c_{new} c_{new}^T - I_4) \tag{6}$$

But we did not consider this operation in simulation study. Simulation results are not presented in this paper for the reason of the paper length limitation. The results came up to our expectation.

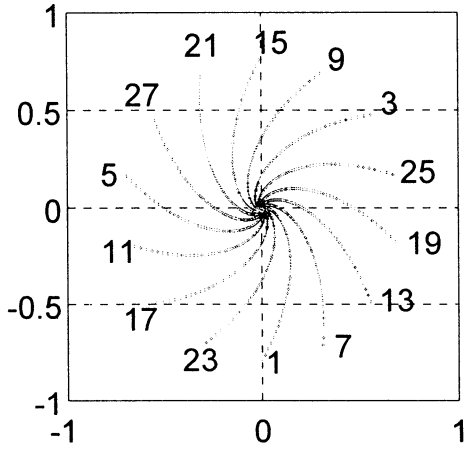


Fig. 3 The linearly transformed gyrating trajectory of Fig. 2(d).

4 Conclusion

We devise a memory model which mimics the responding process in biological systems. Initially the memory wanders chaotically and when a stimulus comes, one of the pre-stored patterns are retrieved by switching the parameters obtained through the associative memory. The holonomic mobile robot regulations are investigated. This method is also directly applicable to the nonholonomic mobile vehicle parking problem by the rearrangement of attractors.

References

[1] Yu. Y. Andreyev, A. S. Dmitriev, L. O. Chua and C. W. Wu, "Associative and Random Access Memory Using One–Dimensional Maps," *International Journal of Bifurcation and Chaos*, Vol. 2, No. 3, pp. 483–504, 1992.

[2] Yu. Y. Andreyev, A. S. Dmitriev and D. A. Kuminov, L. O. Chua and C. W. Wu, "1–D Maps, Chaos and Neural Networks for Information Processing," *International Journal of Bifurcation and Chaos*, Vol. 6, No. 4, pp. 627–646, 1996.

[3] Yu. Y. Andreyev, Yu. L. Belsky, A. S. Dmitiev, and D. A. Kuminov, "Information Processing Using Dynamical Chaos: Neural Networks Implementation," *IEEE Transactions on Neural Networks*, Vol. 7, N. 2, pp. 290–299, 1996.

[4] D. Challou, D. Boley, M. Gini and V. Kumar, "A Parallel Formulation of Informed Randomized Search for Robot Motion Planning Problem," in *Proc. IEEE Int'l Conf. Robotics and Automation*, Vol. 1, pp. 709–714, Nagoya, Japan, May 21-27, 1995.

[5] C. Choi and J.J. Lee, "Chaotic Local Search Algorithm," in *Proc. AROB*, pp. 14–17, Feb. 18–20, Beppu, Japan, 1996.

[6] C. Choi and J.J. Lee, "Finding Multiple Local Minima Using Chaotic Jump," accepted and to be published in *International Journal of Intelligent Control and Systems*, 1997.

[7] Walter J. Freeman, "Tutorial on Neurobiology: From Single Neurons to Brain Chaos," *International Journal of Bifurcation and Chaos*, Vol. 2, No. 3, pp. 451–482, 1992.

[8] T.D. Eom, S.W. Kim, C. Choi and J.J. Lee, "New Skill Learning Paradigm Using Various Kinds of Neurons," in *Proc. IROS*, Vol. 3, pp. 1157–1164, Nov. 4–8, Osaka, Japan, 1996.

REALIZATION OF ARTIFICIAL INTELLIGENCE FOR INTEGRATIVE EEG INTERPRETATION

Masatoshi NAKAMURA, Takenao SUGI
Department of Electrical Engineering,
Saga University,
Honjomachi, Saga 840, Japan

Hiroshi SHIBASAKI, Akio IKEDA
Department of Brain Pathophysiology,
Faculty of Medicine, Kyoto University,
Sakyo-ku, Kyoto 606, Japan

Abstract

A full automatic interpretation of awake electroencephalogram (EEG) had been developed by the authors. The automatic integrative EEG interpretation consists of four main parts: quantitative EEG interpretation [7], EEG report making, preprocessing of EEG data [8] and adaptable EEG interpretation [9]. The automatic integrative EEG interpretation reveals essentially the same findings as the electroencephalographer's (EEGer's), and then would be applicable in clinical use as an assistant tool for EEGer. The method had been developed through collaboration works between the engineering field (Saga University) and the medical field (Kyoto University). This work can be understood as a realization of artificial intelligence. The procedure for realization of artificial intelligence be applicable in other fields of systems control.

Keywords: automatic integrative EEG interpretation, awake EEG, realization of artificial intelligence, neural network

1 INTRODUCTION

The electroencephalogram (EEG) is a summation of electrical activities generated by nerve cells in the cerebral cortex and recorded from the scalp. As the EEG reflects at least a part of the functional status of the subject's brain, the EEG interpreted by electroencephalographers (EEGers) is used as a diagnostic aid in diseases affecting the brain. Automatic analyses of the EEG have been attempted by many investigators. Among various kinds of automatic EEG interpretations previously reported, spike detections in either the scalp-recorded or depth-recorded EEG in epileptic patients has drawn particular attention [3],[4]. Automatic analyses of sleep stages by using EEG data have also been implemented [10], and a high accuracy in the recognition was achieved. Automatic diagnosis of awake EEGs has been investigated by Kowada et al [6]. Although, an awake background EEG is a basis of the subject's EEGs, most previous studies of automatic diagnosis for awake background EEG were restricted to a certain aspect of EEG characteristics such as the dominant rhythm or slow waves. A full automatic integrative interpretation of an EEG record had not been accomplished, mainly because a large number of complex items are involved in the interpretation of the awake background EEG. The automatic integrative interpretation of awake background EEG has been developed through an intimate collaborations between the engineering site and the medical site of the authors.

This paper describes the whole story of the automatic interpretation method of awake background EEG, and shows how the methods have been developed through the collaboration work between the both fields. The way of developing of the automatic EEG interpretation is extended in general, and summarized as a methodology of realization of artificial intelligence, which will be adopted to a wide range of fields for systems control, such as power systems, plant systems and mechatronic systems.

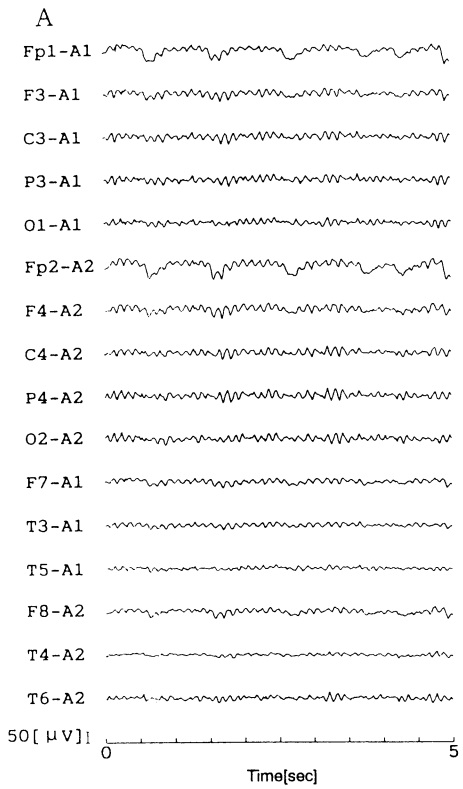
2 EEG RECORDING AND INSPECTION

2.1 EEG recording

All EEGs were recorded in a quiet, dimly lit room where a subject was placed in a supine position in a bed with the eyes closed. Exploring cup electrodes were fixed to the scalp at points Fp_1, F_3, C_3, P_3, O_1, Fp_2, F_4, C_4, P_4, O_2, F_7, T_3, T_5, F_8, T_4 and T_6 of the international $10-20$ system, and all electrodes were referenced to the ipsilateral ear electrode (A_1 or A_2). The 16-channel EEGs were recorded by an electroencephalograph with the time constant of 0.3 sec and a high frequency cut-off at 120 Hz(-3dB) at a paper speed of 3cm/sec and a sensitivity of 0.5cm/50μV.

2.2 Visual inspection of EEGs by a qualified EEGer

The procedure of visual inspection of EEGs was sequentially categorized into 16 items and was arranged



B 1 B 2

	Items	EEGer Estimate	EEGer Score	Automatic Estimate	Automatic Score
	Dominant rhythm				
1	Existence	Yes	0	Yes	0
2	Organization	Mild.ab.	1	0.9	1
3	Asymmetry[1]	0	0	0.2	0
4	Frequency[Hz]	9.5	0	9.2	0
5	Asymmetry[Hz]	0	0	0.1	0
6	Amplitude[μV]	40	0	31.7	0
7	Asymmetry[%]	0	0	20.0	0
8	Extension[μV]	Fp, AT	2	8.3	0
	Beta rhythm				
9	Amplitude[μV]	5	0	10.3	0
10	Asymmetry[%]	0	0	20.2	0
	Theta rhythm				
11	Duration [%]	4	1	3.6	1
12	Electrodes	indefinite		$P_{z,12}, F_{z,4,7,8}$ C_z, P_4, O_2	
	Delta rhythm				
13	Duration [%]	0	0	0.0	0
14	Electrodes	---		---	
	Non-dominant alpha rhythm				
15	Duration [%]	0	0	21.1	1
16	Electrodes	---		$F_z, C_{z,4}, P_{z,4}$ $O_{1,2}, T_{z,8}$	

C 1 EEGer's report

Mildly abnormal record

because of

(1)poorly organized background activity,
(2)too anteriorly extended α rhythm,
(3)occasional θ waves diffusely.

C 2 Automatic EEG report

Mildly abnormal waking record

because of

(1)poorly organized background activity,
(2)occasional θ waves bianteriorly, and
(3)occasional waves of α frequency posteriorly.

[Artifact detection]
Blink artifacts were present in the following segments(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), and were taken into account in the interpretation.

[Vigilance level]
Waking record

Figure 1: EEG record and its EEG interpretation. (a) Time series of 5 sec. long out of 50 sec record. (b) the quantitative interpretation for 16 items. (c) the corresponding EEG report in a written form. The remarks on artifact and vigilance level in the automatic EEG report (c) were correct by described using the preprocessing with the artifact detection and the vigilance level detection procedures. (cited from [8])

in order of interpretation for the purpose of developing an automatic EEG interpretation (Fig. 1). The 50 sec long EEG record of each subject was inspected quantitatively according to the following standard criteria, without any prior knowledge of the clinical picture of each subject. First of all, the dominant rhythm was defined as a dominant EEG activity that consisted of waves with an approximately constant period and with the maximum amplitude at the occipital or parieto-occipital regions (O_1, O_2, P_1, P_2) of the head. As the first step, the dominant rhythm was checked as to whether it was recognizable or absent. If the dominant rhythm was identified, it was then analyzed with respect to its organization, frequency and amplitude. Differences of these 3 quantities of the dominant rhythm between the corresponding sites of the two hemispheres were defined as 'asymmetry'. Extension of the dominant rhythm to anterior regions of the head was also taken into account. For beta rhythm, the amplitude and asymmetry in its amount within the band(13Hz<) at each electrode were taken into account. As regards the waves of theta(4-8 Hz) and delta(<4 Hz) frequency and the activity of alpha frequency (8-13Hz) other than the dominant rhythm, which was defined as 'non-dominant alpha rhythm,' the proportional duration of the respective rhythm at each electrode was estimated and expressed as a percentage, and the electrodes active for each component were listed. Every item of EEG was graded into 4 scores: normal (0), mildly abnormal (1), moderately abnormal (2) and markedly abnormal (3). The criteria (threshold value) for each score were quantitatively specified for each item [7].

2.3 Computation of periodogram parameters

The EEG records that were interpreted by the EEGer as described above were divided into 10 segments of 5 sec duration each. The digital data of the EEG time series were then transformed into the Fourier components by the FFT method [2], and the periodogram of each segment was obtained for each of the 16 channels as the squares of the Fourier components for each frequency of 0.2Hz. Features of the periodogram for each 5sec EEG segment for each in-

dividual channel were expressed by the periodogram parameters for each of the 5 frequency : the dominant rhythm (d), and beta (β), theta (θ), delta (δ) and non-dominant alpha (α) bands. The periodogram consisted of 11 parameters : the amount of the periodogram within the respective bands (S_d, S_β, S_θ, S_δ and S_α), the peak frequency (f_d, f_β, f_θ, f_δ and f_α), and the standard deviation for alpha band(σ_α). Thus, the time and spatial features of the 16 channel EEG record of 50 sec duration for each subject were condensed into 1760 periodogram parameters (11 parameters $\times$ 16 channel $\times$ 10 segments). Based on the periodogram parameters, we developed an automatic EEG interpretation which was equivalent to the EEG interpretation done by the EEGer.

3 AUTOMATIC EEG INTERPRETATION

3.1 Quantitative EEG interpretation

By using the periodogram parameters thus obtained, EEG parameters were constructed so that could closely conform to the procedures of visual EEG interpretation by the EEGer [7].

EEG parameters were computed for each segment data for each subject by using the periodogram parameters and the formulae as mentioned above, and the final estimates of the EEG parameters were obtained by averaging the EEG parameters of all 10 segments. Then the scores for each item were calculated based on the criteria (threshold values) which were employed for the visual inspection of each item [7] and were compared with the results obtained by the EEGer's visual inspection for each subject (Fig. 1(b)). The automatic interpretation revealed essentially the same findings as the EEGer's.

3.2 EEG report making

Although the automatic quantitative EEG interpretation was found to be in good agreement with the visual interpretation by the EEGer, the style for expressing the results of the quantitative EEG interpretation was different from that of EEG reports which was usually written by EEGers (Fig. 1(c1)), and therefore was not easily understandable to other medical doctors.

For the development of an automatic report making, the necessary terminology for the EEG report and the rules for the EEG report making were established [8] by analyzing the relationship between the EEG reports and the quantitative EEG interpretation done by the EEGer. Thus the automatic EEG report was obtained by using the defined terminology and the EEG report making rule based on the results of the automatic quantitative EEG interpretation. The integrative grade of EEG (first line of the EEG report (Fig. 1(c2)) was determined by taking into account the grading scores of each item for the integrative EEG interpretation. The appropriate term for the integrative grading of EEG was selected from the terminologies described by summing up the corresponding weights for each item for quantitative EEG interpretation. The weights for each item [8] were determined so as to attain the conformity between the results of automatic interpretation and those by the qualified EEGer for all subjects. Fig. 1(c) shows quantitative EEG interpretation for each item and EEG report, comparing the EEGer's reports. Although a slight difference of expression between the EEGer's report and the automatic report was found, total meaning of the automatic EEG reports was equivalent to that of the EEGer's report.

3.3 Pre-processing of EEG interpretation

While applying the automatic EEG interpretation method to clinical use, we have encountered two problems: contamination of various artifacts such as eye blinks, EMG (electromyogram) artifacts and electrode artifacts in the EEG record, and change in the vigilance level of the subjects during the recording. If the EEG records were automatically interpreted without paying any attention to the artifacts, misinterpretation of the EEG records is expected to occur not infrequently. Moreover, as EEG records during drowsy state are clearly different from those during wakefulness, the detection of the change in the vigilance level is an important task for EEGers for correctly interpreting awake EEGs. Therefore, the automatic detection of these artifacts and the reduced vigilance level were required as the pre-processing for the automatic EEG interpretation. The equations for detecting artifacts and reduced vigilance level were determined such that they would conform to the procedures that the EEGer actually adopts for the visual inspection of the EEG record [8].

The quality of the automatic EEG interpretation was shown to be improved by the newly developed preprocessing method even in the record contaminated with artifacts or obtained during the drowsy state of the subjects [8], and thus the present method will be effective for clinical use.

3.4 Adaptation of EEG interpretation

In view of the fact that the visual EEG interpretation could be a subjective task and may vary among the EEGers, an automatic EEG interpretation system which is adaptable to each EEGer was required. The adaptable automatic EEG interpretation was accomplished by using a constructive neural network with forgetting factor (combination of Ash's algorithm, [1] and Ishikawa's algorithm, [5]). The artificial neural

network (ANN) was constructed so as to give the integrative interpretation of the EEG based on the intermediate judgment of 13 items. The ANN consists of 3 layers initially fully connected: input, hidden and output layer. The input layer contains 13 units that represents the items considered in the quantitative interpretation. The hidden layer, initially, is formed by a variable number of units. The output layer consists of one unit that represents the final judgment that is scored again from 0 to 3 (normal(0), mildly abnormal(1), moderately abnormal(2) and markedly abnormal (3)) for the input pattern introduced. The final judgment made by the EEGer was used as the teaching signal for the ANN. The judgment emitted by the ANN was considered to be correct if the absolute value of the difference between the teaching signal and the output of the ANN was less than 0.5. The training of the ANN was performed such that the output of the neural network produced values similar to those of the final judgment made by the EEGer.

The testing consisted in verifying the generalization capability of the ANN or its ability to return appropriate results with data which were not used to train it. The testing of this ANN gave 100 % of accuracy for the data interpreted by EEGer-A. However, the generalization performance of the ANN, using the data of EEGer-B only produced 55% of accuracy, indicating that ANN-A was not good at interpreting records for EGGer-B. Since ANN-A could not well generalize for the data interpreted by EEGer-B, the adaptation process was performed. To adapt the ANN, the network was trained again by using both data interpreted by EEGer A and B. The parts that changed by the adaptation process were the amount of linking connections and the magnitude of the weights.

The adapted network (ANN-B) was able to interpret data either from EEGer A or B. When the generalization performance was verified, an improvement was obtained with 96.5% of accuracy over the training and testing data [9]. The automatic EEG interpretation of adapted one would be applicable in the clinical use as an assistant tool for EEGer and physician. As all the procedures were programmed in a personal computer equipped with AD (analogue to digital) converter, the automatic EEG interpretation was implemented in almost real time if the adaptation process was completed.

4 CONCLUSION

The automatic interpretation of awake background EEG, was summarized and was explained how the method have been developed through a collaboration works between the engineering field and medical field. This work can be understood as realization of artificial intelligence. In order to realize the artificial intelligence the designers are required to keep the following points in mind: 1) exact understanding of the objective and the human intelligence, 2) long and intimate discussion with collaborators, 3) good human relationship between collaborators. The methodology will hopefully be adopted to a wide range of different fields of systems control.

References

[1] T. Ash, "Dynamic node creation in back propagation networks". Connection Science, 1(4) , 365-375, 1989.

[2] J. W. Cooley, and J. W. Tukey "An algorithm for the machine calculation of complex Fourier series", Math. Comput., 19, 297-301, 1965.

[3] J. D. Frost Jr. "Automatic recognition and characterization of epileptiform discharges in the human EEG." J. of Clin. Neurophysiol., 2(3), 231-249, 1985.

[4] J. Gotman, "Practical use of computer-assisted EEG interpretation in epilepsy" J . of Clin. Neurophysiol., 2(3), 251-265, 1985.

[5] M. Ishikawa, "A structural learning algorithm with forgetting of link weights." Cognitive Science Section, Information Division, TR 90-7, 1-17, 1990.

[6] M. Kowada, S. Sato, H. Hiraga, T. Tomono and K. Yahata, "Automatic pattern recognition of clinical EEG: A method for waveform recognition." Clin. Electroenceph., 13, 647-656 1971(Japanese).

[7] M. Nakamura, H. Shibasaki, K. Imajoh, et al., "Automatic EEG Interpretation: A New Computer Assisted System for the Automatic Integrative Interpretation of Awake Background EEG," Electroenceph. clin. Neurophysiol., 82:423-431, 1992.

[8] M. Nakamura, T. Sugi, A. Ikeda, R. Kakigi and H. Shibasaki, "Clinical application of automatic integrative interpretation of awake background EEG: quantitative interpretation report making and detection of artifacts and reduced vigilance level", Electroencephalography and Clinical Neurophysiology 98, 103-112, 1996.

[9] M. Nakamura, Y. Chen, T. Sugi, A. Ikeda and H. Shibasaki, "Neural Network Configuration for Automatic EEG Interpretation Adaptable to Individual Electroencephalographer", Eleventh International Conference on Systems Engineering (ICSE'96), 31-36, 1996

[10] J. R. Smith, I. Karacan & M. Yang, "Automatic analysis of the human sleep EEG", Waking and Sleeping, 2:75-82, 1978

Several Variants of Fuzzy Classifier Systems for Pattern Classification Problems with Continuous Attributes

Hisao Ishibuchi, Tomoharu Nakashima and Tadahiko Murata
Department of Industrial Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka 593, Japan
{hisaoi, nakashi, murata}@ie.osakafu-u.ac.jp

Abstract: This paper examines the performance of several variants of our fuzzy classifier system by computer simulations on a real-world pattern classification problem with many continuous attributes. Those variants are different from each other in the following points: (i) rule generation procedures, (ii) population update mechanisms, (iii) definitions of the fitness function of each fuzzy if-then rule, and (iv) learning mechanisms of fuzzy if-then rules. For example, an initial fuzzy if-then rule is directly generated from each training pattern in some variants while it is randomly generated in the simplest version of our fuzzy classifier system. In other variants, the population size (*i.e.*, the number of fuzzy if-then rules) is adjusted based on the classification performance of each fuzzy if-then.
Keywords: Pattern classification, fuzzy rule-based systems, fuzzy rule generation, genetics-based machine learning.

1. Introduction

Fuzzy systems based on fuzzy if-then rules have been successfully applied to various control problems [1,2]. Fuzzy if-then rules in those fuzzy systems were usually derived from human experts as linguistic knowledge. Recently various approaches have been proposed for automatically generating fuzzy if-then rules from numerical data [3-5]. Genetic algorithms [6,7] have been widely utilized for generating fuzzy if-then rules and tuning membership functions [8-12].

For multi-dimensional pattern classification problems with many continuous attributes, we have already proposed a fuzzy classifier system [13,14]. In this paper, we examine the performance of various variants of our fuzzy classifier system by computer simulations on wine data (178 samples with 13 continuous attributes from three classes). This data set is available via anonymous ftp from *ics.uci.edu* in directory */pub/machine-learning-databases*.

2. Fuzzy Rule-Based Pattern Classification

2.1 Pattern Classification Problem

Let us consider a c-class pattern classification problem in an n-dimensional pattern space $[0,1]^n$. We assume that m training patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1,2,...,m$, are given from the c classes ($c << m$). In computer simulations of this paper, attribute values are normalized into the unit interval [0,1].

2.2 Fuzzy If-Then Rules

In this paper, we use fuzzy if-then rules of the following type for our pattern classification problem:

$$\text{Rule } R_j: \text{If } x_{p1} \text{ is } A_{j1} \text{ and } ... \text{ and } x_{pn} \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF = CF_j, \quad (1)$$

where R_j is the label of the j-th fuzzy if-then rule, $A_{j1},...,A_{jn}$ are antecedent fuzzy sets on the unit interval [0,1], C_j is the consequent class, and CF_j is the grade of certainty. As the antecedent fuzzy sets $A_{j1},...,A_{jn}$, we use the five linguistic values (*i.e.*, S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, and L: *large*) in Fig. 1. We also use a special linguistic value "DC: *don't care*" in Fig. 1. Thus the six linguistic values in Fig. 1 including "DC: *don't care*" are used as the antecedent fuzzy sets in our fuzzy classifier system. The consequent class C_j and the grade of certainty CF_j are automatically determined from the given training patterns by a heuristic rule generation procedure [11-14].

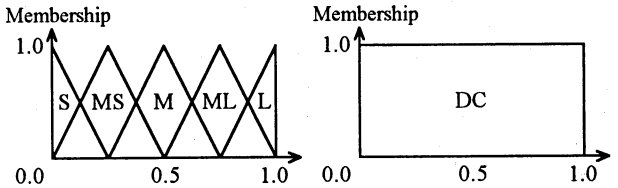


Fig. 1 Membership functions of six linguistic values (S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, L: *large*, and DC: *don't care*).

2.3 Fuzzy Reasoning

Let us denote a set of fuzzy if-then rules by S. In our fuzzy classifier system, each population corresponds to a rule set. An input pattern $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$ is classified by a single winner rule in S, which is determined as follows:

$$\mu_{\hat{j}}(\mathbf{x}_p) \cdot CF_{\hat{j}} = \max\{\mu_j(\mathbf{x}_p) \cdot CF_j \mid R_j \in S\}, \quad (2)$$

where $\mu_j(\mathbf{x}_p)$ is the compatibility grade of the input pattern $\mathbf{x}_p$ with the fuzzy if-then rule R_j. In this paper, we use the following product operation for defining the compatibility grade:

$$\mu_j(\mathbf{x}_p) = \mu_{j1}(x_{p1}) \cdot \mu_{j2}(x_{p2}) \cdot \ \cdots \ \cdot \mu_{jn}(x_{pn}), \quad (3)$$

where $\mu_{ji}(\cdot)$ is the membership function of the antecedent fuzzy set A_{ji} of the fuzzy if-then rule R_j.

The fuzzy reasoning method based on a single winner fuzzy if-then rule makes the credit assignment of our fuzzy classifier system very simple. This is because the reward for the correct classification or the punishment for the misclassification is assigned to the single winner rule.

3. Fuzzy Classifier System

3.1 Basic Algorithm

The basic algorithm of our fuzzy classifier system can be written as follows:

Step 1: Randomly generate an initial population of fuzzy if-then rules.

Step 2: Evaluate the fitness value of each fuzzy if-then rule.

Step 3: Generate new fuzzy if-then rules from the current population by genetic operations.

Step 4: Replace poor fuzzy if-then rules in the current population with the newly generated ones.

Step 5: Return to Step 2 if a pre-specified stopping condition is not satisfied.

Each fuzzy if-then rule in our fuzzy classifier system is represented by a string and handled as an individual. Because the consequent class C_j and the grade of certainty CF_j are automatically determined by the heuristic rule generation procedure [11-14], each fuzzy if-then rule is denoted by a string of its antecedent fuzzy sets as follows (see Fig. 2 and Fig. 3):

$$R_j: A_{j1}A_{j2} \cdots A_{jn}. \quad (4)$$

In the simplest version of our fuzzy classifier system, first a pre-specified number of initial fuzzy if-then rules (say, N_{pop} fuzzy if-then rules) are generated by randomly choosing their antecedent fuzzy sets from the given six linguistic values in Fig. 1. Then the generated fuzzy if-then rules are evaluated by classifying all the given training patterns. When a training pattern is correctly classified, a unit reward is given to the winner fuzzy if-then rule that is responsible for the correct classification. Thus the fitness value is defined as follows after all the given training patterns are classified by the fuzzy if-then rules in the current population.

$$fitness(R_j) = NCP(R_j), \quad (5)$$

where $NCP(R_j)$ is the number of correctly classified training patterns by R_j.

A pair of fuzzy if-then rules are selected from the current population according to their fitness values. Two fuzzy if-then rules are newly generated by a crossover operation (see Fig. 2) and a mutation operation (see Fig. 3) from the selected fuzzy if-then rules. These genetic operations (*i.e.*, selection, crossover and mutation) are iterated to generate a pre-specified number of fuzzy if-then rules (say, N_{rep} fuzzy if-then rules). The newly generated fuzzy if-then rules are replaced with the worst N_{rep} fuzzy if-then rules in the current population. As shown in the above algorithm, these procedures (*i.e.*, fitness evaluation, genetic operations, and population update) are repeated until a pre-specified stopping condition is satisfied.

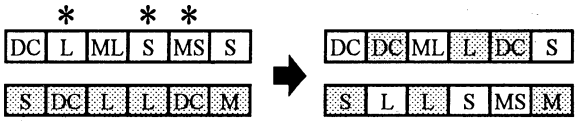


Fig. 2 Crossover operation.

Fig. 3 Mutation operation.

We applied the above-mentioned simplest version of our fuzzy classifier system to the wine data. We used the following parameter specifications:

Population size: $N_{\text{pop}} = 60$, Crossover probability: 1.0,
Mutation probability: 0.1, Replaced rules: $N_{\text{rep}} = 12$,
Stopping condition: 1000 generations.

In the execution of the fuzzy classifier system, we monitored the classification rate at each generation. Let us denote the classification rate at the t-th generation by $r(t)$. We also denote the highest classification rate until the t-th generation by $r^*(t)$: $r^*(t) = \max\{r(1), r(2), ..., r(t)\}$. We performed independent ten trials, and obtained a 98.5% average classification rate. That is, the 98.5% classification

rate was the average value of $r^*(1000)$ over the ten trials.

3.2 *Introduction of Misclassification Penalty*

We also performed the same computer simulations using the following fitness function:

$$fitness(R_j) = \omega_{\mathrm{NCP}} \cdot NCP(R_j) - \omega_{\mathrm{NMP}} \cdot NMP(R_j), \quad (6)$$

where $NCP(R_j)$ is the number of correctly classified training patterns by R_j, ω_{NCP} is the reward for the correct classification, $NMP(R_j)$ is the number of misclassified training patterns, and ω_{NMP} is the penalty for the misclassification. In computer simulations, we specified these parameters as $\omega_{\mathrm{NCP}} = 1$ and $\omega_{\mathrm{NMP}} = 10$.

By independent ten trials, we obtained a 100% average classification rate after 1000 generations. In Fig. 4, we show the average performance at each generation in the early stage of evolution: the average value of $r^*(t)$ over the ten trials for $t \le 100$.

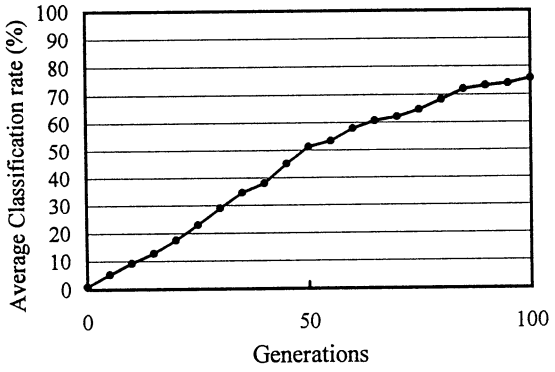


Fig. 4 Simulation results by our fuzzy classifier system.

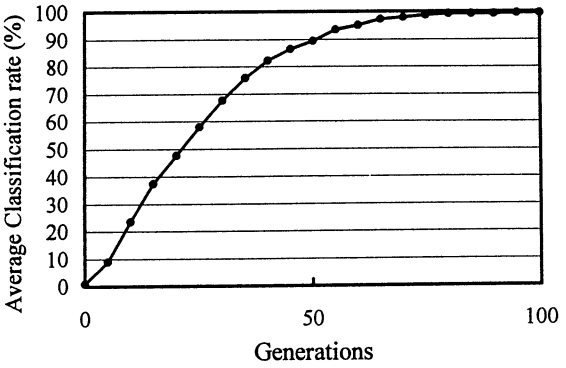


Fig. 5 Simulation results with the learning procedure.

3.3 *Learning of Fuzzy If-Then Rules*

In the simplest version of our fuzzy classifier system, the grade of certainty CF_j is determined by the heuristic rule generation procedure [11-14]. A learning procedure [15] of CF_j can be incorporated in our fuzzy classifier system. In the same manner as in Fig. 4, we applied our fuzzy classifier system with the learning procedure to the wine data. Simulation results are summarized in Fig. 5 where the learning procedure was iterated ten times at each generation.

3.4 *Heuristic Initial Population*

Initial fuzzy if-then rules can be generated directly from the training patterns in the following manner. First we randomly select N_{pop} training patterns. Then, for each of the selected training patterns, we construct the most compatible fuzzy if-then rule using only the five linguistic values excluding "*don't care.*" By this procedure, N_{pop} fuzzy if-then rules are generated. Finally, using the probability p_{DC}, we replace each antecedent fuzzy set of the generated N_{pop} fuzzy if-then rules with "*don't care.*" The obtained N_{pop} fuzzy if-then rules are used as an initial population of the fuzzy classifier system.

In the same manner as in Fig. 4, we applied the fuzzy classifier system with this initial population to the wine data. We did not use the learning procedure of the grade of certainty. Simulation results are summarized in Fig. 6.

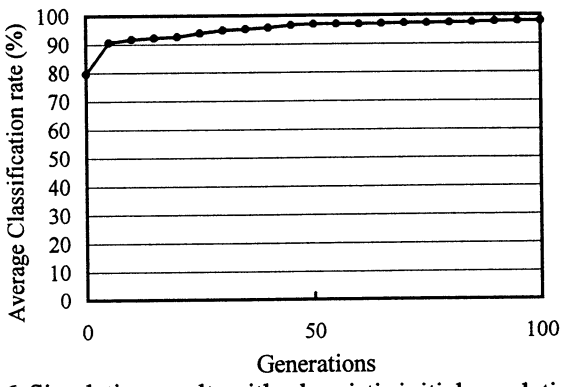


Fig. 6 Simulation results with a heuristic initial population.

3.5 *Heuristic Rule Generation*

The heuristic procedure for generating initial fuzzy if-then rules can be also employed for generating new fuzzy if-then rules for the population update. Let us denote the number of rejected patterns in the current population by m_{reject}. We specify the number of fuzzy if-then rules that are directly generated from those rejected patterns by $\min\{m_{\mathrm{reject}}, N_{\mathrm{rep}} / 2\}$. The other fuzzy if-then rules are generated by the genetic operations as in the simplest version of the fuzzy classifier system.

Simulation results by our fuzzy classifier system with this heuristic population update are summarized in Fig. 7 where the heuristic initial population was used and the learning procedure of the grade of certainty was not used.

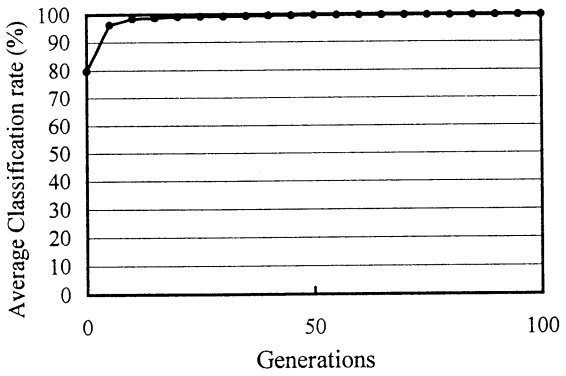


Fig. 7 Simulation results with heuristic population update.

3.6 Variable Population Size

The population size can be adjusted based on the classification performance of each fuzzy if-then rule. That is, the fixed number of new fuzzy if-then rules (*i.e.*, N_{rep} fuzzy if-then rules) are added to the current population and all the fuzzy if-then rules with non-positive fitness values are removed. In the same manner as in Fig. 7, we applied the fuzzy classifier system with the variable population size to the wine data. Simulation results are summarized in Fig. 8 where the initial population size was 60. At 1000th generation, the average population size was 112.4.

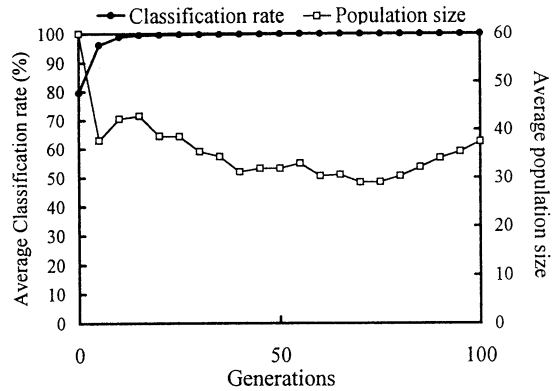


Fig. 8 Simulation results with the variable population size.

4. Conclusion

In this paper, we examined the performance of various variants of our fuzzy classifier system by computer simulations on the wine data with 13 continuous attributes. We observed from simulation results that direct rule generations from training patterns significantly improved the performance of the fuzzy classifier system.

References

[1] M. Sugeno, "An introductory survey of fuzzy control," *Information Sciences* **36** (1985) 59-83.

[2] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller," *IEEE Trans. on SMC* **20** (1990) 404-435.

[3] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. on SMC* **15** (1985) 116-132.

[4] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples", *IEEE Trans. on SMC* **22** (1992) 1414-1427.

[5] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. on Fuzzy Systems* **1** (1993) 7-31.

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

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

[8] C. L. Karr and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Trans. on Fuzzy Systems* **1**, (1993), 46-53.

[9] B. Carse *et al.*, "Evolving fuzzy rule based controllers using genetic algorithms," *Fuzzy Sets and Systems* **80** (1996) 273-293.

[10] Y. Yuan, and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," *Fuzzy Sets and Systems* **84**, (1996), 1-19.

[11] H.Ishibuchi *et al.*, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets and Systems* **65** (1994) 237-253.

[12] H. Ishibuchi *et al.*, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. on Fuzzy Systems* **3** (1995) 260-270.

[13] H. Ishibuchi *et al.*, "A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems," *Proc. 2nd ICEC* (1995) 759-764.

[14] H. Ishibuchi *et al.*, "Genetic-algorithm-based approaches to the design of fuzzy systems for multi-dimensional pattern classification problems," *Proc. 3rd ICEC* (1996) 229-234.

[15] K. Nozaki *et al.*, "Adaptive fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems* **4** (1996) 238-250.

Improving the performance of Q-Learning by Fuzzy Logic

Hisao Ishibuchi, Chi-Hyon Oh and Tomoharu Nakashima
Department of Industrial Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka 593, Japan
{hisaoi, oh, nakashi}@ie.osakafu-u.ac.jp

Abstract: This paper examines the performance of fuzzy Q-learning where a Q-value for each state-action pair is calculated by a fuzzy inference system with fuzzy if-then rules. The main advantage of the fuzzy Q-learning is the applicability to the case of continuous states and/or continuous actions where the application of the traditional Q-learning with look-up tables is difficult. The usefulness of the fuzzy Q-learning in such case has been already demonstrated in the literature. The main aim of this paper is to examine the performance of the fuzzy Q-learning by computer simulations on a simple grid world problem to which the traditional Q-learning can be successfully applied. In our computer simulations, the fuzzy Q-learning and the traditional Q-learning are compared with each other.

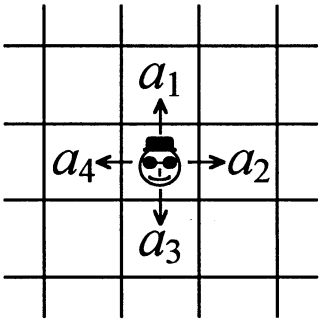
1. Introduction

Q-learning [1] is one of the most well-known reinforcement learning schemes. A Q-value is assigned to each state-action pair, and it is updated based on a reinforcement signal (*i.e.*, reward or punishment) given from the environment after a particular action or after a series of particular actions. In the traditional Q-learning, Q-values are stored in the form of look-up tables. Fuzzy Q-learning [2-5] is an extension of the Q-learning to the case of continuous states and/or continuous actions where the application of the traditional Q-learning is difficult. In the fuzzy Q-learning, a Q-value is calculated by a fuzzy inference system with fuzzy if-then rules (see, for example, Lee [6] for the fuzzy inference). The usefulness of the fuzzy Q-learning for such case has been already demonstrated in the literature [2-5].

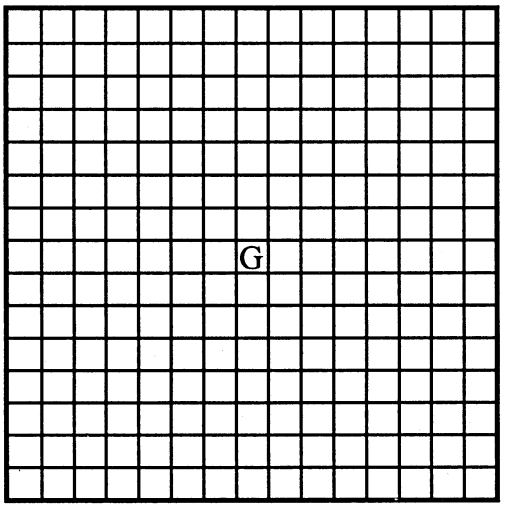
The main aim of this paper is to examine the performance of the fuzzy Q-learning by comparing it with the traditional Q-learning. In this paper, first we briefly explain a simple grid world problem that is used for evaluating the performance of the traditional Q-learning and the fuzzy Q-learning. In the grid world problem, a single agent tries to reach a goal in a 15×15 grid world. Then we explain the basic mechanisms of the fuzzy Q-learning where a Q-value for each state-action pair is calculated by a fuzzy inference system with fuzzy if-then rules. Finally we compare the traditional Q-learning with the fuzzy Q-learning by computer simulations on the grid world problem.

2. Grid World Problem

Let us consider a 15×15 grid world in Fig. 1 where a single goal is placed at the center of the grid world. As shown in Fig. 1, an agent moves by choosing one of the four possible directions at each time step. The agent starts from a random initial position. When the agent reaches the goal,



(a) Four moves of the agent.



(b) A 15×15 grid.

Fig. 1 A simple grid world problem.

it receives a unit award (*i.e.*, the reinforcement signal is +1). On the other hand, the agent hits against the wall, it receives a unit punishment (*i.e.*, the reinforcement signal is −1) and dies. When the agent does not reach the goal nor the wall during 100 time steps, it also dies without receiving any reinforcement signal.

3. Fuzzy Q-Learning

For calculating a Q-value of each action at each position in the two-dimensional grid world problem, we use the following fuzzy if-then rules:

$$\text{Rule } R_{ij}: \text{If } x_1 \text{ is } A_{j1} \text{ and } x_2 \text{ is } A_{j2} \text{ then } Q(\mathbf{x}, a_i) = Q_{ij}, \quad i = 1,2,3,4;\ j = 1,2,...,N, \tag{1}$$

where R_{ij} is the rule label, i indexes actions, a_i is the i-th action, j indexes rules, $\mathbf{x} = (x_1, x_2)$ is a state (*i.e.*, position) in the two-dimensional grid world, A_{j1} and A_{j2} are antecedent fuzzy sets, Q_{ij} is a consequent Q-value, and N is the total number of fuzzy if-then rules for each action. In this paper, we use three linguistic values (*i.e.*, S: *small*, M: *medium*, and L: *large*) in Fig. 2 as the antecedent fuzzy sets A_{j1} and A_{j2}. Thus the grid world is partitioned into nine fuzzy subspaces as shown in Fig. 3. In computer simulations of this paper, we use the eight fuzzy if-then rules $R_{i1} \sim R_{i8}$ in Fig. 3 for calculating the Q-value of each action (*i.e.*, $N = 8$ in (1)).

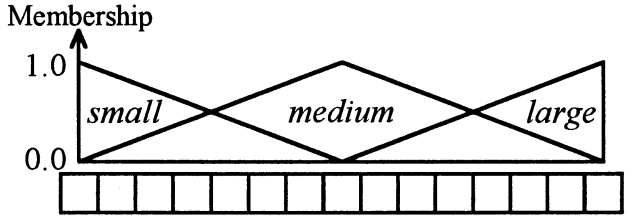


Fig. 2 Antecedent fuzzy sets.

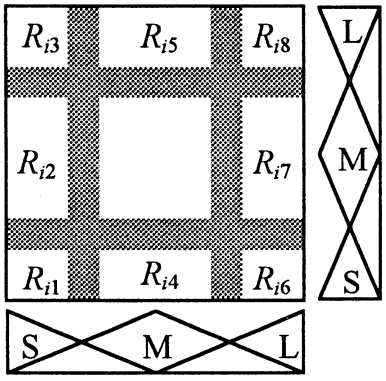


Fig. 3 Fuzzy partition and fuzzy if-then rules.

These eight fuzzy if-then rules for the action a_i ($i = 1,2,3,4$) are written as follows:

R_{i1}: If x_1 is *small* and x_2 is *small* then $Q(\mathbf{x}, a_i) = Q_{i1}$,
R_{i2}: If x_1 is *small* and x_2 is *medium* then $Q(\mathbf{x}, a_i) = Q_{i2}$,
. . . . . .
R_{i8}: If x_1 is *large* and x_2 is *large* then $Q(\mathbf{x}, a_i) = Q_{i8}$.

The fuzzy Q-learning requires less memory storage than the conventional one. That is, the size of a fuzzy rule table is much smaller than that of a look-up table. For example, the look-up table has $4 \times 15 \times 15 = 900$ entries (*i.e.*, 900 Q-values) for our grid world problem with the four actions while the fuzzy rule table consists of $4 \times 8 = 32$ fuzzy if-then rules. A small number of fuzzy if-then rules can handle the entire grid world because they have high interpolation ability based on the fuzzy logic.

The Q-value of each action at each position is calculated from the N fuzzy if-then rules in (1). Let us define the compatibility of the position $\mathbf{x} = (x_1, x_2)$ with the fuzzy if-then rule R_{ij} by the product operator as follows:

$$\mu_{ij}(\mathbf{x}) = A_{j1}(x_1) \cdot A_{j2}(x_2), \quad i = 1,2,3,4;\ j = 1,2,...,N, \tag{2}$$

where $A_{j1}(\cdot)$ and $A_{j2}(\cdot)$ denote the membership functions of the antecedent fuzzy sets A_{j1} and A_{j2}, respectively. The Q-value of the action a_i at the position $\mathbf{x} = (x_1, x_2)$ is calculated as follows:

$$Q(\mathbf{x}, a_i) = \frac{\sum_{j=1}^{N} \mu_{ij}(\mathbf{x}) \cdot Q_{ij}}{\sum_{j=1}^{N} \mu_{ij}(\mathbf{x})}, \quad i = 1,2,3,4. \tag{3}$$

Each action is selected at the position $\mathbf{x} = (x_1, x_2)$ with the following probability:

$$P(a_i) = \frac{Q(\mathbf{x}, a_i) - Q_{\min}}{\sum_{i=1}^{4} \{Q(\mathbf{x}, a_i) - Q_{\min}\}}, \quad i = 1,2,3,4, \tag{4}$$

where

$$Q_{\min} = \min\{Q(\mathbf{x}, a_i) \mid i = 1,2,3,4\}. \tag{5}$$

When the agent moves from $\mathbf{x}^t = (x_1^t, x_2^t)$ to $\mathbf{y}^t = (y_1^t, y_2^t)$ by an action a^t at the time step t, the consequent Q-value of each fuzzy if-then rule is updated as

$$Q_{ij}^t = \begin{cases} Q_{ij}^{t-1} + \Delta Q_{ij}^t, & \text{if } a^t = a_i, \\ Q_{ij}^{t-1}, & \text{if } a^t \neq a_i, \end{cases} \quad (6)$$

$$\Delta Q_{ij}^t = \alpha \cdot [r^t + \gamma \cdot V^{t-1}(\mathbf{y}^t) - Q^{t-1}(\mathbf{x}^t, a^t)] \cdot \frac{\mu_{ij}(\mathbf{x}^t)}{\sum_{j=1}^{N} \mu_{ij}(\mathbf{x}^t)}, \quad j = 1,2,...,N, \quad (7)$$

where α is a learning rate, r^t is a reinforcement signal for the action a^t at time t, γ is a discount rate, and $V^{t-1}(\mathbf{y}^t)$ is defined as

$$V^{t-1}(\mathbf{y}^t) = \max\{Q^{t-1}(\mathbf{y}^t, a_i) \mid i = 1,2,3,4\}. \quad (8)$$

From the update rule of the consequent Q-value of each fuzzy if-then rule in (6)-(8), we can see that multiple fuzzy if-then rules are adjusted by a single action. That is, all the fuzzy if-then rules for the action a^t that have non-negative compatibility grades with the state $\mathbf{x}^t = (x_1^t, x_2^t)$ are adjusted by (6)-(8). This is one of the main characteristic features of the fuzzy Q-learning in comparison with the conventional one where only a single Q-value is updated by a single action. Because multiple fuzzy if-then rules are adjusted by a single action in the fuzzy Q-learning, it requires much less learning iterations than the conventional Q-learning.

Advantages of the fuzzy Q-learning over the conventional one can be summarized as follows:

(1) The size of a fuzzy rule table is much smaller than that of a look-up table.

(2) The learning speed of the fuzzy Q-learning is much faster than the conventional one.

(3) The fuzzy Q-learning has the ability to handle continuous states and/or continuous actions.

4. Computer Simulations

We applied the fuzzy Q-learning to the grid world problem with the following parameter specifications:

Initial value of Q_{ij}: 1.0,
Learning rate: $\alpha = 0.2$,
Discount rate: $\gamma = 0.9$.

We also applied the conventional Q-learning to the same problem with the same parameter specifications. As we have already explained, each trial (*i.e.*, each iteration of learning) was terminated after 100 moves from a randomly initialized position. When the agent reached the goal or the wall, the trial was terminated before 100 moves. Every ten iterations of the learning, we examined the performance of the fuzzy Q-learning as follows by fixing the consequent Q-value of each fuzzy if-then rule:

[Performance Examination Procedure]

Step 1: From all the positions in the grid world, the agent moves according to the greedy policy (*i.e.*, the agent chooses the action with the maximum Q-value at each position).

Step 2: Calculate the successful rate that is the percentage of positions from which the agent could reach the goal in Step 1.

The performance of the conventional Q-learning was also examined in the same manner. Simulation results are summarized in Fig. 4 where the successful rates after every ten iterations of each learning scheme are depicted. From this figure, we can see that the fuzzy Q-learning required much less iterations for obtaining high successful rates. We also measured the CPU time required for 1000 iterations of each learning scheme:

Fuzzy Q-learning: 3.4 seconds,
Conventional Q-learning: 1.8 seconds.

These results shows that the fuzzy Q-learning did not require long computation time while it performed the fuzzy inference for each move of the agent. This is because (i) the number of fuzzy if-then rules were very small (*i.e.*, only eight rules) and (ii) the fuzzy inference did not require long computation time.

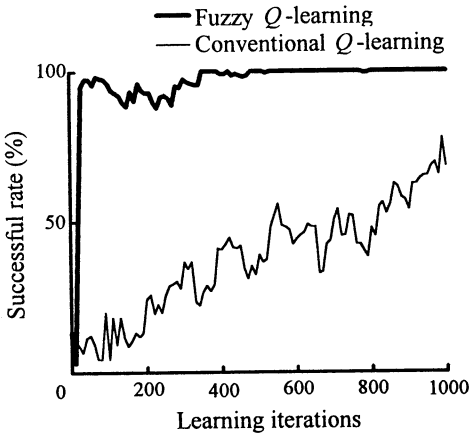


Fig. 4 Simulation results for the learning rate 0.2.

The fuzzy Q-learning has some disadvantages while it seems to clearly outperform the conventional one in Fig. 4.

One disadvantage is the lack of the convergence. When the learning rate was large, we observed the oscillation of the performance of the fuzzy Q-learning during the learning. Simulation results for the learning rate 0.5 are summarized in Fig. 5.

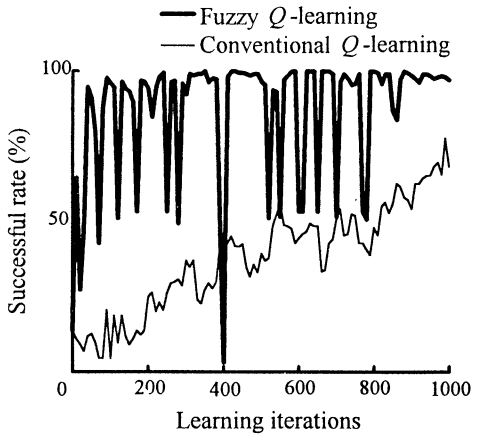


Fig. 5 Simulation results for the learning rate 0.5.

Another disadvantage of the fuzzy Q-learning is the strong dependency of the performance on the choice of fuzzy if-then rules. For example, the performance of the fuzzy Q-learning was drastically deteriorated when we used the nine fuzzy if-then rules corresponding to the fuzzy partition in Fig. 3 (*i.e.*, $R_{i1} \sim R_{i8}$ in Fig. 3 and the additional fuzzy if-then rule corresponding to the center region of the grid world). Simulation results with the learning rate 0.2 are summarized in Fig. 6.

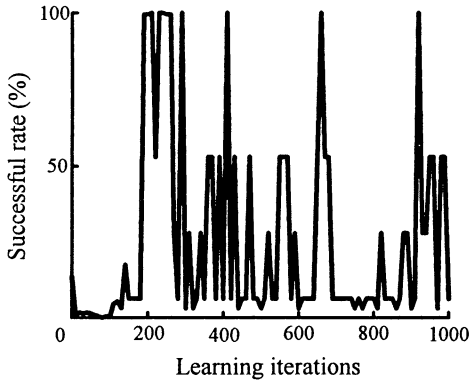


Fig. 6 Simulation results with the nine fuzzy if-then rules.

5. Conclusion

In this paper, we demonstrated the high performance of the fuzzy Q-learning by comparing it with the conventional Q-learning using the simple grid world problem. The high performance of the fuzzy Q-learning was realized by the following characteristic feature: multiple fuzzy if-then rules were adjusted by a single move while only a single Q-value was adjusted in the conventional Q-learning. This characteristic feature also caused the disadvantage of the fuzzy Q-learning: the lack of the convergence. We also demonstrated the deterioration of the performance of the fuzzy Q-learning when the choice of fuzzy if-then rules was inappropriate.

As we demonstrated in Fig. 4, the fuzzy Q-learning clearly outperformed the conventional one when the learning rate and the fuzzy if-then rules were appropriately specified. The fuzzy Q-learning has also other advantages: the applicability to continuous states and/or actions.

References

[1] C.J.C.H. Watkins and P. Dayan, "Q-Learning," *Machine Learning*, Vol.8, pp.279-292, 1992

[2] T. Horiuchi, A. Fujino, O. Katai and T. Sawaragi, "Fuzzy Interporation-Based Q-Learning with Continuous States and Actions," *Proc. of 5th IEEE International Conference on Fuzzy Systems*, pp.594-600, New Orleans, September, 1996.

[3] L. Jouffe and P.-Y. Glorennec, "Comparison between Connectionist and Fuzzy Q-Learning," *Proc. of 4th International Conference on Soft Computing*, pp.557-560, Iizuka, October, 1996.

[4] L. Jouffe, "Fuzzy Inference System Learning by Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics* (submitted).

[5] H. Ishibuchi, H. Miyamoto, C.-H. Oh and T. Nakashima, "The Analysis of Transportation Problems using Fuzzy Q-Leaning," *Proc. of the 6th Intelligent System Symposium*, pp.179-182, 1996 (in Japanese).

[6] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, pp. 404-435, 1990.

Characterization of Behavior of Abstract Rewriting System on Multisets using Normalization Property

Yasuhiro Suzuki and Hiroshi Tanaka
Department of Information Medicine
Tokyo Medical and Dental University
1-5-45 Yushima, Bunkyo-ku Tokyo 113 Japan
Email: {suzuki.com@mri.tmd.ac.jp, tanaka@cim.tmd.ac.jp}

Abstract

We develop an abstract computational model, called ARMS, to deal with systems with large degree of freedom like liquid and confirm that it can simulate the emergence of the complex behavior of cycles such as emerge of autocatalytic cycles, their period-doubling and fusion of cycles often found in the emergence of life. Furthermore, we clarify the conditions in which ARMS generates cycles through the theoretical investigation about ARMS.

1 Introduction

Symbolic computation model which can deal with systems having very large degree of freedom such as liquid has not been proposed so far. However, in the research of complex adaptive systems, specially that about the emergent dynamics of life, the system which we deal with becomes necessarily of large degree of freedom like in the study of Eigen and Shuster [2]. Hence, we have been engaged in the computational study in which an abstract computational model, ARMS [6], using the abstract rewriting system [1] is developed so as to describe the system with large degree of freedom, specially to model the abstract form of dynamic behaviors found in the emergence of life and have confirmed the complex cyclic or hypercylic behaviors such as the emergence of autocatalytic cycle structure or period-doubling of those cycles. In this paper, we further develop this approach and provide the sufficient and necessary conditions in which ARMS generates cycles based on the theory of abstract rewriting system.

2 Model

2.1 Abstract Rewriting System

An abstract rewriting system abstracts the algebraic characteristics of rewriting. By introducing this formal structure, several characteristics of rewriting can be discussed in the common framework. Thus, this concept is applied to various formal methods in mathematics and computer science. We first begin with the definition of the abstract rewriting system.

Definition 1 (Abstract Rewriting System) *An abstract rewriting system(ARS for short) is defined as a pair (A, R), where A denotes a set of elements and R denotes a set of binary relations on A, respectively. Instead of aRb, we write $a \to b$ ($a,b \in A$), and we denote that $a \to b$ is rewriting step.* □

We read the "a $\to$ b" as *a rewrites b* and we call *c* a *reduct* of a.

On ARS the term which can not be reduced is the result of computation. We call them *normal form.*

Definition 2 (Normal Form) *If there does not exist b such as $a \to b$ and $b \in A$, then $a \in A$ is called normal form.* □

We define a property of ARS below.

Definition 3 (Normalization prosperity) *Let (A, R) be an ARS. If there are no infinite rewrite sequences starting at* **a** *then* **a** *is called strongly normalizing. If all its elements in A are strongly normalizing, then we call the ARS is strongly normalizing.* □

If the ARS of the element a is not strongly normalizing, we have the infinite rewrite sequence.

2.2 ARMS

Using the concepts of abstract rewriting system, we introduce a new abstract rewriting system on multisets (ARMS).

A finite set of alphabets A includes a singleton empty set ϕ. A is a total ordered set. Let S be a finite multisets, $|S|$=m is the number of element of S. m called base number.

Definition 4 (term) *The multisets $t \in A^k (1 \leq k \leq n) \in \Sigma$ is a term. Where n is a finite number, A^k is a Cartesian product of $A_1 \ldots A_k$, $A_k = A_1 \times A_2 \ldots \times A_k$. Σ is the set of term and n is maximum term length. A term s is a subterm t, if s is a subset of t. A subterm s of t is proper if $s \neq t$.*

Definition 5 (rewriting rule) *A rewrite rule is an relation l R r(l,r $\in \Sigma$) between terms that satisfy the following two conditions:*

- $|l| = |r|$,
- $|l|, |r| \leq$ *maximum term length n.*

Rewrite rules l R r will be written as $l \rightarrow r$.

Definition 6 (ARMS) *An Abstract Rewriting System on Multisets (ARMS for short) is a pair (T, Ru) consisting of a term T and a set Ru of rewrite rules.*

Definition 7 (Rewriting on ARMS) *Let (T, Ru) is an ARMS. We rewrite $s \rightarrow_{Ru} t$ if there exists a rewrite rule $l \rightarrow r \in Ru$ such that $l \in s$ and $t = s - l \cup r$. The subterm l of s called a redex(an abbreviation of reducible expression).*

ARMS can introduce the inputs into system. We express the inputs as a rule, for example, $\phi \rightarrow a$.

2.3 How ARMS works

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

An Example In this example, we assume that a will be input after each rewriting step. Let R be a set of the following rules:

$$aa \rightarrow \phi e : r_1, \quad b \rightarrow c : r_2, \quad d \rightarrow e : r_3$$

The initial state is set to $\{\phi, \phi, \phi\}$. **State1:**$\{\phi, \phi, \phi\}$, A rule r_1 is applied, but it fails, so only input of symbol "a" will be done in this step. Then, the next state is set to $\{a\}$ after the input. **State2:**$\{\phi, \phi, a\}$, In this state ARMS tries r_2, but it fails, so only input of symbol a will be done again. **State3:**$\{\phi, a, a\}$, ARMS tries r_3, but it fails again, so only a will be input. **State4:**$\{a, a, a\}$, In State 4 r_1 is applied again, and it succeeds. Thus, this state is transformed into: **State4:**$\{b, a, a\}$. It is notable that in State 3, r_1 can be applied, but ARMS do not so, because only one rule is applied in each rewriting step.

Typical Examples In this paragraph, we present an illustrative example which generates cycles. On this typical example we use following rules in descending order:

$$\{a\,a\,a \rightarrow \phi\,\phi\,b : r_1,\ b\,a \rightarrow \phi\,c : r_2,\ \phi\,c \rightarrow d\,d : r_3,\ d\,e \rightarrow \phi\,a : r_4,\ d \rightarrow e : r_5,\ d \rightarrow c : r_6\}.$$

whose state transition is shown below. After 18 steps, the system is stable in forming a cycle, whose period length is 5 steps.

1.	$\{\phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi\}$	
2.	$\{\phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi, a\}$	
↓		16 steps
18.	$\{\phi, \phi, a, a, a, a, a, b, c, e\}$	
19.	$\{\phi, \phi, a, a, a, a, a, c, c, e\}$	
20.	$\{a, a, a, a, a, a, c, d, d, e\}$	
21.	$\{\phi, a, a, a, a, a, a, a, c, d\}$	
22.	$\{a, a, a, a, a, a, a, a, c, e\}$	
23.	$\{\phi, \phi, a, a, a, a, a, b, c, e\}$	
24.	$\{\phi, \phi, a, a, a, a, a, c, c, e\}$	
25.	$\{a, a, a, a, a, a, c, d, d, e\}$	
26.	$\{\phi, a, a, a, a, a, a, a, c, d\}$	
27.	$\{a, a, a, a, a, a, a, a, c, e\}$	
	$\ldots$	

3 Simulation of ARMS

In this model, a multiset is taken as a role of a test tube, which contains "symbols", which correspond to chemical compounds. Then, rewriting rules, corresponding to chemical reaction formulae, act on this multiset, and the applied symbols are transformed into

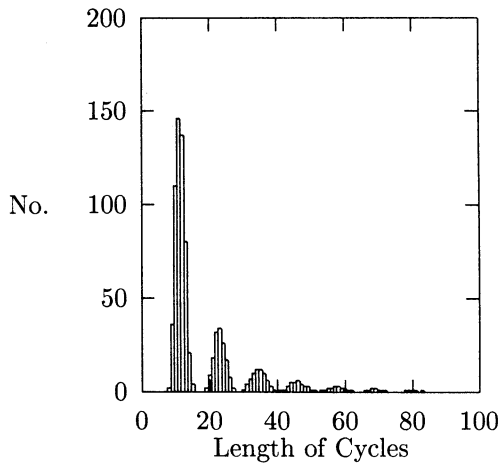


Table 1: An Illustrative Example of the simulations. The system shows bifurcation

other symbols, corresponding to chemical reactions. By use of this model, the condition of a cycle emergence, its robustness and how this system works under a probabilistic condition are examined by using computer simulations.

Computational experiments are made under the following initial conditions:

(1) only five symbols, $\{a, b, c, d, e\}$ are used to describe rewriting rules,

(2) a input symbol is only $\{a\}$,

(3) the cardinality of a multiset is 10,

(4) six rules are used for rewriting steps, and

(5) two important parameters, the frequency of inputs and randomness of rule application are given for each simulation,

where the former four conditions are fixed and two parameters in the last condition are set to variables.

Although those settings are very simple, the experiments give the following two interesting results. The first one shows that a cycle will emerge even under a simple initial condition, compared with Kauffman's network model [5] and Fontana's λ-calculus model [3], which needs large-scale computation to generate cyclic structure from a given system. The second one shows that complex behavior of a cycle, such as fusion of several cycles, is observed simply when input randomness is introduced (table1). The details of this simulation explained in detail in [6].

4 Discussion

Through above simulation we observed various cycles emerge. Interestingly, the empirical result is proved by theoretical studies based on the characteristics of an abstract rewriting system.

If the ARS is not strongly normalizing, we have the infinite rewrite sequence, that is, the ARS generate cycle. Because of ARMS is also ARS, if an ARMS does not hold strong normalization prosperity, the system generate cycle. Therefore, to know the conditions which ARMS generating cycle, we need to know that how conditions make the ARMS be *strongly normalizing.* As its contradiction, we obtain the condition when ARMS generate cycle.

4.1 Condition of Cycle Emergence

Preliminary

Definition 8 (Well-founded order) *Let R be a relation on a set A. If there are no infinite descending sequences $a_1 R a_2 R a_3 \ldots$ of elements of A, we say that R is well-founded.*

For example, if R is a relation $>$ on the set of natural number, then $>$ is well-founded, but on the set of integer, $>$ is not well-founded.

A well-founded order is a powerful method to obtain a proof of termination on ARS. On an ARMS, with lexicographic order as below, we can introduce orders between terms.

Definition 9 (Lexicographic Order) *Let $(A_i, R_i)(i = 1, 2, 3, \ldots)$ be sets equipped with total orders R_i. We define a binary relation R on A,*

$$A = \cup_{m=1}^{n} \{< a_1, a_2, \ldots a_n > | a_i \in A_i (i = 1, 2, \ldots, n)\}$$

(where n is a finite natural number, $< \ldots >$ is a element of Cartesian product of $A_1 \ldots A_n$), as follows: $< a_1, a_2, \ldots a_n > R < b_1, b_2, \ldots b_m >$ if and only if $\exists i$, $1 \leq i \leq m, 1 \leq i \leq n$, $(a_1 = b_1) \wedge \ldots (a_{i-1} = b_{i-1}) \wedge (a_i < b_i)$, or $m \leq n$ and $(a_1 = b_1) \wedge \ldots (a_n = a_m)$. We denote lexicographic order $\succ_{lex}$.

For example, The Relation $>$ such that, a $>$ b $>$ c $\ldots >$ z on A, $\{aaac\} \succ_{lex} \{aaba\}$.

Definition 10 *A binary relation on terms is called a rewrite relation.*

If $\{aaac\} \rightarrow_{Ru} \{aaba\}$, by $ac \rightarrow ab \in Ru$, then the rewrite relation between $\{aaac\} and \{aaab\}$ is $\{aaac\} \succ_{lex} \{aaba\}$.

Definition 11 *A proper order on terms that is also a rewrite relation is called a rewrite order. A reduction order is a well-founded rewrite order.*

There exists a proper order $\succ$ such that $s \succ t(s, t \succ \Sigma)$, and if $\succ$ is well-founded order then $s \succ t$ is a reduction order.

On lexicographic order, we cannot say that if $< a_1, a_2 > \succ_{lex} < b_1, b_2 >$ then $< a_2, a_1 > \succ_{lex} < b_1, b_2 >$ (quasi order), then we introduce sorted.

Definition 12 (Sorted) *On ARMS, if the all alphabets in term t sorted in descending order by arbitrary total order $\succ$ on alphabets, then we say that the term is* **sorted**.

We assume that every terms are sorted below.

4.2 Termination of ARMS

Lemma 1 *On ARMS, a rewrite relation $s \succ t$, if $s \rightarrow_{Ru} t$ with $l \rightarrow r \in Ru$ and $l \succ_{lex} r$*

Proof. Clear, by definition of rewriting. □.

Lemma 2 *An ARMS is terminating if and only if there exists a well founded order R on terms such that $\{\rightarrow_{Ru}\} \subseteq R$.*

Proof. ($\Rightarrow$) Immediate by definition of well founded order. ($\Leftarrow$) If there exits infinite rewrite sequence, it contradicts definition of well-founded order. □

Theorem 1 *An ARMS is terminating if and only if there exist a reduction order $\succ$ on terms such that $l \succ r$ for every rewrite rule $l \rightarrow r \in Ru$.*

Proof.($\Rightarrow$) Clear. ($\Leftarrow$) According to above lemma 1, it suffices to show $\{\rightarrow_R\} \subseteq \succ$. By assumption $l \succ r$ for every rewrite rule $l \rightarrow r \in Ru$. If we take $\succ_{lex}$ on sorted terms instead of $\succ$, then by lemma 2 l $\succ$ r. Hence $\{\rightarrow_R\} \subseteq \succ$. □.

By above theorem, we obtain a following corollary as a condition of the system generates cycles.

Corollary 1 *An ARMS generate cycles if and only if the set of terms Σ is finite set and there does not exist a reduction order $\succ$ on terms such that $l \succ r$ for any rewrite rule $l \rightarrow r \in Ru$.*

Proof. By the contradiction of Theorem 1 (and *Köning's tree lemma* [4]). □

5 Conclusion

In the previous studies [6] we have conducted various simulations about the behaviors of ARMS and confirmed the emergence of complex cycles. In this contribution, we clarify the conditions in which these cycles emerge. Terminating property is found to be essential to cycle emergence, whereas another essential property of ARS, the confluent property of ARS is a principal factor to make cycles more complex.

Definition 13 (Confluent prosperity) *Let (A, R) be an ARS. If for all element b,c $\in$ A with $b \leftarrow \ldots \leftarrow a \rightarrow \ldots \rightarrow c$, and $b \rightarrow \ldots \leftarrow c$, then we say that a is confluent. If all its elements in A are confluent then we call the ARS is confluent.* □

In the future study we will continue to investigate about the confluent property of ARMS to clarify the detail behavior of ARMS.

Acknowledgments

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

References

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

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

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

[4] J. H. Gallier Logic for Computer Science p89, John Wiley & Sons, 1987.

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

[6] Y. Suzuki, S. Tsumoto and H. Tanaka, Symbolic Chemical System based on Abstract Rewriting System on Multisets, Proceedings of Alife V, MIT Press, 1996.

Effect of Complexity on Learning Ability of Recurrent Neural Networks

N. Honma
Col. of Medical Sciences
Tohoku University
Sendai 980-77, JAPAN

K. Kitagawa
Systems Planning Sub Gr.
NTT DATA Corporation
Tokyo 163-02, JAPAN

K. Abe
Faculty of Eng.
Tohoku University
Sendai 980-77, JAPAN

Abstract

The paper demonstrates that recurrent neural networks can be used effectively for estimation problems of unknown complicated nonlinear dynamics. The emphasis of this paper is on distinguished properties of dynamics at the edge of chaos which exists between ordered behavior and chaotic behavior.

We evaluate the advantages of the distinguished dynamics in recurrent neural networks by using new stochastic parameters defined as combinations of the standard parameters. Then a learning method based on the advantage is proposed. The core of this method is to keep the complexity of the network dynamics to the distinguished dynamics phase. In this method the standard parameters of neurons are changed by the core part and also according to the global error measure calculated by the well known simple back propagation algorithm.

Some simulation studies show the excellent learning ability at the edge of chaos and the effect of properties of the advantages on learning.

Key words: *Recurrent neural networks, Chaos, Complex systems, Learning.*

1 Introduction

Recurrent neural networks with dynamics can be useful for identification of nonlinear complicated dynamical systems. Therefore various learning algorithms for the networks have been proposed and discussed. Most of these algorithms[6, 8, 5], however, are based on the back-propagation algorithm for feedforward neural networks. Since the properties of their recurrent network dynamics are not considered in these algorithms, the learning in these networks has many difficulties. For example, these algorithms require numerous memories and are computationally expensive.

Unnikrishnan and Venugopal[7] proposed an another alternative algorithm for both of recurrent and feedforward neural networks, called Alopex. It can train the networks without respect to the network dynamics, but it is still computationally expensive since the changes of parameters such as connection strengths are calculated according to a stochastic error measure.

In this paper, properties of their network dynamics are investigated and relations between the complexity of dynamics and the learning ability are revealed by using new parameters. In general, it is difficult to determine such effectual parameters by strict analysis. We determine them by expanding the results for the discrete recurrent networks reported in our previous paper[1].

Then we propose a new algorithm for recurrent neural networks which can calculate the changes of parameters according to two independent criteria: the goal of learning and an *autonomous criterion* based on properties of the network dynamics. Some examples show that the networks in a distinguished dynamics phase can have excellent learning ability.

2 Recurrent neural networks

To consider the various interactions, we use *fully connected recurrent neural networks.* The networks consist of neurons whose internal states x_i, $i = 1, 2, \cdots, N$, are governed by

$$x_i(t+1) = \frac{1}{1+\exp(\alpha_i s_i(t))}, \quad (1)$$

$$s_i(t) = \sum_{j=1}^{N} w_{ij} y_j(t) + w_{iu} u_i(t) + \theta_i, \quad (2)$$

$$y_i(t) = x_i(t), \quad (3)$$

$t = 1, 2, \cdots$, where s_i, y_i and u_i denote inputs, outputs and external inputs, respectively. N is the number of

neurons. w_{ij} and w_{iu}, α_i and θ_i represent synaptic connection strengths, the gain of sigmoids and thresholds, respectively.

Most of recurrent networks can be expected to have various dynamics from disordered or chaotic behaviors to ordered or simple behaviors by changing these parameters[1]. In the following section, we will introduce integral parameters defined by above networks' configuration parameters to characterize the wide ranges of dynamics regions, and then we prepare the networks belonging to each of dynamics regions by using the introduced parameters, and last we investigate properties of these dynamics regions.

3 Properties of dynamics

For large scale recurrent neural networks composed of a number of neurons, it is difficult to establish a strict formula of the relations among the wide range dynamics regions related to the configuration parameters. Therefore, we tried to analyze it stochastically by modifying the parameters for the discrete recurrent networks such as random Boolean networks[2], cellular automata[4] and Holon networks[1].

In the large networks, output sequences of most of neurons approach to random sequences approximately[3]. For simplicity, suppose that the external inputs can be ignored, namely $w_{iu} = 0$, the distributions of inputs of neurons, G_i, are given as

$$G_i : \begin{cases} \sigma_i^2 &= \frac{1}{12}\sum_j^N w_{ij}^2, \\ m_i &= \frac{1}{2}\sum_j^N w_{ij} + \theta_i, \end{cases} \tag{4}$$

where σ_i^2 and m_i are the variance and the average, respectively.

3.1 Parameters of complexity

The parameters introduced here, $\overline{\sigma^2}$ and r_{in}, are related to the connection strengths among elements of the networks and variety of conveyed signals through the connections as same as for the discrete networks[1]. The average of σ_i^2, $\overline{\sigma^2}$, is defined by

$$\overline{\sigma^2} \doteq \frac{1}{N}\sum_i^N \sigma_i^2. \tag{5}$$

The parameter $\overline{\sigma^2}$ means the macroscopic strength of interaction and nonlinearity of the networks.

We measure the complexity of the network dynamics using mutual information MI_i and Lyapunov exponent λ. The mutual information between $y_i(t)$ and $y_i(t-1)$, and the average, $\overline{MI}$, are given by

$$\begin{aligned} MI_i(y_i(t); y_i(t-1)) &= H(y_i(t)) + H(y_i(t-1)) \\ &\quad -H(y_i(t), y_i(t-1)), \end{aligned} \tag{6}$$

$$\overline{MI} = \frac{1}{N}\sum_{i=1}^N MI_i, \tag{7}$$

where H represents Shannon's entropy. The entropy is calculated by digitizing $y_i(t)$ to 16-discrete values.

Figure 1 shows $\overline{MI}$ and λ as functions of $\overline{\sigma^2}$. In this figure, N is 100 and the θ_i are appropriate values such that m_i are equal to 0. First, note that the measure $\overline{MI}$ is small at large values of $\overline{\sigma^2}$. This means that the future values of the signals y_i are independent of the past values, that is, the network dynamics are chaotic($\lambda > 0$). Also $\overline{MI}$ takes a small value at small values of $\overline{\sigma^2}$. The dynamics, however, are very ordered($\lambda < 0$) because the entropy of the most of neurons is small.

In the area of middle values of $\overline{\sigma^2}$ which exists between above two cases, $\overline{MI}$ is large. We call this distinguished dynamics area the *edge of chaos* since this area corresponds to around $\lambda = 0$(see figure 1).

Secondly, note that the λ increases as $\overline{\sigma^2}$ increases. These results depend on an another parameter, r_{in}, which implies the ratio of inhibitory connection strengths to the total connections. The parameter r_{in} is defined by

$$r_{in} \doteq \frac{\sum_{i,j}^N w_{ij}^-}{\sum_{i,j}^N w_{ij}^+ + \sum_{i,j}^N w_{ij}^-}, \tag{8}$$

where w_{ij}^+ and w_{ij}^- denote absolute values of w_{ij} with the positive sign and the negative sign, respectively.

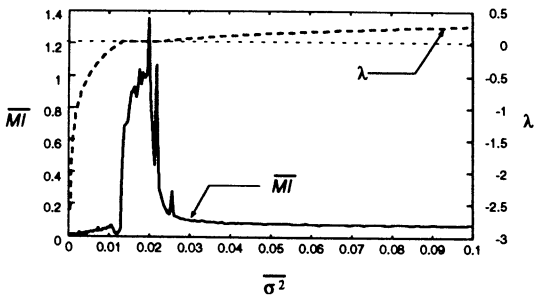


Figure 1: Average single neuron mutual information $\overline{MI}$ and Lyapunov exponent λ over $\overline{\sigma^2}$ space.

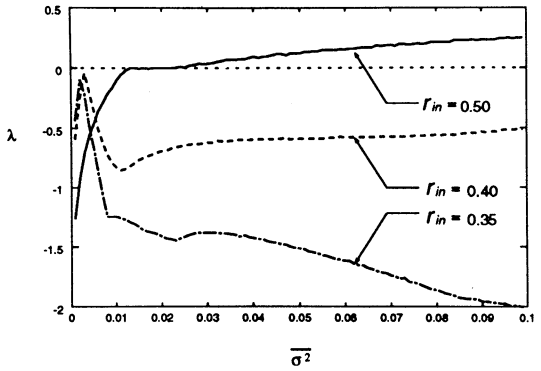


Figure 2: Lyapunov exponent λ over variance $\overline{\sigma^2}$ space for some r_{in} values.

For example, the λ are changed depend on the r_{in} as in figure 2. In the case of $r_{in} = 0.5$, the signals in the networks are most heterogeneous.

4 Learning at the edge of chaos

Here we propose a new learning algorithm for recurrent neural networks. The core of this algorithm is based on the following properties of the dynamics.

4.1 Learning ability at each of dynamics phases

Figure 3 shows the relation between the two measurements of complexity $\overline{MI}$ and λ. We can see that there are three phases of dynamics; the ordered phase where $\lambda < 0$ and $\overline{MI}$ take small values, the chaotic phase where $\lambda > 0$ and $\overline{MI}$ are small values, and the phase of the edge of chaos where $\lambda \simeq 0$ and $\overline{MI}$ take

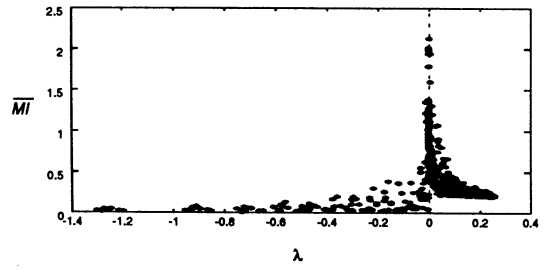


Figure 3: Average single neuron mutual information $\overline{MI}$ over Lyapunov exponent λ.

various values from small to large. The chaotic phase is too complicated to learn the various dynamics. The ordered phase, however, is too simple.

On the other hand, we can expect excellent learning and adaptation ability at the edge of chaos since this phase has various dynamics[3]. This is advantages of the distinguished dynamics phase.

4.2 Emergent learning algorithm(ELA)

Most of the network dynamics are not in the edge of chaos by the well-known simple back-propagation algorithm[8] since the changes of the parameters are calculated according to only the error measure. In our proposed algorithm(ELA), the changes are given by

$$\delta w_{ij} \doteq \begin{cases} \delta^e w_{ij} + c_1 \lambda^3 |\delta^e w_{ij}|, & ({}^i r_{in} \leq 0.5), \\ \delta^e w_{ij} - c_1 \lambda^3 |\delta^e w_{ij}|, & ({}^i r_{in} > 0.5), \end{cases} \quad (9)$$

where c_1 is a constant. $\delta^e w_{ij}$ is the changes calculated by the back-propagation algorithm. ${}^i r_{in}$ denoting the ratio of negative connection strengths are given by

$${}^i r_{in} = \frac{\sum_j^N w_{ij}^-}{\sum_j^N w_{ij}^+ + \sum_j^N w_{ij}^-}. \quad (10)$$

The Lyapunov exponent λ is kept to 0 or approximately by the second term in (9). This term is based on the relation described in section 3.

5 Simulation results

We have tested the learning algorithm for recurrent neural networks on solving temporal XOR problems with various delays[7]. The target output is given by

$$y(t) = u(t-d) \oplus u(t-d-1). \quad (11)$$

The task is to make the network output at time t, the XOR Boolean function of the input at time $t-d$, and the input at time $t-(d+1)$. For this, the network needs to store values from $d+1$ time-steps in the past.

In this simulation, the estimation value $\hat{y}(t)$ are the outputs of feedforward neural networks whose inputs are the outputs of the recurrent networks $y_i(t)$. The goal of learning is to minimize the estimation error $e(\tau)$ given as

$$e(\tau) = \frac{1}{D} \sum_{t=\tau-D+1}^{\tau} |y(t) - \hat{y}(t)|, \quad (12)$$

where $\tau = 1, 2, \cdots$ are the learning steps.

We compared the proposed algorithm with the back-propagation algorithm. In both of two case, the initial dynamics phase was set to the ordered phase. The estimation error e and the Lyapunov exponent λ as functions of learning step τ by the proposed algorithm is shown by the dotted lines in figure 4, and the solid lines show the e and the λ by the back-propagation algorithm. Note that the network dynamics are changed from the ordered phase to the edge of chaos and kept to the phase by the proposed algorithm, and that the error begin to decrease after the phase transition.

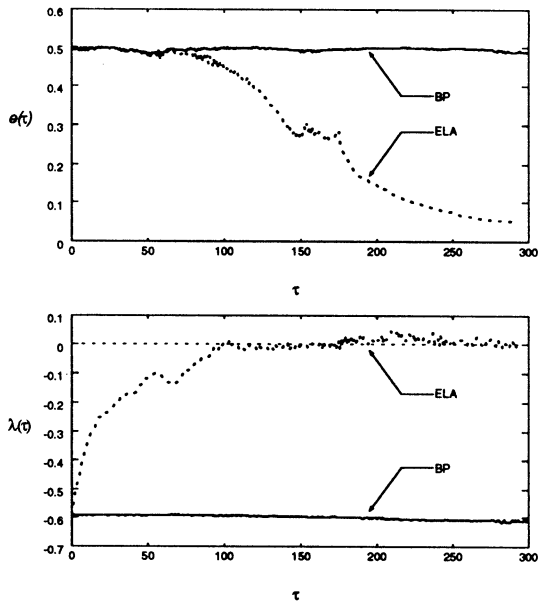


Figure 4: Error e and Lyapunov exponent λ as functions of learning step τ for temporal XOR problem with delay. The case of learning by Emergent Learning Algorithm(ELA) are shown as dotted lines and solid lines show the case of Back-Propagation Algorithm(BP).

As is clear from this result, to keep the network dynamics to the edge of chaos is efficient for learning.

6 Conclusion

It is shown that the dynamics of recurrent neural networks can be controlled by the parameters of interaction among neurons and that three distinctive dynamics phases are revealed: the ordered dynamics phase, the chaotic dynamics phase and the distinguished dynamics phase at the edge of chaos. The simulation, comparing with the back-propagation algorithm, shows that the proposed algorithm based on the advantages of the distinguished phase is effectual one for recurrent neural networks.

When large scale recurrent neural networks are trained by a back-propagation algorithm, the algorithm is computationally expensive: required memories increase exponentially as the number of neurons increases[8]. On the other hand, the proposed algorithm don't need such numerous memories. Further research is to give a mathematical theory of the effect of the dynamical complexity on the recurrent neural networks learning.

References

[1] N. Honma, K. Abe, M. Sato and H. Takeda, "On Emergent Evolution of Holon Networks Using an Autonomous Decentralized Method", Proc. of 13th World Congress of IFAC, vol. I, pp.237-242, 1996.

[2] S. A. Kauffman, "Antichaos and Adaptation", Scientific American, *August*, pp.64-70, 1991.

[3] K. Kitagawa, N. Honma and K. Abe, "An Organization Method by Operating Dynamics for Recurrent Neural Networks", Proc. of 34th SICE Annual Conf., vol. 2, pp.1073-1074, 1995(in Japanese).

[4] C. G. Langton, "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation", Physica D 42, pp.12-37, 1990.

[5] B. A. Pearlmutter, "Learning State Space Trajectories in Recurrent Neural Networks", Neural Computation, vol. 1, No. 2, pp.263-269, 1989.

[6] F.J.Pineda, "Generalization of Back-Propagation to Recurrent Neural Networks", Physical Review Letters, vol. 59, No. 19, pp.2229-2232, 1987.

[7] K.P.Unnikrishnan and K.P.Venugopal, "Alopex: A Correlation-Based Learning Algorithm for Feedforward and Recurrent Neural Networks", Neural Computation, vol. 6, pp.469-490, 1994.

[8] R. J. Williams and D. Zipser, "A learning Algorithm for Continually Running Fully Recurrent Neural Networks", Neural Computation, vol. 1, No. 2, pp.270-280, 1989.

Design of a Complex System based on the Maximum Entropy Principle

K.Tsuchiya K.Tsujita
Dept. of Aeronautics and Astronautics
Kyoto University
Kyoto 606-01

Abstract

A complex system is a system composed of a lot of dynamical elements with mutual interactions. This paper proposes an unified approach of design of an information processing system using a complex system. The method of design is based on the Maximum Entropy Principle. After explaining in details, the proposed method is applied to a design of a spatial filter using a complex system.

1 Introduction

A complex system is a system composed of a lot of dynamical elements with mutual interactions. The complex system can emerge spatial and temporal patterns under a given environment. The emerged pattern has a rigidity, i.e., is structural stable against perturbations of the environment and also has a flexibility, i.e., adapts to changes of the environment. One of the current topics of the system engineering is to realize an information processing system by utilizing spatial and temporal patterns emerged on the complex system[1]. This class of information processing system may work well in a dynamical environment since the spatial and temporal pattern has characteristics of rigidity against and flexibility to the changes of the environment. This paper proposes a design method of an information processing system with a complex system.

The dynamics of a complex system is characterized by two dynamics; One is the dynamics of the elements(element dynamics) and the other is the dynamics of the parameters in the interactions between elements (rule dynamics). On the hand, the objectives of the complex system as an information processing system are specified in terms of the order parameters, the functions of the states of the elements. Then, the design of a complex system is to compose the element dynamics and the rule dynamics so that the order parameters behave in a specified manner. Since the number of the parameters in the interactions is larger than the number of order parameters, the values of the parameters cannot be determined uniquely. Additional conditions are needed. The method of design proposed in this paper employs the Maximum Entropy Principle as a design criterion[2]. In Sec. 2, the proposed method is explained and is applied to a design of a spatial filter in Sec. 3.

2 Method of Design

2.1 Design of the Element Dynamics

The state variable of element i is denoted by u_i $(i = 1, \cdots N)$ and the order parameter k which is a function of the state variables u_i is denoted by $f^{(k)}(u_i)$ $(k = 1, \cdots, M)$. The probability density function of the variables u_i is denoted by $p(u_i)$ and the absolute entropy S of the system is defined by

$$S = -\int_{-\infty}^{\infty} p(u_i) \ln p(u_i) du_i \qquad (2.1)$$

As an information processing system, the behaviors of the order parameters $f^{(k)}(u_i)$ may be specified as

$$\int_{-\infty}^{\infty} p(u_i) f^{(k)}(u_i) du_i = f^{(k)} \quad (k = 1, \cdots, M) \tag{2.2}$$

The probability density function $p(u_i)$ with the constraints (2.2) is determined by the Maximum Entropy Principle as follows:

$$S = -\int_{-\infty}^{\infty} p(u_i) \ln p(u_i) du_i \rightarrow max \tag{2.3}$$

under the conditions

$$\int_{-\infty}^{\infty} p(u_i) f^{(k)}(u_i) du_i = f^{(k)} \quad (k = 1, \cdots, M)$$
$$\int_{-\infty}^{\infty} p(u_i) du_i = 1$$

The solution of Eq. (2.3) is expressed with Lagrange's multiplier λ_k as

$$p(u_i) = Z^{-1} \exp[-\sum_k \lambda_k f^{(k)}(u_i)] \tag{2.4}$$

where, Z is the partition function.

The stochastic process which produces a stationary distribution (2.4) is given by the following stochastic differential equations for u_i

$$\frac{du_i}{dt} = \frac{\partial E(u_i, \lambda_k)}{\partial u_i} + w_i \tag{2.5}$$

where

$$E(u_i, \lambda_k) = -\sum_k \lambda_k f^{(k)}(u_i)$$

w_i is a white Gaussian noise

Equation (2.5) gives the dynamics of the elements, i.e., the element dynamics. The element dynamics (2.5) have Lagrange's multipliers λ_k as parameters. The values of parameters λ_k can be determined by the constraints (2.2) theoretically. But, from a practical view point, it is suitable to determine the values of parameters by a trial and error manner in an actual environment. The tuning algorithm of the parameters are given by the following rule dynamics.

2.2 Design of the Rule Dynamics

In order to evaluate the performance of the complex system as an information processing system, a criterion must be given. Here, as a criterion, a nominal probability density function $\hat{p}(u_i)$ is assumed to be given. By the use of the probability density functions $p(u_i)$ and $\hat{p}(u_i)$, the relative entropy $K(p, \hat{p}, \lambda_k)$ is defined as

$$K(p, \hat{p}, \lambda_k) = \int_{-\infty}^{\infty} \hat{p} \ln(\hat{p}/p) du_i \tag{2.6}$$

The relative entropy $K(p, \hat{p}, \lambda_k)$ is a measure of the distance between the probability density function $p(u_i)$ and $\hat{p}(u_i)$. The parameters λ_k are tuned so as to decrease the relative entropy $K(p, \hat{p}, \lambda_k)$. The dynamics of the parameter λ_k are given as follows:

$$\tau_k \frac{d}{dt} \lambda_k = -\frac{\partial}{\partial \lambda_k} K(p, \hat{p}, \lambda_k) \tag{2.7}$$

where τ_k is a time constant which is introduced so that the dynamics of the parameters are tuned to be slow compared with the dynamics of the elements.

It may be noted that the element dynamics is a Markov Process because the absolute entropy $S(p)$ increases during the process while the rule dynamics is a Bayes Process because the relative entropy $K(p, \hat{p}, \lambda_k)$ decreases during the process; A complex system as an information processing system becomes to be a dual dynamical system where a Markov Process and a Bayes Process interact each other (Fig. 1).

3 Design of a Spatial Filter[3]

Here, a spatial filter is designed by the use of the method proposed in Sec. 2. Consider a complex system composed of elements ij whose state variables u_{ij} $(i, j = 1, \cdots, N)$. Element ij receives an input signal v_{ij} $(i, j = 1, \cdots, N)$ from the environment. The objective of the complex system is to filter the input signal and estimate the original one. The objective is expressed by the following

constraints

$$\int_{-\infty}^{\infty} p(u_{ij}) \sum_{ij} (u_{ij} - v_{ij})^2 du_{ij} = \sigma_1^2$$
$$\int_{-\infty}^{\infty} p(u_{ij}) \sum_{ij} f(t_{ij}, s_{ij}) du_{ij} = \sigma_2^2 \quad (3.1)$$

where $p(u_{ij})$ is a probability density function of variables u_{ij}, $t_{ij} = u_{ij} - u_{i-1,j}$, $s_{ij} = u_{ij} - u_{i,j-1}$

The first equation of Eq. (3.1) represents a priori information about the noise in the input signal v_{ij}. On the other hand, the second equation of Eq. (3.1) represents a priori information about continuity of the environment; In the case where the environment is assumed to be continuous, function $f(t_{ij}, s_{ij})$ is set to be

$$f(t_{ij}, s_{ij}) = t_{ij}^2 + s_{ij}^2 \quad (3.2)$$

While, in the case where the environment is assumed to be discontinuous, function $f(t_{ij}, s_{ij})$ is set to be

$$f(t_{ij}, s_{ij}) = \ln[\frac{1}{\Delta + t_{ij}^2}] + \ln[\frac{1}{\Delta + s_{ij}^2}] \quad (3.3)$$

By the use of the Maximum Entropy Principle, under the constraints (3.1), an unbiased estimate of the probability density function $p(u_{ij})$ is given by

$$p(u_{ij}) = Z^{-1} \exp[-\lambda_1 \sum_{ij} (u_{ij} - v_{ij})^2 + \lambda_2 \sum_{ij} f(t_{ij}, s_{ij})] \quad (3.4)$$

where parameters λ_1 and λ_2 are Lagrange's multipliers, and Z is the partition function. The dynamics of the elements are designed by the use of Eq. (2.5). In the case of Eq. (3.3), the element dynamics are written as

$$\frac{du_{ij}}{dt} = -\lambda_k (u_{ij} - v_{ij}) - [\frac{t_{ij}}{\Delta + t_{ij}} - \frac{t_{i+1,j}}{\Delta + t_{i+1,j}} + \frac{s_{ij}}{\Delta + s_{ij}} - \frac{s_{i,j+1}}{\Delta + s_{i,j+1}}] + w_{ij} \quad (3.5)$$

On the other hand, the dynamics of the parameters are designed by the use of Eq. (2.7). In this case, the dynamics of parameters λ and Δ are given as follows

$$\tau_\lambda \frac{d\lambda}{dt} = \int_{-\infty}^{\infty} \hat{p}(u_{ij})(u_{ij} - v_{ij})^2 \, du_{ij} - \int_{-\infty}^{\infty} p(u_{ij})(u_{ij} - v_{ij})^2 \, du_{ij}$$
$$\tau_\Delta \frac{d\Delta}{dt} = \int_{-\infty}^{\infty} \hat{p}(u_{ij})[\frac{1}{\Delta + t_{ij}^2} + \frac{1}{\Delta + s_{ij}^2}] \, du_{ij} - \int_{-\infty}^{\infty} p(u_{ij})[\frac{1}{\Delta + t_{ij}^2} + \frac{1}{\Delta + s_{ij}^2}] \, du_{ij} \quad (3.6)$$

Finally, after tuning the parameters λ and Δ, the performances of the spatial filter designed are shown in Fig. 2. The filter estimates successfully the original signal with discontinuities.

4 Conclusion

In this paper, a method of design of a complex system has been proposed. The objectives of the complex system are specified in terms of the order parameters. The element dynamics is designed so that the order parameters behave in a specified manner. On the hand, the values of the parameters in the element dynamics are tuned in the environment by a trial and error manner. As a result, the complex system designed is to be a dual dynamical system where the element dynamics which is a Markov Process and the rule dynamics which is a Bayes Process interact each other.

References

[1] H.Haken, "Synergetic Computers and Cognition," Springer-Verlag, 1990

[2] J.N.Kapun and H.K.Kesavan, "Entropy Optimization Principle with Applications," Academic Press, 1992

[3] K.Tsuchiya et.al, "Bayesian Estimation of Random Field with Discontinuities," Trans. of the Society of Instrument and Control Engineers, Vol.28, No.4, pp.514-518, 1992

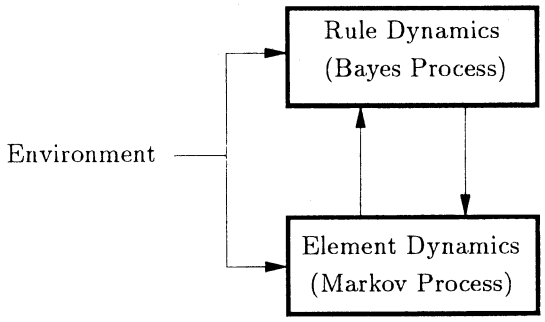


Fig. 1. A Complex System

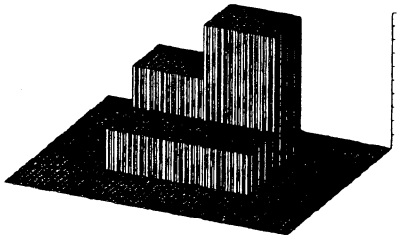


Fig.2 (a)

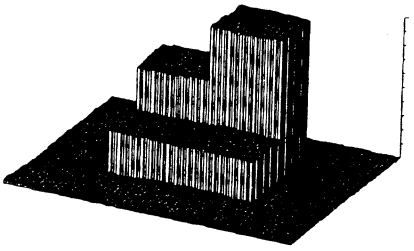


Fig.2 (c)

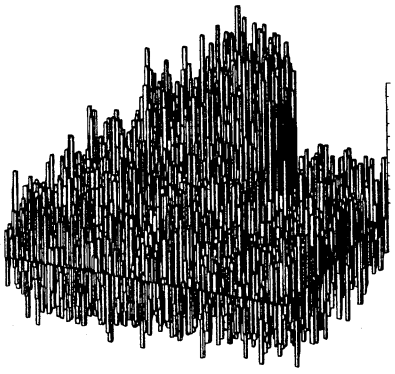


Fig.2 (b)

Fig. 2. Spatial Filter

(a) Original Signal

(b) Input Signal

(c) Estimated Signal

Self-Organization of Complex Adaptive Systems Consisting of Rational Agents

Akira Namatame
Dept. of Computer Science
National Defense Academy
Yokosuka, 239, JAPAN
Tel : + 81-468-41-3810 (ext. 2432)
Fax : +81-468-44-5911
E-mail: nama@cc.nda.ac.jp

Keywords: self-organization, complex systems, rational agent, dynamic game

Abstract

In this paper, we aim at providing a general theoretical framework for designing of complex adaptive systems consisting of rational agent. We term such an agent with a selfish-motivation as a rational agent. They are selfish in the sense that they only do what they want to do and what they think is in their own best interest. We develop the unified model of the mutual interactions among rational agents. We formulate the dynamic competitive and cooperative decision-making problems among rational agents as dynamic games. We obtain and analyze asymptotic equilibrium behaviors of rational agents. This asymptotic equilibrium solution is considered as the coordinated equilibrium behaviors and achieved as the meta-rule through learning among the selfish interests of individuals agents.

1 Introduction

The ability to evolve is the most important property of the living-systems. Evolution and learning are the two most fundamental adaptation process and their relationship is very complex. In the theory of complex adaptive systems, adaptation is viewed as the modification of agents behaviors. However, we need to understand how to get the architecture of a system and an agent, as a component of a system, suited for evolution. The evolution of those large-scale and complex systems found in social, economic, and ecological systems suggests us that we also need to focus on the competition and cooperation among rational agents as basic elements of complex adaptive systems. Such a large-scale adaptive system, a collective designation for nonlinear systems, is defined as the set of a large number of rational agents. In a large-scale complex adaptive system composed of those many rational agents, two types of strategic behaviors may occur: agents mutually interact and behave as a group to optimize the goal of a group, while at the same time, each rational agent also behaves to optimizes its own utility. For an individual rational agent, it behaves to improve its own utility based on its local observation. This ability is based on principles of the individual rationality. We also consider an agent who behaves with the social common goal. By a social goal we mean a goal that is not achievable by any single agent alone but is achievable by a group of agents. The key element that distinguishes a social goal from an agent's individual goal is that they require cooperation. Then how will the evolution of individually rational behaviors proceed? How will the structure of its own goal (utility) functions of a decentralized agent affect as the dynamics of the evolution of their rational behaviors?

We aim at providing a general theoretical framework for analyzing strategic behaviors of as both competitive and cooperative behaviors of rational agents. We especially discuss why and how rational agents form their groups, and how each rational agent learns to behave as both a decentralized agent and a social unit. It is shown the social competence in the form of the cooperative behaviors can be emerged from the interactions of the strategic behaviors. We describe the model of the interactions leading to a coordinated socially optimal behavior. The evolution is guided by both the selfish behavior and the social behavior of each rational agent. The behaviors of complex adaptive systems can be modeled to be emerged from those local and global purposive interactions of rational agents.

We also study the evolution of the organization of a society of rational agents. When thinking about the nature of an agent, we should think about levels of analysis or levels of granularity. We might think about a particular autonomous entity, as an agent, but we might think about an entire organization as an agent. So there needs some aspect of compositionality. That is, we need to be able to put things together and call them an agent and would like to have a recursive or comparable structure that would allow us to build agents at different levels of granularity. The overall system is constructed from many agents. We describe a way of organizing the set of multiple agents, into the structured organization. In a hierarchical organization, an agent could be made of a number of other agents with many levels. Such a hierarchical organization has many roles and defines the behavior of agents in a particular context, how they relate to the environment as well as to each other. The mechanism has a strong similarity to the nature self-organizing and growing process. The growth starts from the set of the unstructured flat organization, and the set of agents is let to self-organize into the whole organization with the nesting hierarchical structure. The growth process is guided by the efficient self-interest seeking behavior of each rational agent, and the overall

organization is constructed from interaction among those agents.

2 Formulation of Complex Adaptive Systems Consisting of Rational Agents

Notions of self-interested behaviors are the foundation of many fields. They are selfish in the sense that they only do what they want to do and what they think is in their own best interests, as determined by their own goals and motivations. Different agents necessarily have different sets of goals, motivations, or cognitive states by virtue of their different histories, the different resources they use, different setting they participate in, and so on. Each agent is able to perform meaningful tasks alone, but which requires the presence of other agents to fully complete its mission. They are selfish in that they only do what they want to do and what they think is in their own best interests, as determined by their motivations and the environment. The global behavior of those agents, is determined by the local interactions of their constituent parts. These interactions merit careful study in order to understand the global behavior of a society. The agents may have different preferences, conflicting objectives or they may have the same goal but wish to decentralize the decision-making process in order to alleviate the heavy burden [8].

We investigate the dynamic adaptive behavior of interdependent agents in a dynamic environment. At each time period, each rational agent faces the problem of choosing its optimal strategy, $u_{i(t)}$, i =1, 2 , ..., n, in order to maximize its objective. The objective of each agent also depends on all other agents' strategies

$$u(i,t) \triangleq (u_1(t),\mathrm{K}\ ,u_{i-1}(t),u_i(t),u_{i+1}(t),\mathrm{K}\ ,u_n(t))\ ,$$

and the current status of the environment. We assume that the cognitive status of each individual agent is aggregately expressed by the state-variable (or cognitive variable) x, which is an element of a compact subset X of R^n. Therefore, each agent's objective is defined by the function

$$g_i\{x_1(t),\ldots,x_n(t),u_i(t)\} = g_i\{x(t),u_i(t)\} \qquad (2.1)$$

The agents may have different preferences, conflicting objectives or they may have the same goal but wish to decentralize the decision-making process in order to alleviate the heavy burden [15]. The utility function of each rational agent depends on both his own decision and all other rational agents' decisions. With the myopic object, each rational agent is modeled to seek a strategy ui(t) of optimizing the object defined over one period of time. As a specific example, we consider the following linear quadratic dynamic competitive problem. The utility function of each agent at the time period t is given as

$$g_i\{x_1(t),\ldots,x_n(t),u_i(t)\} \qquad (2.2)$$
$$= x_i(t)\{a_i - \sum_{j=1}^{n} b_{ij}x_j(t)\} - k_i\{u_i(t)\}^2/2$$

The cognitive state of each rational agent is assumed to be evolving over time, and the dynamics are governed by the equation,

$$x_i(t+1) = f_i\{x_1(t),\ldots,x_n(t),u_i(t+1)\} \qquad (2.3)$$
$$= f_i\{x(t),u_i(t+1)\}$$

This dynamic equation may be interpreted as the dynamic transfer rule of the cognitive state of each rationa lagent. The dynamics are introduced using the state variables representing each individual cognitive states and the dynamic system of equations which governs the change of those cognitive states over time. Putting each agent's cognitive system together, we have the following linear cognitive state equations in a vector form:

$$x(t+1) = Ax(t) + u(t+1) \qquad (2.4)$$

where x(t) and u(t) are column vectors in R^n and A is a nxn matrix.

An agent is said to have a dominant strategy (decision) if it is individually best for its regardless of the decisions chosen by the other agents. If noncooperative behavior is assumed, each agent has a strong incentive to choose his dominant strategy. However, it is rare that such a dominant strategy exists for each agent, and the equilibrium situation is realized by the set of those dominate strategies. We term the set of decisions satisfying the conditions of the individual rationality of all agents as a competitive equilibrium solution (Nash equilibrium) [4][5]. With the long-term object, each rational agent is modeled to seek a sequence of strategies ei = { ui(t)}: t = 0,1,2,...}, which maximizes the accumulated utilities defined over an infinite time horizon defined as

$$J_i(e_1, e_2, \ldots, e_n)$$
$$= \sum_{t=0}^{\infty} \alpha^t g_i(x(t), u_1(t), \ldots, u_i(t), \ldots, u_n(t)) \qquad (2.5)$$

where a (0< a < 1), represents a discount factor.

An equilibrium solution of the above dynamic problem is defined as a set of infinite sequences of feasible strategies with the property that no agent can increase its accumulated objective defined in (2.3) by a unilateral change in a sequence of its own strategies.

Definition 2.1: A set of infinite sequences of feasible strategies (e1*,e2*...,en*) is defined to be an equilibrium solution of a dynamic game if it satisfies

$$J_i(e_1^*, e_2^*, \ldots, e_i^*, \ldots, e_n^*)$$
$$\geq J_i(e_1^*, e_2^*, \ldots, e_i, \ldots, e_n^*) \qquad (2.6)$$

for any sequence of feasible strategies e_i.

A particular important equilibrium solution of a dynamic problem is a steady-state solution.

Definition 2.2: Let $ei^*=\{ui^*(t) : t =0,1,2,..\}$be the sequence of the equilibrium strategies of each agent. If each $ui^*(t)$ converges to some fixed ui^* as t is taken to be infinity, a set of the stationary strategies $\{ui^*\}$, i=1,2,...,n, is defined to be a steady-state equilibrium solution.

Decision making problem in multi-agent is a dynamic process and the actions of the rational agents must be coordinated to achieve globally consistent and good actions. Thus, the coordination among rational agents is always an important consideration in designing a multi-agent model. Team theory or groupware is concerned with

the analysis of a decision problem in which multiple decision-makers attempt cooperatively to attain the same goal [9]. The key element that distinguishes a common goal from an agent's individual goal is that they require cooperation. By a common goal we mean a goal that is not achievable by any single agent alone but is achievable by a group of agents. We define such a common object as the summation of the utilities of the individual agents. Especially the myopic common object is defined as

$$S(x(t),u_1(t),..u_i(t),...,u_n(t)) = \sum_{i=1}^{n} g_i\{x(t),u_1(t),..u_i(t),...,u_n(t)\} \quad (2.7)$$

Similarily the long-term common object is defined as

$$S(e_1,e_2,..e_i,...,e_n) = \sum_{i=1}^{n} J_i\{e_1,e_2,..e_i,...,e_n\} \quad (2.8)$$

3 Analyses of Equilibrium Solutions and Organization Formation in a Society of Rational Agents

A society is defined as one with more than one autonomous selfish agent, in which different agents are responsible for different decisions and make those decisions on the basis of different information, and in which the outcome to the society depends jointly on those selfish behaviors. Rational agents may have different preferences, conflicting objectives or they may have the same goal in a decentralized society. The very basic question is then why and who do selfish agents form organization, and cooperate than to act independently by them selves. They may form organization because of the joint interest for efficient resource acquisition or allocation or they may form organization for possible cost or risk sharing. The selfish agents benefit from the cooperative behaviors where they can improve their own objects. In this section, we are concerned with the problem of the organization formation from self-interested behaviors of individual agents.

In the previous sections, we formulated the interactions among decentralized agents of goal-seeking based on its self-interest. The competitive equilibrium solution in which each agent seeks its myopic object in (2.2) with the dynamic cognitive equation in (2.4) is obtained as

$$(B + B_1 + K(I - A))x = a \quad (3.1)$$

The cooperative solution with the myopic object is obtained as

$$(B + B^T + K(I - A))x = a \quad (3.2)$$

where B is a nxn matrix with the (i,j)-th element is b_{ij}, B_1 is a diagonal matrix with b_{ii} for the i-th diagonal element. B^T is the transpose matrix of B. K is a nxn diagonal matrix with diagonal elements k_i, i=1,2,...n.

The competitive equilibrium solution in which each agent seeks its long-tem object in (2.5) with the dynamic cognitive equation in (2.4) is obtained as

$$(B + B_1 + H(I - A))x = a \quad (3.3)$$

The cooperative solution with the long-term object is obtained as

$$(B + B^T + H(I - A))x = a \quad (3.4)$$

H is a nxn diagonal matrix with a dialogonal matrix with the i-th diagonall element is

$$hi = d_i\{1/a - (a_i)\}. \quad (3.5)$$

As a special homogeneous case, we consider the case in which the interaction matrix B is symmetric with the diagonal elements, i.e., b_{ii}=d, and the off-diagonal elements as, b_{ij}=b, (0<b<d), i,j=1,2,...,n. The column vectors a also have the same elements, i.e., a_i=a, i=1,2,...,n. Then, the competitive solution and the utility rceived at equlibrium of each agent is obtained as

$$x_i^\circ(n) = a^2/(2d + k_\lambda + b(n-1)) \quad (3.6)$$

$$g_i^\circ(n) = (d + k_\lambda - k/2)\{x_i^\circ(n)\}^2 \quad (3.7)$$

where the paprameter κ_λ is given as

Mypoic object:

$$k_\lambda = k\delta \quad (3.8)$$

Long-term object:

$$k_\lambda = k\delta\{1/\alpha - (1-\delta)\} \quad (3.9)$$

Similarily the cooperative solution and the utility rceived at equlibrium of each agent is :

$$x_i^*(n) = a^2/(2d + k_\lambda + 2b(n-1)) \quad (3.10)$$

$$g_i^*(n) = (d + k_\lambda - k/2)\{x_i^*(n)\}^2 + b(n-1)\{x_i^*(n)\}^2 \quad (3.11)$$

As the number of agents increases, the ratio of the cooperative equlibrium solution to competitive equlibrium solution converges to 2, i.e.,

$$x_i^*(n)/x_i^\circ(n) \approx 2 \quad (3.12)$$

As a special case, we consider the static interdependent problem in which each agent's utility function is given as

$$U_i(x_1,...,x_i,...,x_n) = x_i(a_i - \sum_{j=1}^{n} b_{ij}x_j) - c_i x_i \quad (3.13)$$

The social values under both the competitive and cooperative equilibrium solutions are defined as the summations of the utilities of all decentralized agents which are given as:

$$G^\circ(n) \equiv \sum_{i=1}^{n} g_i^\circ(n) = a^2 dn/\{2d + b(n-1)\}^2 \quad (3.14)$$

$$G^*(n) \equiv \sum_{i=1}^{n} g_i^*(n) = a^2 n/4\{d + b(n-1)\}$$

The social value of the competitive solution increases if the number of agents is small, and then it decrease if the number of agents increases, and it converges to zero as the number of the decentralized agents in a society increases. Such a turning point is given at the number of agents given as

$$2 \le n \le (2d/b) - 1 \quad (3.15)$$

On the other hand, the social value of the cooperative solution converges to $(a-c)^2/4b$ and this value represents the cooperative effect.

We now consider the case competitive situations at the several sites. That is, each agent in the same society has the opportunity of competiting each other in several places. Then the question arrises whether they should to cooperate to monopolyse in different site, or they shoul competete at all possible site. If they agree to partispate in diffrent area, and they behave as monoply, the total gain of the socity of n agents is given as

$$n\overline{G} = a^2 n/4d \quad (3.16)$$

If they agree to partispate in every area, and they cooperate in each site, the total gain of the socity of n agents is given as

$$nG^{\bullet}(n) = a^2dn^2/4\{d + b(n-1)\} \quad (3.17)$$

If they agree to partispate in every area, and they compete in each site, the total gain of the socity of n agents is given as

$$nG^{\bullet}(n) = a^2dn^2/\{2d + b(n-1)\}^2 \quad (3.18)$$

Therefore, the ratio of those social values are given as

$$G^{\bullet}(n)/\overline{G} = nd/\{d + b(n-1)\} \quad (3.19)$$

The social value of the cooperative solution converges d/b , which is always greater than 1. On the other hand,

$$G^{\circ}(n)/\overline{G} = n/\{1 + (n-1)b/2d\}^2 \quad (3.20)$$

he social value of the competitive solution is greater than the mutual cooperation for monopoly if the number of agents in a society increases is given at the number of agents given as

$$2 \le n \le (2d/b) - 1 \quad (3.21)$$

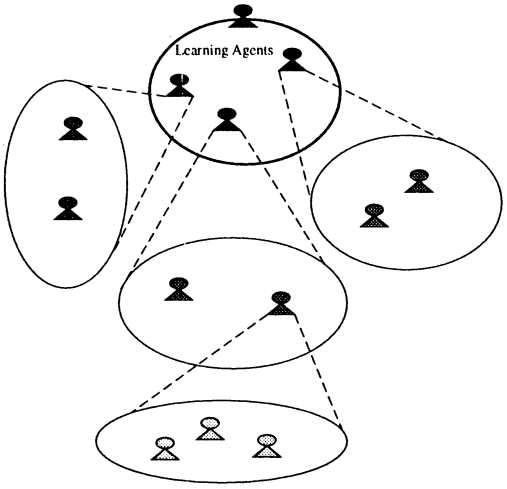


Fig.3 Self-organization of a society of composite agents

In the next, we consider the problem of the organization formation. They may form organization because of the joint interest for efficient resource acquisition or allocation. We consider the two types of organization, namely flat organization and hierarchical organization. With the flat organization with kxn agents, the total utilities at competitive equilibrium of the agents in the whole organization is given as

$$G_1^0 = G^0(kn) = a^2dkn/(2d + b(kn-1))^2 \quad (3.22)$$

On the other hand, with the hierachical organization with k agents in each n group, the total utilities at competitive equilibrium of the agents in the whole organization is given as

$$G_2^0 = kG^0(n) = a^2dkn/(2d + b(n-1))^2 \quad (3.23)$$

Therefore their ratio is given as

$$G_2^0/G_1^0 = \left(\frac{2d + b(nk-1)}{2d + b(n-1)}\right)^2 \quad (3.24)$$

$$\approx O(k^2)$$

Each agent receives higher utility by forming hierachical organization. They may form hierachical organization because of the joint interest for efficient resource allocation and the selfish agents benefit from the hierachical organization with the nesting structure where they can improve their own objects.

4. Conclusion

This paper described a research program for studying social interactions leading to group behavior. The goal of the research is to understand the types of simples local interactions which produce complex and purposive group behaviors. We formulated and analyzed the adaptive behavior of independent agents in a competitive environment. Each agents does not need to express his objective or cognitive rule explicitly, nor to have a priori knowledge of the abilities of other agents. Each agent adapts his action both to the actions of other agents, and thus allowing previously unknown agents to be easily brought together to customize a team responsible for a specific mission. Agents emerge to cooperate toward attaining some coordinated actions. We showed that the implicit objects and cognitive rule of autonomous agents that lead to coordinated social behaviors can be identified through complex and purposive local interactions. It was also shown that those implicit object and cognitive rule of each social agent can be learned by the recurrent neural network model using the set of the strategies observed in the mutual interaction process as the training examples. The unified theory of the cognitive states, the acquaintance and mutual interaction using strategies is used to develop a theory of social cooperation for multi-agent systems, and to account of how the concepts of social role and social structure within multi-agent systems are formulated.

References

[1] Barto,A.G,Sutton," Learning and sequential decision-making", Learning and Computational Neuro Science, MIT Press,1991

[2] Basar,T, & Olsder,G., Dynamic Noncooperative Game Theory, Academic Press ,1982

[3] Carley,K, & Prietula,M.: Computational Organization Theory, Lawrence Erallbawn, 1994

[4] Fudenberg,D,E,and Tirole.J, Game Theory, MIT Press ,1991.

[5] Gasse,L.,Social conceptions of knowledge and action:DAI foundations and open systems semantics, in Artificial Intelligence, Vol.47, pp.107-135,1991.

[6] Marschak,J,& Radner,R,: Decentralized Theory of Teams, Yale Univ. Press,1972.

[7] Shoham,Y: Agent-oriented Programming, Artificial Intelligence, Vol.60, pp51-92,1993,

[8] Tenney. R.,& Sandell,N.R., Strategies for Distributed Decision making, IEEE Trans. Automatic Control, Vol.AC-19,,pp.236-247,1974.

Self-Organization based on Non-Linear Non-Equilibrium Dynamics of Autonomous Agents

Satoshi KURIHARA
kurihara@square.brl.ntt.co.jp

Rikio ONAI
onai@square.brl.ntt.co.jp

NTT Basic Research Laboratories

Abstract

This paper discusses a study on the mechanism for self-organization. A global order is organized by the simple and locally coordinated actions of autonomous agents using only local information. In other words, the autonomous agents coordinate without complex or globally coordinated actions which use global communication and high level strategies.

The fundamental factors for establishing the global order using self-organization are a "dissipative structure", an "autocatalysis mechanism", and "intentional fluctuations". If an environment where agents exist has a dissipative structure and those agents have some sort of autocatalysis and intentional fluctuation mechanisms within themselves, it is possible to form a global order for agents using their simple and locally coordinated actions.

"The blind-hunger dilemma" is used as an example to simulate the self-organization and coordinated actions of agents and to show the validity of our approach.

Keywords: Self-organization; Dissipative structure; Autocatalysis mechanism; Intentional fluctuation; Autonomous agent; Coordination of agent.

1 Introduction

In our self-organization mechanism, non-linear dynamics are applied to the agents. Instead of using reductionism (where the whole is understood by splitting it up into many parts and understanding each part), constructivism, which is a bottom-up approach, is adopted. Since it is difficult for us to understand natural phenomena (so-called complex systems) through reductionism, constructivism seems to be very promising. In reductionism, it may be possible to separate a whole into parts and understand each of these, however one of the most important aspects of information, i.e. the mutual interaction among the parts, is lost by separation.

In our self-organization mechanism, the mutual interaction of agents is random at first, but becomes increasingly orderly. The advantage of this approach is that the system can interact with the environment flexibly and robustly by using features of non-linear non-equilibrium dynamics.

There have been many proposals on the coordination of agents in multi-agent systems [10], and usually agents coordinate by using complex and globally-coordinated actions which use global communication and high level strategies. However, we believe that self-organization by simple and local coordination is a very important factor in establishing a global order.

Simulations are used, using "the blind-hunger dilemma" as an example [1], to demonstrate the validity of our approach. In the blind-hunger dilemma, agents, which are modelled as ants, try to efficiently get energy from one small energy supply base. Each ant can only obtain quite local information, not global information.

From the results of simulation, it was confirmed that if an environment comprised of ants had a dissipative structure and each ant had autocatalysis and intentional fluctuation mechanisms, then the ants could efficiently get energy by organizing a global order, even

if the ants only had very simple mechanisms.

For example, we can apply our self-organization mechanism to the following situations: (1) to autonomous robots working in complex and dynamic environments. In such environments, robots cannot communicate globally, and moreover, the quantity of information they can exchange is quite limited; (2) to control software agents (they are intelligent interface system between the Internet and users) when quite a lot of agents try to access one information resource.

Section 2 describes our self-organization mechanism and Section 3 verifies our approach on the basis of simulation. The results are in Section 4. The conclusion and future work are in Section 5.

2 Self-Organization Mechanism

The word "self-organization" can be used for several different levels, but in this paper, it is used for a low level, where global order is established from a situation where there is no structure at the beginning. Some examples are the phenomena of life and thermal convection.

To organize a global order, a dissipative structure [4] and the autocatalysis mechanism are especially important. A dissipative structure is the order and structure which are generated and maintained in a non-linear non-equilibrium state where there is a flow of energy. However self-organization can occur more efficiently if each agent has a autocatalysis mechanism. This mechanism is a process where once change starts, the change becomes accelerated. In chemical reactions, an autocatalysis mechanism is necessary for a cycle of reactions to occur. A system made as a whole, has a dissipative structure, and by incorporating an autocatalysis mechanism into each agent, self-organization can occur efficiently. The resulting order and structure would have characteristics such as *stability against disturbances, independence on the initial state, synchronism by coordination*, and *divergency* [5]. The theory of life and evolution can also be explained by this mechanism.

In related research [2] [3], the sorting of broods in ant colonies inspired the proposal for a distributed sorting algorithm for use by teams of robots. The robots move randomly, do not communicate, have no hierarchical organization, but can distinguish between objects of two types with a certain degree of error. The dynamics of this type of robot are the same as in our approach.

3 Verification of Our Approach

3.1 The Blind-Hunger Dilemma

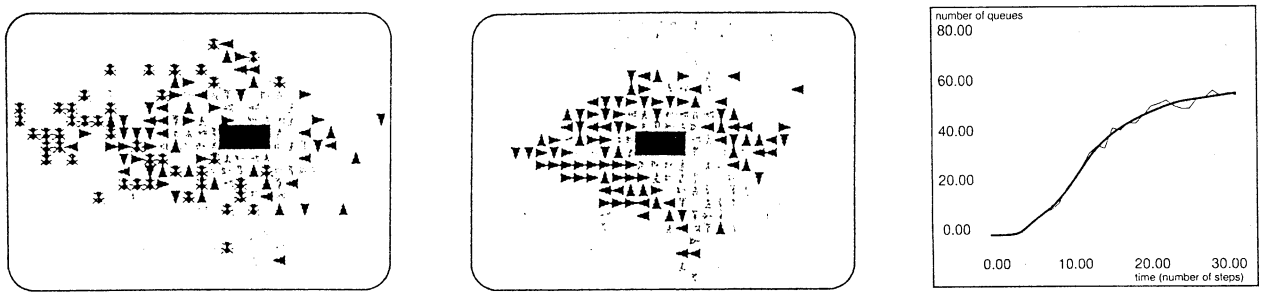
In the blind-hunger dilemma, there are many agents which must get energy. However, since there is only one small energy supply base, an efficient method, or the coordinated actions of agents, is needed [1].

In conventional methods such as global coordinated planning, each agent is given several high level strategies from which to choose the most suitable one through the exchange of information with other agents. The search for the best strategy may use a stochastic model [6] [9], or there may be several agents each of which has a certain strategy [7].

In our approach, an agent does not have high level strategies. Each agent has "conditioned response mechanisms" (afterwards called "habits") which represent the types of behavior it may take. The habit itself will not directly attain the goal of the system. Rather, the goal can only be attained by achieving a global order through the self-organization of agents.

3.2 Ant World

All ants move within a certain area (ant world), and for each time unit, each ant can move either up, down, left, or right for one step (see Figure 1). Each ant has its own energy, which decreases as time goes by. If an ant's energy goes below a threshold, it goes from "normal-mode" to "hungry-mode." It must then go to the energy supply base to get new energy. Each ant can perceive only the direction to the energy supply base. It cannot perceive the exact location. It can only see its neighbors and only know the direction the neighbors are facing and how hungry they are. Each ant wants to reach the supply base quickly, but it would not be appropriate for each ant to act completely on its own because all ants need to get new energy periodically.



(black: hungry-mode ant. dark gray: normal-mode ant. X: dead ant.) If ants do not take any coordinated action, no order is organized (left figure, TYPE I) and most ants exhaust their energy and die. The center figure (TYPE II, III) shows our approach. An order is organized among ants. Hungry-mode ants surround the supply base except for the section to the right. This section becomes an "escape route" for nomral-mode ants. The right figure shows autocatalytic phenomena in the formation of orders.

Figure 1: Simulation of ant world

To comfirm the effectiveness of our approach, we prepared four kinds of TYPEs which ants depend on to move:

1. TYPE I: The ant does not take coordinated action. An ant in the normal-mode moves randomly. However one in the hungry-mode tries to move directly to the energy supply base. If it cannot go where it wants to, it moves in a random direction.

2. TYPE II: Ants in hungry-mode take coordinated action, and ants in normal-mode move randomly. If a hungry-mode ant-A wants to move where there is already an ant-B, and if ant-B is also in hungry-mode and closer to the energy supply base than ant-A, then ant-A starts following ant-B.

3. TYPE III: Ants in both modes take coordinated action. The action for hungry-mode ants is the same as in TYPE II. Regarding normal-mode ants, if there are ant-Xs around the normal-mode ant, and they are in the hungry-mode, then the normal-mode ant moves toward a direction away from the one ant-Xs are facing.

4. TYPE IV: Ants in hungry-mode take coordinated action as in TYPE II and TYPE III, but ants in normal-mode do not take locally coordinated action. They try to move to a place which is a certain distance away from the energy supply base and around which there are few hungry-mode ants. This means that only this type of ants can perceive the location of the supply base and all other ants. Consequently, TYPE IV does not satisfy our criterion of ants only using local information. We have used TYPE IV to compare our approach with the conventional global and coordinated communication approach.

A dissipative structure is realized by the flow of energy from the energy supply base to ants. The continuous flow of energy is established by the unlimited supply of energy available from the supply base, and the energy consumed by each ant. A non-equilibrium state is formed by the different amounts of energy that each ant has and their distribution.

The autocatalysis mechanism is realized in TYPE II, III, and IV. We think that the action of following another ant will form a queue in a autocatalysic way (mentioned in the next section).

We will discuss intentional fluctuations in Section 4.3.

4 Simulation Results

Simulations were done for each type. Some states of ants are shown in Figure 1. Figure 2 (upper left and upper right) shows the number of normal-mode agents and dead agents. Figure 2 (lower left) shows the total amount of energy that all agents possess.

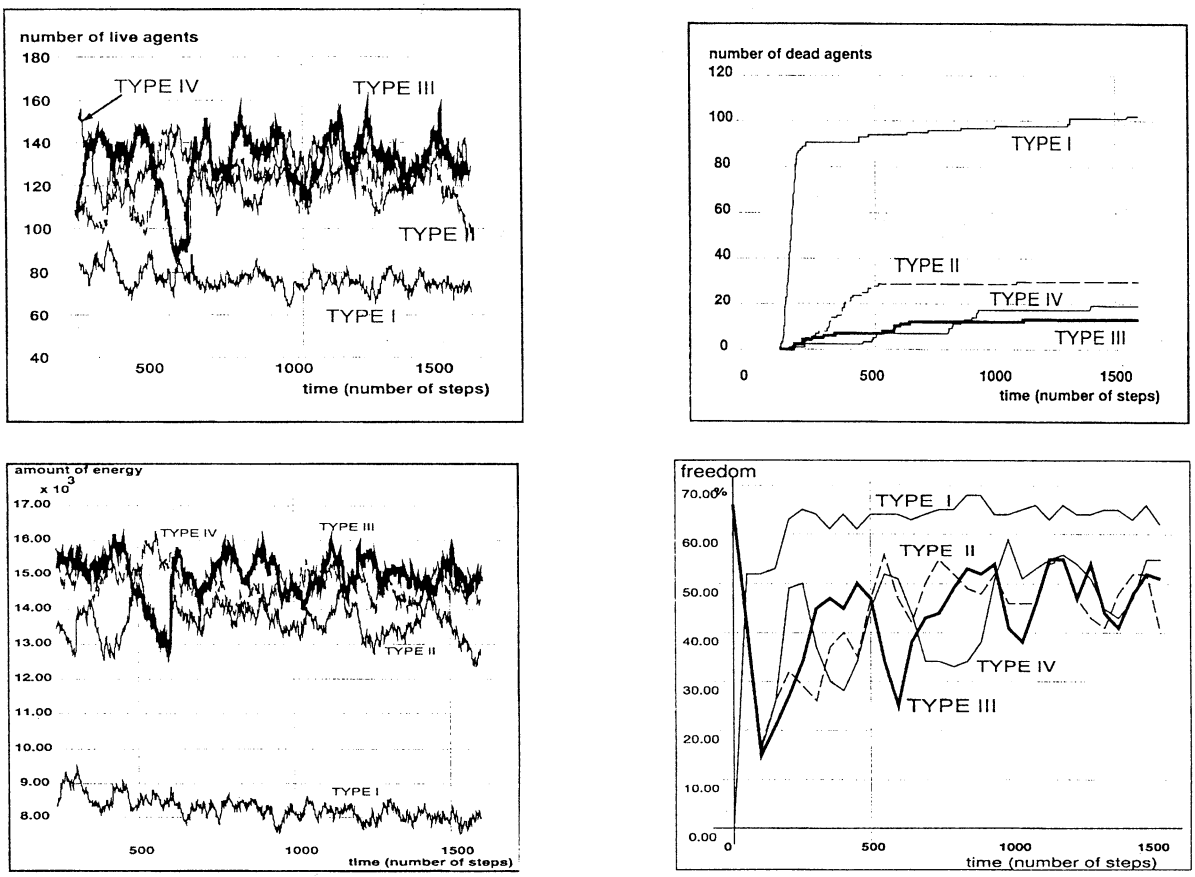


Figure 2: Results of simulation

In TYPEs II, III, and IV the hungry-mode agents were able to get a higher average amount of energy and to achieve a more drastic decrease in the number of dead agents than TYPE I, with TYPE III having the smallest.

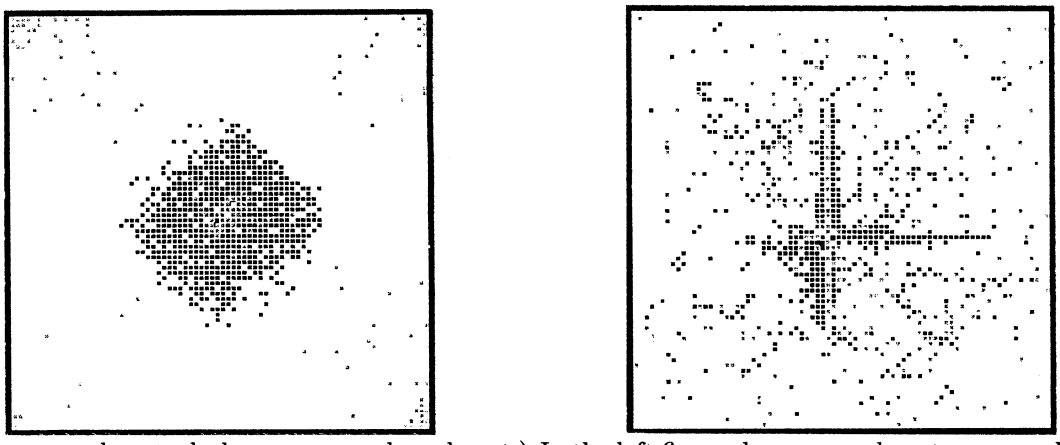
4.1 Autocatalysis Mechanism

From the results of simulation, we were able to confirm that autocatalysis existed in the formation of queues. As the number of queues increased, and lengthened, the number of agents in queues also increased. The queues formed with respect to time are shown in Figure 1 (right). Queues formed very rapidly at first, but after a while, they did not form as rapidly, i.e. they became saturated. By forming queues, an order was established allowing normal-mode agents to get away or "escape" from the energy supply base, and by having the agent at the top of the queue finish getting new energy, the other agents in the queue also had a very good chance of getting new energy.

The TYPE III results were the best followed by TYPE II and TYPE I. TYPE II did not perform as well as TYPE III because it do not use coordinated action when trying to get away from the energy supply base.

4.2 Dissipative Structure

Figure 2 (lower right) shows the change in the average percentage of freedom that an agent has. If an agent can move either up, down, left, or right, then the percentage is 100%, but if an agent can only move in two directions (due to other agents being next to it, or due to the direction of the energy supply base), then the percentage is 50%. When order is formed, the agent's movements become restricted, hence the percentage of freedom goes down. When order starts to disintegrate due to some outside factors, then the percentage goes up. The cycle of formation, destruction, and re-formation of an order can be seen from the percentage going up and going down in Figure 2



(black: hungry-mode ant. dark gray: normal-mode ant.) In the left figure, hungry-mode ants surround the energy supply base entirely and normal-mode ants cannot escape from the supply base. Consequently, most ants lose their energy and die. However in the right figure, many ants are located far from the energy supply base and the "escape route" for nomral-mode ants can be organized by a small number of ants.

Figure 3: Verification of intentional fluctuations

(lower right). In other words, a certain regularity is formed due to the dissipative structure of energy. This regularity stabilizes the organization of order, and as a result holds down the number of dead agents.

TYPE IV which is based on the conventional global and coordinated communication approach could also achieve higher energy and a more drastic decrease in the number of dead agents than TYPE I. At the beginning, we surmised that TYPE IV would get the best score. However, TYPE III achieved the best score by a narrow margin. We can explain the causes by using Figure 2 (lower right). In Figure 2 (lower right), the cycle of freedom was not formed as clearly in TYPE IV as in TYPE III. This means that efficient self-organization was not formed in TYPE IV because normal-mode agents did not take coordinated action when trying to get away from the energy supply base. Moreover, TYPE IV had the problem that if there were an increasing number of ants in the ants world, communication costs would increase exponentialy and performance would rapidly decrease.

4.3 Intensional Fluctuation

We could verify the basic validity of our self-organization mechanism by simulation. However, with an increasing number of ants, the energy supply base was completely surrounded, so even if ants used our approach they could not get energy as efficiently as TYPE I. At this point, we added a new factor to the dissipative structure and the autocatalysis mechanism. The new factor was the "intensional fluctuation mechanism." We also added a new habit TYPE V, which was similar to TYPE III but the following habit was added: when hungry-mode ant-Xs which could not move anywhere for a fixed time they behaved like normal-mode ants for another fixed time. Through this habit, an organization was created whereby the number of hungry-mode ants which left the energy supply base increased autocatalyticly and the surrounding situation was avoided. As a result, the self-organization of TYPE III mentioned in Section 4 was re-organized. The intentional suicide of hungry-mode ants (they behaved normally even though they were in hungry-mode) formed a fluctuation and made efficient energy supply possible (see Figure 3).

5 Conclusion and Future Work

This paper discussed a mechanism for self-organization, which only used quite local information, and not global communication or high level strategies. A "dissipative structure", an "autocatalysis mechanism", and "intentional fluctuations" were considered as the most important factors for self-organization. Furthermore, it was necessary to have decentralization and autonomy. A global order was organized from the locally-

coordinated actions of agents by satisfying these factors. Once a global order was formed, each agent conformed to it. However, since the environment fluctuates, the original order cannot be maintained. In this case, a new order would have to be established to adapt to the dynamic environment naturally.

An important point is that each agent does not directly work for a global goal. The global order is organized as a result of each independent agent moving with only local information. In the simulations, an escape route was not established due to hungry-mode agents wanting to make one. This was established as a result of the coordinated actions of each agent. It would be impossible to control each agent to attain a global goal with a large number of agents and a quite complex environment, i.e. the traditional coordinated planning approach would not be suitable. Consequently, we think that our approach would be reasonable in such cases.

This problem can be applied to cases where many robots must work together and where global communication is impossible due to a dynamic and complex environment. In these environments, there would be less risk in using many small, robust coordinated robots than in one large high-level one.

In future work, it is necessary for agents to have a mechanism which can create new habits to adapt to a new environment. In other words, a mechanism for learning and evolution should be considered. Also, agents should have some sort of genetic code so that metabolisms can be simulated.

Finally, our approach can be used for conventional methods in multi-agent systems such as planning, inference, and so on, and applied to research on autonomous robots, where flexible and robust interaction with a dymanic environment is necessary. It can also be applied in human-machine interaction, where flexible interaction with humans, which can be thought of as a typical complex system, is desired [8].

We are currently continuing real-world simulation by using autonomous robots (Khepera robot and B14) to confirm the effectiveness of our approach.

Acknowledgements

We would like to thank to our manager, Dr. Kenichiro Ishii and the researchers of the Semantic Computing Research Group.

References

[1] C. Numaoka: Introducing the Blind Hunger Dilemma: Agent Properties and Performance, *Proceedings of ICMAS'95*, pp. 290 – 296, 1995.

[2] J. L. Deneubourg: The Dynamics of Collective Sorting Robot-like Ants and Ant-like Robots, *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animals*, pp. 356 – 363, 1991.

[3] S. Goss and J. L. Deneubourg: Harvesting By A Group Of Robots, *Proceedings of the First European Conference on Artificial Life: Toward a Practice of Autonomous Systems*, 1992.

[4] G. Nicolis and I. Prigogine: Self-organization in Nonequilibrium Systems, Wiley & Sons, 1977.

[5] H. Haken: Instability of Self-Organizing System and Devices, Springer-Verlag GmH & Co. KG, 1983.

[6] M. Kubo, and Y. Kakazu: Evaluation of Coordinated Motions of Multi-Agent Systems on Competition for Food between Ant Colonies, *Journal of Information Processing Society of Japan*, Vol. 35, No. 8, pp. 1555 – 1566, 1994 (in Japanese).

[7] M. P. Georgeff and A. L. Lansky: Reactive Reasoning and Planning, *Proceedings of AAAI*, pp. 677–682, 1987.

[8] S. Kurihara, and M. Okada, R. Nakatsu: A Coordinative Structure in Man-Machine Interactions," *Proceedings of HCI'95*, 1995.

[9] S. Yamada: Controlling Deliberation with Success Probability in a Dynamic Environment, *Proceedings of AIPS'96*, pp. 251–258, 1996.

[10] V. Lesser (general chair), *Proceedings of ICMAS'95*, 1995.

Dual Dynamics: Designing Behavior Systems for Autonomous Robots

Herbert Jaeger
GMD, FIT.KI
D-53754 Sankt Augustin

Thomas Christaller
GMD, FIT.KI
D-53754 Sankt Augustin

Abstract

This paper describes the "dual dynamics" (DD) formal scheme for robotic behavior control systems. Behaviors are designed as dynamical systems which are specified in ordinary differential equations. A key idea for the DD scheme is that a robotic agent can work in different "modes", which lead to qualitatively different behavioral patterns. Intuitively, modes can be likened to "moods". Mathematically, transitions between modes are bifurcations in the control system.

Key words: behavior-based robotics, control, dynamical systems.

1 Introduction

The "behavior based" approach to designing mobile robots [3] [8] has been very fertile. However, the field suffers somewhat from a certain lack of formal theory, which in turn hampers the understanding and the design of robots with increasingly complex behavioral repertoires. This paper introduces a formal model of complete behavior control architectures for mobile robots, the "dual dynamics" (DD) scheme.

Behaviors are construed as dynamical systems, which interact through shared variables. There is no global control; the overall functionality of the system arises from the interactions of the individual behavior subsystems.

The basic assumption on which DD rests is that a situated agent can work in different "moods". The notion of "moods" is related to *behavior systems* in ethology [1], i.e. patterns of activities that can be identified on statistical and functional grounds. In animals, "moods" may be linked to relatively slowly changing somatic (hormonal, neurological) conditions. "Moods" allow the agent to deeply tune in to situations, by establishing expectations, facilitating particular motor responses and inhibiting others, etc.

While "moods", with all their phenomenal richness, will hardly lend themselves to a formal definition, the DD approach captures one essential aspect of them in a mathematically precise way. Namely, different "moods" are taken to represent *qualitatively different* "regimes" of the behavior control system, i.e. regions in some space of control parameters where a dynamical system exhibits unique, stable qualitative properties. The theory of dynamical systems actually has no commonly used term for such regions, since mathematicians are more interested in the transitions (bifurcations) between these regions than in the regions themselves. In this article, we call such regions in control parameter space, "modes".

DD helps to advance behavior-based robotics in two ways: (i) theoretically, by offering a rigorous model of behavior control systems in terms of dynamical systems and bifurcations thereof, and (ii) practically, in that the differential equations of DD models can be run directly as actual control programs on robots. In that case, the mathematical transparency of a DD model greatly aids in the design-test-redesign development cycle.

The DD approach has been partially inspired by ethological (e.g. [1]) and biocybernetical (e.g. [4]) models of hierarchical behavior control systems. Methodologically, it is related to current efforts to exploit dynamical systems theory for behavior-based robotics (e.g. [2] [7]).

The main part of this paper presents a concise description of the DD model (section 2). A brief report of a DD-based robot is given in section 3. Section 4 concludes with a discussion.

2 The DD scheme

In this section, we present first an informal sketch of the DD scheme, then provide the mathematical specification.

2.1 Informal sketch

The basic building blocks of a DD robot architecture are *behaviors*. They are ordered in levels (fig. 1a). At the bottom level, one finds *elementary* behaviors: sensomotoric coordinations with direct access to external sensor data and actuators. Typical examples are `move_forward` and `turn_left`. At higher levels, there are increasingly *complex* behaviors, which also have access to sensoric information but cannot directly activate actuators: their effect is to regulate modes. Typical examples of complex behaviors are `work` and `replenish_energy`.

Two aspects of an elementary behavior's performance are cleanly kept apart: first, *what* the behavior does while it is active, and second, *when* (and how strongly) it becomes active. The first aspect is referred to as *target dynamics*, the second, as *activation dynamics*. Both aspects are dealt with in separate dynamical systems – hence the catchword, "dual dynamics". The target dynamics yields target trajectories for all actuators that are relevant for the behavior. The activation dynamics, by contrast, maintains a single activation variable. Essentially, the activation parameter of an elementary behavior becomes effective by disinhibiting the effect of the target dynamics on the effectors (cf. fig1b).

Unlike in many robotic control schemes, an elementary behavior is not "called to execute" from top-down. The level of elementary behaviors is fully operative on its own and would continue to work even if the higher levels were cut off. The effect of higher levels is not to "select actions", but to change the overall dynamics of the entire elementary level, by inducing bifurcations in that level.

For instance, assume that the robot is in a `replenish_energy` mode, i.e., the high-level behavior `replenish_energy` has a high activation, and other high-level behaviors like `work` have a low activation. In this condition, an elementary behavior `turn_left` might activate itself when a charging station is perceived at the left. When the mode changes to `work` (i.e., the activation of `work` dominates other high-level activations), the activation dynamics of `turn_left` would undergo a bifurcation such that the behavior would become active when an opportunity to work is perceived at the left.

Only the activation dynamics subsystem undergoes bifurcations in a properly designed DD scheme. The fact that bifurcations (which are inherently difficult to master from a designer's perspective) are confined to these one-dimensional subsystems contributes greatly to the transparency of DD behavior control systems.

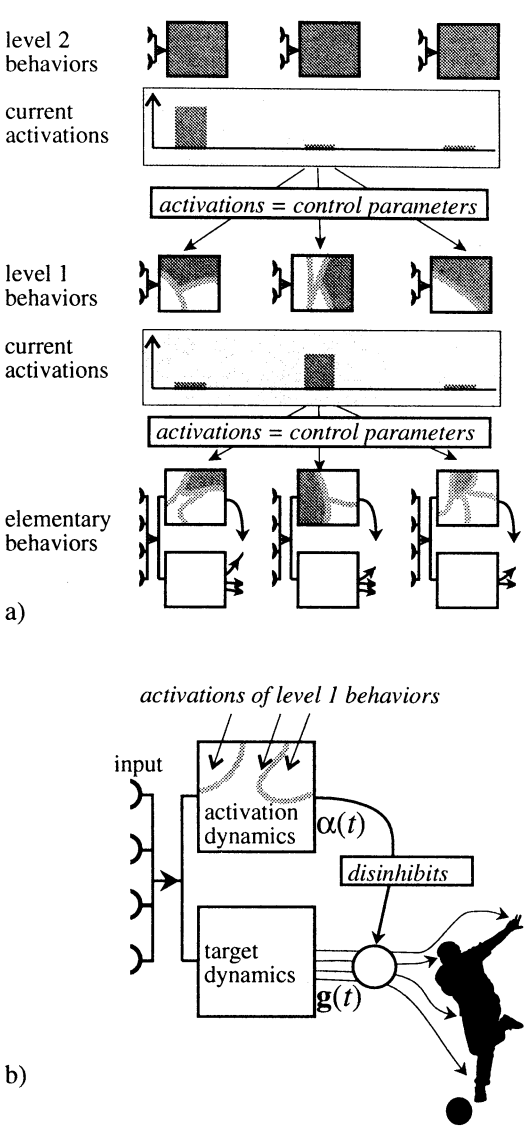


Figure 1: (a) Global structure of a DD behavior control system. At any time, every behavior has an activation. Activations of higher-level behaviors (depicted in shaded boxes) act as control parameters for the activation dynamics of lower levels. The dynamical system which maintains a behavior's activation can undergo bifurcations; this in indicated by depicting these systems as stylized "phase diagrams" (boxes with irregular partitions). A mode of the entire system is thus determined by the activations of all higher-level behaviors. (b) The target and activation subsystems of an elementary behavior.

2.2 The formal model

Here we present a core version of the formal DD model. A more elaborate version is described in [5].

First we describe a single elementary behavior.

The target dynamics of an elementary behavior B_j yields a vector-valued target trajectory $\mathbf{g}_j(t)$, where each vector component represents the target trajectory of a particular effector. The target dynamics is expressed via ordinary differential equations:

$$\dot{\mathbf{g}}_j = G(\mathbf{g}_j, \alpha_j, I_j(t)) \quad (1)$$

The quantity α_j is the activation of B_j (see below). $I_j(t)$ represents time-varying input to B_j, such as sensor input.

The activation dynamics of B_j consists in the trajectory of a single parameter, α_j, which determines whether the behavior is inhibited ($\alpha_j \sim 0$) or active ($\alpha_j \sim 1$), or something in between.

The activation dynamics of a behavior can bifurcate, yielding different modes. This is achieved by utilizing the activations $\alpha'_1, \ldots, \alpha'_m$ of the level-1 complex behaviors $B'_1, \ldots, B'_m$ as control parameters. The activation dynamics of B_j looks as follows:

$$\begin{aligned}\dot{\alpha}_j = {} & \alpha'_1 T_{j,1}(\alpha_j, \mathbf{g}_j, I_j(t)) + \ldots + \\ & + \alpha'_m T_{j,m}(\alpha_j, \mathbf{g}_j, I_j(t)) - \alpha_j k \prod_{i=1,\ldots,m} (1 - \alpha'_i)^2\end{aligned} \quad (2)$$

$T_{j,i}(\alpha_j, \mathbf{g}_j, I_j(t))$ is a function of α_j, the behavior's target trajectory $\mathbf{g}_j$, and possibly other input quantities $I_j(t)$. Each $T_{j,i}$ corresponds to a particular mode of (2), which is entered when α'_i is roughly equal to 1, and the other α'_k roughly equal to 0. If α'_i goes to zero and another α'_r rises, the dynamics of (2) changes from being determined by $T_{j,i}$ to being determined by $T_{j,r}$, which will typically result in a bifurcation of (2). The terms $T_{j,i}$ can be designed explicitly and independently from one another.

The decay term $-\alpha_j k \prod_{i=1,\ldots,m}(1 - \alpha'_i)^2$ brings α_j back to zero when all α'_i vanish to zero.

As to the question of what kind of input $I_j(t)$ is permissible for an elementary behavior B_j, DD has an iron rule: *the only input which comes top-down from higher-level behaviors is the mode control parameters α'_i in the activation dynamics.* This rule implicitly allows to use every conceivable non-top-down source for $I_j(t)$, e.g., sensor input, activations of other elementary behaviors, or their target trajectories.

In order to yield an output signal from the behavior to the actuators, $\mathbf{g}_j(t)$ and $\alpha_j(t)$ are combined via the following product assignment:

$$\dot{\mathbf{u}}_j = k_j \alpha_j (\mathbf{g}_j - \hat{\mathbf{z}}_j), \quad (3)$$

where $\hat{\mathbf{z}}_j$ is the estimated current state of the actuators, k_j is a gain constant, and $\mathbf{u}_j$ is the signal issued from the behavior to the actuators. This *product term* implements a simple closed loop control (of P-controller type) which tries to make the actuators follow the target trajectory $\mathbf{g}_j$. If this propertional control proves inefficient, the product term can be augmented to more complex control schemes, e.g. PID-control. The DD scheme is not committed to a particular kind of control realized in this term.

Taken all together, the DD model of an elementary behavior consists of the equations (1), (2), and a suitable version of (3).

Now let us turn to the complete picture. The DD scheme allows to construct multi-level architectures with any number of levels. Since higher levels – the top level excepted – are formally similar to each other, we can restrict this presentation to the case of a three-level architecture with l second-level complex behaviors $B''_1, \ldots, B''_l$, m first-level complex behaviors $B'_1, \ldots, B'_m$, and n elementary behaviors $B_1, \ldots, B_n$.

$$\begin{aligned}\dot{\alpha}'_i = {} & \alpha''_1 T'_{i,1}(\alpha'_i, I'_i(t)) + \ldots + \\ & + \alpha''_l T'_{i,l}(\alpha'_i, I'_i) - \alpha'_i k \prod_{h=1,\ldots,l} (1 - \alpha''_h)^{r'_i}\end{aligned} \quad (4)$$

The activation dynamics of a second-level complex behavior B''_h has the form

$$\dot{\alpha}''_h = T''_h(\alpha''_h, I''_h(t)) \quad (5)$$

The iron rule concerning inputs mentioned in the previous section transfers to complex behaviors. Thus, $I'_i(t)$ and $I''_h(t)$ can be virtually anything provided it does not come from higher levels.

3 Practical experiences with DD

A reference implementation of the DD scheme has been carried out as part of cooperative work between GMD and the VUB AI Lab. The VUB AI Lab offers a large experimental arena which has been devised as a model of an ecosystem [6]. Robots in this arena have to "survive" by performing a robotic analogue of a "parasite fighting" task in an energy-efficient fashion, and by recharging in due intervals. The arena is

a challenging environment since it offers no controlled lighting conditions (lighting ranges from direct sunshine in the morning to distant artificial lighting at night hours), while the robots have to rely on simple light sensors for achieving their tasks.

One of the Lab's Lego-based 2df robots, the "Black Knight"(BK) was used as a platform for a reference implementation for the DD scheme[1]. A repertoire of behaviors that is standard for many robots running in the "ecosystem" arena was re-implemented using the DD scheme.

A detailed account of experiences with designing BK's behavior control system is not feasible here[2]. Suffice it to say that specifying the behavior control system in terms of some 50 differential equations took the first author a week, programming and debugging 10 days, and fine-tuning the many, many time constants in these equations another 2 days. These times include learning the basics of C, and overcoming the misconception of reliable sensor readings – this author was at that time a complete novice in practical robotics.

4 Discussion

The DD scheme has several obvious limitations, among them the following two. First, it is a very general scheme which helps to organize a complete system of many behaviors, but it does not say anything specific about how the target and activation dynamics of each behavior should be designed. Second, the mathematical format of ordinary differential equations is confined to handling sensoric input which consists in a few variables. Specifically, this excludes video input.

The advantages of DD lie in its conceptual and mathematical clarity, which in turn leads to transparent, and hence efficient, design. After all, DD made it possible for a newcomer to practical robotics to finish a design and implementation job within three weeks, achieving a behavioral complexity that is state of the art in behavior-based robotics. This holds some good promise for DD to lead beyond the limitations in behavioral complexity with which the field seems to have gone stuck since some years.

[1] Cf. ftp://ftp.gmd.de/GMD/ai-research/Publications/1996-/jaeger.96.dd14.c for the heavily documented code.

[2] A first detailed report on a particular behavior is available in ftp://ftp.gmd.de/GMD/ai-research/Publications/1996/jaeger.96.brain.ps.gz

Acknowledgements

The authors feel deeply grateful toward Luc Steels and his collaborators for a very rewarding collaboration.

References

[1] G.P. Baerends. On drive, conflict and instinct, and the functional organization of behavior. In M.A. Corner and D.F. Swaab, editors, *Perspectives in Brain Research. Proc. of the 9th Int. Summer School of Brain Research, Amsterdam, August 1975*, pages 427–447. Elsevier, Amsterdam, 1975.

[2] R. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1/2):173–216, 1995.

[3] R. A. Brooks. Intelligence without reason. A.I. Memo 1293, MIT AI Lab, 1991.

[4] J.-P. Ewert, T.W. Beneke, E. Schürg-Pfeiffer, W.W. Schwippert, and A. Weerasuriya. Sensorimotor processes that underlie feeding behavior in tetrapods. In V.L. Bels, M. Chardon, and P. Vandewalle, editors, *Biomechanics of Feeding in Vertebrates*, volume 18 of *Comparative and Environmental Physiology*, pages 119–161. Springer Verlag, 1994.

[5] H. Jaeger. The dual dynamics design scheme for behavior-based robots: A tutorial. Arbeitspapiere der GMD 966, GMD – Forschungszentrum Informationstechnik GmbH, St. Augustin, 1996.

[6] D. McFarland. Towards robot cooperation. In D. Cliff, editor, *From Animals to Animats III: Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior*, pages 440–444. Bradford/MIT Press, 1994.

[7] G. Schöner, M. Dose, and C. Engels. Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2):213–246, 1995.

[8] L. Steels. Building agents out of autonomous behavior systems. In L. Steels and R.A. Brooks, editors, *The "Artificial Life" Route to "Artificial Intelligence": Building Situated Embodied Agents.* Lawrence Erlbaum, 1993.

A study on an autonomous mobile robot control with behavior network

Keisuke Kondo Ikuko Nishikawa Hidekatsu Tokumaru

Department of Computer Science
Faculty of Science and Engineering
Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu Shiga 525-77 Japan

Abstract

This paper describes controllers of an autonomous robot with behavior network. Actions of the robot are classified into eight behaviors according to difference between right and left motor speeds. Each behavior corresponds to an output unit of two-layered neural network. Threshold in this neural network takes two values according to the previous output value. The robot selects one behavior. In our experiment, the robot can avoid obstacle and wall after learning.

1 Introduction

The conventional approach to control an autonomous robot is to decompose the problem into a series of functional units. For example, it decomposes the problem into some subset of sensing, perception, modeling, making of action plan, task execution, and motor control (Figure 1 (a)).

Behavior-Based approach proposed by Brooks builds intelligent control systems where many individual modules directly generate some part of the behavior of the robot (Figure 1 (b)). The programs has layered, but non-hierarchical structure, with lower levels taking care of more primitive activities and higher levels taking care of more sophisticated ones. While robots built by this approach have been demonstrated learning, progress in learning new behaviors has proved rather difficult.

This paper describes controllers of an autonomous robot with behavior network. Actions of the robot are classified into eight behaviors according to difference between right and left motor speeds. Each behavior corresponds to an output unit of two-layered neural network. Robot selects one behavior. These behaviors cooperate and compete each other through learning processes. In our experiment, the robot can avoid obstacle and wall after learning.

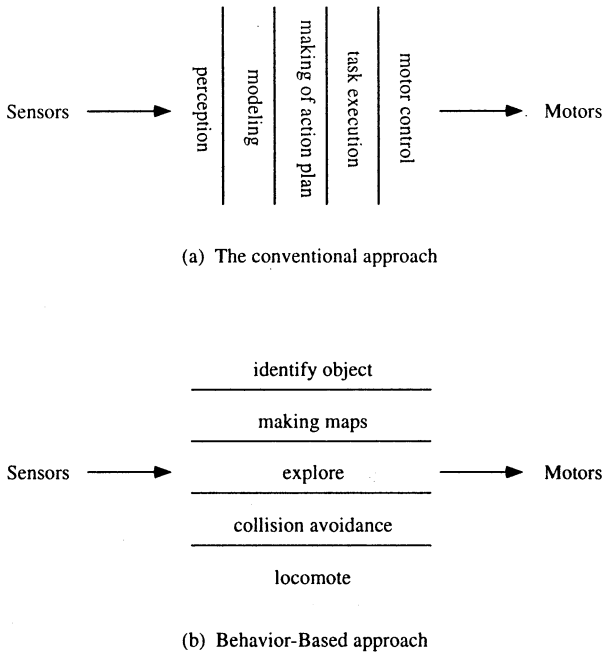


Figure 1: Control systems for an autonomous mobile robot

2 Experiment environment

In our experiment, Khepera (Mondana, Franzi & Ienne, 1993, Lami/EPFL) is used as an autonomous robot. It is a miniature robot with a diameter of 55 mm, height of 30 mm and weight of 70 g. It is supported by two wheels, each controlled by a DC motor with an incremental encoder. It has eight infrared proximity sensors placed around its body (six on the front and two on the back : Figure 2) which are based on emission and reception of infrared light. It can be controlled by an external computer through cable.

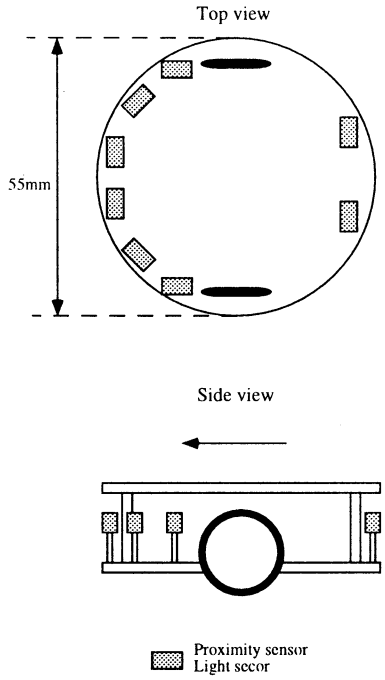


Figure 2: Position of the sensor

The simulation environment field is 70 cm × 70 cm large. There are four columnar obstacles as shown in Figure 3.

3 Behavior Network

Actions of Khepera are classified into eight behaviors according to difference between right and left motor speeds. They are shown in Figure 4.

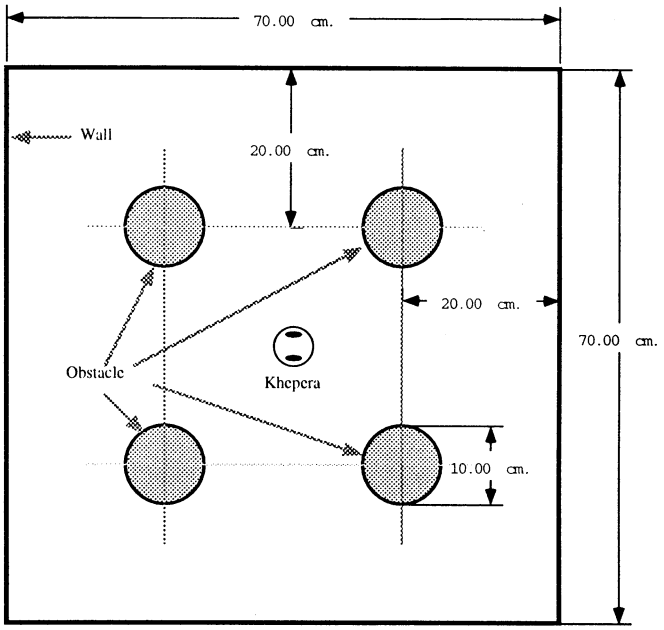
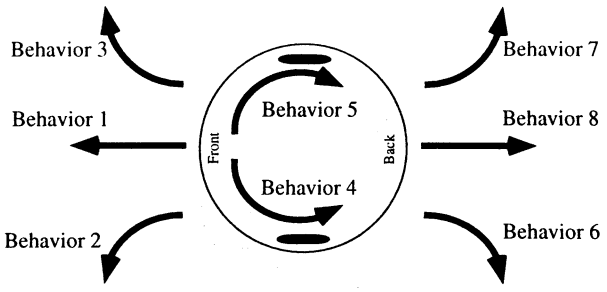


Figure 3: Simulation filed



Behavior 1 : Left_motor=Right_motor>0
Behavior 2 : Right_motor>Left_motor>0
Behavior 3 : Left_motor>Right_motor>0
Behavior 4 : Right_motor>0>Left_motor
Behavior 5 : Left_motor>0>Right_motor
Behavior 6 : 0>Left_motor>Right_motor
Behavior 7 : 0>Right_motor>Left_motor
Behavior 8 : Left_motor=Right_motor<0

Figure 4: Eight types of behavior

One behavior is selected for Khepera by a two-layered neural network. The input units x_i consist of two motor units x_0, x_1 and eight sensor units $x_2, \ldots, x_9$. The output units z_j consist of eight behaviors (Figure 5). Connection between input and output units is given by (1).

Output value z_j is further transformed into y_j by the following thresholding. Threshold takes two values $h_1, h_2 (h_1 > h_2 > 0)$ according the previous value of y_j. If $y_j(t-1)$ equals to 0, then the threshold value is h_1, otherwise the threshold is smaller value h_2. The behavior with maximum value y_j is chose as the present action $J(t)$ of Khepera. When y_j, which attains maximum value once becomes not maximum, then y_j is forced to be 0 for five time steps, as a refractory period.

$$z_j(t) = \sum_i w_{ij}(t) x_i(t) \tag{1}$$

$$\Theta(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

$$y_j(t) = \begin{cases} z_j(t)\Theta(z_j(t) - h_1) & : \text{ for } y_j(t-1) = 0 \\ z_j(t)\Theta(z_j(t) - h_2) & : \text{ otherwise} \end{cases} \tag{3}$$

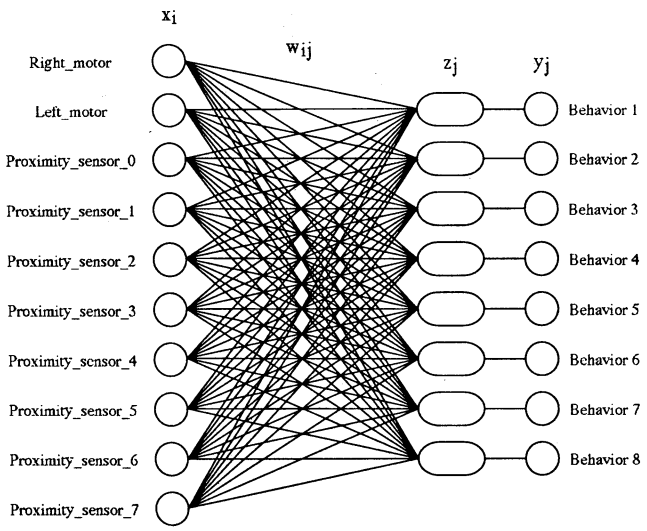


Figure 5: Behavior network

4 Evaluation Function

The robot has two modes depending on the proximity sensor inputs. One is normal mode and the other is danger mode. When at least one infrared sensor takes maximum value, Khepera is set to danger mode. In this mode, Khepera is not more than 2 cm from obstacle. Otherwise it is in normal mode.

The evaluation function at time t $R(t)$ is defined as a summation of four variables $V(t), S(t), M(t)$ and $C(t)$. $V(t)$ measures the time-averaged rotation speed of two wheels, $S(t)$ is the difference between the present and the previous values of the total input from eight sensors, $M(t)$ describes the change between two modes, and $C(t)$ is a penalty for collision to the wall or obstacle.

$$V(t) = \frac{\sum_{k=0}^{19} x_0(t-k)}{20} + \frac{\sum_{k=0}^{19} x_1(t-k)}{20} \tag{4}$$

$$S(t) = \sum_{i=2}^{9} x_i(t-1) - \sum_{i=2}^{9} x_i(t) \tag{5}$$

$$M(t) = mode(t) - mode(t-1) \tag{6}$$

$$mode(t) = \begin{cases} 0 & : \text{ normal mode} \\ -1 & : \text{ danger mode} \end{cases} \tag{7}$$

$$C(t) = \begin{cases} -1 & \text{in the case of the collision} \\ & \text{to wall or obstacle} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$R(t) = V(t) + S(t) + M(t) + C(t) \tag{9}$$

1 According to $R(t)$, learning rule for the weight $w_{ij}(t)$ is given by the following formula;

$$w_{ij}(t+1) = w_{ij}(t) + \alpha r(t) \sum_{n=0}^{10} \gamma^n x_i(t-n) \delta_{iJ(t)} \tag{10}$$

, where $\delta_{iJ(t)}$ stands for Kronecker's delta.

5 Experiment results

Experiment of learning collision avoidance was done using (10). Thought the learning rule is applied through the experiment, weight values seem to converge around 10^4 time steps. Collision occurs seven times for $1 \sim 1 \times 10^4$ steps but only once for each $1 \times 10^4 \sim 2 \times 10^4$ steps and $2 \times 10^4 \sim 3 \times 10^4$ steps. After then no collision occurs during the experiment (which finished at 25×10^4).

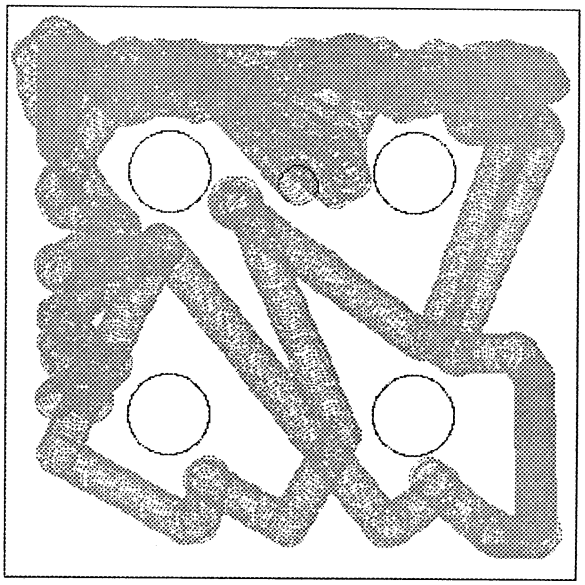


Figure 6: Behavior of Khepera for $21847 \sim 32000$

6 Conclusions

This paper describes controllers of an autonomous robot with behavior network. Thought the learning rule is applied through the experiment. The robot can avoid obstacle and wall after learning in our experiment. As the next step, the learning rule is applied in a general environment.

References

[1] Rodney A. Brooks, "A Robust Layered Control System For a Mobile Robot," *IEEE Journal of Robotics and Automation*,Vol. RA-2, No. 1, pp.14-23, March 1986.

[2] Dario Floreano, Francesco Mondada, "Evolution of Homing Navigation in a Real Mobile Robot," *IEEE Transaction on Systems, Man, and Cybernetics*,Vol. 26, No. 3, pp.396-407,June 1996.

[3] Pattie Maes, "Behavior-Based Artificial Intelligence," *From Animals to Animates II :Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pp.2-10, 1992.

[4] Francesco Mondada, Dario Floreano, "Evolution of neural control structures: some experiments on mobile robots," *Robotics and Autonomous Systems*,Vol. 16, pp.183-195, 1995.

[5] K-Team, Khepera - user manual, September 1994.

A Proposal of Inertia-only (Friction/Gravity-free) Robots

Suguru Arimoto and Hiroki Koga
Faculty of Engineering , The University of Tokyo
Bunkyo-ku, Tokyo, 113 Japan

Abstract

This paper firstly shows that dynamics of nonlinear mechanical systems such as robot manipulators and mechanical hands are expressed by a nonlinear position-dependent circuit. Existence of such a circuit-theoretic expression corresponds to a variational form derived by Euler-Lagrange's formalism based on the principle of virtual work for mechanical systems. The paper secondly shows a simple method of compensating for both friction (static and Coulomb frictions) and gravity forces and proposes a kind of testbed robots called "Friction/Gravity-free" robots.

key words: Friction/Gravity-free Robots, Impedance Control, Passivity, Euler-Lagrange formalism, Nonlinear Circuits

1 Introduction

Dynamics of motion of most mechanical systems are governed by Lagrange's equation of motion. It is well known that Lagrange's equation of motion for robotic arms with revolute-type joints is nonlinear and there are strong dynamic couplings between joint variables. To cope with this difficulty, the so-called "computed torque" method was proposed in 1970's and two fast computation algorithms for computing the left-hand side of Lagrange's equation were proposed independently by Hollerbach[1] and Luh et al.[2]. They showed that the computational complexity of the proposed algorithms is proportional to n, the degree of freedom of the system, and in 1980 much of the literature on development of VLSI implementations of the algorithms were published. However, there still remain various robot control problems that can not be overcome by the computed torque method. The most important issues are 1)basic physical parameters such as link inertia moments are uncertain, 2)frictional terms are unknown, and 3)dynamics become more complicated when the system has physical interaction with an object or task environment. It should be also pointed out that each nonlinear term in mechanical systems has its own physical properties and therefore better controllers can be designed by gaining physical insights into such nonlinear terms and their interrelations and taking advantage of these physical properties.

This paper firstly shows that dynamics of nonlinear mechanical systems such as robotic arms and mechanical hands are expressed by a nonlinear circuit similar to electric circuits. It is pointed out in many text books that dynamics of a simple mechanical system consisting of mass, dashpot, and spring have analogy to a linear lumped-parameter electric LCR circuit. However, it has been believed implicitly that this analogy is valid for linear dynamics. In this paper, not only more general and nonlinear dynamics with free endpoint but also nonlinear dynamics under holonomic constraint can be expressed by a kind of nonlinear circuits called "nonlinear position-dependent circuits". It is claimed in this paper that dynamics of a mechanical system have an expression of nonlinear position-dependent circuit if its dynamics can be formulated by a variational form derived by the Euler-Lagrange formalism. This is regarded as a converse to the fact claimed by Meisel[3] that any electric circuit without couplings via transformers can be treated by the EL formalism.

One of advantages that circuit-theoretic expressions of dynamics for nonlinear mechanical systems have is that the concept of impedance which plays a crucial role in linear electric circuits can be generalized by referring to "passivity relation" for an input-output pair of torque (corresponding to voltage) and velocity (corresponding to current). It is shown that the passivity relation exhibits an interrelation of energy flows in the system and the concept of impedance matching (see Auslander[4]) also can be understood by seeing the energy flow directly expressed in the circuit-theoretic representation of its dynamics. Therefore, nonlinear position-dependent circuits are considered to be an energy flow-based network of mechanical dynamics. Another merit of such circuit-theoretic expressions is to ease the design methodology of controllers and make it more tractable and intuitive. As one of controller designs for mechanical systems via nonlinear position-dependent circuits, we present a simple method for compensating both gravity forces and Coulomb (and static) frictions by adding time-varying capacitor-like modules with positivity to the original circuit expressing motion of the system. This implies a possibility of realization of "Gravity/Friction-free" robots whose motions are governed virtually by only terms due to the inertia. We present a preliminary experimental result by using a DD robot that demonstrates the effectiveness of the proposed method.

2 Variational Form and Nonlinear Position-dependent Circuit

Consider dynamics of a robotic arm with all revolute joints whose endpoint is constrained holonomically on a surface (see Fig. 1). Denote the kinetic energy, potential energy, and the surface equation by $K(q,\dot{q}) = \frac{1}{2}\dot{q}^{\mathrm{T}}H(q)\dot{q}$, $P(q)$, and $\phi(x(q)) = 0$ respectively, where $q = (q_1, \cdots, q_n)^{\mathrm{T}}$ signifies the joint vector consisting of joint angles and $x = (x_1, x_2, x_3)^{\mathrm{T}}$ does the cartesian coordinates. We assume that the inertia matrix $H(q)$ includes contributions of inertias due to internal load distributions in its diagonal components. We also assume that each friction force of the motor and the transmission mechanism at its corresponding joint can be denoted by $r_i(\dot{q}_i)$ and the contact friction arises in the direction counter to $\dot{x}$ with magnitude $\xi(\|\dot{x}\|)$. Then it is possible to define

$$D_0(\dot{q}) = \sum_{i=1}^{n}\int_0^{\dot{q}_i} r_i(z)dz, \quad D_1(\dot{x}) = \int_0^{\|\dot{x}\|}\xi(z)dz, \tag{2.1}$$

and to describe a variational form for the system of Fig. 1 in the following way:

$$\int_{t_1}^{t_2}[\delta\{K - P - \lambda\phi(x(q))\} + \{F^{\mathrm{T}} - \frac{\partial D_0(\dot{q})}{\partial\dot{q}}\}\delta q - \{\frac{\partial D_1(\dot{x})}{\partial\dot{x}}\}\delta x]\mathrm{d}t = 0. \tag{2.2}$$

which follows from applying the EL formalism to dynamics of the system described by Fig. 1. Here, F denotes the generalized force vector whose components are torques produced from actuators and λ does the Lagrange multiplier, which corresponds to the magnitude equivalent to the contact force f times $\|\phi_x\|^{-1}$, that is, $\lambda = f/\|\phi_x\|$, where $\phi_x = \partial\phi/\partial x$. Since δq is arbitrary, the variational form (2.2) implies the following Lagrange's equation

$$\left\{H(q)\frac{\mathrm{d}}{\mathrm{d}t} + \frac{1}{2}\dot{H}(q)\right\}\dot{q} + S(q,\dot{q})\dot{q} + r(\dot{q}) + g(q) = -J_x^{\mathrm{T}}(q)\{\bar{f} + \xi(\|\dot{x}\|)\vec{\dot{x}}\} + F \tag{2.3}$$

where $J_x = \partial x/\partial q$, $\vec{\dot{x}}$ denotes the unit vector with the same direction as $\dot{x}$, and $\bar{f} = (f/\|\phi_x\|)\phi_x^{\mathrm{T}}$ denotes the contact force vector arises in the direction of the normal to the surface at the contact point $x = x(q)$. It has been shown by one of the authors [5] [6] that dynamics of eq. (2.3) can be expressed by a circuit shown in Fig. 2. As is seen in Fig. 2, the circuit expressing motion of the original robot dynamics in terms of joint coordinates q is coupled with the circuit in terms of cartesian coordinates x via the Jacobian matrix $J_x(q)$. In eq. (2.3), the coefficient matrix $S(q,\dot{q})$ is equal to $\frac{1}{2}\dot{H}(q)\dot{q} - \{\partial K(q,\dot{q})/\partial\dot{q}\}^{\mathrm{T}}$, which

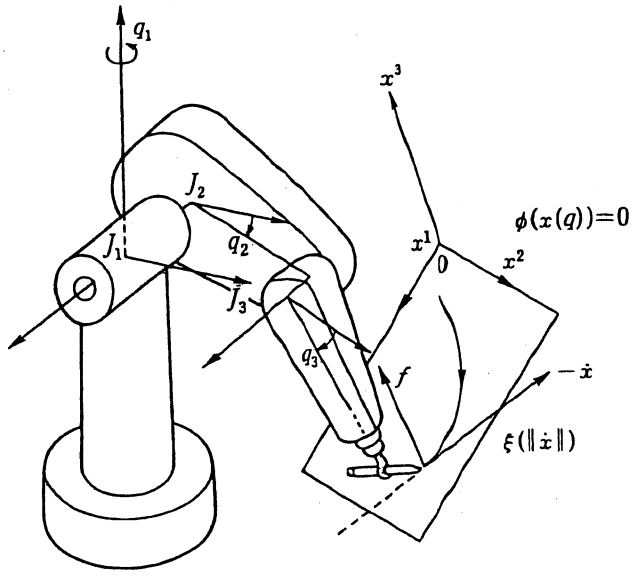


Fig.1 A robot task under holomic tool-endpoint constraint.

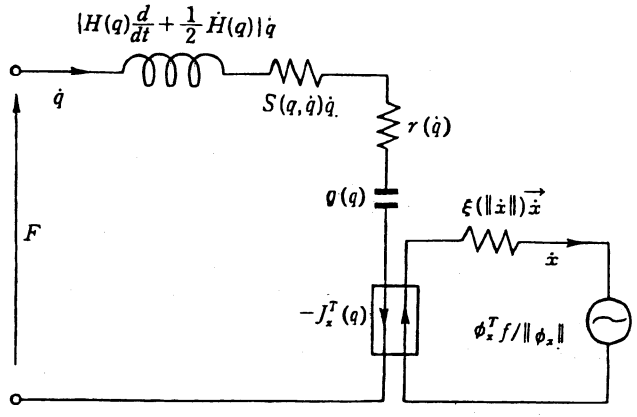


Fig.2 A circuit representation of the dynamics under geometric constraint described by eqn(2.3).

can be described in detail as

$$S_{ij}(q,\dot{q}) = \frac{1}{2}\left\{\frac{\partial}{\partial q_j}(\sum_{k=1}^{n}\dot{q}_k H_{ik}) - \frac{\partial}{\partial q_i}(\sum_{k=1}^{n}\dot{q}_k H_{jk})\right\} \tag{2.4}$$

This shows that S is skew-symmetric. The energy relation for the circuit of Fig. 2 can be easily obtained by taking an inner product between input torque F and output angular velocity $\dot{q}$, which results in

$$\int_0^t \dot{q}^{\mathrm{T}}(\tau)F(\tau)\mathrm{d}\tau = E(t) - E(0) + \int_0^t\{\dot{q}^{\mathrm{T}}(\tau)r(\dot{q}) + \xi(\|\dot{x}\|)\dot{x}^{\mathrm{T}}\vec{\dot{x}}\}\mathrm{d}\tau \tag{2.5}$$

where $r(\dot{q}) = (r_1(\dot{q}_1), \cdots, r_n(\dot{q}_n))^{\mathrm{T}}$ and the E denotes

the total energy such that

$$E(t) = \frac{1}{2}\dot{q}^{\mathrm{T}}(t)H(q(t))\dot{q}(t) + P(q(t)). \qquad (2.6)$$

Since $\dot{x}^{\mathrm{T}}\vec{x} \geq 0$, $\dot{q}^{\mathrm{T}}r(\dot{q}) \geq 0$ according to physical properties of frictions, and it is possible to choose the constant of the potential so that $\min_q P(q) = 0$, it follows from eq. (2.5) that

$$\int_0^t \dot{q}^{\mathrm{T}}(\tau)F(\tau)\mathrm{d}\tau \geq -E(0) = -\gamma_0^2. \qquad (2.7)$$

We say that the input-output pair $\{F, \dot{q}\}$ satisfies passivity.

3 Friction Compensation

In this section we firstly show that, under the existence of static and Coulomb frictions at each joint, an ordinary PD feedback with gravity compensation for setpoint position control leads to a trapping of motion at some immovable state within a finite time without attaining the given target position. On the contrary, if the gap between the stick force level and the level of Coulomb friction at each joint is infinitely small, then introduction of regressors for uncertain parameters of gravity forces and Coulomb frictions as time-varying capacitors in the nonlinear position-dependent circuit implies global asymptotic stability of a PID type setpoint position control without incurring any offset. It should be emphasized that, in the loop of PI feedback, saturated position error signals should be used in order to keep passivity of the concerned circuit between input torque vector and output vector. This compensation of gravity force and Coulomb frictional force without using any force/torque sensor can be established by a quite simple circuit block including time-varying capacitors and a time-invariant capacitor that can be coupled with the original nonlinear position-dependent circuit expressing original Lagange's equation of motion. Note that, if there arises at least one immovable axis caused by static and Coulomb frictions, then the validity of Lagrange's equation of motion for the total system is collapsed. Fortunately, some rigorous analysis based on Lyapunov's second method can be applied to the overall system even if there arise discontinuities in acceleration signals, owing to the fact that discontinuities of acceleration of one of axes arise only if the velocity signal at the axis vanishes.

Consider the case that the tool endpoint of a robotic arm can move freely in space. The dynamics can be described by Lagrange's equation as

$$\left\{H(q)\frac{\mathrm{d}}{\mathrm{d}t} + \frac{1}{2}\dot{H}(q)\right\}\dot{q} + S(q,\dot{q})\dot{q} + r(\dot{q}) + g(q) = F. \qquad (3.1)$$

We assume that

$$r(\dot{q}) = B_0\dot{q} + \bar{C}\mathrm{sgn}(\dot{q}) \qquad (3.2)$$

in which the first term of the right hand side denotes viscous frictions and the second term Coulomb frictions, where $\mathrm{sgn}(\dot{q}) = (\mathrm{sgn}(\dot{q}_1), \cdots, \mathrm{sgn}(\dot{q}_n))^{\mathrm{T}}$ with $\mathrm{sgn}(\theta) = +1$ for $\theta > 0$ and -1 for $\theta < 0$. We reasonably assume that both B_0 and $\bar{C}$ are diagonal matrices with positive diagonal entries. Further we assume the existence of static friction at the level $\bar{c}_i + \delta_i$ (the stick force) with $\delta_i > 0$ at the ith joint, which means that the ith joint can not start to move from the rest position unless the exertion of the total torque beyonds the magnitude $\bar{c}_i + \delta_i$ (see Fig. 3). We further assume that static friction at joint i stops the motion around the joint axis if all the following conditions are satisfied:

i) At instant $t = t_0$ the angular velocity $\dot{q}_i(t)$ vanishes, i.e., $\dot{q}_i(t_0 + 0) = 0$.

ii) $|f_i(t_0 - 0)| < \bar{c}_i + \varepsilon_i$ for some $\varepsilon_i > 0$ and $|f_i(t_0 + 0)| \leq \bar{c}_i + \delta_i$, where ε_i is very small relative to δ_i and f_i denotes the ith component of the left hand side of eq. (3.1) minus F except the leading term $H_{ii}(q)\ddot{q}_i$ and Coulomb friction $\bar{c}_i\mathrm{sgn}(\dot{q}_i)$.

In the conditions above, $f_i(t_0 - 0)$ means the limit of $f_i(t)$ as t increases towards t_0 and $f_i(t_0 + 0)$ the limit of $f_i(t)$ as t decreases toward t_0. If conditions i) and ii) above are satisfied, then $\ddot{q}_i(t_0 + 0) = 0$ and there exists a value $\varepsilon > 0$ such that $\ddot{q}_i(t) = 0$ for any $t \in [t_0, t_0 + \varepsilon]$. If a typical PD servo-loop with off-line compensation for the gravity term

$$u = g(q_d) - A(q - q_d) - B_1\dot{q} \qquad (3.3)$$

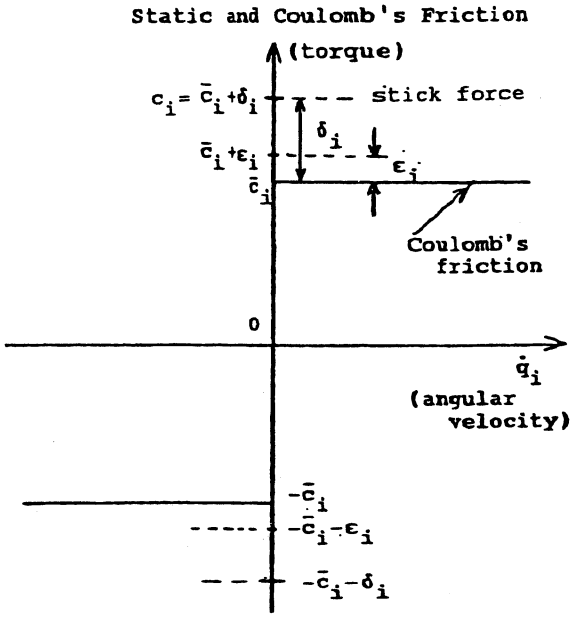


Fig. 3 Torque levels of static and Coulomb's frictions.

for a given target position q_d is used for setpoint position control, then the equilibrium state ($q = q_d, \dot{q} = 0$) becomes globally asymptotically stable provided that there is none of static and Coulomb frictions in any joint and position gain matrix A is chosen enough large to withstand the gravity force, that is, to satisfy the following inequalities simultaneously:

$$P(q) - P(q_d) - \Delta q^{\mathrm{T}} g(q_d) + \frac{1}{2}\Delta q^{\mathrm{T}} A \Delta q \geq \varepsilon \|\Delta q\|^2, \tag{3.4}$$

$$\Delta q^{\mathrm{T}}\{A\Delta q + g(q) - g(q_d)\} \geq \varepsilon \|\Delta q\|^2. \tag{3.5}$$

In contrast with this, when there are static and Coulomb frictions at every joint, then it arises a phenomenon of finite-time trapping as stated in the following:

Proposition 1 If all $\bar{c}_i$ are positive and $A > 0$ satisfies both inequalities (3.4) and (3.5), then for any initial state $(q(0), \dot{q}(0))$ there is a finite time $t_s \geq 0$ such that for any $t(\geq t_s)$ $\dot{q}(t) \equiv 0$ and $q(t) \equiv$ const. at an immovable state, that is, the ith component of $\{g(q) - g(q_d) + A\Delta q\}$ is fixed in $(-\bar{c}_i - \delta_i, \bar{c}_i + \delta_i)$ for all i after $t \geq t_s$.

To elude the finite-time trapping in a manifold $\dot{q} = 0$ without attaining the target position, it is known widely among research workers engaged in practical experiments that the use of a regressor for Coulomb frictions is quite effective in practice. Motivated by this fact, we consider a pair of regressors, $Z(q_d, \dot{q}) = (Z_0(q_d), Z_1(\dot{q}))$, where $Z_0(q_d)$ is a regressor for the gravity term $g(q_d)$ such that $Z_0(q_d)\Theta_0 = g(q_d)$ with an uncertain parameter vector Θ_0 consisting of link masses and a payload, and $Z_1(\dot{q}) = \mathrm{diag}(\mathrm{sgn}(\dot{q}_1), \cdots, \mathrm{sgn}(\dot{q}_n))$ such that $\bar{C}\mathrm{sgn}(\dot{q}) = Z_1(\dot{q})\Theta_1$ with a parameter vector $\Theta_1 = (\bar{c}_1, \cdots, \bar{c}_n)^{\mathrm{T}}$ of uncertain coefficients $\bar{c}_i$ of Coulomb frictions. If we denote $\Theta = (\Theta_0^{\mathrm{T}}, \Theta_1^{\mathrm{T}})^{\mathrm{T}}$, then

$$Z(q_d, \dot{q})\Theta = \bar{C}\mathrm{sgn}(\dot{q}) + g(q_d). \tag{3.6}$$

Since Θ is unknown, we use an estimater $\hat{\Theta}(t)$ at time t which is updated by the law

$$\hat{\Theta}(t) = \hat{\Theta}(0) - \int_0^t \Gamma^{-1} Z^{\mathrm{T}}(q_d, \dot{q}(\tau)) y(\tau) \mathrm{d}\tau. \tag{3.7}$$

Here, y is defined as

$$y = \dot{q} + \alpha s(\Delta q) \tag{3.8}$$

with a displacement vector $s(\Delta q) = (s_1(\Delta q_1), \cdots, s_n(\Delta q_n))^{\mathrm{T}}$ where $s_i(\Delta q_i)$ is a nonlinear saturated function as shown in Fig. 4. By the use of the estimater $\hat{\Theta}(t)$ for unknown parameters Θ, we consider a simple distributed feedback control

$$u = -A\Delta q - B_1 \dot{q} - \beta y - C\int_0^t y(\tau)\mathrm{d}\tau + Z(q_d, \dot{q})\hat{\Theta} \tag{3.9}$$

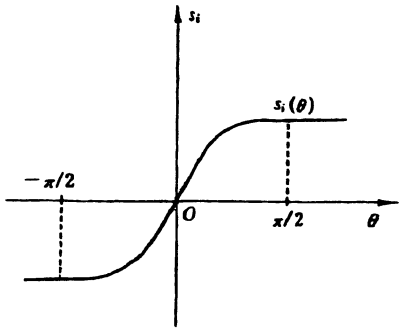


Fig. 4 A profile of saturated function $s_i(\theta)$.

with $\beta > 0$ and $C > 0$. Substituting this into eq. (3.1) yields

$$\left\{(H_0 + H(q))\frac{\mathrm{d}}{\mathrm{d}t} + \frac{1}{2}\dot{H}(q)\right\}\dot{q}\{B + S(q,\dot{q})\}\dot{q} + A\Delta q + \beta y + g(q) - g(q_d) + C\int_0^t y(\tau)\mathrm{d}\tau + Z(q_d, \dot{q})\Delta\Theta = 0 \tag{3.10}$$

where $\Delta\Theta = \Theta - \hat{\Theta}$. When there is no trapping at any joint, an inner product of the last term on the left hand side of eq. (3.10) with y leads to

$$y^{\mathrm{T}} Z(q_d, \dot{q})\Delta\Theta = -\left\{\frac{\mathrm{d}}{\mathrm{d}t}\Gamma\hat{\Theta}\right\}^{\mathrm{T}}\Delta\Theta = \frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\{\Delta\Theta^{\mathrm{T}}\Gamma\Delta\Theta\} \tag{3.11}$$

and the fifth term on the right hand side of eq. (3.9) can be written in the form

$$Z(q_d, \dot{q}(t))\hat{\Theta}(t) = Z(t)\hat{\Theta}(0) - \int_0^t Z(t)\Gamma^{-1}Z^{\mathrm{T}}(\tau)y(\tau)\mathrm{d}\tau \tag{3.12}$$

where $Z(t) = Z(q_d, \dot{q}(t))$. Evidently, the matrix $K(t,\tau) = Z(t)\Gamma^{-1}Z^{\mathrm{T}}(\tau)$ can be regarded as a kernel operator with positivity. Hence, eq. (3.10) can be expressed by a nonlinear position-dependent circuit as shown in Fig. 5.

It should be remarked again that the above argument is valid for only the case that all angular velocities $\dot{q}_i$ $(i = 1, \cdots, n)$ are non-zero, that is, there is no trapping of motion at any joint. If one joint axis stops rotational movement by being trapped due to its static and Coulomb frictions, the update of parameter estimate at its corresponding component should be stopped because the corresponding component of eq. (3.10) is not valid, too. Hence, we introduce a diagonal matrix $S(\dot{q})$ defined by

$$S_{ii}(\dot{q}) = \begin{cases} 1 & \text{if } \dot{q}_i \neq 0, \\ 0 & \text{if } \dot{q}_i = 0, \end{cases} \tag{3.13}$$

and, instead of eq. (3.7), we set

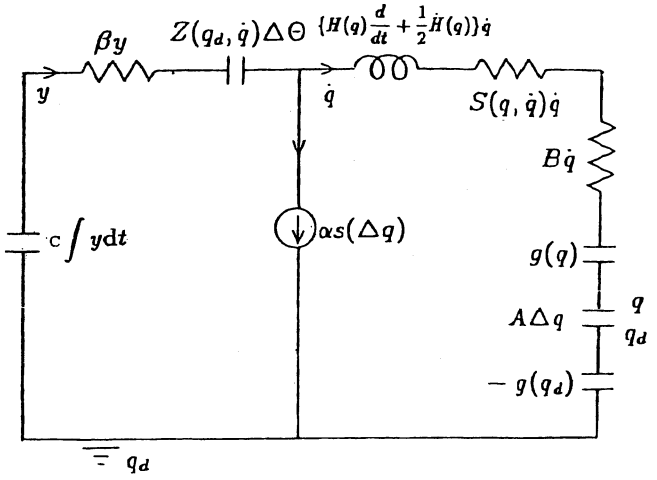


Fig.5 This circuit is equivalent to dynamics of eqn(3.7).

$$\hat{\Theta}(t) = \hat{\Theta}(0) - \int_0^t \Gamma^{-1} Z^{\mathrm{T}}(q_d, \dot{q}(\tau)) S(\dot{q}(\tau)) y(\tau) \mathrm{d}\tau. \tag{3.14}$$

When the gap between the magnitudes of static friction (stick force) and Coulomb friction at every joint is quite small, that is, $\delta_i \cong 0$ and $\varepsilon_i \cong 0$ for all $i = 1, \cdots, n$, as seen in direct drive robots, it is then possible to state the following:

Proposition 2 By taking $\alpha > 0$ appropriately and assuming the zero gap between the magnitudes of static friction and Coulomb friction at every joint (i.e., $\delta_i = \varepsilon_i = 0$ for all i in Fig. 3), the solution to eq. (3.10) together with the update law of eq. (3.14) for any initial state $(q(0), \dot{q}(0))$ and a bounded initial guess $\hat{\Theta}(0)$ of unknown parameters Θ (say, $\hat{\Theta}(0) = 0$) converges asymptotically to the equilibrium state $(q_d, 0)$ as $t \to \infty$, regardless whether $\hat{\Theta}(t)$ converges or not to its true vector value.

To see the physical essence of the proposed method, consider the circuit of Fig. 6 that expresses the simplest case of such motion with the single degree of freedom. During a time interval $[t_0, t_1]$ when $\dot{q}_i(t) \neq 0$ for any $t \in (t_0, t_1)$, the capacitors as regressors for compensation for gravity and Coulomb frictions do work and thereby the squared parameter estimation error $\Gamma\Delta\Theta^2/2$ together with the kinetic energy and the potential concerning the position displacement decreases. Once the motion is trapped at $t = t_1$ in an immovable state $\dot{q}_i(t) = 0$, then connection between terminals P and Q in the circuit of Fig. 6 is switched off and the current does not flow in the right hand closed loop of the circuit. Instead of it, the current flows in the left hand closed loop due to the current source $\alpha s(\Delta q_i)$ as long as $\Delta q_i \neq 0$ and after a while the charge at the capacitor with capacitance C^{-1} (compliance C) accumulates enough to switch on the connection between terminals P and Q, that is, the magnitude of the torque between P and Q becomes large enough to beyond the level of stick force and to start to move again.

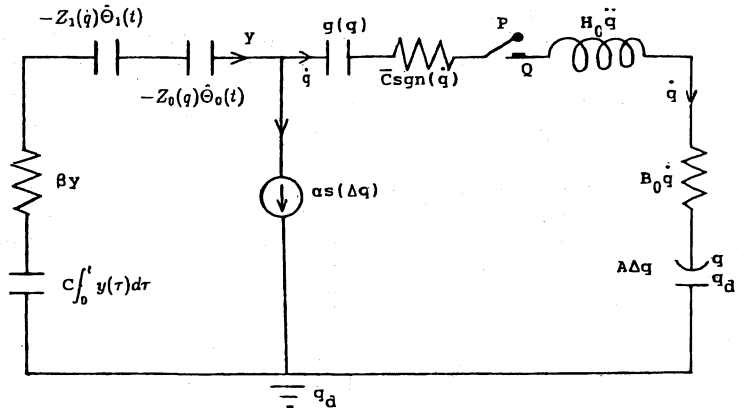


Fig.6 A circuit expressing dynamics of single degree of freedom under static and Coulomb's frictions.

4 Gravity/Friction-free Robots

Although it is not possible in theory to guarantee the convergence of parameter estimation error $\Delta\Theta$ within a single maneuvering of the robot for setpoint regulation, it is expected in practice that the quadratic error of parameter estimate $\frac{1}{2}\Delta\Theta^{\mathrm{T}}\Gamma\Delta\Theta$ may decrease with increasing t. In particular, estimation error $\Delta\Theta_1$ for magnitudes of Coulomb frictions may decrease, too, with increasing t during its maneuvering. This means that the level of the maximum torque to trap the ith axis in an immovable mode decreases during maneuvering the robot. Hence, chattering arised from discontinuity of accelerations $\ddot{q}(t)$ is expected to diminish with progression of time t. Note that, in the case of above Proposition 2, chattering arises at acceleration level, differently from that, in most of sliding mode controls for robots, chattering arises at input signals.

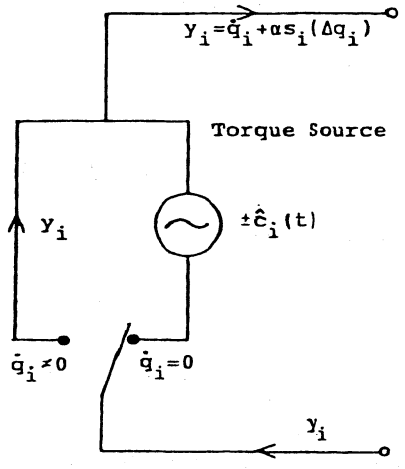


Fig.7 A circuit module for estimation for the magnitude of stick force $c_i(=\bar{c}_i+\delta_i)$.

When the level of stick force is higher by $\delta_i > 0$ than the magnitude of Coulomb's force at the ith joint, is it possible to estimate $c_i (= \bar{c}_i + \delta_i)$ for $i = 1, \cdots, n$ during maneuvering the robot? Fortunately, the recent rapid progress of digital technology and power electronics liberates design of controller modules (see [4]). Instead of the capacitor with compliance C in Fig. 6, we propose a circuit module described by Fig. 7, in which two branches work alternatively according to $\dot{q}_i \neq 0$ or $\dot{q}_i = 0$, that is, during $\dot{q}_i \neq 0$ the current y flows through the left hand branch without causing any voltage drop and during $\dot{q}_i = 0$ the right hand branch works as a torque source which emanates the estimated torque $\hat{c}_i(t)$ for c_i, where $\hat{c}_i(t)$ is regulated by the following rule:

$$\hat{c}_i(t) = \begin{cases} \hat{c}_i(t_{-1}) - \bar{\varepsilon}_i, \\ \quad \text{if } \hat{c}_i(t_0 + 0) = \hat{c}(t_{-1}) \text{ is enough} \\ \quad \text{large to start to rotate the axis,} \\ \hat{c}_i(t_{-1}) + C_i \int_{t_0}^{t} \alpha |s_i(\Delta q_i(\tau))| \mathrm{d}\tau, \\ \quad \text{otherwise.} \end{cases} \tag{4.1}$$

Here, t_{-1} denotes the latest switching instant when the ith axis started to rotate and $t_0 (> t_{-1})$ does the present switching instant when the axis stops to move. The signature of the torque source $(\pm\hat{c}_i(t))$ denotes the direction of rotation and switches according to the signature of Δq_i.

According to a preliminary experiment when one single axis is drived by a direct drive motor of DR series (DR1015B, manufactured by Yokogawa-Hokushin Inc.) as shown in Fig. 8 together with Table 1, it is observed that the setpoint position control from the initial position $q(0) = 0$ (radian) to the target position $q_d = \pi/4$ (radian) did work smoothly and parameter estimates for Coulomb's friction and viscous friction converged quickly with a similar speed of convergence of the position error. In this experiment we also introduced a regressor for estimation of the magnitude of stick force by using the rule described by eq. (4.1), where $\bar{\varepsilon}_i$ is taken to be considerably small relative to δ_i (say, $\bar{\varepsilon}_i \approx 0.1 \times \delta_i$). This update rule for estimation of the static force induces chattering at acceleration level, which does not affect much the position error and the estimation errors for parameters of viscous and Coulomb frictions as seen in Fig. 8(a) and (b).

References

[1] J. M. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 10, pp. 730-736, 1980.

[2] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators", *Trans. ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 102, pp. 69-76, 1980.

$$s(\theta) = \begin{cases} 1 & \theta > \pi/2 \\ \sin\theta & |\theta| \leq \pi/2 \\ -1 & \theta < -\pi/2 \end{cases}$$

α=1.0 (rad / s)	Γ(Coulomb)=4.0 (mN)
P-gain=1.0 (mN)	γ^{-1}=60.0 (s/mN)
D-gain=2.0 (smN)	Γ(viscous)=7.0 (smN)
	Γ(gravity)=2.0 (kgm)

Table 1 A table of constants and the saturated function $s(\theta)$ used in the prelinanary experiment.

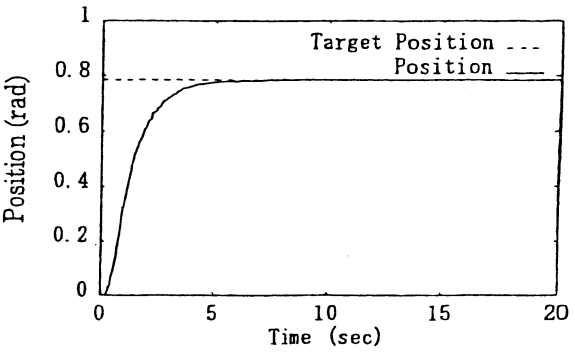


Fig. 8 (a) Transient response the position signal.

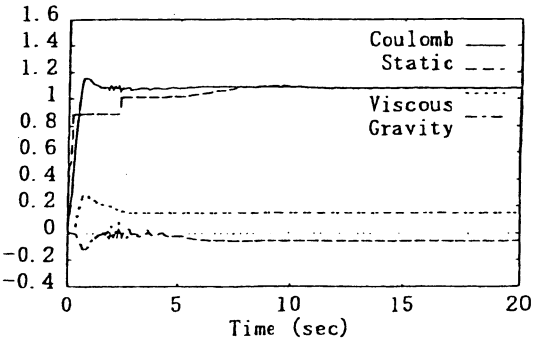


Fig. 8 (b) Transient behaviors of estimators for unknown parameters.

[3] J. Meisel, *Principle of Electromechanical Energy Conversion*, McGraw-Hill, New York, USA, 1964.

[4] D. M. Auslander, "The computer as liberator: The rise of mechanical system control", *Trans. of ASME J. of DSMC*, Vol. 115, pp. 234-238, 1993.

[5] S. Arimoto and T. Nakayama, "Another language for describing motions of mechatronics systems: A nonlinear position-dependent circuit theory", *IEEE/ASME Trans. on Mechatronics*, Vol. 1, No. 2, pp. 168-180, 1996.

[6] S. Arimoto. *Control Theory of Nonlinear Mechanical Systems: A Passivity-based and Circuit-theoretic Approach*, Oxford University Press, Oxford, UK, 1996.

HOJO-Brain for Motion Control of Robots and Biological Systems

Sankai Y., Fujiwara K., Watanabe K., Moriyama H.

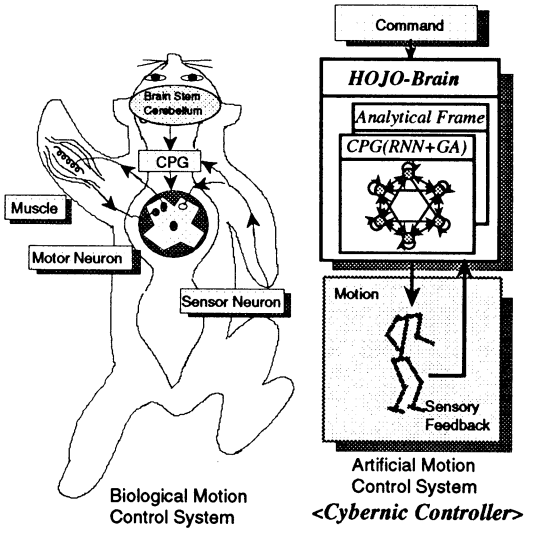
Institute of Engineering Mechanics, University of Tsukuba, Tsukuba, 305, Japan

Abstract

The purpose of this research is to propose and develop a control method in the robotic field and the bio-medical field, which is configured by the robotic/biological simulator, analytical control frame, sensory feedback and the artificial CPG which is constructed by recurrent neural network (RNN) and genetic algorithm (GA). We call such controller "HOJO-Brain", which means supplementary brain for motion control. We apply this method in the robotic field and the bio-medical field. In the robotic field, this HOJO-Brain would be applied to a 5-DOF legged locomotion robot. In the bio-medical field, this HOJO- Brain is applied to animals as the FES (Functional ElectroStimulation) controller. This FES control system with HOJO-Brain would have a possibility to realize more effective and emergent motion control for severely physically handicapped persons such as the quadriplegia . By the computer simulations and the simple actual experiments using animals, we could confirm the fine adaptivity and emergence for the motion control.

1. Introduction

Patients who have damaged spinal cords or physically handicapped person who has motility disturbance by cerebral infantile palsy are restrained to the wheel chair or the bed and inconvenient life is forced on them. In these cases, FES technology gives us some possibilities to control their arms or legs which can not be controlled by themselves. But it might be difficult to control them. Because, we can not decide control trajectory or motion pattern under unknown environment. Of course, parameters and muscular properties of living body would be unknown and nonlinear. In the robotic field, we can find similar situations. Some parameters of multi-link robot system have nonlinear or time-variant properties and it would be difficult to identify them in many cases. Moreover, it is also difficult to design the controller of the robot under unknown environments. The purpose of this research is to propose and develop a control method in the robotic field and the bio-medical field, which is configured by the robotic/biological simulator, analytical motion frame, sensory feedback and the artificial CPG[1–6] which is constructed by recurrent neural network (RNN) and genetic algorithm (GA).



<Concept of this Research>

How to prepare the motion pattern or desired trajectory ?
How to prepare the adaptive controller for each person or different circumstances ?

Adaptive
Emergent

Decentralized

Cybernic Controller
=Recurrent Neural Network (RNN)
+ Genetic Algorithm (GA)
+ Analytical Frame (Controller)

Fig.1 Basic concept of this research

E-mail address: sankai@kz.tsukuba.ac.jp
Tel. +81 295 53 5151 Fax. +81 298 53 5207
www:http://sanlab.kz.tsukuba.ac.jp

We call such controller "HOJO-Brain", which means supplementary brain for motion control. In this research, it is the one to develop the control strategy by the supplementary brain "HOJO-Brain" artificially added besides the motion control system which the living body originally possesses for the physically handicapped person who has the trouble in the signal transmission from the motion control system, based on the conception of newly building in an adaptive supplementary brain "HOJO-Brain". We apply this method in the robotic fields and the bio-medical field. In the robotic field, this HOJO-Brain is applied for a 5-DOF legged locomotion robot. In the bio-medical field, this HOJO-Brain is applied to animals as the FES (Functional Electrical Stimulation) controller[3]. The special mark of this research shown in Fig.1 is *cybernic controller* which consists of analytical motion modes and *self-organized artificial CPG/PGs = Recurrent Neural Network(RNN) + Genetic Algorithm(GA)*.

2. Method and Results

The configuration of HOJO-Brain is shown in Fig.2. The artificial central pattern generator (CPG/PGs) and the analytical controller are main parts of this system.

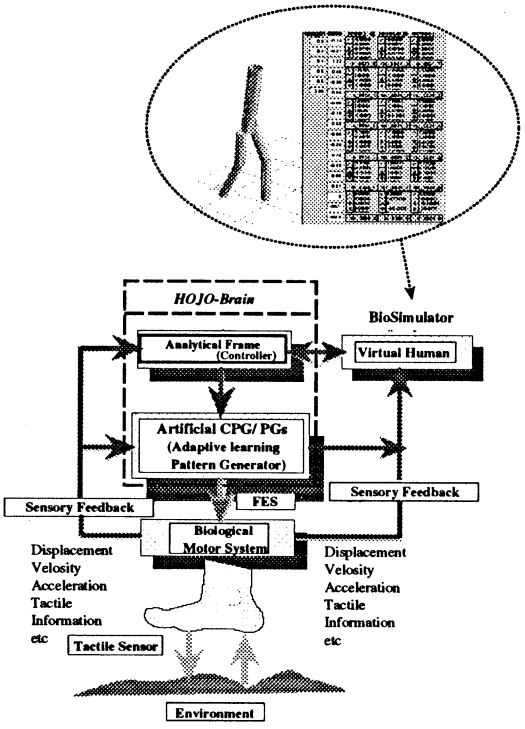


Fig.2 Configuration of "HOJO-Brain"=CPG/PGs and Analytical Controller

Sensory signal such as displacement, velocity, acceleration, tactile information are fed back to the CPG/PGs and the controller to perform self-organization, parameter tuning in some motion modes, state feedback. In this level, we can choose the status in the CPG/PGs, i.e., we can use the plain CPG (no learning) or pre-learned CPG by use of the simulator described as virtual human or biological system or robot system. The purpose of this virtual biosimulator is to realize pre-learning and to reduce the actual repetition. It is very similar to the image training.

The neuron model adopted here is recurrent neural network (RNN). Furthermore, each neuron has a time constant and mutual linkages. Therefore, this network can learn the dynamic motion pattern. The state equation of *i-th* neuron is shown in Fig.3. The synaptic weights and time constant are modified by using genetic algorithm (GA). In this research, RNN weight matrix as the gene is not integer but real number.

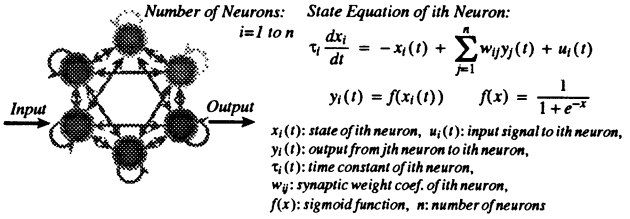


Fig.3 Architecture of neural network and state equation

GA calculation and the algorithm in this research are shown in Fig.4. Parameters in the recurrent neural network (RNN) are treated as the genetic information. Time constants, synaptic weights and thresholds of neurons are cossovered and mutated. These genetic parameters are floating numbers and packaged in gene matrices. To confirm the actual performance of the CPG/PGs, experiments have been performed for twenty frogs as shown in Fig.5. Every spinal cord of frog's lower back was cut and every leg below knee was connected to FES controller. In these experiment, the degree of freedom is one. Conditions in these experiments are set as follows; RNN[number of neuron unit=8, reset inner states of CPG=0], GA[number of pieces=64, mutation rate in pre-learning= 0.15(cf. |error|>10% then 0.3), mutation rate in after-learning=0.05(cf. |error|>10% then 0.3), performance index of fitness=only regulation(desired value=45 degree)].

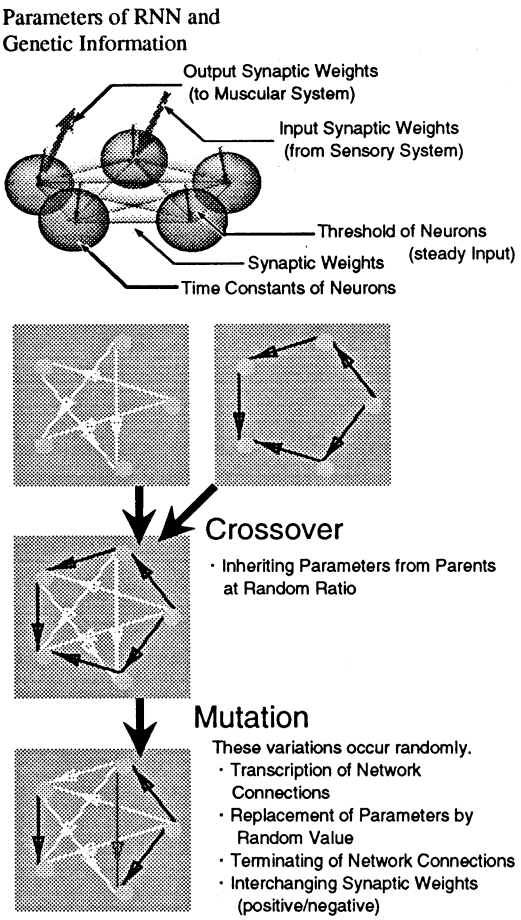


Fig.4 Genetic Algorithm in this research

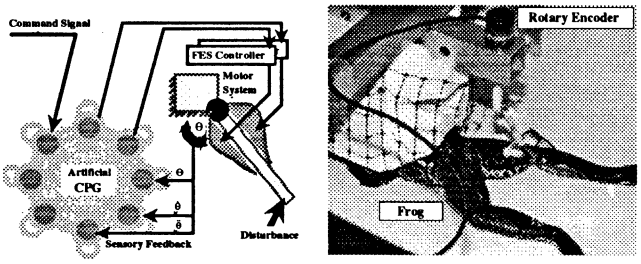


Fig.5 Experiment

Results are shown in Fig.6. In this case, pre-learning of the artificial CPG by use of the simulator was performed previously before the actual experiment. In the simulation, we confirm the advantage of this method for parameter perturbation (e.g. 30%). Of course, parameters of living frogs are unknown and different.

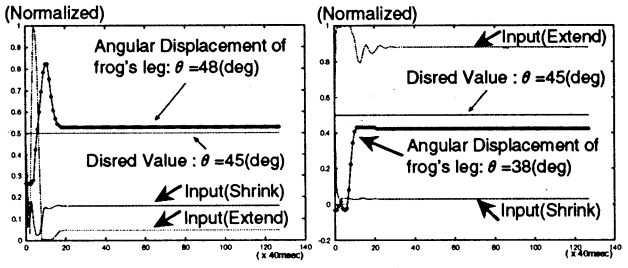


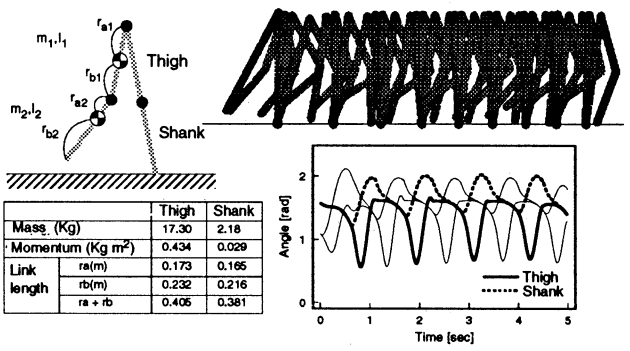
Fig.6 Results of adaptation after 15--19 generations

After experiments, relatively fine results are obtained only 15--19 generations as shown in Fig.6. In the left figure in Fig.6, we can find a overshoot, but amplitudes of input signals are relatively small.

In the right figure(Fig.6), there is no overshoot but amplitudes of input signals are relatively large. In this experiment, performance index is only regulation value. If we need other indices, we can prepare some indices for judgment of fitness, e.g., "regulation, overshoot, input amplitude".

Next, we try to apply this controller to the biped locomotion robot in simulations. In this simulation shown in Fig.7(a), the model of biped robot has 3-DOF and the knee is restrained (0 degree to 180 degree: free). The control frames, i.e., strategy and modes are designed by designers, but parameters and motion timing are searched by modified hill climbing methods. The control frame is shown in Fig.7(b). After virtual trial and error, stable gait(walking) control is obtained. Searched parameters are also shown in Fig.7(b).

Under this analytical frame, parameters for gait control are searched within about 10 minutes by a personal computer.



		Thigh	Shank
Mass (Kg)		17.30	2.18
Momentum (Kg m²)		0.434	0.029
Link length	ra(m)	0.173	0.165
	rb(m)	0.232	0.216
	ra + rb	0.405	0.381

Fig.7(a) Simulation of biped locomotion robot

Strategy:

$\tau_h = K_1(v_{hr} - v_h)$

$\tau_k = K_2(v_{hr} - v_h) + K_3(\theta_{kr} - \theta_k) + D_1(\dot{\theta}_{kr} - \dot{\theta}_k) + T_1 + K_4(v_{hr} - v_h)$

Fitness: $J = A_1 x_h + A_2 y_h + A_3 n$

τ_h: *hip joint torque*, θ_h: *hip joint angle*, $\dot{\theta}_h$: *hip joint angular velosity*

τ_k: *knee joint torque*, θ_k: *knee joint angle*, $\dot{\theta}_k$: *knee joint angular velosity*

v_h: *hip joint velosity(horizontal dir. x)*, x_h: *hip joint position(horizontal dir. x)*

h_t: *hight of toe*, n: *number of steps*, y_h: *hip joint position(vertical dir. y)*

<Mode1> the first half of right leg: free, left leg: support

<Mode2> the latter half of right leg: free, left leg: support

<Mode3> both of legs: support, right leg: forward

<Mode4> the first half of left leg: free, left leg: support

<Mode5> the latter half of leg: free, right leg: support

<Mode6> both of legs: support, left leg: forward

$(K_1, K_2, K_3, K_4, T_1, D_1, v_{hr}, v_{kr})$

<Mode1>=(7.1,−49.9,509,—,—,49.5,0.87,0.11)

<Mode2>=(6.2,−49.2,495,12.5,−1.9,51.1,0.76,0.1)

<Mode3>=(−6.5,50.1,503,7.9,0.78,50.5,0.88,0.11)

<Mode4>=(7.0,49.2,500,—,—,48.7,0.75,0.1)

<Mode5>=(−6.3,50.9,504,−10.2,0.54,50.2,1.2,0.11)

<Mode1>=(6.1,−50.7,502,−8.4,−1.2,49.9,0.82,0.11)

Fig.7(b) Control strategy for gait : walking control

It is necessary to be able to deal with the control system of possible practical use when unknown disturbance and system parameter are indefinite. In this section, some simulations are performed to confirm the performance of the HOJO-Brain controller. The aim of this simulation is to obtain effective attitude/posture control methods such as not only straddling or holding but also walking when the unknown disturbance is added to biped robot or human. 5-DOF model is adopted here. Control frame is shown in Fig.8.

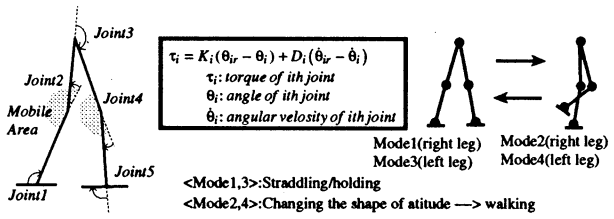


Fig.8 Control strategy of gait : posture/attitude and walking control

Parameters(K,D) and motion timing are searched by the similar way mentioned above. Mode1,3 are the straddling or holding states and Mode2,4 are the states of changing the shape of attitude. In the simulation, joint(1,2,4,5) have torque saturation assumed 7[Nm] and joint3 has torque saturation assumed 14[Nm]. In Fig.9, state of the mode is switching according to the disturbance and biped system stand still after disturbance. In case of continuous disturbance shown in Fig.10, the biped system is walking according to the disturbance. This disturbance is similar to power assist. In case of cyclic disturbance shown in Fig.11, gait control mode is switched according to the state of the attitude.

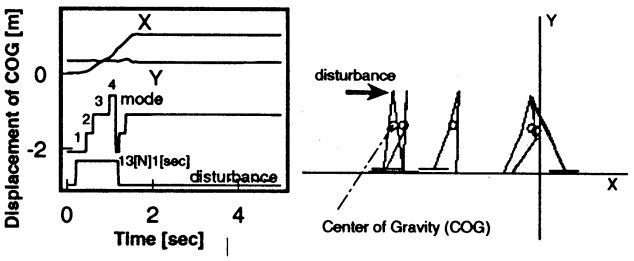


Fig.9 Simulation: disturbance 13[N]1[sec]

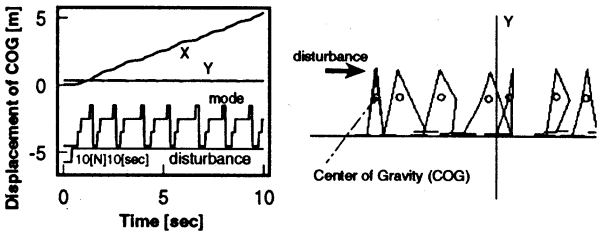


Fig.10 Simulation: disturbance 10[N]10[sec]

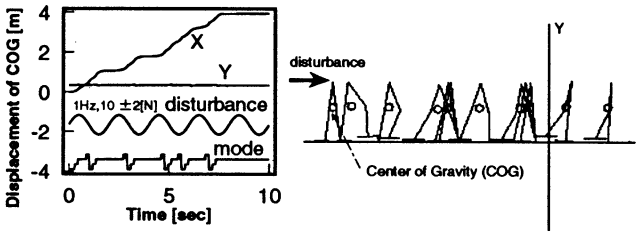


Fig.11 Simulation: disturbance 1Hz, 10[N]±2[N],10[sec]

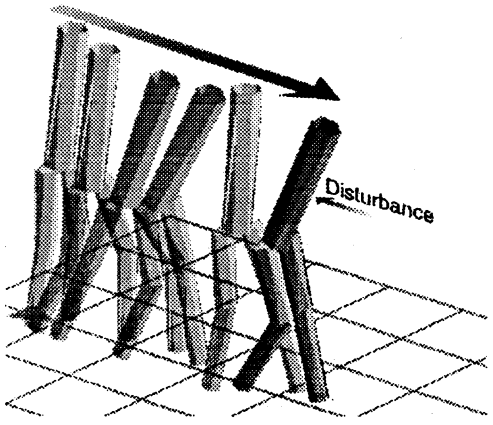


Fig.12 Gait Simulation by using the Artificial CPG

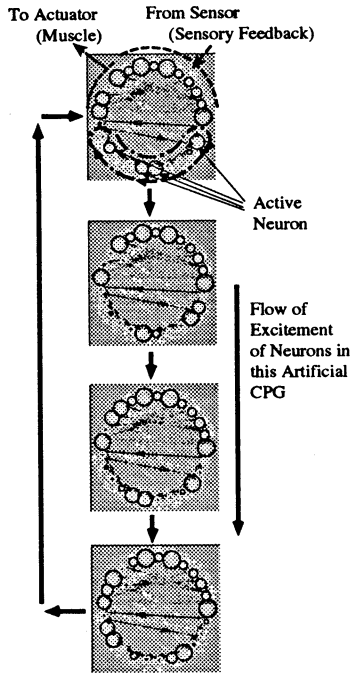


Fig.13 Transition of Excitement of Neurons in this CPG corresponding to the simulation in Fig.12

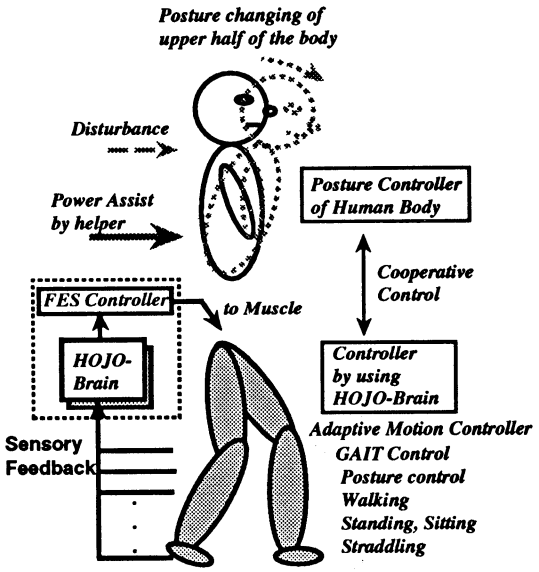


Fig.14 A future application of supplementary brain "HOJO-Brain"

3. Discussion and Conclusion

We could confirm the excellent adaptivity and emergence for the motion control by using this supplementary brain "HOJO-Brain", i.e., cybernic controller which consists of CPG/PGs and analytical controller. The combination of analytical controller and CPG/PGs would be suitable structure. Especially, it must be practical good use to apply this HOJO-Brain for the physically handicapped person who has the trouble in the lumbar vertebra (Fig.14). In this example, the upper half of the body and the lower half of the body will have a separate control system. The lower half of the body would be controlled with HOJO-Brain even when the upper half of the body swings or some disturbance joins. And patient's attitude/posture will be maintained stably. This HOJO-Brain concept might be regarded as essential for the total motion control.

Acknowledgment

Acknowledgements are due to members in our laboratory for their technical assistance. Thanks to Venture Buisiness Laboratory in University of Tsukuba, some parts of this research are achieved.

References

[1] Sankai Y.: A Biocybernetic System in the Artificial Organ Control and Bio-Robotics, International Centre of Biocybernetics, MCB Lecture Note,Vol.23, pp66-99,1995.

[2] Sankai Y., Ohta M.,:A Basic Concept of Super Rabbit, RoManSy9, pp349-359, 1993.

[3] Sankai Y.: FES Controller with Artificial CPG for Biological Systems, IFES95, pp267–270, 1995.

[4] Sankai Y.: Gait Control Method of Robot and Bio– logical Systems by use of HOJO-Brain, RoManSy11, (in press), 1996.

[5] Sankai Y., Shirai M., Yasuhara K.:Neural Based Sensing and Control for One-Legged Robot, JASME ICAM'93, ADVANCED MECHATRONICS, pp924-930, 1993.

[6] Sankai Y., Sugiura H., Satake T.:On a Basic Concept of Neural Network control of One-Legged Robot, Proc.of JRS, Vol.9(2), pp641–642, 1991.

[7] Guyton: Textbook of Medical Physiology, Igaku Shoin / Saunders, 628-629, 1981.

Intelligent Control with New Imaging Processing Strategy for a Mobile Vehicle

Masanori Sugisaka*, Xin Wang*, Ju-Jang Lee**

*Department of Electrical and Electronic Engineering, Oita University, 700 Dannoharu, Oita 870-11 Japan

**Korea Advanced Institute of Science and Technology, Taejon 305-701 Korea

Abstract In this paper, an intelligent Control with a new imaging processing stagety which is used to decide the control inputs to the fuzzy controller, is presented. This method enabled the mobile vehicle to idetify the position change of specified guideline types better and improved the running result of the mobile vehicle.

Keywords Mobile Vehicle, Imaging Processing, Fuzzy Controller.

1. INTRODUCTION

The intelligent control problems have been researched by many people[1][2][3][4]. These researches have dealt with the motion planners and smoothing trajectories and so on. The intelligent control systems are defined as having an ability to emulate human capabilities, such as planning, learning and adaptation. The intelligent control technology is in a brilliant prospect in the next-generation automobiles, for instance enhancing driver's safety, reducing the driving stress, traction control, autonomous operation, antiskid braking systems, and noise compensation control etc. [5]. Other applications like Intelligent Highways Systems(IHS) are also being developed. Most industrial controls are nonlinear dynamical processes. So it is given an additional complexity to conventional controller because of the well-known modeling procedure.

The fuzzy logic control(FLC), which is one of the major developments of fuzzy set theory and was primarily designed to represent the knowledge that is hardly to be expressed by quantitative measures, has proven effective for complex, nonlinear and imprecisely defined processes for which standard model-based control techniques are impractical or impossible.

This paper present a control strategy for a mobile vehicle which is based on the fuzzy control theories from the practical viewpoint.

The controlled plant in this paper has a structure like a real car[6], is called an Intelligent Mobile Vehicle(MV). Our purpose is make the MV run along with a guideline on the road on the basis of the image information. The image information which is captured by a CCD camera and its relevant processor is divided into four parts, according to the distance of the guideline on the road in front of the MV. The reference position is located at center of the MV.

2. SATEMENT OF THE CONTROL PROBLEM

The system of the MV is a simplified model of the constraints on the movement of a real car(see Figure 1). We 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. We denoted by **v** the speed of the driving wheels, and by ϕ the angle between the front wheels and the main direction of the MV. Thus the motion of the MV can be described by the following model[1].

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos\theta \\ sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1/d \end{pmatrix} v \cdot tan\phi \quad (1)$$

with

$$| v | \leq v_{max}, \ | \phi | \leq \phi_{max}.$$

where v is the magnitude of the speed **v** and 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.

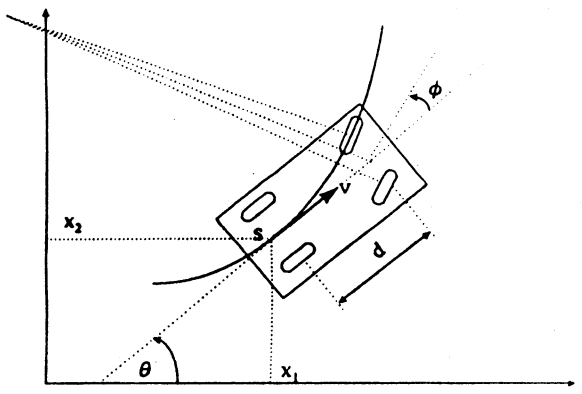


Figure 1: The Running Model

This system can be viewed as a nonlinear system where controls are v and ϕ. The constraint on the turning radius is expressed by $\mid \phi \mid \leq \phi_{max}$, where ϕ_{max} is strictly positive real. To the MV $\phi_{max} = \pi/5$, the velocity of the MV is upper bounded by v_{max}=2.0m/s.

The constraint on the control must hold for all t in the domain of definition. Laumond and Joacobs[1] have proven such a system is controllable. Let us consider a reference trajectory of the vehicle motion:

$$q^d(t) = col\{x_1^d(t), x_2^d(t), \theta^d(t)\} \in D \qquad (2)$$

where D is a domain of possible configurations of the MV. We assume that the vehicle state vector $q(t)$=col$\{x_1(t), x_2(t), \theta(t)\}$ is not belong to the reference trajectory(2). Then the running problem is equivalent to drive the MV from an state $q(t)$ into a desired state(location) $q^d(t)$ with the control inputs v and ϕ. For the clarity of exposition, we concentrate an forward running in this paper and assume v>0 during the running process.

3. CONTROL SYSTEM

There are many methods to practice from the state $q(t)$ to the state $q^d(t)$. The fuzzy control is used in this paper. The main idea of fuzzy system or fuzzy controller is to imitate the human reasoning process to ill-defined plants, and is able to realize the imprecision of the reasoning process without using exact quantitative analysis.

3.1 FUZZY LOGIC CONTROL

The principle of fuzzy logic systems is to express human knowledge in the form of linguistic **if** $\cdots$ **then(condition/action)** rules [7][8]. The common form is

$$Rule_i : if(x_1 = A_1^i, x_2 = A_2^i \cdots, x_l = A_l^i) \text{ then } (y_1 = B_1^i, y_2 = B_2^i, \cdots, y_m = B_m^i)$$

where $x_j \in U_p$ and $y_k \in V_q$(j=1, 2, $\cdots$, l; k=1, 2, $\cdots$, m) are input and output variables, and A_j^i and B_k^i are their actual values for the ith rule respectively, and i=1, 2, $\cdots$, n. These value generally depict the variables by linguistic terms, which are represented by fuzzy sets defined on the corresponding universes of discourse U_p and V_q, such as *positive big*(PB), *positive medium*(PM), *positive small*(PS), *zero*(ZE), *negative small*(NS), *negative medium*(NM), *negative big*(NB) etc.. It is common to use a membership function of a fuzzy set to represent the set itself. For example, μ_F represents the membership function of a fuzzy set F. The design of a TISO (two inputs and single output) controller will be based on a number of fuzzy rules.

$Rule_i$: if x_1 is A_1^i and x_2 is A_2^i then y is B^i (i=1, 2, 3,$\cdots$, n)

and

$$R_i=(A_1^i \times A_2^i \times B^i)$$

If there are n linguistic rules in a system, the fuzzy relation

$$R = R_1 \bigcup R_2 \bigcup \cdots \bigcup R_n = \bigcup_{i=1}^{n} R_i$$

and the fuzzy reasoning result

$$B^p = \mathrm{R} \circ (A_1^i \times A_2^i)$$

where the symbol $\circ$ is a composition rule. Simple arguments show

$$\mu_{B^p}(y) = \max_{x_1, x_2} \left[\mu_R(x_1, x_2, y) \wedge \mu_{A_1^i \times A_2^i}(x_1, x_2)\right]$$

Using the center of gravity method, the output of fuzzy controller y^p

$$y^p = \frac{\sum_{i=1}^{n} y^i \mu_{B^i}(y)}{\sum_{i=1}^{n} \mu_{B^i}(y)} \qquad (3)$$

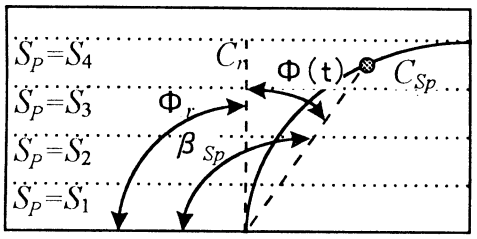


Figure 2: The image information

3.2 CONTROLLER DESIGN

Firstly, the real guideline have to be converted into available image information, and then the image information which is captured by a CCD camera and the relevant processor is divided into four parts $S_p(p = 1, 2, 3, 4)$ according to the distance of the guideline on the road in front of the MV (see Figure 2). The center of gravity $C_{S_p}(C_{S_p} \in S_p)$ is abstracted independently. The variables of the fuzzy controller, having a two-input and two-output, are chosen as follows:

(a). Let the angle of C_{S_p} be $\beta_{S_p}(\geq 0)$, and the variable $\Phi_\beta(t)$ has the following releation

$$\Phi_\beta(t) = \bigvee_{p=1}^{4}(\beta_{s_p} \wedge \bar{\beta}_{s_{p+1}} \wedge \bar{\beta}_{s_{p+2}} \cdots \wedge \bar{\beta}_{s_{p+q}}) \quad (4)$$

$$(p + q \leq 4, q = 0, 1, 2, 3, 4)$$

with

$$\beta_{s_{p+q}} \cdot \bar{\beta}_{s_{p+q}} = 0, \quad (5)$$

$$\bar{\beta}_{s_{p+q}} + \beta_{s_{p+q}} = max\{\beta_{s_{p+q}}, \bar{\beta}_{s_{p+q}}\}. \quad (6)$$

$$\bar{\beta}_{s_{p+q}}|_{\beta_{s_{p+q}}=0} > max\{\beta_{s_{p+j}}\}. \quad (7)$$

$$(j = 0, 1, 2, 3, 4; j \leq q)$$

One of the two inputs $\Phi(t)$ is the angle between the center of gravity of the guideline and reference position.

$$\Phi(t) = \Phi_\beta(t) - \Phi_r \quad (8)$$

t: time;
Φ_r: reference angle.

(b). The other input is the rate of the errors $\Delta\Phi(t)$ between the interval t-Δt and t

$$\Delta\Phi(t) = \Phi(t) - \Phi(t - \Delta t)$$

$$= \Phi_\beta(t) - \Phi_\beta(t - \Delta t) \quad (9)$$

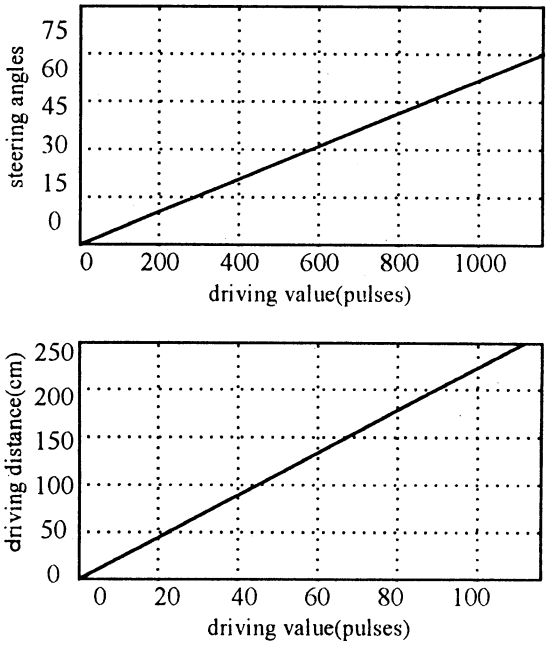


Figure 3: The feature of main units

(c). The outputs of the fuzzy controller are the two control variable, the steering angle ϕ and speed v.

It should be noted that the shape of the spacial object will continuously change following image processing [9]. The role of the fuzzy controller is therefore to identify these changes and make the correct decision on both the control variables, the driving speed v and angle of steering ϕ.

4. EXPERIMENT

4.1 THE FEATURE OF THE MAIN UNITS

The driving motor is controlled by means of Pluse Width Modulated (PWM). The principle of PWM amplifier is to convert a DC voltage into rectangle voltage to input into a DC motor by controlling the width of the rectangle voltage to adjust the average voltage of the motor. A DC stepping motor is used in steering. Both of the feature of the units are shown in Figure 3.

4.2 EXPERIMENT RESULT

The MV consists of a microcomputer NEC PC98 21Ap, some I/O boards including a video interface SUPER CVI. The MV's two rear wheels can

rotate forwards and backwards about the axle shaft and are driven by a 24V DC motor using cross helical gears. The MV's front wheels are driven by a 24V DC stepping motor for motion to the left or right, with transmission being via gear wheels. Two 6V DC motors are employed to rotate the CCD camera in the horizontal and vertical directions. The location of the front wheels and camera are detected by three encoders. The guideline type is red with 50mm in width, and include direct lines, curves and urgent turnings. The experiment are shown in Figure 4. The minimum speed is not less than 0.5m/s.The maximum speed of the MV is about 2.0m/s.

5. CONCLUSIONS

A fuzzy controller has been successfully applied to enable a mobile vehicle to follow specified guideline type runs. However, in order to accomplish more smooth running, the interruption control has to be installed in addition to the fuzzy controller. This control is now under consideration. New technologies will also be applied to the MV control system to enable the mobile vehicle to recognize and track moving objects and avoid obstacles using advanced intelligence.

6. REFERENCE

[1] J. P. Laumand, "A Motion Planner for Nonholomic Mobile Robots", *IEEE Trans. on Robotics and Automation*, 10(5):577-593(1994).

[2] S. Fleury, "Primitives for Smoothing Mobile Robot Trajectories", *IEEE Trans. on Robotics and Automation*, 11(3):441-448(1995).

[3] Y. M. Enab, "Intelligent controller design for the ship steering problem", *IEE Proc. Control Theory Appl.* 143(1):17-24(1996).

[4] D. Gorinevsky, A. Kapitanovsky, And A. Goldengerg, Neural Network Architecture For Trajectory Generation And Control Of Automated Car Parking, *IEEE Trans. On Control System Tech.* 4(1): (1996).

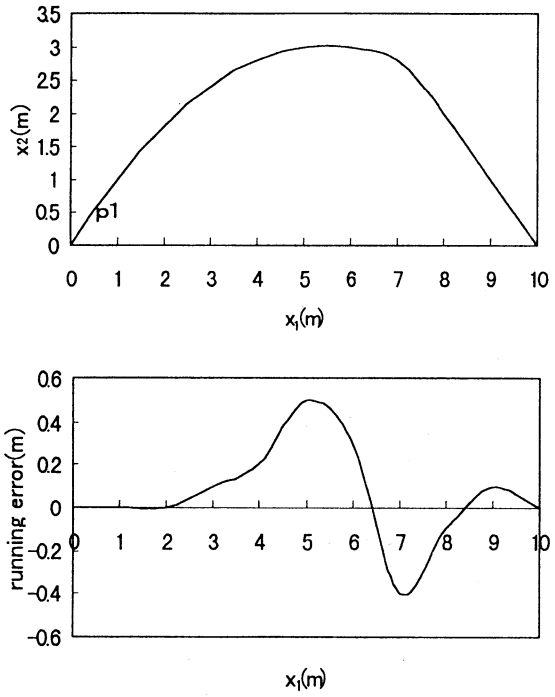


Figure 4: The running road and result of the MV

[5] D. A. Linkens, H. O. Nyongesa,: Learning System In Intelligent Control: An Appraisal Of Fuzzy, Neural And Genetic Algorithm Control Application, *IEE Proc. Control Theory Appl.*, 143(4), 367-386(1996).

[6] M. Sugisaka, X. Wang, "The Development of a Practical Control Method for an Intelligent Mobile Vehicle", *Proc. of International Symposium on Artificial Life and Robotics*, 234-237(1996).

[7] M. Sugeno, *Fuzzy Control*, Nikkankougyo Sinnbunshya, in Japanese, 74-84,1989.

[8] M. Madan, K. Rommohan , R. Ronald , *Advances in Fuzzy Set Theory and Applications*, North-Holland Publishing Company, 1990.

[9] Y. Anzai, *Pattern Recognition and Machine Learning*, Academic Press, 1992.

NEURAL NETROKS FOR CONTROL IN AN ARTIFICIAL BRAIN OF RECOGNITION AND TRACKING SYSTEM

Masanori Sugisaka[1], Noboru Tonoya[1,2], and Toshiyuki Furuta[3]

[1]Department of Electrical Engineering
Oita University, Oita, 870-11 Japan
e-mail msugi@cc.oita-u.ac.jp, fax 001-81-975-7820, tel 001-81-975-54-7831
[1,2]Presently, Fujitsu, Ltd.
[3]Research and Development Center CCRG, RICOH CO. Ltd, Yokohama 223, Japan

Abstract

This paper presents a new information processing machine which is called artificial brain(ABrain) and considers the structure of artificial neural networks constructed in a RICOH neurocomputer RN-2000 in the ABrain, in order to track given trajectories which are produced in a microcomputer or a light moved by hand in a recognition and tracking system.

1. Introduction

We have already considered the neural network structure for controlling the duty ratios of DC motors in X(horizontal) and Y(vertical) axes in the recognition and tracking system of a moving object[1],[2]. The neural networks used in the control studies were constructed in a RICOH neurocomputer RN-2000[3],[4]. It was verified that both the input and the structure of the artificial neural network in RN-2000 changes according to the information processed from sensors. For the purpose of tracking a moving object, the input to the neural network is the error betweeen the moving object and the center of the CCD camera[1]. On the other hand in order to recognize the shape of the moving object, the input to the neural network is the moment invariants of the image of the moving object from the CCD camera[2]. Based on this observation, we will propose a new information processing machine which is called artificial brain(ABrain) constructed by both hardware and software.

In this paper at first we forcus our attention on the investigation of the structure of the artificial neural networks constructed in the primitive ABrain for generating control signals in the recognition and tracking system. We show the experimental results for tracking the moving object by changing the structure of the neural network in RN-2000 which is used to learn a propotional control law. Secondly we briefly discuss the general structure of the ABrain. Finally we explain that the general ABrain consists of mainly the ABrain for recognition, the ABrain for thinking, and the ABrain for action or behavior

2. Neural Networks for Tracking

We discuss which structure of the neural networks in the primitive ABrain, which consists of a Von-Neumann computer, one neurocomputer RN-2000, their interface, and the recognition and tracking system shown in Fig.1, is best from a point of view of control performance. By changing the structure of the neural network, more precisely, the number of neurons in both input and output layers, the control performances of tracking both a desired value produced by the Von-Neumann computer and an object(light) moved by hand are

compared in order to decide the best structure of the neural network in the ABrain.

The RICOH neurocomputer RN-2000 in Fig.1 has 4 layers where each layer has sixteen neurons[3]. As the input to the neuron in the input layer, both the integer ranged from 0 to 127 and two digits(0 and 127) are considered. Therefore, we constructed three types of neural networks shown in Fig.2.

The propotional control law between the error and the duty ratio for X and Y axes DC motors is learned by the neural networks shown above. The input values obtained by the bit transformation technique[5],[6] are used for the neural network 1. The neural network 1 is used at first for the X axis DC motor and secondly for the Y axis motor. For this case, the neural network 1 is not able to predict the output values which are not learned. It is necessary for the neural network to learn the whole data. The resut is shown in Fig.3a.

The input values for the neural network 2 is integer ranged from 0 to 127. For this case, the neural network 2 is able to predict the output values which are not learned. It is not necessary for the neural network 2 to learn the whole data. Therefore only two pair of the input and output values are used as the learning data. The result is shown in Fig.3b.

The input values for the neural network 3 is integer ranged from 0 to 127. The neural network 3 produces the duty ratios for the X and Y axes DC motors simultaneously so that there are two neurons in the input and output layers. The neural network 3 has the same prediction ability as the neural network 2. Therefore, four pair of the input and output values are used as the learning data. The result is shown in Fig.3c.

3. Structure of ABrain

The configulation of an ABrain for robots is shown in Fig.4. This configulation has a general structure.

The ABrain consists of

1. various sensors which receive different information,

2. a self-organizing circuit which organizes the structure of artificial neural networks and numbers of both neurons and neurocomputers in the ABrain,

3. several parallel neurocomputers and Von-Neumann computer,

4. a circuit for generating internal performance criterion in order to optimize the total neural network systems including the parallel neurocomputers and a Von-Neumann computer,

5. and decision circuit which determines control actions to drivers or actuaters.

We showed the general structure of the ABrain in Fig.4. The functions of information processing in the ABrain based on our former experimental results are embedded in the hardware configulation in Fig.4.

The ABrain for the recognizing an object is called an ABrain for recognition and consists of a neural network where the inputs to the neurons in the input layer are feautures(eg, moment invariants) of the object. The ABrain for thinking how behave for it is called an ABrain for thinking and consists of a neural network where the inputs to the neurons in the input layer are genetic information and/or knowledge obtained from the experiment and learning. The ABrain for deciding an action or a behaivor is called an ABrain for action or behavior. It consists of a neural network where the input to the neurons in the input layer is the difference or error between the present state and the desired state.

The three ABrains stated above constitute the main part of the ABrain shown in Fig.4. The information received from a sensor is processed in the three ABrains(recognition, thinking, action or behavior) sequentially or in parallel. The design of such a sophisticated ABrain stated above is now under our investigation.

4. Conclusions

In this paper we proposed a new information processing machine which is called the ABrain. We have already developed a primitive ABrain for tracking and recognition[1],[2]. Based on both the results[1].[2] and the experimental results given in this papers, we obtained the guideline of designing more sophisticiated ABrain than the primitive ABrain developed by our group. We will apply the ABrain which consists of the hardware and software developed in this study for various engineering fields.

References

[1] M. Sugisaka, "Neurocomputer control for tracking of moving objects," *Proc. of Int. Symp. on Artificial Life and Robotics*, 243-247, 1996.

[2] M. Sugisaka, "Pattern recognition using neurocomputer," *Proc. of Int. Symp. on Artificial Life and Robotics*,248-251, 1996.

[3] H. Eguchi, T. Furuta, H. Horiguchi and S. Oteki "Neural network hardware with learning function utilizing pulse-density modulation", *Trans. of the Institute of Electronics, Information and Communication Engineers, Vol.J74C-II*, **5**, 369-376, 1991.

[4] S. Oteki, A. Hashimoto, T. Furuta, S. Motomura, T. Watanabe, D. G. Stork and E. Eguchi, "A digital neural network VLSI with on-chip learning using stochastic pulse encoding", *Proc. of 1993 Int. Joint Con. on Neural Networks*, 3039-3045, 1993.

[5] M. Sugisaka, S. Motomura, T. Kitaguchi, and H. Eguchi, "Hardware Based Neural Identification:Linear Dynamical Systems", *Proc. of Int. Workshop on Intelligent Systems and Innovative Computation-The 6th Bellman Continuum-*, 124-131, 1994

[6] M. Sugisaka, "Hardware based neural identification: nonlinear dynamical systems", *Proc. of the 2nd Asia-Pacific Conf. on Control and Measurement, Wuhan-Chongqing, P. R. China*, 255-259, 1995.

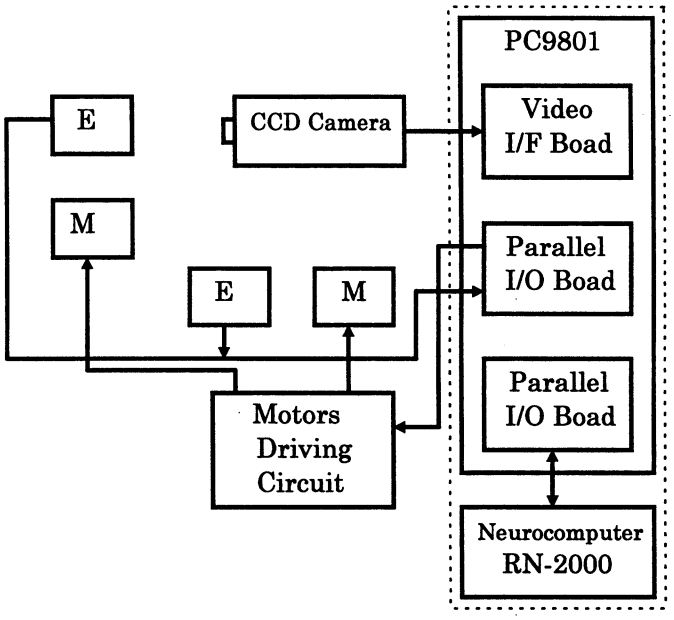
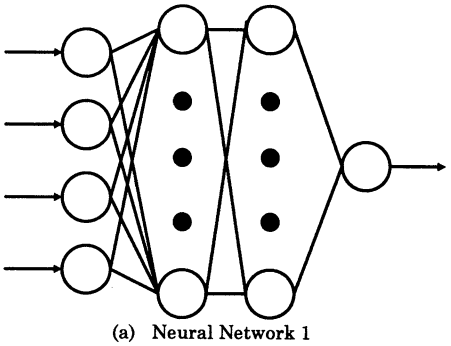
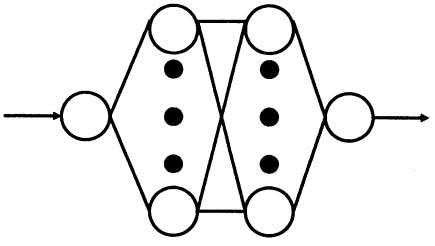


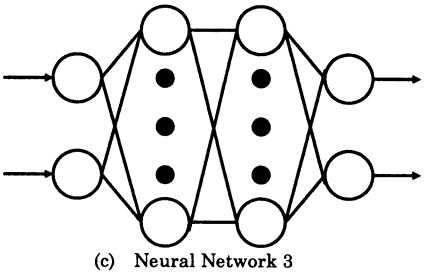
Fig.1 Recognition and Tracking System with ABrain



(a) Neural Network 1



(b) Neural Nerwork 2



(c) Neural Network 3

Fig.2 Neural Networks in ABrain

number of learning : 2378 (times/pattern)
time of learning : 1min 7sec
meanvalue of errors : 1.64

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

Fig.3a Outputs of Neural Network 1

number of learning : 84 (times/pattern)
time of learning : 2sec
meanvalue of errors : 1.5

Errors		Teaching signals		Learned
True	Normalized	True	Normalized	results
0	0	0	0	0
10	127	100	127	121

Input	0	1	2	3	4	5	6	7	8	9	10
Output	0	3	12	28	35	52	59	75	86	92	94

Fig.3b Outputs of Neural Network 2

number of learning : 322 (times/pattern)
time of learning : 4sec
meanvalue of errors : 1.62

Errors				Teaching signals				Learned	
True		Normalized		True		Normalized		results	
X	Y	X	Y	X	Y	X	Y	X	Y
0	0	0	0	0	0	0	0	0	0
0	10	0	127	0	100	0	127	3	127
10	0	127	0	100	0	127	0	123	6
10	10	127	127	100	100	127	127	126	123

X	Input	0	1	2	3	4	5	6	7	8	9	10
	Output	0	8	17	29	39	49	59	68	79	88	94
Y	Input	0	1	2	3	4	5	6	7	8	9	10
	Output	0	9	19	29	38	51	59	71	79	91	96

Fig.3c Outputs of Neural Network 3

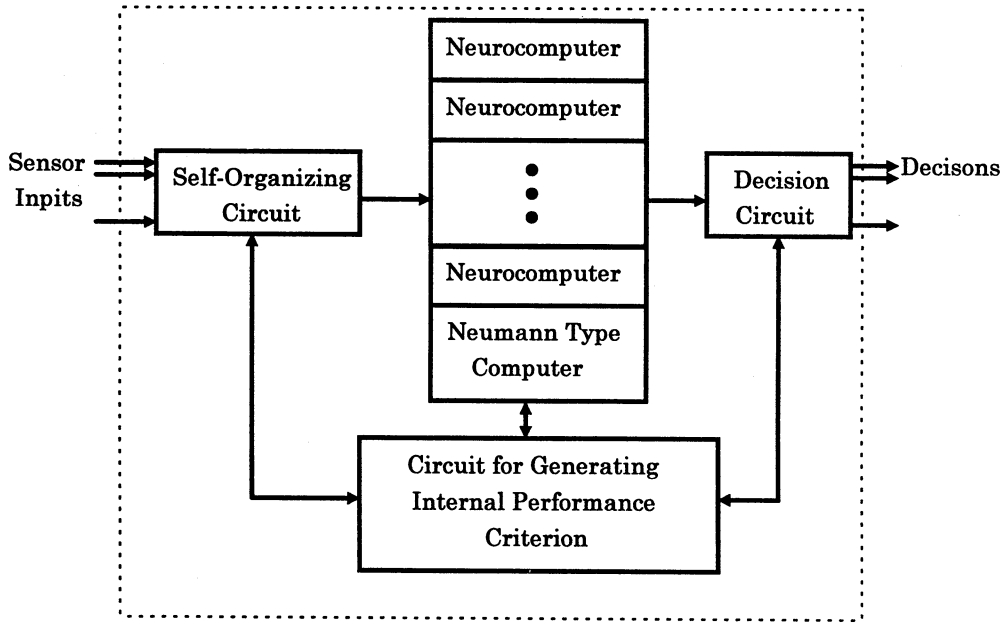


Fig. 4 Configulation of Artificial Brain

Learning Model for Adaptive Behaviors as an Organized Group of Swarm Robots

Keiki Takadama
ATR Human Information Processing Research Labs. / University of Tokyo

Koichiro Hajiri
ATR Media Integration & Communications Research Labs. / Ritsumeikan University

Tatsuya Nomura
ATR Human Information Processing Research Labs.

Michio Okada
ATR Media Integration & Communications Research Labs.

Katsunori Shimohara
ATR Human Information Processing Research Labs.

2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan, Tel:+81-774-95-1093 Fax:+81-774-95-1008
Email: takadama@hip.atr.co.jp/keiki@ai.rcast.u-tokyo.ac.jp

Abstract

This paper proposes a learning model in which individual robots generate and acquire adaptive behaviors as an organized group of swarm robots from local interactions between them. In this paper, we focus upon emergent processes in which swarm robots dynamically form an organized group, and autonomously generate adaptive behaviors without an outside control, rather than control the collective behaviors of the robots. A simulation of a task which is to combine beams to construct a truss in space, using swarm robots, shows not only that individuals can adapt themselves to the group, but also that an organized structure can emerge to solve the task effectively and efficiently. As a result that the swarm robots in our model self-organize their own adaptive behaviors, we can find that roles in robots emerge according to the number of robots and that the robots do not need as many time steps to construct a truss as those with a global function.

Keywords: adaptive behavior, emergence, organizational learning, self-organization, evolution, swarm robots

1 Introduction

In decentralized approaches, it is important for swarm robots to have some mechanism for self-evaluating their global group behaviors. The easiest way to evaluate actions is to employ a global function as an outside control. But, in order to get the result of the global function, the robots must communicate with a control system such as a space station, or must broadcast information to each other. This makes communication costs high. Furthermore, it is also difficult to determine an appropriate global function in general, *e.g.*, the global function which is different according to the number of robots. Therefore, in our model, we employ only local functions, instead of a global function. This differs from the usual practice of employing a global function for group behavior as used in previous studies on swarm robots.

Rather than controlling swarm robots from the outside, this paper discusses whether it is possible for them to autonomously control their own global behaviors based on local functions in the process of adapting to the group. Therefore, we propose a learning model in which individual robots autonomously learn adaptive behaviors with only local functions, and we show the effect of introducing this learning model.

2 Model for Adaptive Behaviors

Our learning model –*Organizational Learning*– for adaptive behaviors with only local functions has two mechanisms: a "*Behavior Generating Mechanism*" and a "*Role Self-organizing Mechanism*".

2.1 Organizational Learning

When human beings self-evaluate their contribution to collaborative works, they do not use a global function, because they cannot know all of the information and cannot determine an appropriate global function. Therefore, human beings use some degree of adaptation to the group with only local functions. In the process of adaptation, they learn effective roles as global group behaviors. In this paper, we define this kind of learning for adaptation as "*Organizational Learning*", and we introduce this learning to swarm robots.

2.2 Behavior Generating Mechanism

In our model, robot behavior generating mechanisms (BGM) can generate roles as a sequence of actions, due to local interactions between swarm robots. The robots behave according to their roles, and adapt themselves to a group by repeating such interactions.

As shown in Fig. 1, each robot has a Classifier System (CS) [3], which is a rule-based machine learning system. In CS, each *if-then* rule named Classifier(CF) is composed of three parts: Condition Part, Action Part and Strength Part. The condition part corresponds to the *if* clause, and usually contains the present robot action such as A,B and other input information such as 0,1. # indicates don't care. The action part corresponds to the *then* clause, and indicates the

next robot action. Finally, the strength part indicates the strength of the CF, which affects the behaviors of the robots.

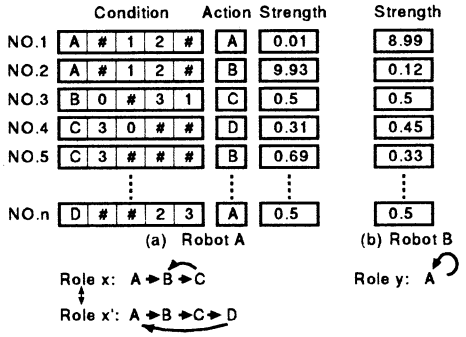


Figure 1: Behavior Generating Mechanism

Each robot has the same CF except for the strength value, which is changed in the process of adapting to the group. For every adaptation, BGM generates roles as a sequence of actions. For example, robot A acquires role x as shown in Fig. 1(a) which indicates the action of A,B,C,B,C,$\cdots$, and robot B acquires role y as shown in Fig. 1(b) which indicates the continuation of action such as A,A,$\cdots$. If a strength value of No. 4 in robot A is higher than 0.69, role x' emerges instead of role x, because a No. 4 CF has a higher probability of being selected than a No. 5 CF. The difference between role x and x' is whether a robot select to action of B again or D after C, and it is caused as a result of local interactions between the robots.

2.3 Role Self-organizing Mechanism

The generated roles in BGM affect the collective behaviors as a group, and then promotes the dynamical re-organization of the group. In our model, the role self-organizing mechanism (RSM) enables a group to re-allocate the roles of its members, and to form a new structure for achieving a given goal.

We propose RSM which is an extension of the "*profit sharing*" in CS, in which the mechanism distributes rewards to all of the CFs used to achieve a given goal. In profit sharing, rewards are not distributed to CFs until the robots achieve the goal and the reward is only a plus. In our model, RSM distributes rewards to only the CFs which are used to achieve the goal, and which win in a roulette selection. In other words, even if the CF is used to achieve the goal, rewards are not distributed to a CF which does not compete with other CFs, *i.e.*, which is only one CF matched to the condition.

Furthermore, RSM also distributes rewards to CFs in the case of failure. An equation in the case of success indicates the reinforcement of all CFs in the same way as profit sharing. On the other hand, equations in the case of failure indicate the reinforcement of CFs except for the last CF which is weakened. These equalities are formed by following the manner of human beings. That is to say, in the case of failure, human beings do not carry out all of the actions which they select, but they try to carry out almost all actions again with a few other actions especially in the last sequence of actions. We find this tendency to be strong in the first several failures. In addition to behaviors according to this tendency, RSM has another mechanism which reflects the way that human beings sometime change to behave with other actions not in the last sequence of actions. This mechanism works because RSM selects a CF with probability.

- **Success: (Can achieve a goal)**
  $ST(x) = ST(x) + 1, \quad where\ x = 1, 2, \cdots, n$
- **Failure: (Cannot achieve a goal)**
  $ST(x) = ST(x) + 1, \quad where\ x = 1, 2, \cdots, n-1$
  $ST(x) = ST(x) - 2, \quad where\ x = n$

ST is the strength of the CF and is set in the order of contribution to achieving a goal, n is the max number of CFs which compete with other CFs. By continuing to fail when CF(n) is used, $ST(n')$ whose CF(n') compete with CF(n) becomes bigger than $ST(n)$. As a result, CF(n') is selected instead of CF(n).

3 Swarm Robots in a Space

3.1 Example

We assume tasks which require swarm robots to cooperate for accomplishment in a space through radio communications. In this example, we assume the task as construction of a truss by combining beams as quickly as possible. In order to accomplish this task, three robots must cooperate: two robots each hold a beam and set the exact angle between the two beams, and another robot welds the two beams. All robots are given the same tasks, and know the order of setting a beam or welding it from the left to the right side in order to avoid unnecessary actions. There are other assumptions as follows.

- The robots have the necessary functions to lift a beam and weld it, and can also move or rotate.
- In order to avoid to bump into other robots, the robots take into consideration their width and the length of the beam when they move.
- Each robot can communicate with other robots within its scope.
- The robots do not know their effective roles beforehand.

3.2 Robots

As shown in Fig. 2, we can distinguish between two types of robots. The robot shown in Fig. 2(a)

has a beam and goes to the beam location. The robot shown in Fig. 2(b), on the other hand, has nothing and goes to the welding location or returns to the station. All robots can hold a beam at the station and release it at the beam location and can also weld two beams at the welding location. The robots never release a beam until it is welded, and they cannot cross a beam over other robots. The robots can also acquire the state and location information about beams and welding within the scope of the LOOK radius. Furthermore, the robots can update this information by communicating with other robots which are observed within the scope of the LOOK radius.

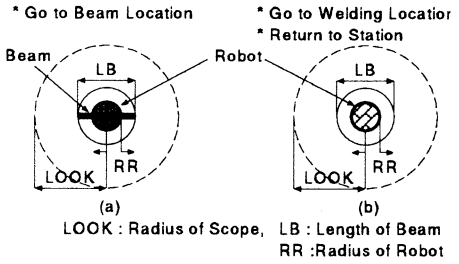


Figure 2: ROBOT

Robots must learn to acquire adaptive behaviors especially at the station and at the welding location after the two beams are welded. At the former place, the robots learn to select whether they hold the beam or go to the welding location, and at the latter place, the robots learn to select whether they go to another welding location or return to the station.

3.3 Global and Local Functions

When robots are able to use a global function, the station knows the newest WB (Waiting for a Beam) and WW (Waiting for welding) through communication between the station and the robots. WB is the number of robots which stay to wait for beams to be welded, and WW is the number of robots which stay to wait for the welding. We do not count the WB or WW in robots which are described with an arrow indicated moving as shown in Fig. 3. The robots can always achieve their task because they know the newest information from the station, but communication costs are high. When robots use only local functions, on the other hand, they acquire a new role or change their own role respectively with RSM, where x is the number of robots.

- **Global Function (GF):**
  $GF := WB - WW$
- **Local Function (LF):**
  $LF(x) := RSM(x)$

Using a global function at the beam and at the welding location, the robots continue to act successfully during $GF < 1$ as shown in Fig. 3(a) (WB=1 WW=4). When $GF = 1$, the robots which go to the welding location must change to return to the station. This is because robots with beams cannot go to the beam location when $GF > 1$, as another welding robots stay to wait for beams in the course to the beam location as shown in Fig. 3(b) (WB=3 WW=1). At the station, the robots must hold the beams when $GF \geq 1$, because there are many welding robots in the welding location. On the other hand, the robots must go to the welding location when $GF < 1$, because there are many robots with beams in the beam location.

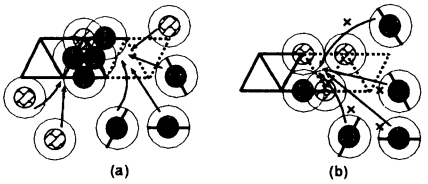


Figure 3: Construct a Truss

Using only local functions, the robots act according to the RSM local functions. The result of the RSM is different between robots, because the sequence of actions is different for each robot.

4 Simulation

4.1 Experiments

We performed experiments to compare the following four cases in terms of time steps and effective roles as adaptive behaviors. In these experiments, we assume the time steps to be the steps which are needed to achieve a task, and the enforcement counts to be the number of repeating tries from the beginning when the robots are not able to achieve a task. All of the robots are given the same task, which is to construct a truss by combing 13 beams.

- Case 1: 5 Robots (Using Global Function)
- Case 2: 5 Robots (Using Local Function)
- Case 3: 10 Robots (Using Global Function)
- Case 4: 10 Robots (Using Local Function)

4.2 Results

In the experiments using a global function as an outside control, robots were consistently able to achieve the task because all of the robots knew the newest information from the station. But, the time steps did not decrease as the enforcement count increased because the global function was not changed. The robots did not acquire roles because they behave according to the global function. As shown in Fig. 4, it took 262 steps with 5 robots and 150 steps with 10 robots to achieve the task. In this figure, the x-axis shows the enforcement count and the y-axis shows the time steps for archiving the goal.

In the experiments with only local functions, the robots were not able to achieve the task with the first several enforcement counts, but they were able to achieve it as the count increased. This was because each robot adapted itself to the group by acquiring a new role or changing its own role. As shown in Fig. 4, it took 249 steps with 5 robots and 132 steps with 10 robots to finally achieve the task. The time steps with both 5 and 10 robots became less than the time steps with the global function.

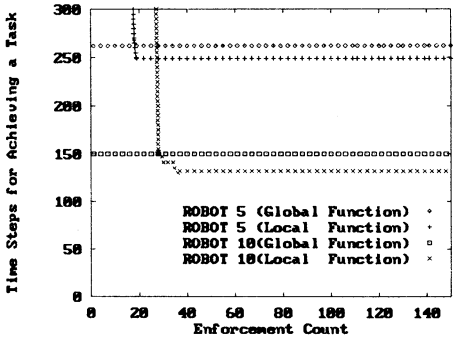


Figure 4: Time Steps for Achieving a Goal

With local functions, a lot of roles emerged according to the number of robots because the rules (CF) were changed as the robots can adapt to the group. As shown in Table 1, we were able to find two types of roles with 5 robots and four types of roles with 10 robots. With 5 robots, some robots carried beams from the station to the beam location, and other robots returned to the station after they welded a beam. With 10 robots, we found more complex roles than with 5 robots. It was interesting for the 10 robots to acquire organizational behaviors in which only one robot went to the welding location and the remaining 9 robots each went to the beam location with a beam.

Table 1: Roles

5 robots		10 robots	
number	role	number	role
3 robots	B → S	1 robot	W → S → W
2 robots	W → S	4 robots	B → S
		3 robots	B → W
		2 robots	B → S → W

S : Return to Station, B : Go to Beam Location
W: Go to Welding Location

4.3 Discussion

Using our learning model for adaptive behaviors without an outside control, the following observations can be made from the experimental results.

- Robots evolve to extend their possibility of adaptive behaviors by continuing to acquire a new role or to change their own role through adaptation to the group.
- The roles of the robots emerge according to the number of robots. As a result, an organized group which solve a task effectively emerges.
- In the case of a task for which a global function can not be appropriately determined, the robots in our model were able to perform to achieve a task with less time steps than a model using a global function.

5 Conclusion

This paper has proposed a learning model in which individual robots autonomously learn adaptive behaviors with only local functions. We have discussed the effect of our model by comparing the time steps with a model using a global function as an outside control and by investigating the acquired roles. As a result, our model shows an effect for the problems in which a global function can not be appropriately determined. Our model also enables swarm robots to evolve to extend their possibility of adaptive behaviors by continuing to acquire a new role or to change their own role through adaptation to the group. The following research must be done in the future.

- Determine the effect of our model for adaptive behaviors with more robots than those using in this research.
- Determine the effect of our model in group reconstruction when new robots are added or broken robots are removed.
- Acquire optimal adaptive behaviors.
- Determine the effect of our model without local communications.

References

[1] Aiba H., Terano T.: "*A Computational Model for Distributed Knowledge Systems with Learning Mechanism*" Expert Systems with Applications, 10(3/4), pp. 417-427, 1996.

[2] Bull L., Fogarty, T. C., Snaith M.: "*Evolution in Multi-agent Systems: Evolving Communicating Classifier Systems for Gait in a Quadrupedal Robot*", ICGA 95, pp. 382-388, 1995.

[3] Holland, J. H.: "*Properties of the Bucket Brigade Algorithm*", ICGA 85, pp. 1-7, 1985.

[4] Maeda Y.: "*Evolutionary Algorithm for Behavior Learning of Multi Agent Robots*", International Fuzzy Systems and Intelligent Control Conference, pp. 360-367, 1996.

[5] Takadama K.: "*A Study on Self-Generation of Individuality in an Artificial Organism Population*", Alife V, Poster Presentations, pp. 109-116, 1996.

[6] Takadama K., Nakasuka S.: "*Emergent Strategy for Adaptive Behaviors with Self-Organizational Individuality*", SEAL'96, pp. 161-168, 1996.

Hippocampal Neural Network Model Performing Navigation by Homing Vector Field Adhesion to Sensor Map

M. Matsuoka, S. Hosogi, Y. Maeda
Fujitsu Laboratories LTD.
140 Miyamoto, Numazu-shi,Shizuoka, 410-03, Japan

Abstract

We propose an artificial neural network models for autonomous agents, i.e., mobile robots, to learn maps of environments and acquire the ability to perform home-navigation autonomously. The networks consists of two sub-networks, each of which has a similar structure with hippocampal lamellar neuronal circuits. Hebbian learning procedures self-organize the first sub-network to output distributed sinusoidal activity of cells by accumulating motor information generated during movement, and the second sub-network to output localized activity by prototyping sensory information. These patterns represent a homing vector providing relative coordinates of the agent from a starting point, and a place code corresponding uniquely to a point of environment, respectively. By attaching homing vectors to the sensor map, the homing vector is associated with the sensory stimuli. Then the agents can perform home-navigation autonomously by the association.

1 Introduction

Autonomous navigation is the fundamental ability for natural and artificial agents to perform adaptive behaviors in its environments [1]. The aim of this research is to develop basic methods for self-orientation that serves agents with such ability. Self-orientation is a biological mechanism that provides intelligent agents with the ability of adapting to dynamically changing environments and executing autonomous movements. The mechanism is based on neuronal processes that relates an agent itself with environments and evaluates the relations to perform movements. Mammalian hippocampal neuronal networks are considered to have such mechanisms, since lesions of the hippocampus cause disfunction of performing self-orientation movements in navigation tasks[2].

After O'Keefe discovered in rat's hippocampus a "place cell" responding selectively when the rat occupies a specific place in the environment[3], hippocampal models have been proposed to elucidate the neuronal information processing for the ability [2][4][5]. Although some of the models can be utilized as the base of the self-orientation mechanism, they are likely to be refined from the engineering point of view in the context of the engineering application.

In this paper, we have developed an artificial neural network with hippocampus-like structure and biologically plausible Hebbian learning rules. The present model can encode motor and sensor inputs into integrated representations for homing navigation from any arbitrary locations. By using a simulated mobile robot (R3, IS Robotics), we test our hippocampal navigation circuit on homing navigation tasks.

2 The Neural Network Model

The hippocampal circuits form a lamellar structure composed of units which are 4-layered neural networks with one-directional excitatory connections, EC $\rightarrow$ DG $\rightarrow$ CA3 $\rightarrow$ CA1 [6]. The weights between layers are found to have high modifiability with Hebbian learning rule. In this paper we call the unit as an Hippocampal Neural Network (HNN) whose structure is shown in Figure 1. We regard the unit HNN as a basic information processor and the functional role can be decomposed into three parts as follows: an encoding circuit $\mathcal{C}_E$ (EC $\rightarrow$ DG), a representation circuit $\mathcal{C}_R$ (DG $\rightarrow$ CA3), and an integration circuit $\mathcal{C}_I$ (CA3 $\rightarrow$ CA1). Each cell in the layers has the output $f(\vec{x}) = 1/(1+e^{-\alpha(\vec{w}\vec{x}+\beta)})$. Several kind of functional cells, i.e., a gaussian cell, or a bounded linear cell, are obtained by changing weights from inhibitory intervening cells and excitatory self-recurrent connections. In fact, pyramidal cells in the hippocampus receive inhibition from intervening cells and excitation form self-recurrent connections [6].

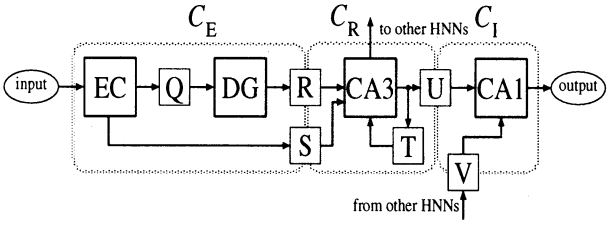


Figure 1. *A Hippocampal Neural Network. The network consists of 4 layers of (functional) cells, EC, DG, CA3 and CA1. The weights of the cells are denoted by Q, R, S, T, U and V.*

Figure 2 shows that our hippocampal navigation circuit is constructed from two units; the HNN^H for calculating homing vectors and the HNN^S for building sensory maps. On the integration circuits $\mathcal{C}_I^H$, an

homing vector is associated with a sensory prototype signal from the $\mathcal{C}_R^S$. By using a structural restriction of HNN and characteristics of inputs, both of the modifiable weights in the $\mathcal{C}_E$ and the $\mathcal{C}_R$ can be learned by Hebbian learning rule.

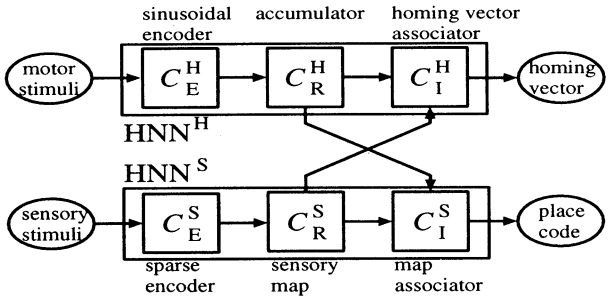


Figure 2. *The hippocampal circuit for navigation.*

3 Learning Rules

3.1 Learning rules for $\mathcal{C}_E$

Let $\vec{W}^{(n)}$ be the output of the EC layer of M cells at time n. Let $\vec{X}^{(n)}$ be the output of the DG layer of N cells at time n. The weights $Q = (Q_{ij})$ from the EC layer to the DG layer are organized by the following rule

$$\Delta Q_{ij}^{(n)} = (-Q_{ij}^{(n)} + a_1 X_i^{(n)} W_j^{(n)}) X_i^{(n)}.$$

As the learning proceeds, the output patterns of the DG become sparse and non-topological even if the output patterns of the EC are dense and ordered topologically.

The weights $S = (S_{ij})$ from the EC layer to the DG layer are learned by the following rule

$$\Delta S_{ij}^{(n)} = a_2(\cos(\Delta\phi^{(n)}) - S_{ij}^{(n)}) X_i^{(n-1)} W_j^{(n)},$$

where $\vec{X}^{(n-1)} = Q\vec{W}^{(n-1)}$ and $\cos(\Delta\phi_n)$ is due to the θ-rhythm [7]. The correlation between the output $\vec{W}^{(n)}$ of the EC layer and the output $\vec{X}^{(n-1)}$ corresponds to the conditional stimuli as in the Grossberg's learning rule [8]. The θ-rhythm signal is compared to the unconditional stimuli. If the θ-rhythm exists, then $S_{ij} \simeq \cos\{2\pi(\frac{i}{N} - \frac{j}{M})\}$ is obtained by the learning rule. If $\cos(\Delta\phi^{(n)}) = 0$, then $S_{ij} \simeq 0$ for any input to the $\mathcal{C}_E$ in the learning.

Let $R = (R_{ij})$ be the weights from the DG layer to the CA3 layer. Let $\vec{Y}^{(n)}$ be the output of the CA3 layer of N cells at time n. In the presence of the θ-rhythm, the weights S are formed to affect not a few on the CA3 activity that is, in result, not determined only by the DG. Therefore the correlation between the DG and the CA3 is declined and then the weights R are not organized to represent prototypes. The absence of the θ-rhythm means that the weights S are null and thus the activation of the CA3 is determined only by the DG. Assume $T = (T_{ij})$ be "Mexican hat" connection weights composed of self-excitations and lateral inhibitions. The weight R_{ij} is modified if the DG's output $X_j^{(n)}$ contributes to the activation of the CA3's output $Y_i^{(n)}$. Then the weights R are organized to represent prototypes of the sensory stimulus [10].

3.2 Learning rule for $\mathcal{C}_R$

The CA3's self-recurrent weights $T = (T_{ij})$ stabilize an output pattern of itself or store the output as a short term memory. The following learning rule is applied when the transformation by the weights T cannot preserve the output pattern of the CA3.

$$\Delta T_{ij}^{(n)} = a_3(Y_i^{(n)} - Y_i^{(n-1)}) Y_j^{(n-1)}.$$

In the presence of the weights S, the weights R does not represent prototypes of the output of the DG. Besides, the accumulation due to the weights S is occurred to the CA3. The outputs of the CA3 no longer represent prototypes (place code) as its localized activations and consequently are not conservative to the transformation by the Mexican hat connection. Thus the learning rule is applied to T to retain the functionality of the CA3 as a short term memory. If an element of weights R is a small random value, the application of the learning rule results in $T_{ij} \simeq \frac{2}{N}\cos\{\frac{2\pi}{N}(i-j)\}$ which is an identical map to the sinusoidal vectors of the frequency $\frac{2\pi}{N}$ [9]. In the case that the weights T are the Mexican hat and the weights S do not exist in the $\mathcal{C}_E$, the learning rule is not executed. Since each of prototypes is conservative to the transformation by the weights T.

The prototyping circuit of the HNN^S, for the purpose of the high accuracy navigation, should be substituted by the well-ordered gaussian cells with the Neural Gas Algorithm (NGA)[11]. The error analysis of the original prototyping circuit is difficult for its nonlinearity, while the sensory map generated by the NGA are well studied with radial basis functions.

3.3 Learning rule for $\mathcal{C}_I$

Let $U = (U_{ij})$ be the weights from the CA3 layer to the CA1 layer. The weights are set to $U_{ij} = \delta_{ij}$ so that the output of the CA3 is one-to-one corresponding to that of the CA1. Let $V = (V_{ij})$ be the weights from the external CA3 in the other HNN to the internal CA1. The output of the internal CA3 is denoted by $\vec{Y}_1^{(n)}$ and that of the external CA3 by $\vec{Y}_2^{(n)}$. The learning rule for the C$_I$ is denoted by

$$\Delta V_{ij}^{(n)} = a_4(Y_{i1}^{(n)} - V_{ij}^{(n)}) Y_{j2}^{(n)}.$$

The application of the learning rule results in that an internal signal $\vec{Y}_1$ is associated with an external signals $\vec{Y}_2$ expected to be occurred simultaneously when the agent exploring the environment.

4 Homing vector field and sensory map

The agent produces motor signals and receives sensory signals during exploration around the environment. Due to the existence or absence of the θ-rhythm in the learning, in addition to the different representations of motor and sensory signals, the common HNNs are specialized into processors of motor information and sensory information respectively.

After learning, the motor information processing unit HNN^H acts as follows : The circuit $\mathcal{C}_E^H$ encodes angle and distance into a sinusoidally distributed representation of cells [7]. The representation is added to the pattern on $\mathcal{C}_R^H$, where the homing vector is preserved by self-recurrent connections of $\mathcal{C}_R^H$ acting as a short-term memory.

On the sensory information processing unit HNN^S, the function of $\mathcal{C}_R^S$ is modified adaptively to encode sensory inputs into sparse representation. The representation is translated to a localized one with a gaussian receptive field on $\mathcal{C}_R^S$, which corresponds to "place cell". The self-recurrent connections of $\mathcal{C}_R^S$ acts as stabilizer of the activity of the place cell.

An averaged value of homing vectors around each of sensory prototypes are embedded to weights in $\mathcal{C}_I^H$ by Grossberg's learning rule. The embedded vectors finally converge to the true homing vectors on the sensory prototypes. On the CA1 of the HNN^H, the basis of the homing vector field are associated with sensory stimulus without integrating motor stimulus. The homing vector fields is reconstructed as the output pattern of the CA1 from the basis embedded in the weights V in the HNN^H. The output pattern is calculated by

$$\vec{Z}^H = \sum_j a_j^S \vec{V}_j^S,$$

$$a_j^S = \frac{Y_j^S}{\sum_j Y_j^S} \simeq \frac{e^{-\|\vec{X}^S - \vec{R}_j^S\|/\sigma_j^2}}{\sum_j e^{-\|\vec{X}^S - \vec{R}_j^S\|/\sigma_j^2}},$$

where $\vec{Z}^H$ is the output of the CA1 of the HNN^H after learning, $\vec{V}_j^H = (V_{1j}^H, \ldots, V_{Nj})$ is the mean homing vector corresponding to the j-th sensory prototype of $\vec{R}_j^S$, $\vec{X}^S$ is the the DG of the HNN^S induced by a sensory stimuli, and $\vec{R}_j^S$ is the j-th prototype embedded in the weights R^S of the HNN^S. By integrating signals from HNN^H and HNN^S, the homing vector field adhered to sensory maps in HNN^S is obtained in the hippocampal navigation circuit.

5 Results

Each of the layers in the HNN^H is composed of 8 cells. In the HNN^S, the EC is composed of 32 cells, the DG is 128 cells, the CA3 has 25 cells in itself, and the CA1 also has 25 cells. Before the homing navigation task, the robot explores the area of $3m \times 3m$ by the random walk and then 20000 samples are selected from the real environment. The samples consists of pairs of a motor stimuli and a sensory stimuli. The learnings in the HNN^H and HNN^S are executed simultaneously by on-line setting in which each of the samples is utilized for one adaptation. The learning procedures cost about 5 minutes on a S-4/20(125MHz, SPECFP 153.1). A result of the learning is shown in Figure 3.

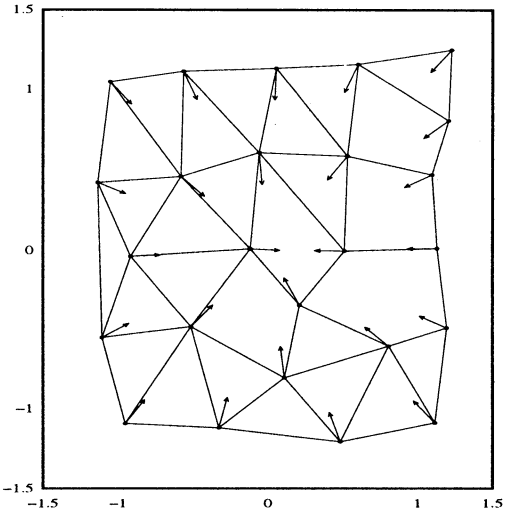


Figure 3. *Homing vectors attached to the sensory map. Lattice points denote sensory prototypes and lattices are adjacent relations between the prototypes. Arrows shows direction of homing vectors attached to the prototypes.*

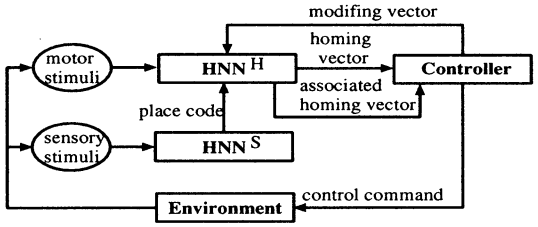


Figure 4. *Model of the homing-navigation from any point in the explored environment.*

Figure 4 shows the diagram for home-navigation by using the vector field. The homing vector $\vec{Z}^H$ is associated on the HNN^H with the sensory stimuli of the current point (, i.e., an S→H association). Find the maximum value $Z_{i_{max}}^H$ of the homing vector and move the agent in the direction of $\phi = \frac{2\pi}{N} i_{max}$ by a distance $r = c\|\vec{Z}^H\|$, (0< c <1). If $Z_{i_{max}}^H$ is nearly equal to 0 then stop the robot else repeat the procedure.

Figure 5 shows the robot moves from arbitrary points to a home position (recursively) according to control commands (r, ϕ) generated by the procedure. The experiments show that the robot makes homing navigation with good approximate precision from any point to the origin in experiment circumstance without pre-programmed information about a task space.

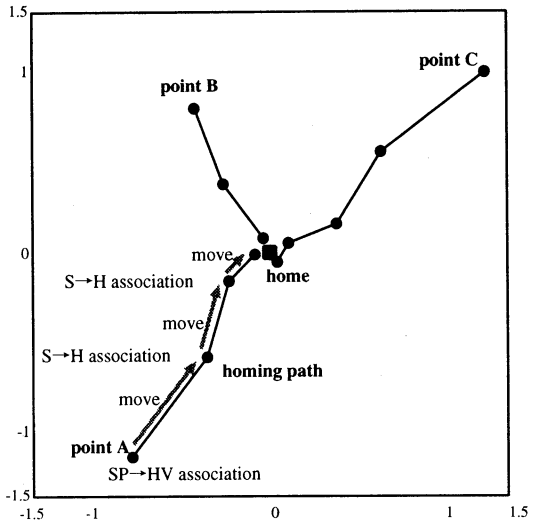


Figure 5. *Homing paths calculated iteratively from the homing vector field adhesion to the sensory map.*

6 Summary

We have proposed the hippocampal navigation model with two sub-networks which are self-organized to process motor and sensory information by Hebbian learning rules. By the associative interactions between the sub-networks, homing vector fields on one sub-network are attached to sensory maps on the other. Such integrated maps endow robots with the capability of selecting information to generate control commands by adapting its operating environments.

We recognize the present network is less accurate in predicting positions in comparison with the methods based on the Extended Kalman Filtering [12][13], since there are no noise model in our network. However, the robot with the hippocampal navigation model can establish a coordinate system autonomously during exploration in the environment. This is a new view point of our approach.

Acknowledgments

This study was performed through Special Coordination Funds of the Science and Technology Agency of the Japanese Government.

References

[1] R. A. Brooks, "A Robust Layered Control System For A Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol.RA-2, pp.14-23, 1986.

[2] R. G. M. Morris, P. Garrud, J. N. P. Rawlins and J. O'Keefe, "Place navigation impaired in rats with hippocampal lesions," *Nature*, Vol.297, pp.681-683, 1982.

[3] J. O'Keefe and J. Dostrovsky, "Hippocampus as a spatial map : Preliminary evidence from unit activity in the freely moving rat," *Brain Research*, Vol.31, pp.171-175, 1971.

[4] B. L. McNaughton, L. L. Chen and E. J. Markus, "Dead Reckoning, Landmark Learning, and the Sense of Direction: A Neurophysiological and Computational Hypothesis," *Journal of Cognitive Neuroscience*, Vol. 2, pp.190-202, 1991.

[5] N. Burgess, M. Recce and J. O'Keefe, "A Model of Hippocampal Function," *Neural Networks*, Vol.7, pp.1065-1081, 1994.

[6] D. G. Amaral, N. Ishizuka and B. Claiborne, "Neurons, numbers, and the hippocampal network," *Progress in Brain Research*, Vol.83, pp.1-11, 1990.

[7] D. S. Touretzky, A. D. Redish and H. S. Wan, "Neural Representation of Space Using Sinusoidal Arrays," *Neural Computation*, Vol.5, pp.869-884, 1993.

[8] S. Grossberg, "Embedding fields : A theory of learning with physiological implicatoins," *Journal of Mathematical Psychology*, Vol.6, pp.209-239, 1969.

[9] T. Wittmann and H. Schwegler, "Path integration – a network model," *Biological Cybernetics*, Vol.73, pp.569-575, 1995.

[10] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag, 1983.

[11] T. M. Martinetz, S. G. Berkovich and K. J. Schulten, "Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction," *IEEE Transactions on Neural Networks*, Vol. 4, pp.558-569, 1993.

[12] J. J. Leonard, H. F. Durrant-Whyte and I. J. Cox, "Dynamic Map Building for an Autonomous Mobile Robot," *The international Journal of Robotics Research*, Vol. 11, pp.286-298, 1992.

[13] A. Kurz, "Constructing Maps for Mobile Robot Navigation Based on Ultrasoinc Range Data," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol.26, pp.233-242, 1996.

Fine Motion Strategy in Three-Dimensional Space Using Skill-Based Backprojection

Akira Nakamura*
Takashi Suehiro**

Tsukasa Ogasawara*
Hideo Tsukune*

*Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan

**Real World Computing Partnership
1-6-1 Takezono, Tsukuba, Ibaraki, 305 Japan

Abstract

Motion of manipulation in a task can be decomposed into several motion primitives called "skills". Skill-based motion planning has the possibility to perform tasks as skillfully as human beings do. On the other hand, the backprojection method performed in configuration space has been often used in fine motion planning. This paper describes fine motion planning in three-dimensional space using skill-based backprojection.

Key words: manipulation skill, fine motion planning, backprojection, configuration space

1 Introduction

In the 21st century, robots will play an active role in assisting people in their daily lives. Such assistant robots need to carry out various tasks skillfully. However, it is often difficult for the present robots to accomplish even tasks which people can do with ease. We have shown that when practicing such tasks it is effective to adopt the concept of "skills" that are motion primitives [1]-[3]. On the other hand, techniques of fine motion planning performed in configuration space such as the backprojection method have been often used for planning tasks such as assembly [4],[5]. Since an object is represented as a point in configuration space, the planning is simplified.

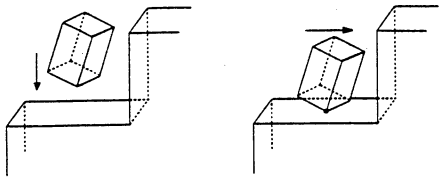
In the First International Symposium on Artificial Life and Robotics (AROB 1), we proposed a fine motion planning using skill-based backprojection [6]. However, the proposed method was restricted to motion tasks in a two-dimensional environment. Robots must be able to perform tasks in a three-dimensional environment in order to work like a human being.

Therefore, in this paper, extending our method proposed in AROB 1, we propose a fine motion planning of tasks in a three-dimensional environment using skill-based backprojection. By this method, we can obtain a subset of the initial position and orientation from which a grasped object can reach the goal by a sequence of skill commands.

In the next section, the concept of manipulation skill is explained. The representation of skill primitives in configuration space is shown in section 3. The construction of backprojection in a task is explained in section 4. An example is shown by using the task of putting a small box into a large box in section 5.

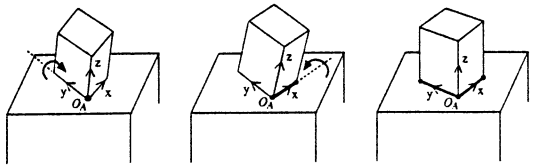
2 Skill Primitives in Manipulation

Motions of a manipulator in a task can be decomposed into several motion primitives, each of which has a particular target state. We call these primitives "skills" [1]-[3], which are derived by analyzing practical manipulation tasks. It is possible to realize various tasks by combinations of these primitives. The concept of skills is particularly important when constraint states between two objects vary in assembly tasks. In this paper, we consider three skill primitives of "move-to-touch," "rotate-to-level" and "rotate-to-insert," which play an important role in such tasks. Typical assembly tasks can be realized by a combination of these three skill primitives. The skill primitives in three-dimensional space are derived by extending those in two-dimensional space proposed in [6].



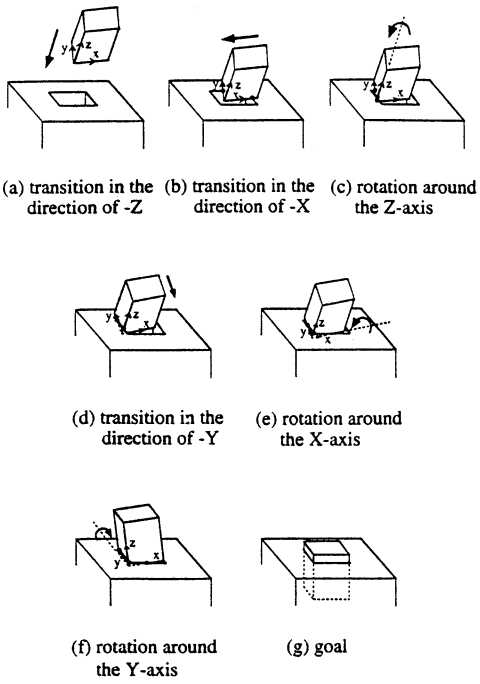
(a) move-to-touch$_f$ skill (b) move-to-touch$_c$ skill

Fig.1 Move-to-touch skill



(a) rotation around the Y-axis (b) rotation around the X-axis (c) face-to-face contact

Fig.2 Rotate-to-level skill



(a) transition in the direction of -Z (b) transition in the direction of -X (c) rotation around the Z-axis

(d) transition in the direction of -Y (e) rotation around the X-axis

(f) rotation around the Y-axis (g) goal

Fig.3 Rotate-to-insert skill

(1) Move-to-touch Skill

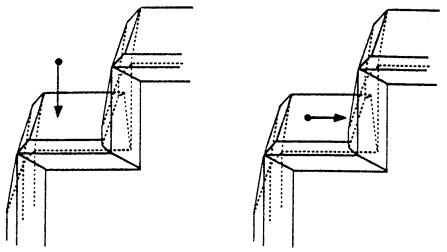
The move-to-touch skill is defined as the transition of a grasped object in a constant direction that continues until a contact occurs. This skill is performed in velocity control mode. Not only the transition in free space (Fig.1(a)), but also sliding while keeping contact in a different direction of motion (Fig.1(b)) is included in this skill. These are represented by the "move-to-touch$_f$" skill and the "move-to-touch$_c$" skill, respectively. The achievement of this skill can be detected by instantaneously increased resistance in the direction of transition.

(2) Rotate-to-level Skill

The rotate-to-level skill is defined as rotation motion around either a contact point or a contact edge in order to match a face of the grasped object with a face of another object. For example, if the grasped object is a rectangular parallelepiped, this skill is performed by a sequence of two rotation motions around different axes as shown in Fig.2(a), (b) and (c). The achievement of each rotation motion can be detected by change of the instantaneous center position.

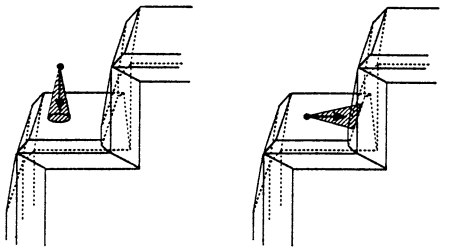
(3) "Rotate-to-insert" Skill

For simplicity, let us explain this skill in the case that a rectangular parallelepiped is inserted. The rotate-to-insert skill is performed as the last two steps of the insertion task (Fig.3). The insertion task in three-dimensional space is constructed by revision of that in two-dimensional space shown in [6]. This insertion task is composed of six motions: (i) transition in the direction of -z (Fig.3(a)), (ii) transition in the direction of -x (Fig.3(b)), (iii) rotation around z-axis (Fig.3(c)), (iv) transition in the direction of -y (Fig.3(d)), (v) rotation around x-axis (Fig.3(e)), and (vi) rotation around y-axis (Fig.3(f)). The motions (i), (ii) and (iv) are performed by move-to-touch skills, and the step (iii) is performed by rotate-to-level skill. In the motions (v) and (vi) the grasped object is raised up by rotation motion, and the number of vertices of the object entered in the hole increases. The rotate-to-insert skill is defined as the last two steps. The achievement of each rotate-to-insert skill is detected by variation of the instantaneous center position.



(a) move-to-touch$_f$ skill (b) move-to-touch$_c$ skill

Fig.4 Move-to-touch skill in C-space



(c) move-to-touch$_f$ skill (d) move-to-touch$_c$ skill

Fig.5 Move-to-touch skill with control uncertainty cone in C-space

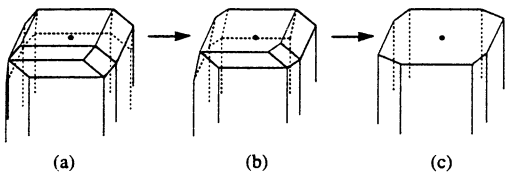
3 Skills Primitives in Configuration Space

In this section, we consider the trajectories in configuration space with respect to each three-dimensional skill primitive. Backprojections are derived as the trajectories in the inverse direction.

It is difficult to deal with planning problems in three-dimensional space because of the six degrees of freedom; three degrees of freedom on position and three degrees of freedom on orientation. Therefore, putting restrictions on the change of orientation, we consider the trajectories in configuration space by using several slices in which orientations do not vary.

(1) Move-to-touch Skill

Trajectories in configuration space by move-to-touch$_f$ and move-to-touch$_c$ skills (Fig.1(a), (b)) are drawn in Fig.4(a), (b), respectively. If velocity errors of manipulation are taken into consideration, trajectories are derived by using control uncertainty cones as shown in Fig.5(a), (b).



(a) (b) (c)

Fig.6 Rotate-to-level skill in C-space

(2) Rotate-to-level Skill

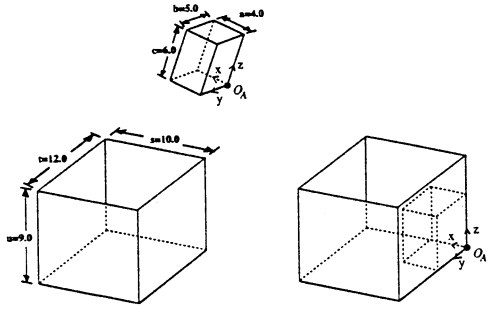
In order to avoid complication of shapes of C-obstacles, we assume that the reference point O_A is a vertex in contact with the surface of the obstacle in advance. In configuration space, Fig.2(a), (b), (c) are represented as shown in Fig.6(a), (b), (c), respectively. The transitions on three slices of orientations are carried out.

(3) Rotate-to-insert Skill

In the skill sequence of Fig.3, if the state of Fig.3(e) is realized, the grasped object attains the goal by two rotate-to-insert skills without fail. Therefore, the initial position and orientation of the object are derived by specifying the orientation in Fig.3(e) and calculating the backprojection of the previous steps according to the above-mentioned procedure in (1), (2).

4 Construction of Backprojection in a Task

In the case that a task is composed of one skill primitive, the initial start region from which the object can attain the goal is derived by backprojection from the goal shown in section 3. If it is composed of several skill primitives, it is derived by calculating backprojections successively according to the skill sequence of the task. C-obstacles of skill primitives with rotation are derived as slices by specifying the angle. If the initial region derived from the value of angle vanishes, backprojections are calculated by substituting another value for it. This is repeated until the initial region becomes appropriate.



(a) start (b) goal

Fig.7 A task of putting a small box at a corner of a large box

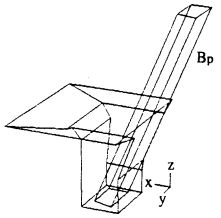


Fig.8 C-obstacles and backprojection

5 Example

Let us consider the task of putting a small box at a corner of a large lidless box from the outside as shown in Fig.7(a),(b). We assume that the task is accomplished by the reverse sequences of the six skills: (i) move-to-touch$_c$ skill in the direction of -y, (ii) rotate-to-level skill around z-axis ($\Delta\ \psi = \pi / 12$), (iii) move-to-touch$_c$ skill in the direction of -x, (iv) rotate-to-level skill around y-axis ($\Delta\ \theta = \pi / 12$), (v) rotate-to-level skill around x-axis ($\Delta\ \phi = -\pi / 12$), and (vi) move-to-touch$_f$ skill in the direction of -z. The backprojection of skill (vi) is derived as shown in Fig.8 and this gives the appropriate initial position of a small box to reach the goal by the sequence of these skill. In this case, uncertainty of control is not taken into account.

6 Conclusion

We have shown a fine motion planning in three-dimensional space using skill-based backprojection. Skill-based backprojection in a two-dimensional environment proposed in AROB 1 was extended to that in a three-dimensional environment. Therefore, it becomes possible to plan manipulation motion which resembles the behavior of the human hand. Generally, it is difficult to do planning in three-dimensional space because of the six degrees of freedom; three degrees of freedom on position and three degrees of freedom on orientation. The workload was reduced by considering configuration space on slices in which orientations do not vary. In future, we will study the precise treatment of errors of control, sensor and model in three-dimensional space, and reinforcement of variety of skill.

Reference

[1] T. Hasegawa, T. Suehiro and K. Takase, "A Model-Based Manipulation System with Skill-Based Execution in Unstructured Environment," '91 Int. Conf. on Advanced Robotics, pp.970-975, Pisa, Italy, June 1991.

[2] T. Hasegawa, T.Suehiro and K. Takase, "A Model-Based Manipulation System with Skill-Based Execution in Unstructured Environment," '91 Int. Conf. on Advanced Robotics, pp.970-975, Pisa, Italy, June 1991.

[3] T. Hasegawa, T.Suehiro and K. Takase, "A Model-Based Manipulation System with Skill-Based Execution," IEEE Trans. on Robotics and Automation, vol.8, no.5, pp.535-544, Oct. 1992.

[4] T. Lozano-Perez, M. T. Mason and R. H. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots," Int. J. Robotics Research, Vol.3, No.1, pp.3-24, 1984.

[5] M. Erdmann, "Using Backprojections for Fine Motion Planning with Uncertainty," Int. J. Robotics Research, Vol.5, No.1, pp.19-45, 1986.

[6] A. Nakamura, T. Ogasawara, T. Suehiro and H. Tsukune, "Fine Motion Strategy for Skill-Based Manipulation," 1st Int. Sym. on Artificial and Robotics pp.158-161, Oita, Japan, Feb. 1996.

Organizational Learning among Decentralized Agents

Takanori Sasaki & Akira Namatame

Dept. of Computer Science
National Defense Academy
Yokosuka, 239, JAPAN
Tel : +81-468-41-3810 (ext. 2432), Fax : +81-468-44-5911
E-mail: sasaki@cc.nda.ac.jp

Keywords: organizational learning, decentralized agent, Knowledge sharing,

Abstract

A decentralized agent is defined as an autonomous software entity with the rational decision-making capability based on its own utility function. We consider the problem of organizational learning of those decentralized agents. We especially specify the problems of knowledge exchange, transfer, and sharing necessary for an efficient organizational decision-making. In the setting of organizational learning, two learning levels may exist: each agent can learn individually based on its own experiences, while at the same time, agents can learn as a group by sharing their private learnt knowledge..We provide the model of organizational learning in which each decentralized agent graudally learns to behave an efficiently with both the selfish-interest and social competence. It is shown that the improved organizational decision-making can be achieved by sharing common knowledge among decentralized agents.

1 Introduction

An organization consists with more than one autonomous agent, in which the outcome to the organization depends jointly on those behaviors of decentralized agents with the selfish-interest. Each decentralized agent has responsible for the same or different decision-making, and makes its decision on the basis of different information. The very simplest organizational decision-making would be an exchange of each agent's decision based on their private knowledge and expereinces. However, decentralized agents may adjust, adapt, and learn from working with other agents in the same organization. The organization of decentralized agents can offer something not available in the individuals. Such an efficient organizational decision-making would then require the exchange of knowledge held by each agent, so that the improved coordination can be achieved by sharing the common knowledge [2][5].

In this paper, we are concerned with so called organizational learning for the efficient group decision-making in which agents learn by sharing their knowledge. In the setting of the organization of agents, two learning levels may exist. Each agent can individually learn as an autonomous entity based on its own experiences. That is each agent learns its way of decision in term of the utility function. Individual learning is defined as the set of mechanism which utilizes adaptive decision-making of selfish-interest seeking of decentralized agents. As a group, organizational learning may take an effect in the form of group decision, and the improved organizational decision-making can be achieved by sharing knowledge and an efficient knowledge transfer among agents. With the organizational learning, each agent needs to adjust, adapt, and learn from the other agents. The agents cooperate to learn by exchanging their learnt knowledge in terms of their utility functions acquired through its own experiences. Organizational learning allows agents behave in order to achieve high-level group goals without the need of cooperative planning, communications, or enforcement for cooperation. Thus, over a time, a group of decentralized agents will be able to learn to cooperate together at an even higher-level learning of efficiency and adaptability.

2 Formulation of Organizational Learning

The notions of self-interested behaviors are the foundation of many fields[3]. Most often, when people in AI use the term agent, they refer to an entity that functions continuously and autonomously in an environment in which other agents exist and other processes take place[4]. A decentralized agent is designated to refer to the entity with the autonomous decision-making capability based on its selfish-interest. They are selfish in the sense that they only do what they want to do and what they think is in their own best interests, as determined by their own goals and motivations. A decentralized agent, endowed with its own decision capability, is specialized to interact with a specific problem domain, and it learns its decision-rule from its observation and experiences. Each agent has a specialized representation of the characteristics of its decision-making in the form of the utility function.

Problem solving and decision making in an organization should be modeled as a dynamic process and the actions of the agents must be coordinated in order to achieve a globally consistent and efficient organizational decision. In an organization of agents, two levels of learning may occur: the agents can learn as a group, while at the same time, each agent can also learn individually[5]. At the individual learning phase, each agent acquires its function that encapsulates a specific set of knowledge obtained as its own experiences. A decision or utility function of each rational agent represents specific knowledge obtained with a different view. At the organizational learning stage, individual agents put forward their learnt knowledge, their own local experiences or private or local knowledge for consideration by other agents. The organizational learning process, as a group, can be deployed in a manner parallel to the individual learning process of each agent. The process of organizational learning is, however, more complex than those in a single agent. Organizational learning may require the exchange of knowledge held by the agents. We then need to provide the mechanism of how an organization of agents share their private knowledge.

The organization of rational agents is denoted by $G=\{A_i : i = 1,2,\ldots,k\}$. An agent has distinct and mutually exclusive decision alternatives. We define the set of alternatives as $W=\{O_1, O_2, \ldots, O_k\}$ and each alternatives is characterized by n attributes, $X_1, X_2, \ldots, X_n$, and each of them has multi-valued attribute with the domain $Dom(X_i)$. A set of alternatives denoted as

$$D=\{X_1:Dom(X_1), X_2:Dom(X_2), \ldots, X_n:Dom(X_n)\} \qquad (2.1)$$

The models of this paper treat binary decisions a positive and negative decision where an agent has distinct and mutually exclusive decision alternatives. In the case where each agent makes its own decision on each alternative, such an alternative is labed by the Boolean value. Especially, an instance $d_t \in \mathbf{D}$ with c_t=1 represents a positive alternative on which an agent makes the positive decision. On the other hand, the alternative d_t with c_t=0 represents a negative alternative on which an agent makes the negative decision. We can also extend such a labeling to the vetor representation of c_t=(c_{t1}, c_{t2},..., c_{tm}). The list of the feature values of an alternative is represented as $d = (d_1, d_2, \ldots, d_n) \in \mathbf{D}$ where the variables d_1, d_2, ..., d_n take the Boolean values. We define the set of the ordered pairs $<d_t, c_t>$ where $d_t \in \mathbf{D}$ and $c_t \in \{0, 1\}$ as the private knowledge space of an agent. As illustlatred in Fig. 1, the problem of organizational learning with knowledge sharing is then formulated as the composite problem of the shared knowledge space **W** = [**D**, **C**] of an organization from the set of the private knowledge spaces $\mathbf{W_i}$=[$\mathbf{D_i}$, $\mathbf{C_i}$], i = 1, 2, ..., n, of each agent, where $\mathbf{UW_i} = \mathbf{W}$. Each agent has learned to be an expert to learn to handle a subset $\mathbf{W_i}$=[$\mathbf{D_i}$, $\mathbf{C_i}$] of the shared knowledge space **W** = [**D**, **C**].

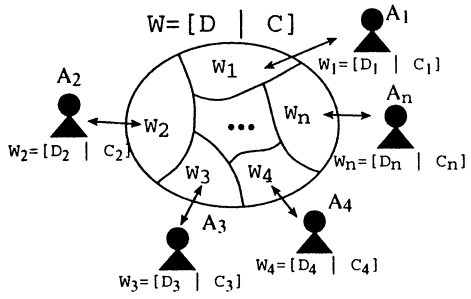


Fig. 1 The concept of the organization decision-making environment

3 Decision-making in an Organization with Individual Learning

In this section, we address the question of how an organization of rational agents learn their intelligent behaviors in order to achieve their independent goals. In an organization, each rational agent cannot simply proceed to perform its decision without considering what other agents may act. The goal of each agent is determined solely by how its decision affects other members in an organization and how the decisions of other agents affect its own goal. We especially ask the following questions: If rational agents make decisions on the basis of his own goal and by incorporating expectations on how his decision will affect other agents, then how will the evolution of individually rational behaviors proceed? How will the structure of his own goal (utility) functions of a rational agent affect as the dynamics of the evolution of their rational behaviors?

In this section, we are concerned with the organizational decision-making problem with individual learning. Each agent in an organization faces the rational decision-making problem of optimizing its own utility without knowing the utlity functions of the others. The agents may have different preferences, interdependent utility functions. Each agent's utility function is described as

$$U_i(x_1,\ldots,x_i,\ldots,x_n) = U_i\{x_i, x(i)\} \tag{3.1}$$

The utility function of each rational agent depends on both its own decision and all other rational agents' decisions. Each agent behaves to optimize its own utility function expressed as the utility function. We define such an optimal decision as the response function which is given as the function of the other agents' decisions $x(i) = (x_1, x_2, .., x_{i-1}, x_{i+1}.., x_n)$,

$$\phi_i\big(x(i)\big) \triangleq \arg \underset{x_i}{Max}\, U_i\big\{x_i, x(i)\big\} \tag{3.2}$$

Such a response function satisfies the condition such that

$$M_i\{x_i, \phi_i(x(i))\} = 0 \qquad i = 1,2,...,n. \tag{3.3}$$

where

$$M_i\{x_i, x(i)\} = \partial U_i\{x_i, x(i)\} / \partial x_i \qquad i = 1,2,...,n.$$

Rational agents are quite apparently both selfish agents and social actors, and they need to present and reason about the knowledge, action, and plans of the other agents in an organization. Without complete kowledge of other agents, however, the rational agent needs to infer other agents' decisions. With the individual learning capability, each agent modifies its decision based on the current and previous performance in order to optimise its own utility [1]. Adaptive decision-making is facilitated by designing the rational agents to be somewhat modified selfish behavior. We call this ability as individual learning. We formulate the individual learning model as the mutual adjustment of individual decisions without knowing the others' utility functions, adjusts its strategy over time. We formulate the mutual ajustement of behaviors of goal-seeking based on its self-interest. The dynamic action selection process must be coordinated to achieve globally consistent and efficient actions. The very simplest cooperation would be an exchange of facts such as the current actions. Then the ajustment process is formulated as

$$\begin{aligned} x_1(t+1) &= \phi_1\big(x(1,t)\big) \\ x_2(t+1) &= \phi_2\big(x(2,t)\big) \\ &\vdots \\ x_n(t+1) &= \phi_n\big(x(n,t)\big) \end{aligned} \tag{3.4}$$

whrere ϕ_i defins each agent's response function and

$$x(i,t) = (x_1(t),\ldots,x_{i-1}(t), x_i(t), x_{i+1}(t),\ldots,x_n(t)) \tag{3.5}$$

This adjustment process generates a partial strategy that governs the actions of the agents . The use of directives by an agent to control another can be viewed as a form of incremental behavior adjustment[6]. Such an ajustment process with individual learning converges to an equilibrium which is defined as the fixed point of the response functions, ϕ i, i=1,2,..,n,

$$\begin{aligned} x_1^* &= \phi_1\big(x^*(1)\big) \\ x_2^* &= \phi_2\big(x^*(2)\big) \\ &\vdots \\ x_n^* &= \phi_n\big(x^*(n)\big) \end{aligned} \tag{3.6}$$

As a specific example, we consider the following quadratic utility function of each agent A_i. i=1,2,...,n, which is given as

$$U_i(x_1,\ldots,x_i,\ldots,x_n) = x_i(a_i - \sum_{j=1}^{n} b_{ij}x_j) - c_i x_i \qquad (3.7)$$

where a_i, b_{ij}, c_i i,j=1,2,...,n, are some positive constants. The response function of each rational agent defined in (3.2) is obtained as follows:

$$\phi_i(x(i)) = (a_i - c_i - \sum_{j\neq i} b_{ij}x_j)/b_{ii} \qquad (3.8)$$

The utility obtained by each agent with individual learning is obtained as

$$\overline{U}_i(x_1,\ldots,x_i,\ldots,x_n) = b_{ii}(\bar{x}_i)^2 \qquad (3.9)$$

We consider the case in which the interaction matrix B is symmetric with the diagonal elements are the same, i.e., b_{ii}=d, and the off-diagonal elements are, b_{ij}=b, (0<b<d), i,j=1,2,...,n. The column vectors a, c also have the same elements, i.e., a_i=a, c_i=c, i=1,2,...,n, respectively. The utility of each rational agent is obtained as follows:

$$U_i^{\circ}(n) = (a-c)^2 d/\{2d + b(n-1)\}^2 \qquad (3.10)$$

The sum of the utilities of the agents in an organization with individual learning is then given as

$$G^{\circ}(n) \equiv \sum_{i=1}^{n} U_i^{\circ}(n) = (a-c)^2 dn/\{2d + b(n-1)\}^2 \qquad (3.11)$$

4 Decision-making in an Organization with Organizational Learning

Decentralized agents can put forward their private knowledge for consideration by other agents based on its own local interactions, and individual learning would require the exchange of actions with other agents. The process of building up intelligent behaviors based on cooperative intentions may be called mutual or organizational learning[2]. Organizational learning is grounded in the actions of many agents' activities taken together, and it not a matter of individual choice. It is one's actions in relation to those of others (vice versa) that maintain its participation. Organizational learning is then formulated as the web of activity emerged from the mutual interactions among rational agents by sharing common kanowledge.

We define $\lambda_i(x(i))$ termed as the shadow price of agent i as

$$\lambda_i(x(i)) = -\sum_{j\neq i}^{n} (\partial^2 U_j / \partial x_j \partial x_i) x_j \qquad (4.1)$$

The shadow price indicates the influence of the decision of agent i to the utility functions of the other agents. We assume that they share them as the common kowledge in the same organization. The organizational learning model describes how each rational agent with sharing the common knowing, adjusts its strategy over time as follows:

$$M_i\{x_i, x(i)\} > \lambda_i\{x(i)\} \quad then \quad x_i := x_i + \delta x_i$$
$$M_i\{x_i, x(i)\} < \lambda_i\{x(i)\} \quad then \quad x_i := x_i - \delta x_i \qquad (4.2)$$

Such an ajustment of the mutual action with sharing the knowledgele converges to the equilibrium solution. At equilibrium , we have

$$M_i\{x_i^*, x^*(i)\} - \lambda_i\{x^*(i)\} = 0 \qquad i = 1,2,\ldots,n. \qquad (4.3)$$

In an organization of rational agents driven by their own selfish motivations, the above adjustment process leads them to learn the efficient group-decision-making rule.

As a specific example, we consider the decision-making of each rational agent with the utility function of the quadratic form in (3.7)

The shared knowledge, termed as the social price defined in (4.1) is given as follows:

$$\lambda_i(x(i)) = -\sum_{j\neq i}^{n} (\partial^2 U_j / \partial x_j \partial x_i) x_j = \sum_{j\neq i}^{n} b_{ji} x_j \qquad (4.4)$$

The utility of each rational agent with organizational learning is given as

$$U_i^*(x_1,\ldots,x_i,\ldots,x_n) = b_{ii}(x_i^*)^2 + \sum_{j\neq i}^{n} b_{ji} x_j^* x_i^* \qquad (4.5)$$

We consider the case in which the interaction matrix B is symmetric as considered in (3.10). The utility of each agent with organizational learning is given as

$$U_i^*(n) = (a-c)^2/4\{d + b(n-1)\} \qquad (4.6)$$

The sum of the utilities of all agents in an organization is given as

$$G^*(n) \equiv \sum_{i=1}^{n} U_i^*(n) = (a-c)^2 n/4\{d + b(n-1)\} \qquad (4.7)$$

Fig. 2 shows the sums of the utilities as the function of the number of agents. By taking the limit with the number of the rational agents, those values have the following properties:

$$\lim_{n\to\infty} G^{\circ}(n) = 0 \qquad (4.8)$$

$$\lim_{n\to\infty} G^*(n) = (a-c)^2/4b \qquad (4.9)$$

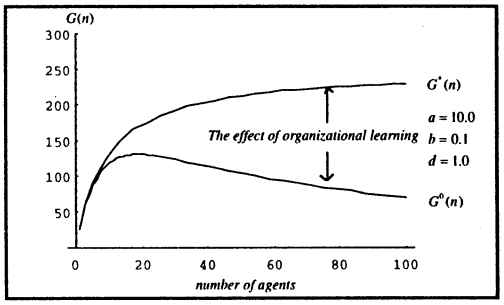


Fig. 2 The effect of organizational learning

5 Application: Decentralized Agents of Supplying Knowlegde Resources

We discuss some problem domains of organizational learning for an efficient group decision-making. Rational agents exist between the users' workstations and the network and decide the best suitable knowledge resources. Many rational agents represent knowledge resources distributed across multiple workstations and file servers on a network. Consider the organiztion of knowledge agents G={K_i, i=1,2,...,n,} supplying certain information services as shown in Fig. 3. The supplies of those knowledge services are based on the specific queries from the out side of the organization. We denote the quantity of knowledge of

knowledge agent K_i that matches to the set of specific queries Q by $s(K_i,Q)$.

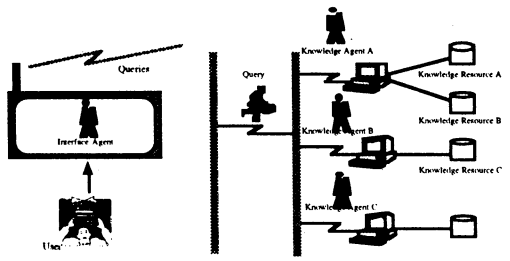


Fig. 3 The supply of knowledge resources by decentralized agents

We formulate the interdependent problem among those knowledge agents as follows: Each knowledge agent K_i, i=1,2,...,n, supplies knowledge at the level of u_i, i=1,2,...,n, where each u_i is given as.

$$u_i = s(K_i,Q)x_i \qquad 0 \le x_i \le 1 \tag{5.1}$$

Each x_i, i=1,2,...,n. denotes the decision strategy of the knowledge agent K_i. Each knowledge agent has its own market value for its knowledge service, and which is assumed to be linear with respect to the levels of the knowledhe supply u_i, i=1,2,...,n, given as

$$p_i = \bar{a}_i - \bar{b}\sum_{i=1}^{n} u_i \tag{5.2}$$

The cost incurred to each knowledge agent is also defined by the linear function. Then the decision problem of each knowledge agent is to choose the level of its knowledge service so that his utility (profit) function

$$U_i(x_1,\ldots,x_i,\ldots,x_n) = (\bar{a}_i - \sum_{j=1}^{n}\bar{b}u_j)u_i - \bar{c}_i u_i \tag{5.3}$$
$$= S(K_i,Q)^2\{(a_i - \sum_{j=1}^{n} b_{ij}x_j)x_i - c_i x_i\}$$

is optimized, where a_i, b_{ij}, c_i i,j=1,2,...,n, are some positive constants defined as

$$a_i = \bar{a}_i / S(K_i,Q)$$
$$b_{ij} = \bar{b}S(K_j,Q)/S(K_i,Q) \tag{5.4}$$
$$c_i = \bar{c}_i / S(K_i,Q)$$

We denote the set of queries made in the past Q ={q_1, q_2, ..., q_k,} and each query is characterized by n attributes, $X_1,X_2,\ldots,X_n$, and which is represented by the list of the feature values $d_t = (d_{t1}, d_{t2}, \ldots, d_{tn}) \in \mathbf{D}$ where the variables $d_1, d_2, \ldots, d_n$ take the Boolean values. Based on each agent's response on each query, such a query is labed by the Boolean value. The success response of agent i to the specific query d_t is labeled by c_{it} Thefore the T×1 column vector C_i epresents agent its histories to the set of queries which is represented by D. The private knowledge of agent i is then described by $\mathbf{W_i}$=[D, C_i], i = 1, 2, ..., n, where each component of $\mathbf{W_i}$ is the ordered pairs $<d_t, c_{it}>$ where $d_t \in \mathbf{D}$ and $c_{it} \in \{0, 1\}$. Especially, a query $d_t \in \mathbf{D}$ with c_{it} =1 represents the positive response of agent i, and the query d_t with c_{it}=0 represents the no response. The problem of organizational learning with knowledge sharing is formulated as the composition of the shared knowledge space $\mathbf{W} = [\mathbf{D} \mid C_1, C_2, \ldots, C_n]$ of an organization from the private knowledge space of each agent $\mathbf{W_i}$=[D, C_i], i = 1, 2, ..., n. By sharing the common knowledge $\mathbf{W} = [\mathbf{D} \mid C_1, C_2, \ldots, C_n]$, each agent can now estimate some positive constants that characterize the utilities of the other agents. Therefore we can approximate the similarity measures to the queries as follows:

$$s(K_i,Q) = \sum_{t=1}^{T} c_{it} \equiv \|C_i\|$$
$$s(K_i,K_j) = \sum_{t=1}^{T} c_{it}c_{jt} \equiv (C_i,C_j) \tag{5.5}$$

Thefore by sharing their private knowledge, each agent can obtain the shadow price as

$$\lambda_i(x(i)) = -\sum_{j\neq i}^{n} (\partial^2 U_j / \partial x_j \partial x_i)x_j$$
$$= -\sum_{j\neq i}^{n} b_{ji} = -\bar{b}\sum_{j\neq i}^{n} \|C_j\|/\|C_i\| \tag{5.6}$$

6. Conclusion

The goal of the research is to understand the types of simples local interactions which produce complex and purposive group behaviors. We formulated and analyzed the social learning process of independent rational agents. We showed that cooperative behaviors can be realized through purposive local interactions based on each individual goal-seeking. Each rational agents does not need to express his objective or utility function, nor to have a priori knowledge of those of others. Each rational agent adapts his action both to the actions of other rational agents, and thus allowing previously unknown rational agents to be easily brought together to customize a group responsible for a specific mission. Intelligence of agents are emerged as intelligent behaved toward attaining some coordinated actions. With the growing complexity of intelligent systems, explicit cooperation of individual subsystems becomes difficult. The overall system is constructed from several interactings rational agents. We discussed how an assembly of agents cooperates to learn as a group. Each rational agent encapsulates a specific set of knowledge obtained from a different problem domain. At the cooperative stage, rational agents put forward their learnt knowledge. Pragmatically, a system will seem intelligent if it can merge new properties that can not simply learnt their own experiences. We showed that the cooperative learning model may provide such an emergent property.

REFERENCES

[1] Barto,A.G,Sutton," Learning and sequential decision-making", Learning and Computational Neuro Science, MIT Press,1991

[2] Carley,K, & Prietula,M.: Computational Organization Theory, Lawrence Erallbawn, 1994

[3] Fudenberg,D,E,and Tirole.J, Game Theory, MIT Press ,1991.

[4] Gasse,L.,Social conceptions of knowledge and action:DAI foundations and open systems semantics, in Artificial Intelligence, Vol.47, pp.107-135,1991.

[5]Sian.S.E. "The Role of Cooperation in Multi-agent Learning Systems", in Cooperative Knowledge-Based Systems, Springer-Verlag, pp. 67-84 (1990).

[6] Tenney. R.,& Sandell,N.R., Strategies for Distributed Decision making, IEEE Trans. Automatic Control, Vol.AC-19,,pp.236-247,1974.

[7] Young,P., The evolution of conventions,Econometrica, Vol.61,No.1, pp.57-84, 1993.

On Ethology inspired Models for Cooperative Mobile Minirobotics

Nadir MOSTEFAI and Alain BOURJAULT
Laboratoire d'Automatique de Besançon, CNRS UMR 6596
25 Rue Alain Savary - 25000 Besançon - France
Phone: 81 40 27 92 / 81 40 28 00
Fax: 81 40 28 09, E-mail: mostefai@ens2m.fr/Alain.Bourjault@ens2m.fr

ABSTRACT

This paper concerns the area of minirobotics especially in the context of multi-agents systems. Each agent is a physical entity that consists of an autonomous mobile minirobot. By using lots of relatively simple minirobots in place of one large complicated and expensive robot, we can do works in the environment at a fraction of the time and cost. Furthermore, we can argue this choice by robustness and flexibility. The idea is to study scenarios inspired from ethology which analyses the behaviours and interactions in insects societies. We are particularly interested in the recruitment mechanisms used in ants colonies. Each minirobot involved in our scenarios has a cognito-reactive architecture since it combines some characteristics of reactivity and cognitivity concepts. An example of a simple object oriented petri-net that models a scenario based on a mass recruitment mechanism is shown. Such model controls the dynamic interactions of an objects set that define the system. Two simulated scenarios with results are illustrated.

Keywords: Ethology, Cooperation, Minirobotics, Modelling, simulation..

1 Introduction

In multirobots systems [1], two main approaches are known : the cognitive approach and the reactive one. In the cognitive approach, the functioning mode for each robot is in the form of perceive--process--act. The processing is itself a composition of functions that are: identification, planning and decision. These functions use knowledge bases which contain domain knowledge (purposes, plans, environmental knowledge) and control knowledge (methods, strategies, their chain). A robot can have an explicit representation of the environment including other robots and may take into account the past activities. However, in practice, such cognitive systems are feasible so far only with a small number of robots. Furthermore, we cannot argue for a reduction of time between mission conception and implementation.

The reactive approach is very interesting. It privileges the behaviour notion upon the knowledge one. The functioning mode is rather in the form of perceive--act. The subsumption concept [2] is the most illustrative methodology in building intelligent reactive control. It consists on a behaviours hierarchy where a behaviour from a superior level subsume a behaviour of lower levels. The concept of reactivity emphasises that there would be no traditional notion of planning and the notion of world modelling is impractical and unnecessary [3].

Our research deals with a cognito-reactive approach. The idea with hybridising both previous approaches is to let a reactive behaviour-based system take care of the real time issues involved in world interactions while a more traditional artificial intelligence system (based on a learning cognitive process) sits on top, making longer term executive decisions as in following a memorised path.

The idea of scaling down the size of the robots inspires missions which will capitalise on several agents. Then, we should not think of agents swarms as machines which are told what to do but rather as autonomous individuals when turned on, do what is in their intelligent control programs.

2 Scenarios modelling

The ethology is our inspiration source for elaborating cooperative scenarios. It leads to the analysis of mechanisms involved in insects social organisation. When we consider the behaviours of the workers in ants societies, we fall into one of the three following foraging categories [4][5]:

- purely individual foraging;
- individual foraging + recruitment;
- individual foraging + recruitment + permanent trail.

In the second category, the workers forage individually, but if one encounters an important food source, it recruits nest mates which initiate the food source's exploitation. Different recruitment mechanisms exist. In mass recruitment, a scout (the recruiter) discovers the food source and returns to the nest laying a chemical trail. In the nest, some of its nest mates (the recruited) detect the trail and follow it to the food source. There they ingest food and return to the nest reinforcing the trail. In tandem recruitment, the scout invites ants at the nest to accompany it back to the food. One recruit succeeds in following the leader, the two animals being in close contact. In group recruitment, the recruiter leads a group of recruits to the food source by means of a short-range chemical attractant. After food ingestion, in each recruitment type, the recruited become recruiters.

The OOPN (Object Oriented Petri Net) combines two principles: the object concept and the Petri-Net tool.

The object oriented approach is rich enough so that it can be applied for various systems. An object is provided with a set of attributes and a set of operations (or methods) enabling to manipulate those attributes values. In [6], the Petri-Nets with objects formalisms are defined as a kind of high level Petri-Net because they integrate the three components of an object: data structure, operations and behaviour. Each transition in the OOPN may be associated with precondition, methods or postcondition . Therefore, any desired level of cognitivity can be integrated in the methods block.

Let us consider the environment (Fig. 1) to model. There is a base B_1 where minirobots r_1 take place. The aim consists on finding the target in order to exploit it (scenario based upon a mass recruitment mechanism). We define a detection zone (dz), an approach zone (az) and the exploitation zone (ez). The parameters (zc) and (rc) refer respectively to the zone capacity (number of minirobots admitted simultaneously in a given area) and to the resource capacity (number of minirobots that can exploit the target simultaneously).

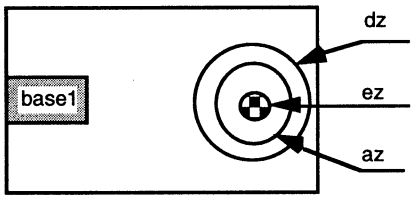


Fig. 1 Environment representation

We can define some objects noted as follows:
- $<r_1>$: minirobots from base B_1
- <zc> : exploitation zone capacity
- <rc> : target resource capacity
- $<call_i>$: target discovery message from $<r_i>$

The set of possible actions is {gf, gb, tl, tr, st, com, expt} where:
gf (resp gb) : go forward (resp go backward)
tl (resp tr) : turn left (resp turn right)
st : stop com : send message expt : exploit

We note $<r_i\text{^}action_i>$ for minirobot r_i doing $action_i$.

We have supposed m minirobots $<r_1>$ in the system. When the zone capacity <zc> is greater than one, it should be represented by several objects in a place. The object <rc> is particularly interesting since it brings out the deadlock problem in the system. If this latter "token" is inhibited, the following transition will never be crossed. An analogous reasoning could be held for the object <zc>.
The cooperation is pointed out by the objects $<call_i>$ that ensure coordination in the system, by communicating a target discovery message (Fig. 2).

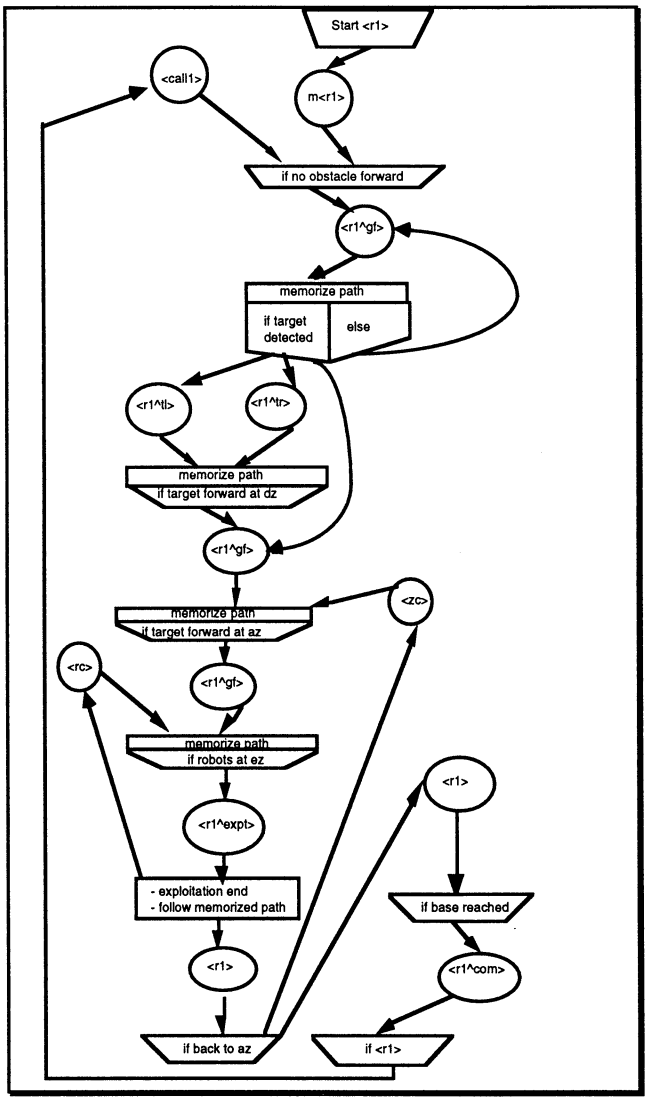


Fig. 2 An OOPN model

3 Simulation

Let's assume we have the environment illustrated in Fig. 3. Its dimensions are 10/10 m and includes three bases where minirobots park (five minirobots in each base). We define two scenarios: the first case is based upon a tandem recruitment mechanism and the second case uses a mass recruitment mechanisms.

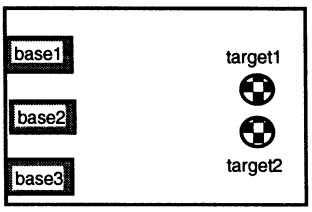


Fig. 3 Simulated environment

3.1 Scenarios examples

A simulation has been done on a SiliconGraphics workstation, using Automod programming language. To complete the list of parameters previously defined, we add v_i as the velocity of the minirobot r_i (the default velocity is fixed to 0,5 m/s).We also define two criterions: the density balancing (D.B) criterion leads a minirobot reaching dz (the detection zone) to choose a target with a minimum of exploiters while for an aggregation criterion (Aggreg), a minirobot chooses a target with a maximum of exploiters. The default criterion is the density balancing.

3.1.1 Case one : Tandem recruitment

Figure 4 shows that the number of minirobots at target1, when using an aggregation criterion is greater than for a density balancing criterion from about time t= 2 minutes. In Figure 5, we note some influence when varying the rc parameter on the total number of minirobots at target2 whose value is sometimes greater for rc=2. Figure 6 gives the average times spent by minirobots r_2 issued from base2 at target1. We vary here the starting times for the minirobots of the different bases. We note two falling peaks when start-per =10 sec while for start-per=30 sec, we have only one falling peak .

3.1.2 Case two : Mass recruitment

Figure 7 shows that the number of minirobots at target1 with an aggregation criterion is greater than with a density balancing criterion. However, we note the number of minirobots (14 over a total of 15) that go to target1 with the aggregation criterion, this means only one minirobot selects target2. In Figure 8, from about 5 minutes, most of time, the number of minirobots at target2 is larger when we increase the rc parameter. Figure 9 shows that the falling peaks are lagged with the increase of velocity and sometimes, the values are slightly different.

3.2 Randomness in scenarios

It is introduced in the targets exploitation. A constant value time (no randomness here) is used to represent a single, recurring event. The uniform distribution needs two arguments: the mean (m) and the deviation (n), values have an equal probability of falling between "m-h" and "m+h". The normal distribution that is represented with a bell shaped curve requires two arguments: the mean (M) and the deviation (N); 78% of all events happen between "M-N" and "M+N", the other 22% of the events occur randomly somewhere on the bell curve outside the previous range. We use the following laws:

law 1 : constant = 50 sec
law 2 : uniform 50, 20 sec
law 3 : normal 50, 10 sec
law 4 : 50 % of time for law 2 and 50% of time for law 3

The results obtained for example at time t= 10 minutes are given in Table 1 and Table 2.

A (resp B): total number of minirobots at target1 (resp target2).
C (resp D): average times spent by r_2 at target1 (resp target2).

parameter / law	A	B	C (sec)	D (sec)
law 1	5	5	54	54,3
law 2	4	6	67,5	53,5
law 3	5	5	39,5	57,1
law 4	6	4	129	61

Table 1: data obtained with tandem recruitment

parameter / law	A	B	C (sec)	D (sec)
law 1	8	5	52,7	48,8
law 2	8	6	41,3	51,4
law 3	7	7	64,2	48,1
law 4	7	6	48,1	59,6

Table 2: data obtained with mass recruitment

4 Conclusion

Cooperative minirobotics systems can be envisaged for many applications. However, defining the concept of cooperation remains vague. In the literature, it has been defined in different fashions. This led us to develop ethology inspired scenarios that could be matched to many real world tasks. We think particularly to planetary exploration or preventive and curative tasks. A model using object oriented petri-net has been developed. It is useful during the design step. It helps to elaborate the control and to analyse the system. Since we have several parameters to take into account in the control strategies, a simulation was therefore necessary. A testbed is under development to put into practice the simulated scenarios. We should start by a scenario that involves two minirobots using a tandem recruitment mechanism.

References

[1] Cao, Y. U., Fukunaga, A. S., Kahng, A. B., and Meng, F., 1995, "Cooperative Mobile Robotics: Antecedents and Directions," Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 226-234.

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

[3] Steels, L. "Towards a theory of emergent functionality." Int Conference on Simulation of Adaptive Behaviours, from Animals to Animats, pp 451 - 461, 1991.

[4] Pasteels, J. and al "Self-Organization mechanisms in ant societies (I) : trail recruitment to newly discovered food sources." Behavior in social insects, vol - 54, 1987.

[5] Deneubourg, J.L. and al "Random behavior, amplification process and number of participants : how they contribute to the foraging properties of ants.", Physica 22D (1986), pp 176 - 186.

[6] Bastide, R., 1992, "Objets cooperatifs : un formalisme pour la modelisation des systemes concurrents," Thesis, Université Paul Sabatier, Toulouse.

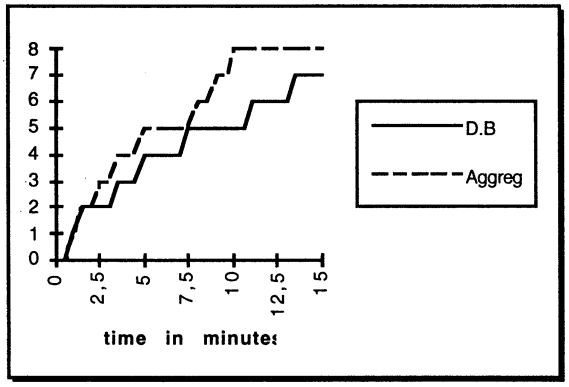


Fig. 4 Minirobots that reach target1 when using D.B or Aggreg criterion (Vi=0.5m/s, rc=1, zc is infinite) in case one.

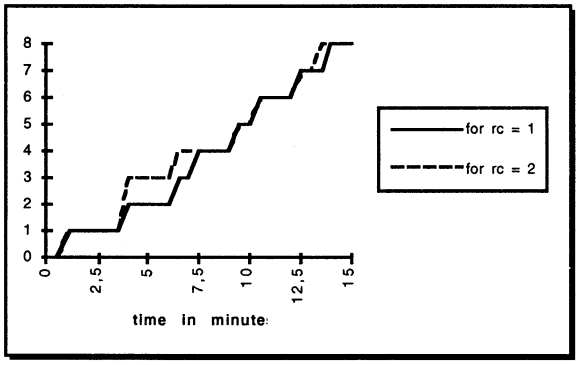


Fig. 5 Minirobots that reach target2 for a different rc parameter in case one.

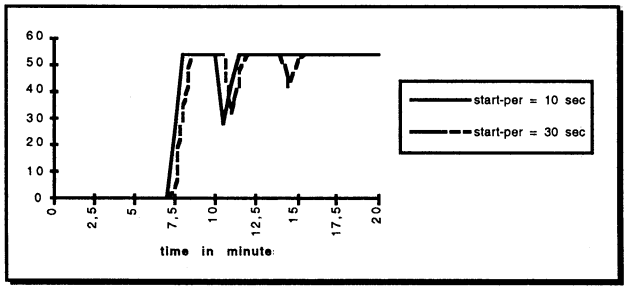


Fig. 6 Average times (in seconds) spent by minirobots r_2 at target1 for different starting times in case one.

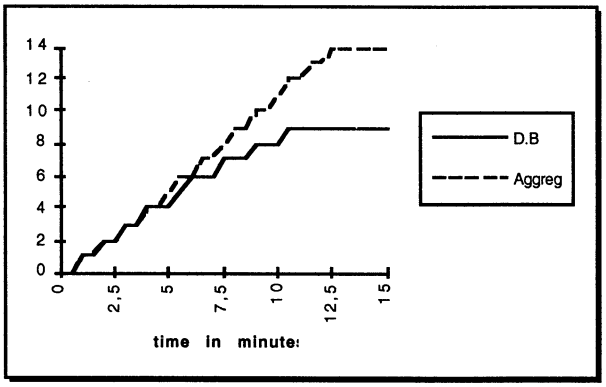


Fig. 7 Minirobots that reach target1 when using D.B or Aggreg criterion (Vi=0.5m/s, rc=1, zc is infinite) in case two.

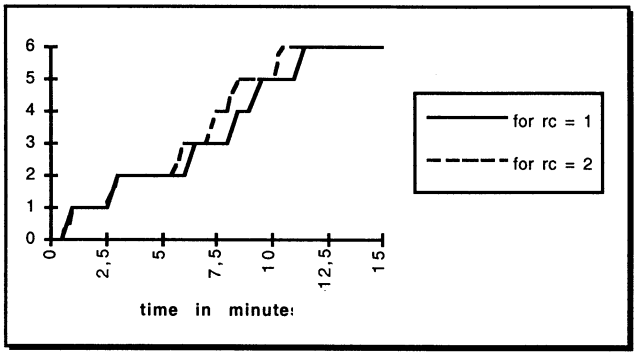


Fig. 8 Minirobots that reach target2 for a different rc parameter in case two.

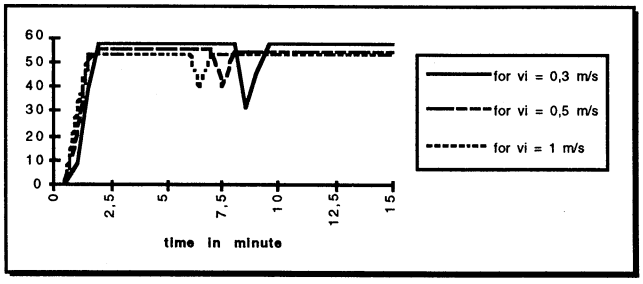


Fig. 9 Average times (in seconds) spent by minirobots r_2 at target2 for different velocities in case two (rc=2).

Self-Generating Algorithm of Evaluation for Cooperative Behavior among Distributed Autonomous Robots

Kazuya OHKAWA
University of Tsukuba
1-1-1 Tennodai, Tsukuba
Ibaraki, 305, Japan
okawa@melcy.mel.go.jp

Takanori SHIBATA
Artificial Intelligence Lab., MIT
545 Technology Square
Cambridge, MA 02139, USA
shibata@ai.mit.edu

Kazuo TANIE
Mechanical Engineering Lab.
1-2 Namiki, Tsukuba
Ibaraki, 305, Japan
m1750@mel.go.jp

Abstract

This paper describes a self-generating algorithm of behavioral evaluation that is a norm to share tasks among robots. This behavioral evaluation is composed of rewards and self-evaluated standards. Rewards are given by the operator when the task is shared by robots, and self-evaluated standards are acquired as the result of executions, such as the case when energy is consumed. Each robot plans its own next behavior, considering its environment and the behavior of other robots. Using the evaluation, the robot can cooperate with other robots depending on the situation even if their purpose is changed during the execution of the tasks.

We performed simulations to study the effectiveness of the proposed algorithm. In this simulation, we applied the algorithm to three robots, each with three behaviors. We confirmed that each robot can create a behavioral evaluation appropriately even if the operator changed rewards in the middle of the simulation.

1 Introduction

When distributed autonomous robots work on common tasks, they can achieve the goal efficiency if each robot avoids conflicts and shares tasks depending on the situation. Therefore, there are many research focusing on robot cooperation for distributed autonomous robots [1-6]. For example, Asama *et al.* studied on communication system for multiple robot system [4], Yuta *et al.* proposed the modest cooperation [5] and Kuniyoshi *et al.* worked on cooperation by observation [6]. However, in most of these research, the operator determines and provides shares and strategies for cooperative behavior in advance. In this case, if the robots' purpose is changed during the execution of their tasks, they cannot adapt to the change because they are not provided the other robots' share of tasks. So, the operator must provide them with new behaviors depending on the situation.

In many biological societies, however, there are many cooperative behaviors without a supervisor for optimization. This is because each of the members observes the situation well, and changes the current behavior to new behavior depending on new situation [6,7].

Being inspired by biological systems, we proposed an algorithm as shown in Figure 1 for generation of spontaneous cooperative behavior among distributed autonomous robots [8]. Each robot equipped with the algorithm has the ability that can decide and select the next behavior depending on behaviors of other robots by using learning function.

In this case, we assumed that the operator of robots provided the evaluation requiring the learning function in advance. However, the operator had to know many information to establish their evaluation properly. And if their purpose was changed, the operator had to modify the evaluation of each of them.

In this paper, therefore, we propose the self-generating algorithm of the evaluation for cooperative behavior among distributed autonomous robots, and show the result of simulations.

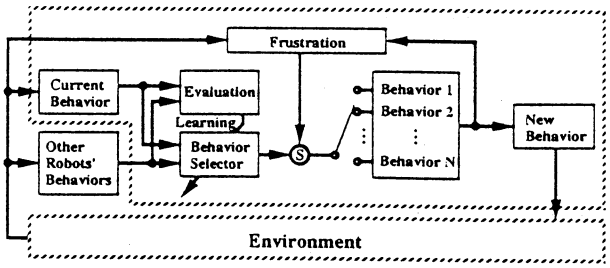


Fig. 1 : Algorithm for Generation of Spontaneous Cooperative Behavior

2 Algorithm for Generation of Spontaneous Cooperative Behavior

We proposed the algorithm as shown in Figure 1 to share tasks by distributed autonomous robots even if the operator implemented each robot's common program to achieve tasks. This algorithm is composed of several modules that are *behavior modules* including elemental behaviors to achieve all of tasks, *selector module* to select the elemental behavior, *sensor module* to obtain the information of its situation, and *evaluation module* to evaluate selected behavior.

Each robot should select the behavior that has the longest value of the evaluation. This behavior is the best selection for the robot, however, it is not for the group of robots. In consequence of their selection, robots make a conflict of the executing task. Therefore, in order to make mutual concessions, we added a learning function that can accumulate the experience of other's behaviors [9-11].

Robots can not make cooperative behavior according to circumstances even if each robot has the learning function. Therefore, we adopted a concept of frustration to evade such situation. By using this function, robots can select their own behavior autonomously when they continue to select the same behavior.

3 Self-Generating Algorithm of Behavioral Evaluation

We provide each robot with a common evaluation for its selected behavior with other robots' behaviors, in advance. The evaluation is shown in Table 1. However, it is difficult to decide appropriate values of the evaluation for all situations because we have to know the tasks, all the situations, functions of robots, and so on. And we must modify the evaluation when their environment or purpose of tasks is changed.

Table 1 : Behavioral Evaluation

Behavioral Evaluation		Other Robot's Behaviors		
		Behavior 1	. . .	Behavior N
Current Behaviors	Behavior 1	E (1,1)	. . .	E (1,N)
	. . .	. . .		. . .
	Behavior N	E (N,1)	. . .	E (N,N)

In many biological societies (animals, humans), however, members can select an appropriate behavior depending on new situation even if a supervisor does not give a evaluation in advance. Because each selects a behavior that is based on self-evaluated standards. The self-evaluated standards correspond to fatigues and fulfilling the achieved tasks. Therefore, each can select a behavior with efficiency, such as achieving tasks with less fatigues.

Considering the capability of animate beings, we propose a self-generating algorithm of behavioral evaluation. In this paper, we use a consumption of energy instead of a concept of fatigues used to achieve tasks. In short, each robot obtains the consumption of energy by using their sensors through a bottom up method. However, if robot use the only consumption of energy, they will not be able to make a cooperative behavior to achieve tasks because each robot selects its own behavior in a selfish manner. In our study, therefore, we assumed that the operator provided rewards for the group of them through a top down method when their task was shared.

Each robot combines rewards and energy consumption, and generates a behavioral evaluation to select the appropriate behavior in the given situation. The following describes the concrete method of generating them.

When behaviors of self and the others are defined by 'a' and 'b' at a certain time, respectively, we can generate the behavioral evaluation E(a,b) using the following equations:

$$E(a,b)=\begin{cases} C(a,b) & R(a,b)=0 \\ C(a,b)-\dfrac{\sum_{i=1}^{n}\sum_{j=1}^{n}C(i,j)}{\sum_{i=1}^{n}\sum_{j=1}^{n}R(i,j)}\times R(a,b) & R(a,b)\neq 0 \end{cases} \quad (1)$$

where C(i,j) and R(i,j) are a consumption of energy used to execute the task and the reward given by the operator, respectively. And 'n' is a number of behaviors that can be selected by the robot.

In this paper, we define that C(i,j) and R(i,j) were modified at every situations. Therefore, the robot can generate the evaluation depending on its situation, even if rewards or consumption of energy is modified.

4 Simulation

We performed simulation experiments in order to study the effectiveness of the proposed algorithm. In this simulation, we applied the algorithm to three robots to confirm that they can share tasks depending on situations.

4.1 Conditions of Simulation

We applied the algorithm as shown in Figure 1 to three robots, named Robot A, B and C. We assumed that they had sensors to observe the others' behaviors. Therefore, each robot can obtain information of the others' current behaviors. This information is not obtained through the communication by both side method. In short, they cannot plan to share tasks in advance.

We assumed that the number of behaviors required to achieve tasks estimated at three, and each robot can select them at will. Each behavior is named Behavior 1, 2 and 3. And we assumed there were two tasks to achieve a purpose of the operator. The operator gives a common reward to each robot when its task is shared by robots. The first task is the case that Behavior 1 is selected by all robots. The second task is the case that they select all of the behaviors. In this case, there are six combinations to obtain a reward, although the combinations are not important for the operator.

Each robot measures the consumption of energy used to execute its own behavior whenever they select the behavior. In this simulation, however, they can not measure it in practice. Therefore, we assumed that each robot consumed the energy of 1, 2 and 3 to execute Behavior 1, 2 and 3 respectively.

All of initial values for behavioral evaluations were set to 0. And initial values used in the learning function were chosen as random parameters.

4.2 Simulation Results

In order to obtain a reward form the operator, each robot has to share tasks depending on situations. However they cannot select appropriate behavior at first because they haven't accumulated experiences to generate their own evaluation. As the result, we confirmed that each robot selected all types of behaviors. And they measured consumption of energy used to execute those behaviors and generated their own behavioral evaluation through their experiences. As the result of leaning based on evaluation, they could create a cooperative behavior when the number of experience trials was 21 times. And they obtain rewards from the operator and repeated its behavioral selection, as shown in Figure 2. We confirmed that each robot understood an appropriate behavior for each other and continued to select Behavior 1.

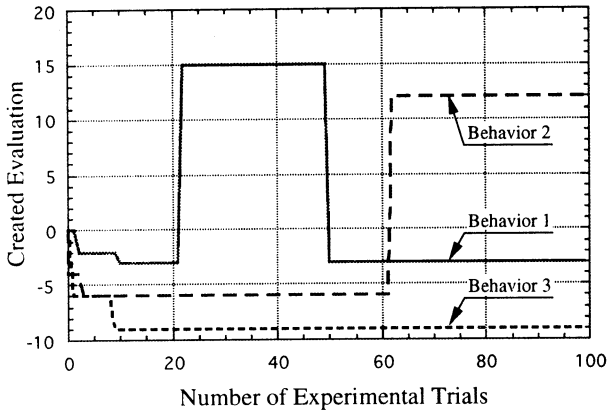
The operator changed the purpose of the task when the number of experience trials passed 50 times. Therefore, each robot was be able to obtain a reword from the operator when each robot selected the behavior differing from others' behavior. However, each robot was not given when they should change to the new behavior and what behavior they should select. Therefore, each robot had to generate the appropriate evaluation by using rewards given the operator, and select its own new behavior by using its generated evaluation.

In this simulation, we confirmed that Robot A and Robot C generated the evaluation as if each behavior was the best for each robot, even if its own behavior was not the best behavior because its consumption of energy was not the least. As a result, they could create a cooperative behavior depending on situations even if the operator changed rewards in the middle of the simulation.

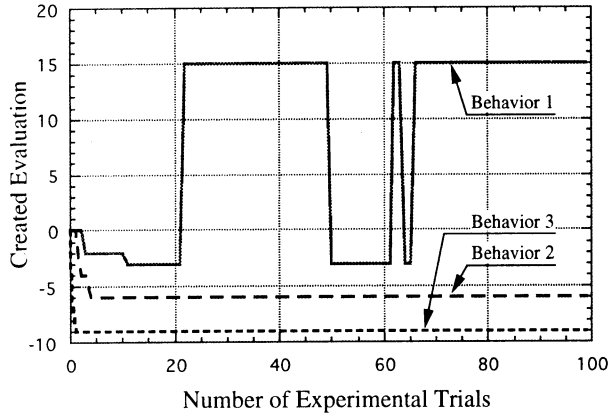
5 Conclusions

This paper proposed a self-generating algorithm of evaluation for cooperative behavior among distributed autonomous robots. This behavioral evaluation is composed of rewards given by the operator and self-evaluated standards acquired as the result of the execution. Each robot learns the others' behaviors by using the evaluation, and select its own behavior. Therefore, robots can create a cooperative behavior as sharing tasks even if the operator gives a common algorithm and reward.

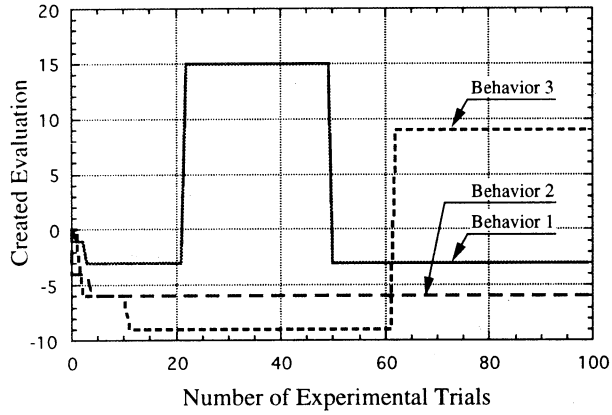
We performed simulations to study the effectiveness of the proposed algorithm. In this simulation, we confirmed that the algorithm resulted in sharing tasks by three robots depending on situations, even if the operator changed the rewards in the middle of the simulation experiment.



(a) Evaluation Generated by Robot A



(b) Evaluation Generated by Robot B



(c) Evaluation Generated by Robot C

Fig. 2 : Simulation Results

Acknowledgments

The authors would like to thank Dr. Arvin Agah for many interesting discussions and helpful comments.

Reference

[1] F. R. Noreils, An Architecture for Cooperative and Autonomous Mobile Robots, IEEE Int'l Conf. on Robotics and Automation, pp.2703-2710 (1992)

[2] T. Fukuda and G. Iritani, Construction Mechanism of Group Behavior with Cooperation, IEEE Int'l Conf. on Robotics and Automation, pp.535-542 (1995)

[3] K. Sekiyama and T. Fukuda, Adaptability Enhancement of Multiple Robots in Dynamical Environment, Proc. of the IEEE Int'l Conf. on Advanced Robotics '95, pp.1656-1661 (1995)

[4] H. Asama, K. Ozaki, H. Itakura, A. Matsumoto, Y. Ishida, and I. Endo, Collision Avoidance among Multiple Mobile Robots Based on Rules and Communication, IEEE Int'l Workshop on Intelligent Robots and Systems '91, pp.1215-1220(1991)

[5] S. Yuta, S. Premvuti, Coordinating Autonomous and Centralized Decision Making to Active Cooperative Behaviors Between Multiple Mobile Robots, IEEE Int'l Workshop on Intelligent Robots and Systems '92, pp.1566-1574 (1992)

[6] Y. Kuniyoshi, S. Rougeaux, and M. Ishii, Cooperation be Observation, -The Framework and Basic Task Patterns-, IEEE Int'l Conf. on Advanced Robotics '94, pp.767-774 (1994)

[7] D. McFarland, Towards Robot Cooperation, From Animal to Animats 3, Int'l Conf. on Simulation of Adaptive Behavior, pp.440-444 (1994)

[8] T. Shibata, K. Ohkawa, and K. Tanie, Spontaneous Coordinated Behavior of Robots through Reinforcement Learning, IEEE Int'l Conf. on Neural Networks '95, pp.2908-2911 (1995)

[9] C. Watkins, Learning from Delayed Rewards, Ph.D. Thesis, Cambridge University (1989)

[10] T. Yamaguchi, M. Miura, and M. Yachida, Cooperative Reinforcement Learning with Spontaneous Mimetism, Int'l Fuzzy Systems Association World Congress, pp.101-104 (1995)

[11] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, Vision-Based Reinforcement Learning for Purposive Behavior Acquisition, IEEE Int'l Conf. on Robotics and Automation, pp.146-153 (1995)

An Analysis of Dynamics of Chaotic Neural Networks and its Application to Combinatorial Optimization

Tohru Ikeguchi† K. Aihara †† M. Hasegawa†

†Department of Applied Electronics
Science University of Tokyo
Chiba 278, Japan

†† Department of Mathematical Engineering and
Information Physics
The University of Tokyo
Tokyo 113, Japan

Abstract

We propose a novel method for combinatorial optimization problems. We combine chaotic neurodynamics which is effective for combinatorial optimization with a heuristic algorithm which is applicable to large scale combinatorial optimization problems. Although the conventional approaches with chaotic neurodynamics were only applied to the problems of at most 10 cities, it is shown that larger problems can be solved by our novel method by chaotic dynamics with high solving abilities.

Keywords: Chaos, Combinatorial Optimazation, Neural Networks

1 Introduction

Deterministic chaos chaos have been discovered in various kinds of sciences, such as fluid systems, chemical reaction systems, control systems, biological systems and so on. As for nerve systems, it has been clarified that both experimentally with squid giant axons and numerically with the Hodgkin - Huxley equations [1] that deterministic chaos is easily and reproducively observed [2]. Then, a model of a chaotic neural networks composed of neuron models with chaotic dynamics has already been proposed [7].

As for one of the applications of the chaotic neural networks, several interesting results are reported on solving combinatorial optimization problems [4, 5, 6]. These approaches are based on the mutual connection neural networks approach proposed by Hopfield and Tank[3]. Although new approaches using chaotic neurodynamics are theoretically expected to be effective for solving the problems of undesirable local minima intrinsic to the mutual connection neural network approach, these methods still have two serious problems. Firstly, if the mutual connection neural network is adopted, $n \times n$ mutual connections should be required in case of n neurons, therefore $n^2 = N^4$ mutual connections for N-city TSP. If N increases, the number of mutual connection gets huge, therefore calculation becomes difficult in a practical situation. Secondly, constructing a closed tour, which is a constraint for solving the TSPs on a neural net approach, is difficult as well as searching the minimum length tour, because the tour is constructed only by the special firing patterns of the neural networks. If the state of neural networks does not satisfy the constraint term, it cannot form even a closed tour.

As for another conventional approach to the TSPs, several heuristic algorithms have been studied [8]. It has been shown that these methods are effective to obtain near optimal solutions of the TSPs. Different from the neural network based approaches, these methods are applicable to large problems, because there is no need to satisfy constraints, which are always satisfied in case of adopting those heuristic algorithms. Moreover, since there is no need to prepare large number of memories, computation amount is smaller than the method based on the mutual connection neural networks. However, the heuristic approaches also have a kind of steepest descent down hill dynamics, so there is a possibility that these algorithms get stuck at undesirable local minima.

In this paper, we adopt chaotic neurodynamics to the local minimum problem of the heuristic algorithms and it is shown that our algorithm can solve large problems effectively in comparison to the conventional neural network based approach.

2 The model of a chaotic neural network

The dynamics of a chaotic neuron with graded output and exponentially decaying refractoriness is de-

scribed by the following equation[7]:

$$x(t+1) = f[A(t) - \alpha \sum_{d=0}^{t} k^d g\{x(t-d)\} - \theta] \quad (1)$$

where $x(t+1)$ is the output which takes an analog value between 0 and 1 at discrete time $t+1$; $A(t)$ is an externally applied input at t; f is the continuous output function; g is the refractory function; α, k, and θ are the scaling parameter of refractoriness, the decay parameter of refractoriness, and the threshold, respectively.

We extended the above neuron model of Eq.(1) to an artificial neural network model, which we call a chaotic neural network, by considering two kinds of input; namely, feedback inputs from component neurons and externally applied inputs[7]. The dynamics of the ith chaotic neuron in the neural network composed of n chaotic neurons can be modeled as follows [7]:

$$\begin{aligned} x_i(t+1) = f_i \Bigg[& \sum_{j=1}^{m} V_{ij} \sum_{d=0}^{t} k_s^d A_j(t-d) \\ & + \sum_{j=1}^{n} W_{ij} \sum_{d=0}^{t} k_m^d h_j\{x_j(t-d)\} \\ & - \alpha \sum_{d=0}^{t} k_r^d g_i\{x_i(t-d)\} - \theta_i \Bigg] \end{aligned} \quad (2)$$

where $x_i(t+1)$ is the output of the ith chaotic neuron at the discrete time $t+1$; $A_j(t-d)$ is the strength of the jth input externally applied at the discrete time $t-d$ and m is the number of the externally applied inputs; V_{ij} is the connection weight from the jth externally applied input to the ith chaotic neuron; W_{ij} is the connection weight from the jth chaotic neuron to the ith chaotic neuron; f_i is the continuous output function of the ith chaotic neuron; h_j is the transfer function of the axon of the jth chaotic neuron for the propagating action potentials; g_i is the refractory function of the ith chaotic neuron. α is the scaling parameter; θ_i is the threshold; k_s, k_m and k_r are the decay parameters for the external inputs, the feedback inputs and the refractoriness, respectively.

By defining three terms in Eq.(2) as internal states by ξ_i,η_i and ζ_i, we can reduce Eq.(2) to the following equations under the assumption of exponential decay[7]:

$$\xi_i(t+1) = k_s \xi_i(t) + \sum_{j=1}^{m} V_{ij} A_j(t), \quad (3)$$

$$\eta_i(t+1) = k_m \eta_i(t) + \sum_{j=1}^{n} W_{ij} h_j\{x_j(t)\}, \quad (4)$$

$$\zeta_i(t+1) = k_r \zeta_i(t) - \alpha g_i\{x_i(t)\} - \theta_i(1-k_r), \quad (5)$$

$$x_i(t+1) = f[\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)]. \quad (6)$$

3 Implementation of a heuristic algorithm to chaotic neural networks

3.1 The algorithm of the 2-opt

The 2-opt is one of the heuristic methods for solving the TSP. The algorithm searches the minimum length tour by changing the rank order of cities. In general, an r-opt algorithm can be defined. In r-opt algorithm, all exchanges of r links are tested until there is no more exchange which improves the current tour length. It is easy to understand that the larger the value r, the more likely it is that the final solution is optimal. However, one of the drawbacks on r-opt algorithm is that the number of executions to test all r order exchanges increases. The reason why we adopt $r = 2$, namely 2-opt algorithm, here is that although 2-opt algorithm is quite simple, it provides near optimum solutions.

Another drawback on 2-opt is that it is very difficult to obtain an exact optimum solution with this algorithm, because this method has only a kind of steepest descent down hill dynamics. Namely, the total tour length only decrease, so the state has many undesirable local minima where the algorithm gets stuck.

On the other hand, it is already known that chaotic dynamics are effective for solving such a local minimum problem [4, 5], so in this paper, we combine the algorithm of the 2-opt with the dynamics of the chaotic neural network model [7].

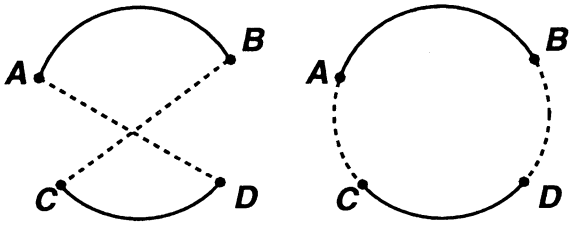


Figure 1: 2-opt example. A, B, C, and D represent cities. Current tour is on the left and tour after exchange is on the right.

3.2 The 2-opt with neural dynamics

In order to implement the 2-opt into the chaotic neural networks, at first, we construct a neural network model which behaves as the conventional 2-opt. On each order-changing process of the 2-opt, there are $n \times n$ ways for constructing new links, so $n \times n$ neurons are prepared and they are arranged on an $n \times n$ grid. In this neural network, the (i,j)th neuron corresponds to a link i-j, that is a link between cities i and j, which means a city j is visited next to a city i. Then, the following equation is defined for a firing of the (i,j) th neuron,

$$x_{ij}(t+1) = D_1(t) - D_{ij}(t), \quad (7)$$

$$\begin{cases} x_{ij}(t) \geq 0 & \text{a link } i\text{-}j \text{ should be connected,} \\ x_{ij}(t) < 0 & \text{no change,} \end{cases} \quad (8)$$

where $D_1(t)$ is a tour length at time t, $D_{ij}(t)$ is the new tour length obtained by applying 2-opt algorithm to make the city j visited next to the city i. This neural network changes its state asynchronously. Let us assume that a city i corresponds to the city A, and a city j to C in Fig.1. In the 2-opt algorithm, if the link A-C should be connected, then the links A-B and C-D must be cut, the link B-D must be connected and the visiting order between the cities B and C must be reversed. Similarly to such a procedure, if the (i,j) th neuron fires at time $t+1$, a city that had been visited next to a city j at time t (corresponding to the city D in Fig.1) must be visited next to the city that had been visited next to a city i at time t (corresponding to the city B in Fig. 1), and the visiting order between the city that was visited next to the city i at time t and the city j must be reversed.

3.3 The 2-opt with chaotic neurodynamics

In this subsection, we modify a neural network version of the 2-opt in the previous subsection to the chaotic neural network version in order to realize chaotic neurodynamical 2-opt. In this paper, we apply external inputs to chaotic neurons in order to control firings for minimizing tour length. To store the minimum lengths in chaotic neural network, $D_1(t) - D_{ij}(t)$ in Eq.(7) can be used as external inputs for the internal states $\xi_{ij}(t)$ of chaotic neurons [7].

Next, in order to control firings of chaotic neurons, we use negative constant weights. Among neurons on the same horizontal or vertical lines on the $n \times n$ grid, only one of them should fire, because only a single city must be visited next to a city, so only a single neuron in a row and a column should fire. Too many executions of the 2-opt algorithm from the city i, which means too many firings in $(i,k)(k = 1, ..., n)$th neurons, leads to too much complicated order changing. Therefore such a constraint for controlling firings is described by the following equation:

$$E_m = \sum_i^n (\sum_k^n x_{ik} - R)^2 + \sum_k^n (\sum_i^n x_{ik} - R)^2, \quad (9)$$

where R is a control parameter for firing rates. It should be noted that Eq.(9) does not necessarily need to be satisfied in a strict sense, because the constraint for forming a tour is already included in the 2-opt algorithm. Namely, Eq.(9) is not an essential aspect of our algorithm, but is introduced for obtaining high solving abilities.

From Eq.(9), the value of the connection weights W^A_{ijkl} between the (i,j) th neuron and the (k,l) th neuron and the value of the threshold a_{ij} of the (i,j) th neuron are obtained as follows,

$$W^A_{ijkl} = -C\{\delta_{ij}(1-\delta_{kl}) + \delta_{kl}(1-\delta_{ij})\}, \quad (10)$$

$$a_{ij} = CR, \quad (11)$$

where C is a positive constant.

Then, in order to obtain more higher solving abilities, we introduce another kind of connections. We make negative connections between the (i,j) th and the (j,i) th neurons. Both neurons are corresponding to the link between cities i and j. So if a link between i and j are to be connected, both neurons would fire, which will make neural network dynamics complicated. In order to avoid such a situation, the following connection is introduced:

$$W^B_{ijkl} = -B\delta_{il}\delta_{jk}, \quad (12)$$

where B is a positive constant.

From the chaotic neural network model [7] and Eqs. (10)~(12), the final order changing rules which we propose are describe by the following equations:

$$\xi_{ij}(t+1) = k_s\xi_{ij}(t) + h(D_1(t) - D_{ij}(t)), \quad (13)$$

$$\eta_{ij}(t+1) = k_m\eta_{ij}(t) - C\sum_{l=1}^n x_{il}(t) - C\sum_{l=1}^n x_{lj}(t) - Bx_{ji}(t). \quad (14)$$

$$\zeta_{ij}(t+1) = k_r\zeta_{ij}(t) - \alpha x_{ij}(t) + CR, \quad (15)$$

$$x_{ij}(t+1) = f[\xi_{ij}(t+1) + \eta_{ij}(t+1) + \zeta_{ij}(t+1)], \quad (16)$$

where $\xi_{ij}(t), \eta_{ij}(t)$ and $\zeta_{ij}(t)$ are internal states of externally applied inputs, feedback inputs, refractoriness

of the (i,j) th neurons, respectively; $x_{ij}(t)$ is the output of the (i,j) th neuron. If an output $x_{ij}(t)$ is larger than a threshold θ, it can be considered that the (i,j) th neuron fires, and the order is changed so as to satisfy that a city j is visited next to a city i.

4 Numerical Results on TSPs

The results of our method applied to a 105-city problem, Lin105 [9], are summarized in Fig.2, with $k_m = 0.0, k_s = 0.0, R = 1.75, \epsilon = 0.001, C = 0.00125, B = \alpha/2$, $h(z) = z, \theta = \frac{1}{2}$ and changing the value of k_r and α.

They are compared with the results of stochastic dynamics which are realized by replacing Eq.(15) by the random neuron model [5].

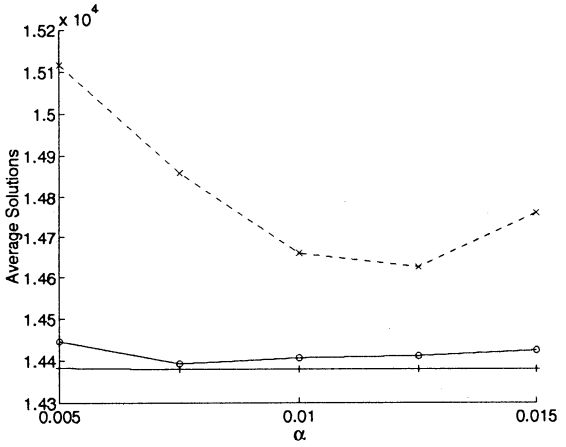


Figure 2: Results for a 105-city problem (Lin105) with chaotic and random neurons in case of changing α. The solutions of each trials are the best solutions in 10,000 iterations and averaged on different initial conditions. Solid line with circles is for $k_r = 0.925$ and with pluses for $k_r = 0.95$. Dashed line with crosses is for random neurons.

From Fig.2, it is clear that using chaotic dynamics leads to higher solving abilities than the stochastic dynamics. Furthermore, the optimum solution, 14,379, can be obtained 100 % by our novel method with several parameters. Although the method using random neurons cannot search the optimum solution every trial with 10,000 iterations, the average iteration for obtaining the optimum solution using chaotic neurons with $k_r = 0.95$ and $\alpha = 0.015$ is only 463.5.

Our method is also applied to a 318-city problem, Lin318 of Ref. [9], and 42,334 is obtained with $k_r = 0.85, k_m = 0.2, k_s = 0.0, R = 0.9, \epsilon = 0.003, \alpha = 0.09, C = 0.09, h(z) = 0.075z, B = 0.045$. Although the solution 42,334 is not optimum, we believe that our method can solve this problem or the larger problems, if we find the optimal parameter values.

In this paper, in order to obtain higher solving abilities, we introduce one more improvement of our method: in each iteration, the original 2-opt method is applied, and the tour form is improved in each iteration. Then, 42,190 is obtained with $k_r = 0.85, k_m = 0.2, k_s = 0.0, R = 0.9, \epsilon = 0.003, \alpha = 0.09, h(z) = 0.1z, B = 0.045, C = 0.09$, with 10,000 cut off time.

5 Conclusion

In this paper, we combine the advantages of novel modern heuristic approaches, which uses chaotic neurodynamics for avoiding undesirable local minima, and the advantages of conventional heuristic methods, which is applicable to large problems. Our neural network model does not necessarily have huge mutual connections, so the size of the neural network can become small, then this framework can be applied to large problems. Furthermore, our neural networks can always construct a closed tour because the constraint for forming a closed tour is already included in the heuristic algorithm. Therefore our method doesn't make a problem complicated.

It is very important to develop an effective algorithm for deciding the parameters to obtain the best performance.

References

[1] A. L. Hodgkin and A. F. Huxley : J. Physiol.(London), 117, 500, 1957.

[2] e.g. K. Aihara and G. Matsumoto : J. Theor. Biol., 95, 697, 1982.

[3] J. J. Hopfield and D. W. Tank: Biol. Cybern., 52, 144, 1985.

[4] H. Nozawa: Chaos, 2, 3, 377, 1992.

[5] T. Yamada, K. Aihara and M. Kotani: Proc. of IEICE NOLTA, 4, 10.3-6, 1173, 1993.

[6] L. Chen and K. Aihara: Neural Networks, 8, 6, 915, 1995.

[7] K. Aihara, et al.:Phys. Lett. A, 144, 333, 1990; K.Aihara: in "Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems," ed. H.Kawakami, 143, World Scientific, 1990.

[8] A.Croes: Opns. Res., 5, 791, 1958; S.Lin: BSTJ, 44, 2245, 1965; S.Lin and B.W.Kernighan: Opns. Res., 21, 498, 1973.

[9] G. Reinelt: "TSPLIB," available via http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html.

Numerical Study of Resonance in a Sinusoidally Driven Chaotic Neuron and Its Network

Shin Mizutani† Takuya Sano‡ Tadasu Uchiyama†
Noboru Sonehara† Katsunori Shimohara†

† NTT Human Interface Laboratories, ‡ NTT Optical Network Systems Laboratories
1-1 Hikarinooka, Yokosuka-Shi, Kanagawa, 239 Japan

Abstract

We study numerically stochastic resonance (SR) like phenomenon in a chaotic neuron model and its global coupling network driven by a weak sinusoid. This resonance phenomenon has a peak at a drive frequency as noise induced SR. However, it has a different mechanism from noise induced SR. Mean field of a summing network can enhance the signal-to-noise ratio (SNR) by the law of large numbers, while mean field of a global coupling network does not enhance the SNR. We discuss the mechanism which occurs resonance in the model.

1 Introduction

Background noise is considered a nuisance. Engineers have always striven to reduce the noise, but in some nonlinear systems, background noise yields the benefit of weak signal detection. This phenomenon is called stochastic resonance (SR) [1, 2, 3]. The basic SR mechanism is synchronization and resonance to an external periodic signal when a bistable system is driven by a subthreshold sinusoid plus some amount of noise. Only a subthreshold signal can not overcome the potential barrier of the bistable system, but the presence of noise helps to move between wells of bistable potential. The power spectrum of the system's time series shows peaks at a drive signal frequency and its harmonics. This indicates that SR has enhanced the detection of weak signal from background noise. Threshold systems also reveal SR, but the result is noise induced threshold crossing rather than resonance because the enhancement occurs in any drive signal frequency [4, 5]. Therefore, excitable media also reveal SR [6]. For these reasons, neuron models, for example, the FitzHugh-Nagumo model [7] and so on, have SR behavior. This SR effect is found in the real sensory neurons of crayfish and obviously provides the useful sensory function of segmentation of noisy backgrounds [8].

In this report, we consider SR in a chaotic system. Few papers have addressed SR in chaotic systems, Nicolis et al. [9] did not report any obvious peak in power spectrum, while Anishchenko et al. [10, 11] considered only noise induced SR. The basic mechanism is bimodal chaos or chaos-chaos intermittency synchronized to an external sinusoid. SR is found in bistable systems of not only fixed points but also any attractor even chaos.

If noise in biological systems can be regarded as deterministic chaos, we can suppose that biological systems may take advantage of resonance by generating chaos as effective noise in a spontaneous manner. Here, we study a chaotic neuron model and its global coupling network driven by a weak sinusoid. The chaotic neuron model and its network were proposed by Aihara et al. and is based on their studies of squid giant axons and the Hodgkin-Huxley equation [12]. The networks of the neuron model have been intensively studied for information processing, especially for escaping local minima for optimization and retrieving associative memory [13, 14, 15]. Recently, some chaos control techniques have also been applied to the network to stabilize the chaotic retrieval process [16]. The neuron model may mirror a threshold system like the McCulloch-Pitts model and an excitable system like the FitzHugh-Nagumo model, but is chaotic rather than bistable. This was the first simple neuron model to provide chaotic response; some parameter combinations trigger chaotic output.

In this report, we show that the chaotic neuron model driven by a weak sinusoid also has obvious peaks in the power spectrum as noise driven SR. The resonance mechanism is different from noise driven SR because the model does not have chaotic multi-attractors

simultaneously at a certain parameter set causing chaos-chaos intermittency as the previous case. Next, we show the dependency of signal-to-noise ratio (SNR) to change a model parameter and the drive signal parameters. After the neuron model results, we show that SNR and its dependency of network size (a number of neurons) in the summing network and also show that SNR and its dependency of coupling weight and a model parameter in the global coupling network. Finally, we conclude the results and discuss the mechanism of resonance.

2 Chaotic Neuron Model and Its Global Coupling Network

The simplest chaotic neuron model has the following dynamics.

$$\begin{aligned} y(t+1) &= ky(t) - \alpha f(y(t)) + a, \\ x(t+1) &= f(y(t+1)), \\ f(z) &= \frac{1}{1+exp(-\frac{z}{\epsilon})}, \end{aligned} \tag{1}$$

where
$y(t)$: internal state of neuron at time t,
$x(t)$: output of neuron at time t,
k : refractory time coefficient,
α : scaling factor for the refractory term ($\alpha \geq 0$),
a : parameter based on the input and the threshold,
ϵ : steepness parameter of sigmoid function.
We investigate this model for simplicity. The neuron has chaotic output when input parameter a yields a positive Lyapunov exponent. A bifurcation diagram and the Lyapunov exponent as a function of parameter a is shown in Ref. [16]. We fix the model parameters at $k = 0.7, \alpha = 1.0, \epsilon = 0.02$ throughout this paper.

This paper studies a homogeneous diffusive global coupling network with homogeneous input. The network has following equation.

$$\begin{aligned} y_i(t+1) &= ky_i(t) - \alpha f(y_i(t)) \\ &+ \sum_{j=1}^{N} w\{f(y_j(t)) - f(y_i(t))\} + a, \\ x_i(t+1) &= f(y_i(t+1)). \end{aligned} \tag{2}$$

3 Fundamental Mechanism of Stochastic Resonance

SR is a noise induced effect in nonlinear systems. Consider an over-damped ball $x(t)$ moving in a double-well potential $U(x)$

$$\frac{dx}{dt} = -\frac{\partial U}{\partial x} + F(t) + A\sin\omega_0 t \tag{3}$$

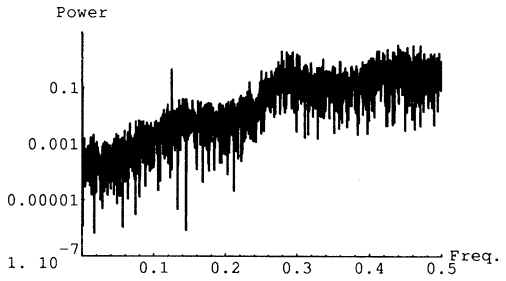
with a weak time-periodic signal $A\sin\omega_0 t$. The ball is weakly forced sinusoidally in time, but can not overcome the potential barrier. It requires a certain mount of force to surmount the barrier. If we add noisy fluctuated force $F(t)$, the ball can occasionally jump over the barrier. Here, $F(t)$ is Gaussian white noise ($\langle F(t)\rangle = 0, \langle F(t)F(t')\rangle = 2D\delta(t-t')$). The time series $x(t)$ of surmounting events has a peak $S(\omega_0)$ at sinusoid frequency ω_0 in the power spectrum. This peak does not appear when the noise is too weak or too large, because the surmounting events does not occur or occurs at random, respectively. There is an optimum amount of noise for any given SNR (for example, SNR$= 10\log_{10}(S(\omega_0)/N)$.). This phenomenon is called SR. Background noise enhances the detection of a weak time-periodic signal. SR also appears in excitable systems and threshold systems.

4 Stochastic Resonance in a Chaotic Neuron Model

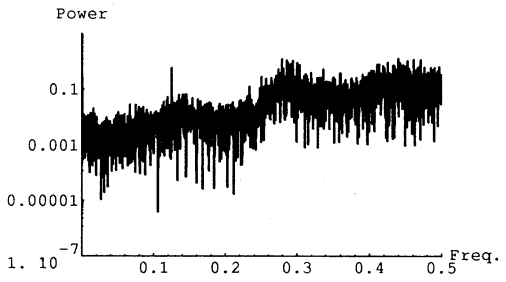
We investigated the map with adding $A\sin 2\pi f_m t$ to a chaotic neuron map in Eq. (1). T_n is n time steps for calculating a power spectrum and f_m is the frequency of the drive signal. We chose the parameters for the chaotic regime as $a = 0.35, A = 0.005, f_m = 0.125, T_n = 4096$. We show typical power spectra of time series of $x(t), y(t)$ obtained by discarding transient time steps. Both spectra have sharp peaks at drive frequency $f_m = 0.125$ in Fig. 1. Higher harmonics are observed when signal amplitude increases. For the sigmoid function, similar to threshold systems, the power spectrum of $x(t)$ has a higher peak than $y(t)$. We investigated the resonance of $y(t)$ rather than $x(t)$ to study a chaotic neuron model as a nonlinear map without the sigmoid function effect.

We calculated the dependence of the SNR on input parameter a, amplitude A and frequency f_m as drive signal parameters. We defined the SNR in time series $y(t)$ as,

$$\text{SNR} = 10\log_{10}(\frac{S(f_m)}{N}). \tag{4}$$

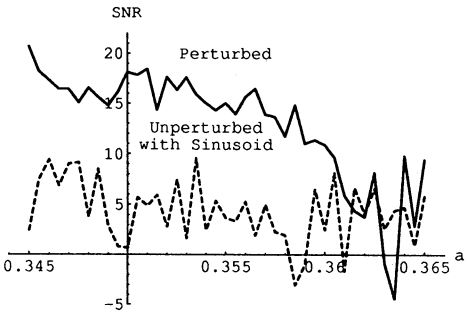


(a) Power spectrum of $x(t)$ time series

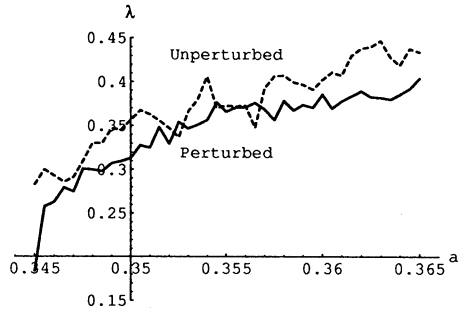


(b) Power spectrum of $y(t)$ time series

Figure 1: Power spectra of a chaotic neuron model driven by a weak sinusoid



(a) SNR vs input parameter a



(b) Lyapunov exponent vs input parameter a

Figure 2: SNR as a function of input parameter $a = [0.345, 0.365]$. In Fig. (a), the solid line shows the SNR for perturbed time series and the dotted line shows the SNR for unperturbed time series with the addition of a weak sinusoid. In Fig. (b), the solid line shows the Lyapunov exponent for perturbed time series and the dotted line shows the Lyapunov exponent for unperturbed time series (weak sinusoid is not added).

Here, $S(f_m)$ is the peak power at drive signal frequency and N is the noise power obtained by averaging 40 frequency points around the signal frequency. We used the same parameters as above except when investigating the dependency on its parameter.

First, Figure 2 shows the SNR vs input parameter a of the model. We chose the region of $a = [0.345, 0.365]$ for increasing the Lyapunov exponent of the unperturbed model as in Fig. 2(b). The perturbed system offers improved SNR compared to unperturbed time series with the addition of weak sinusoid. It has a tendency that there is a range of optimum Lyapunov exponent, which the SNR is improved. However, this situation may simply depend on the amplitude of the chaotic time series, not on the Lyapunov exponent.

Next, Figure 3 shows the SNR vs amplitude A of the drive signal. The perturbed system also offers higher SNR than the unperturbed time series with weak sinusoid. The perturbed system becomes nonchaotic when the drive amplitude still increases.

Finally, we show the SNR vs frequency f_m of the drive signal in Fig. 4. Some frequencies (considered optimum in this paper) yield quite high SNR values. Obviously, this model contains some mechanism which resonates given a weak drive signal. The SNR of the perturbed system is high except in several frequency regions; we note that these regions match those in which the power spectrum of the unperturbed system (without weak sinusoid) takes high values.

5 Stochastic Resonance in a Chaotic Neural Network

We investigated the following chaotic neural network driven by a weak sinusoid.

$$y_i(t+1) = ky_i(t) - (\alpha + Nw)f(y_i(t))$$

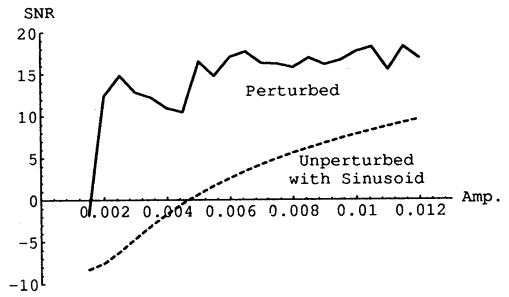


Figure 3: SNR as a function of drive amplitude A. Solid and dotted lines follow Fig. 2.

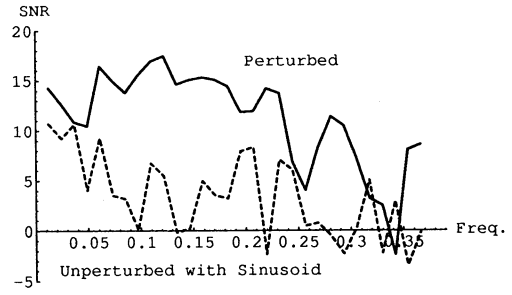
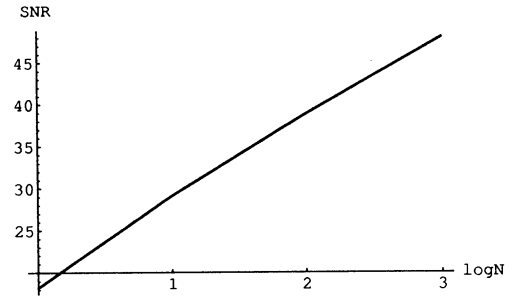


Figure 4: SNR as a function of drive frequency f_m. Solid and dotted lines follow Fig. 2.

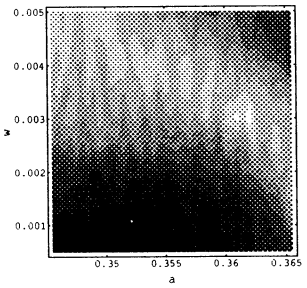
$$+ \sum_{j=1}^{N} wf(y_j(t)) + a + A\sin 2\pi f_m t. \quad (5)$$

Each neuron in a network has same input a and driven by a weak sinusoid $A\sin 2\pi f_m t$. We notice the time series of the mean field $m(t) = \frac{1}{N}\sum_{i=1}^{N} y_i(t)$ and its power spectrum. We classify the network conditions into two cases; one is a summing network and the other is a coupling network.

We calculated the dependency of the SNR against changing network size N and a network parameter, weight w and a neuron parameter, input a at $N = 100$ in Fig. 5. We use the same parameters as the previous case. The SNR is more improved when network size N increases in Fig. 5 (a) [17]. This effect is understood by the law of large numbers. The SNR is decreased by increasing weight w in Fig. 5 (b). We could not observe enhancement in the mean field by global coupling [18].



(a) SNR as a function of network size N



(b) SNR vs weight w and input a at $N = 100$

Figure 5: SNR as a function of network size N and a function of weight w and input a. Shading shows the SNR.

6 Conclusions and Discussions

We investigated a chaotic neuron model and its global coupling network driven by a weak sinusoid by numerical calculations. Our results revealed noise driven SR-like behavior with a power spectrum peak at a drive frequency. The SNR depends on the model parameters and the drive signal parameters in the both cases of the neuron model and the neural network.

We discuss the mechanism of this phenomenon. First, there is the possibility that the approximated system may have a bistable potential. The Hopfield type neuron model has a bistable potential in a certain case [19]. The chaotic neuron model does not have bistable potential as the Hopfield type even if the model with the given parameter set is approximated by a differential equation ($dx/dt = x(t+1)-x(t)$). The SNR of the approximated system can not be estimated by adiabatic theory [20]. Second, it may be explained as noise induced threshold crossing; the equation of the model has the sigmoid function like threshold systems. The SNR of threshold systems is independent

of drive frequency, but the model is dependent on frequency. Third, and more interesting, the perturbation can be regarded as a slow change in a bifurcation parameter a (input parameter). If the phase structure of an attractor is sensitive to the parameter, the slow perturbation, slow compared to the time scale of the attractor, derives a resonance phenomenon. There are some reports that some chaotic dynamics are suppressed by resonant parametric perturbations or weak sinusoid perturbations [21, 22]. To decide whether this phenomenon is *bona fide* resonance or not [4, 5], it is also important to investigate the dependence of a histogram of interspike intervals or residence time on the drive frequency. Finally, we mention that it is also interesting to investigate enhancement by nearest neighbour coupling as array enhanced SR [23].

Acknowledgements

The authors are grateful to Dr. Yukio Tokunaga of NTT Human Interface Laboratories for supporting this research. Thanks are also due to Prof. Kazuyuki Aihara, Dr. Takehiko Horita, Dr. Masaharu Adachi and Dr. Shin Ishi for their valuable suggestions.

References

[1] K. Wiesenfeld and F. Moss. "Stochastic Resonance and the Benefits of Noise: From Ice Ages to Crayfish and SQUIDs". *Nature*, 373:33–36, 1995.

[2] F. Moss and K. Wiesenfeld. "The Benefits of Background Noise". *Sci. Am.*, 273:66–69, 1995.

[3] A.R. Bulsara and L. Grammaitono. "Tuning in to Noise". *Phys. Today*, 49:39–45, 1996.

[4] L. Grammaitono, F. Marchesoni, and S. Santucci. "Stochastic Resonance as *Bona Fide* Resonance". *Phys. Rev. Lett.*, 74:1052–1055, 1995.

[5] L. Grammaitono. "Stochastic Resonance and the Dithering Effect in Threshold Physical Systems". *Phys. Rev. E*, 52:4691–4698, 1995.

[6] P. Jung and G. Mayer-Kress. "Spatiotemporal Stochastic Resonance in Excitable Media". *Phys. Rev. Lett.*, 74:2130–2133, 1995.

[7] A. Longtin. "Stochastic Resonance in Neuron Models". *J. Stat. Phys.*, 70:309–327, 1993.

[8] J. K. Douglass, L. Wilkens, E. Pantazelou, and F. Moss. "Noise Enhancement of Information Transfer in Crayfish Mechanoreceptors by Stochastic Resonance". *Nature*, 365:337–340, 1993.

[9] G. Nicolis, C. Nicolis, and D. McKernan. "Stochastic Resonance in Chaotic Dynamics". *J. Stat. Phys.*, 70:125–139, 1993.

[10] V. S. Anishchenko, A. B. Neiman, and M. A. Safanova. "Stochastic Resonance in Chaotic Systems". *J. Stat. Phys.*, 70:183–196, 1993.

[11] V. S. Anishchenko, M. A. Safanova, and L. O. Chua. "Stochastic Resonance in Chua's Circuit". *Int. J. Bifur. Chaos*, 2:397–401, 1993.

[12] K. Aihara, T. Takabe, and M. Toyoda. "Chaotic Neural Networks". *Phys.Lett. A*, 144:333–340, 1990.

[13] H. Nozawa. "A Neural Network Model as a Globally Coupled Map and Applications Based on Chaos". *Chaos*, 2:377–386, 1992.

[14] M. Hasegawa, T. Ikeguchi, T. Matozaki, and K. Aihara. "Solving Combinatorial Optimization Problems by Nonlinear Neural Dynamics". In *Proc. Int. Conf. on Neural Net.*, pages 3140–3145, 1995.

[15] M. Adachi, K. Aihara, and M. Kotani. "Nonlinear Associative Dynamics in a Chaotic Neural Network". In *Proc. 2nd Int. Conf. Fuzzy Logic and Neural Network*, pages 947–950, 1992.

[16] S. Mizutani, T. Sano, T. Uchiyama, and N. Sonehara. "Controlling Chaos in Chaotic Neural Networks". In *Proc. Int. Conf. on Neural Net.*, pages 3038–3043, 1995.

[17] J. J. Collins, C. C. Chow, and T. Imhoff. "Stochastic Resonance without Tuning". *Nature*, 376:236–238, 1995.

[18] M. E. Inchiosa and A. R. Bulsara. "Coupling Enhanced Stochastic Resonance in Nonlinear Dynamic Elements Driven by a Sinusoid Plus Noise". *Phys. Lett. A*, 200:283–288, 1995.

[19] W. C. Schieve, A. R. Bulsara, and G. M. Davis. "Single Effective Neuron". *Phys. Rev. A*, 43:2613–2623, 1991.

[20] B. McNamara and K. Wiesenfeld. "Theory of Stochastic Resonance". *Phys. Rev. A*, 39:4854–4869, 1989.

[21] R. Lima and M. Pettini. "Suppress of Chaos by Resonant Parametric Perturbations". *Phys. Rev. A*, 41:726–733, 1990.

[22] Y. Braiman and I. Goldhirsh. "Taming Chaotic Dynamics with Weak Periodic Perturbations". *Phys. Rev. Lett.*, 66:2545–2548, 1991.

[23] J. F. Linder, B. K. Meadows, W. L. Ditto, M. E. Inchiosa, and A. R. Bulsara. "Array Enhanced Stochastic Resonance and Spatiotemporal Synchronization". *Phys. Rev. Lett.*, 75:3–6, 1995.

Analog Integrated Chaotic Neuron Circuit and Its Applications

Yoshihiko Horio† and Ken Suyama‡

†Department of Electronic Engineering
Tokyo Denki University, Tokyo, 101, Japan

‡Department of Electrical Engineering
Columbia University, New York, NY 10027, U.S.A

Abstract

The chaotic neuron model [1, 2] is implemented as an integrated circuit form with 2 μm CMOS technology using a switched-capacitor circuit technique. Measurements from the IC chip show that the proposed circuit qualitatively replicates the response of the real neuron. As an illustration, the integrated neurons are used in a combinatorial optimization network to solve a 4-city traveling salesman problem. Moreover, a dynamical associative memory is constructed using the chip.

Keywords: Chaos, Neural Networks, Nonlinear Integrated Circuits.

1 Introduction

Ever since artificial neural networks were suggested [3] and demonstrated [4] in a context of modern VLSI technology, various methods have been proposed in an attempt to search for practical ways to realize artificial neural networks in a monolithic form. Most of these methods are based on a neuron model which consists of simple elements such as threshold or sigmoidal nonlinear functions, summers, and multipliers.

On the other hand, the existence of chaos has been confirmed through physiological experiments on a single neuron [5]–[12] as well as a cluster of neurons [13]–[17]. These reports seem to support the hypothesis that chaos plays an essential role in signal and information processing of biological systems [1, 2, 12, 15]–[22].

Chaotic neural network, which consists of a neuron model with similar nonlinear dynamics observed in squid's giant axons [1, 2, 12], has recently been proposed as a possible implementation scheme for artificial neural networks [1, 2, 18]. It has been shown that chaotic neural network bears chaotic search ability, which helps its solution to escape from local minima and to reach global solution efficiently. This ability is an advantage over artificial neural networks with conventional neurons where they suffer the well-known local minima problem. This chaotic search ability has been observed in engineering applications such as dynamical associative memory [23]–[28] and optimization problems [29]–[32].

Mixed analog-digital VLSI implementation of chaotic neural networks can potentially lead to an attractive alternative for engineering applications such as optimization and dynamical associative memory. Such hardware implementations can also relieve researchers from lengthy computer simulation when they investigate complicated dynamics of even a small chaotic neural network.

This paper briefly discusses the design and performance of an IC implementation of a switched-capacitor (SC) chaotic neuron [33, 34]. Experimental verification of the integrated chaotic neuron will be presented. A 4-city traveling salesman problem and a dynamical associative memory will be demonstrated as engineering applications of the integrated neurons.

2 Chaotic Neuron Model

The chaotic neuron model can be described by the following difference equations [1, 2, 12]:

$$y(t_{n+1}) = ky(t_n) - \alpha f(y(t_n)) + U(t_n) + \theta \quad (1)$$

$$x(t_{n+1}) = f(y(t_{n+1})) \quad (2)$$

where $y(t_n)$ is the internal state of the neuron at discrete time $t_n = nT_c$ where T_c is a clock period, $U(t_n)$ is the sum of stimulations from external inputs and other neurons, θ is an external bias, α is a parameter for refractoriness, k is the damping factor of refractoriness, $f(\cdot)$ is a continuous nonlinear function, and $x(t_n)$ is the analog output of the neuron.

3 Switched-Capacitor Circuit Implementation of Chaotic Neuron Model

A SC circuit implementation of the difference equations (1) and (2) is shown in Fig. 1 [33]. Equation (1) is implemented by a SC integrator with the refractoriness parameters expressed by the following capacitor ratios:

$$k = 1 - \frac{C_k}{C_i}, \qquad \alpha = \frac{C_f}{C_i} \quad (3)$$

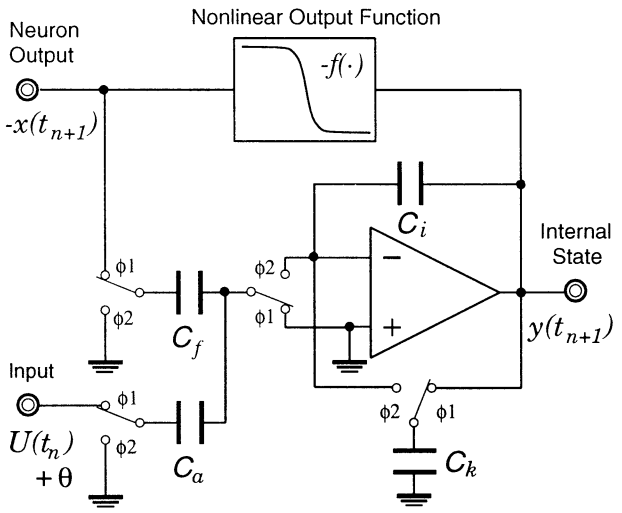


Figure 1: Switched-capacitor chaotic neuron circuit.

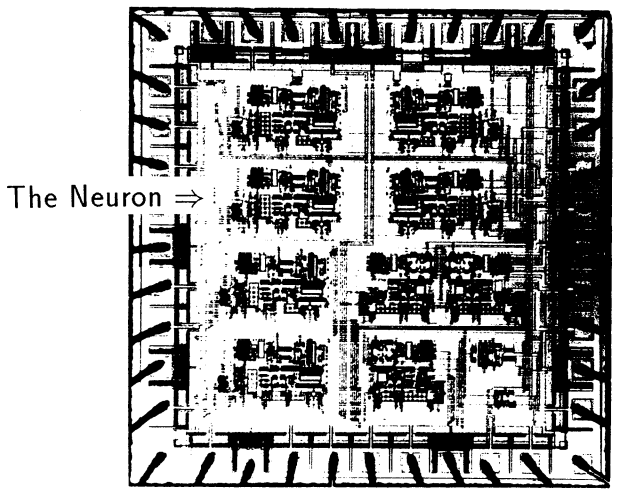


Figure 2: Microphotograph of the prototype SC chaotic neuron chip.

where $k = 0.8$ and $\alpha = 1.0$ were used for this design. These values should be chosen such that the neuron produces response with rich dynamics. Extensive numerical simulation [1, 33] suggested that the values chosen satisfied the requirement. The ratio C_i/C_a is set to 1 so that the input $U(t_n) + \theta$ in Fig. 1 is equivalent to $U(t_n) + \theta$ in (1). $\phi 1$ and $\phi 2$ are two non-overlapping clock waveforms with period T_c. The circuit for the nonlinear output function, $f(\cdot)$, consists of a CMOS inverter, followed by a saturation limiter and a push-pull source follower [33].

3.1 IC Implementation

A photomicrograph of the prototype chip is shown in Fig. 2 [34]. A standard 2-μm double-metal, double-poly, n-well analog CMOS technology [35] was used. Nine SC chaotic neurons were fabricated in the chip. There are actually two types of SC neurons for exploring different design methodologies. The neuron described in this paper is marked in the figure. It occupies the active area of 0.14 mm^2 and actual capacitor sizes are $C_i = C_f = C_a = 1$ pF and $C_k = 0.2$ pF. The supply voltages were ± 5 V. It should be emphasized that no effort was done to optimize the active area for this test chip.

3.2 Measurements

The test setup consists of a clock generation circuit (100 kHz), a sample-and-hold circuit for obtaining one dimensional map of $y(t_n)$ and $y(t_{n+1})$, and a data acquisition circuit. The network is deliberately clocked slowly so that the data acquisition circuit can function properly.

Figure 3(a) shows the measured bifurcation structure of the internal state, $y(t_n)$, of the SC chaotic neuron where the external bias θ was swept from -2.5 V to 2.5 V. In the figure, 300 points are plotted for each value of θ.

Figures 3(b) and (c) show the average firing rate and the Lyapunov exponent, respectively. The Lyapunov exponent and the average firing rate are defined as follows:

$$\lambda = \lim_{N \to +\infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln \left| \frac{dy(t_{n+1})}{dy(t_n)} \right| \quad (4)$$

$$\rho = \lim_{N \to +\infty} \frac{1}{N} \sum_{n=0}^{N-1} h(x(t_n)) \quad (5)$$

where $h(\cdot)$ is a threshold function, which models waveform shaping dynamics of axon [1, 2, 12] (threshold voltage is 0 V), and N is the total number of data points. Note that there are some errors in estimation of the Lyapunov exponents [36]–[38] due to a finite number of data points and other nonideal effects associated with measurements. We are currently developing an algorithm to estimate the Lyapunov exponents accurately from measured data [38].

Various experiments using the integrated SC chaotic neurons have been carried out [27, 31, 32, 37, 39]. In this paper, a 4-city traveling salesman problem (TSP) and a dynamical associative memory using 16 SC chaotic neurons are briefly demonstrated to show the effectiveness of the IC implementation.

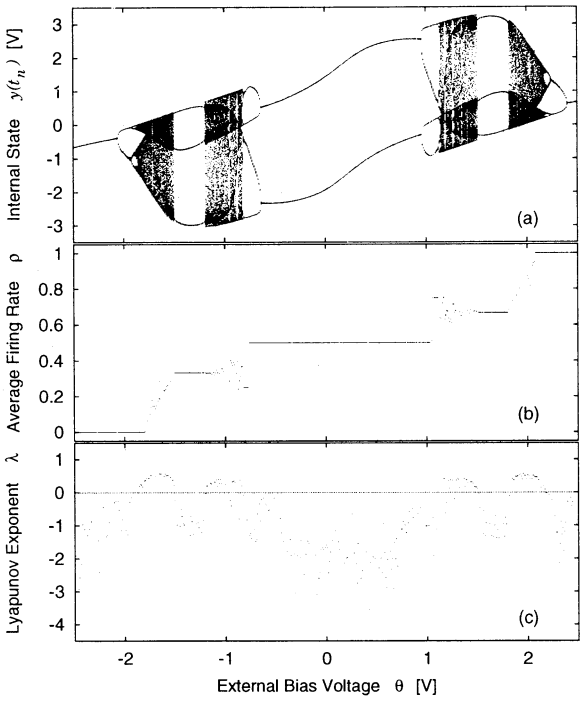


Figure 3: Measured characteristics of the integrated neuron, (a) bifurcation structure, (b) the average firing rate, and (c) the Lyapunov exponent.

4 Traveling Salesman Problem

The TSP problem is formulated by a permutation matrix used in [3]. The coordinates of the first four cities from the Hopfield and Tank data [40] are used here. Figure 4 describes the experimental setup for the TSP network. The comparators are used to model the waveform shaping dynamics of axon. Synaptic weights $w_{ij,lm}$ are implemented by 256 8-bit Digital-to-Analog converters (DACs) where each weight is set sequentially by a computer through the data acquisition board. Initial conditions can be specified through INIT blocks, which consist of switches as shown in the inset of Fig. 4. The synaptic weights and the bias value θ can be obtained by:

$$w_{ij,lm} = -A\delta_{il}(1-\delta_{jm}) - B\delta_{im}(1-\delta_{il}) - Dd_{il}(\delta_{m,j+1}+\delta_{m,j-1}) \quad (6)$$

$$\theta = R(A+B) \quad (7)$$

where $w_{ij,lm}$ is a synaptic weight from neuron lm to neuron ij, A, B and C are constants, $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ otherwise, and R is the firing rate parameter. R is a scaling factor, which can modify the bias value θ for a given set of A and B, in order to vary the dynamics of the network [29, 30].

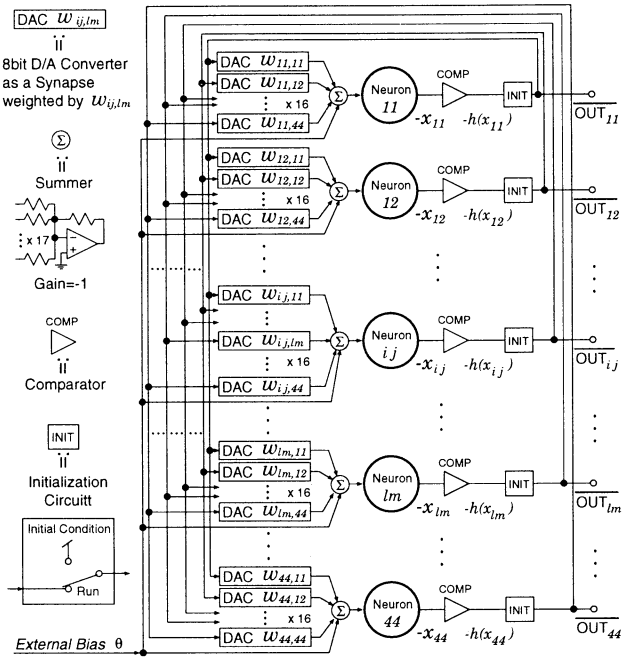


Figure 4: Experimental setup for a 4-city TSP network using the chaotic neuron chips.

4.1 Experimental Results

After the DACs (see Fig. 4) were programmed using the weights computed by (6) for a given problem and an appropriate value of R was chosen, we tested the performance of the network by setting initial conditions randomly. For each initial condition, the outputs were observed and stored for the first 10,000 iterations. The rates of convergence to the best path and the feasible paths for different values of the firing rate parameter R were computed with the 10,000 data points per *one* initial condition where a set of 100 randomly generated initial conditions were used. The results were plotted in Fig. 5 [32]. The "Feasible Paths" in the figure represents both local minima and the global solution.

In order to reveal the underlying dynamics of the TSP network, only the first 10,000 data points were purposely chosen to calculate the convergence rate while the rate would have reached 100 percent for most of the values of R in the range shown in Fig. 5, if the network were allowed to run longer. The results suggest that the chaotic neural network has excellent characteristics to find the global solution. The chaotic searching ability of the network can be adjusted by the firing rate parameter R which changes the dynamics of the network as shown in Fig. 5. We are currently studying the reasons for having such a characteristic

of the convergence rates versus R. More information is available in Ref. [32].

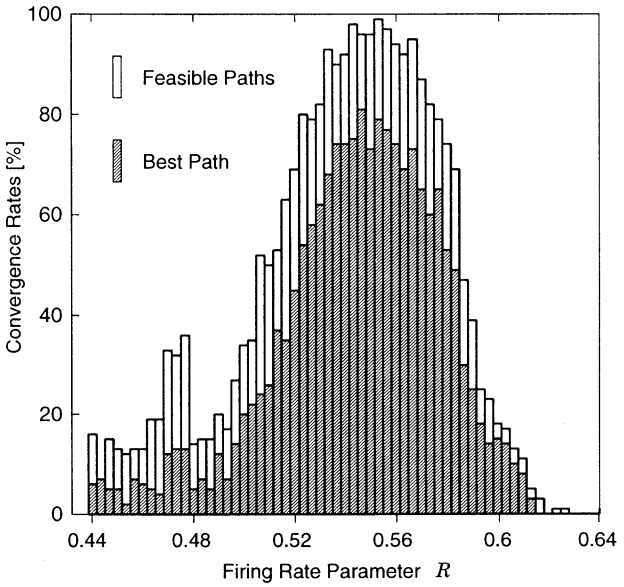


Figure 5: Convergence rates of the 4-city TSP network.

5 Dynamical Associative Memory

The dynamical associative memory in this paper is defined as an associative network composed of chaotic neurons [1, 2]. The network produces chaotic temporal patterns when the refractory effect is stronger than the mutual interactions between neurons [24, 26, 28]. If we store some patterns by the auto-correlation matrix as given in (8), the network "wanders around" the stored patterns and does not stay at the equilibrium because of the chaotic dynamics. This behavior is similar to the chaotic itinerancy [19, 21]. A weight matrix $\boldsymbol{W}$ can be computed by:

$$\boldsymbol{W} = \sum_{n=1}^{p} \boldsymbol{S}_n \boldsymbol{S}_n^t - p\boldsymbol{I}, \tag{8}$$

where $\boldsymbol{S}_n$ is a bipolar binary pattern vector which is stored in the network, p is the number of the patterns, and $\boldsymbol{I}$ is a unit matrix.

We constructed a Hopfield-type [3] 4×4 associative memory network, which is similar to the one in Fig. 4, using SC chaotic neurons in the IC chips. In the following subsection, results from one of the three experiments [27], which have been done using this experimental set up, is described. For the remaining experimental results, see Ref. [27]. The similar experiments have been demonstrated in Refs. [23, 24, 26, 28] with numerical simulations.

5.1 Experimental Results

Three patterns shown in Fig. 6 were stored in the network according to (8). In the figure, a black (white)

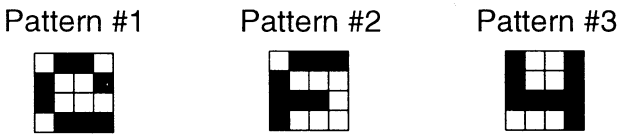


Figure 6: The memorized patterns.

box shows "+1" ("-1") of the pattern vector. A uniform external stimulus was applied to the network by the external bias voltage $\theta_i = 1.710$ V for all i. Figure 7 shows an example of the output pattern sequence, where the network shows non-periodic memory retrieval. A history of the retrieval is shown in Fig. 8. We observed the discretized quasi-energy function [24] defined by

$$QE^h(t_n) = -\frac{1}{2}\sum_{i\neq j}\sum_{j} W_{ij} h(x_i(t_n)) h(x_j(t_n)) - \sum_i \theta_i h(x_i(t_n)), \tag{9}$$

where $h(x) = 1$ $(x \geq 0)$ and $h(x) = -1$ $(x < 0)$. A time evolution of the quasi-energy function is shown in Fig. 9.

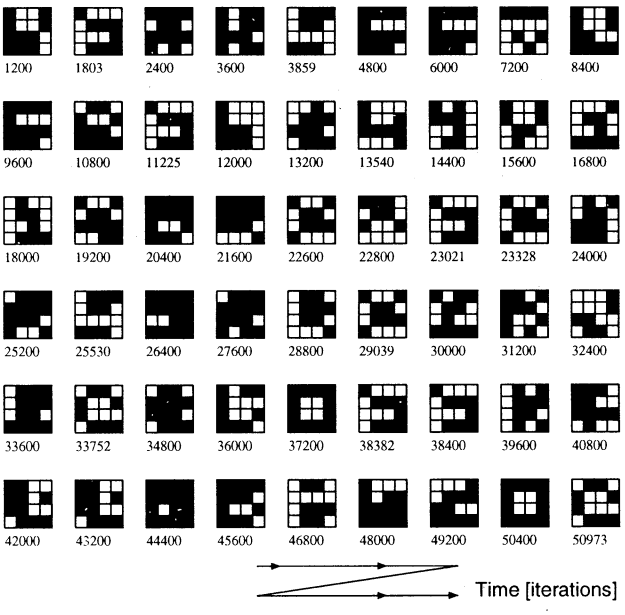


Figure 7: The output pattern sequence with uniform bias.

As a consequence, all of the three patterns and their complementary patterns are retrieved randomly,

but uniformly in frequency. Furthermore, the network does not stay in any equilibrium point. As shown in Fig. 9, the network spends most of the time in chaotic searching mode with this bias condition. This behavior may correspond to the neutral or resting state of neural networks [19, 16].

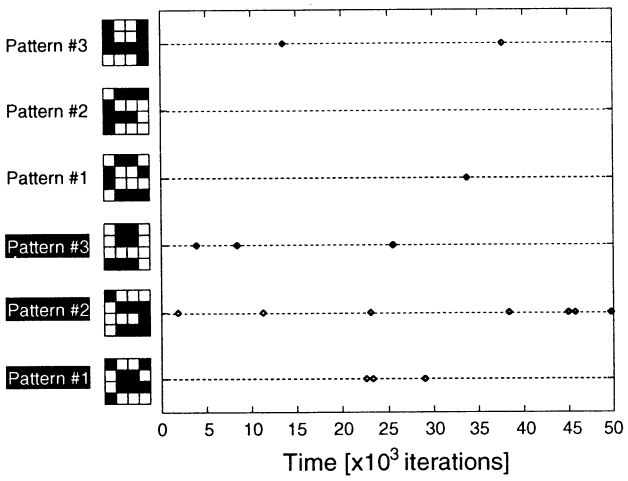


Figure 8: The history of retrieval with uniform bias. (Several points merged so that they look as one point because of low resolution of the printer.)

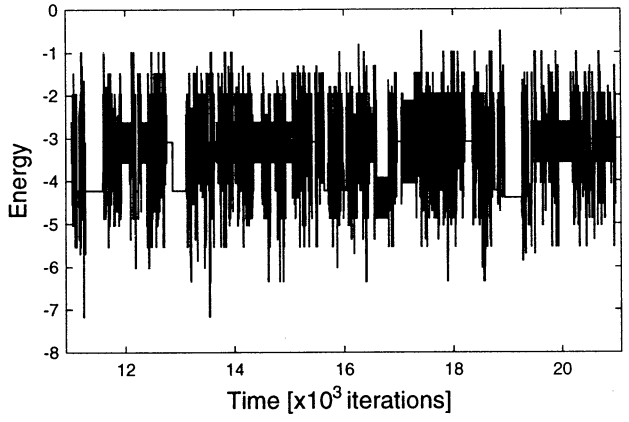


Figure 9: The time evolution of the quasi-energy function with uniform bias.

6 Conclusions

We have presented the design of integrated switched-capacitor chaotic neuron and its performances. The neuron was fabricated with a standard 2-μm, double-metal, double-poly, analog CMOS process. Various measurement results have confirmed the existence of rich nonlinear dynamics. A 4-city TSP and a dynamical associative memory have been used as examples for possible engineering applications. This work has successfully demonstrated a first step to apply analog integrated circuits to a chaotic neural network in which analog signal processing is essential. Currently, we are fabricating programmable integrated synapse IC's so that the resulting network can be used for investigating its dynamics in detail and for extending its ability to solve larger engineering problems.

Acknowledgment

The authors thank K. Aihara from Tokyo University and T. Ikeguchi from Science University of Tokyo for their valuable discussions. This work was supported in part by International Scientific Research Program under Project 08044171.

References

[1] K. Aihara, 'Chaotic neural networks,' in *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems*, H. Kawakami ed., pp. 143–161, World Scientific, Singapore, 1990.

[2] K. Aihara, T. Takabe, and M. Toyoda, 'Chaotic neural networks,' *Phys. Lett. A*, vol. 144, no. 6,7, pp. 333–340, 1990.

[3] J. J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities,' *Proc. National Academy of Sciences*, 79, pp. 2554–2558, 1982.

[4] J. J. Hopfield, 'Neurons with graded response have collective computational properties like those of two-state neurons,' *Proc. National Academy of Sciences*, 81, pp. 3088–3092, 1984.

[5] M. R. Guevara, L. Glass, and A. Shrier, 'Phase locking, period-doubling bifurcation, and irregular dynamics in periodically stimulated cardiac cells,' *Science*, 214, pp. 1350–1353, 1981.

[6] H. Hayashi, N. Nakao, and K. Hirakawa, 'Chaos in the self-sustained oscillation of an excitable biological membrane under sinusoidal stimulation,' *Phys. Lett. A*, vol. 88, pp. 265–266, 1982.

[7] K. Aihara and G. Matsumoto, 'Chaotic oscillations and bifurcations in squid giant axons,' in *Chaos*, A.V. Holden ed., Manchester Unvi. Press, 1986.

[8] H. Degn, A. V. Holden and L.F. Olsen eds., *Chaos in Biological Systems*, NATO ASI Series, Plenum Press, 1987.

[9] G. Matsumoto, K. Aihara, Y. Hanyu, N. Takahashi, S. Yoshizawa, and J. Nagumo, 'Chaos and phase locking in normal squid axons,' *Phys. Lett. A*, vol. 123, no. 4, pp. 162–166, 1987.

[10] G. J. Mpitosos, R. M. Burton Jr., H. C. Creech, and S. O. Soinila, 'Evidence for chaos in spike trains of neurons that generate rhythmic motor patterns,' *Brain Res. Bull.*, 21, pp. 529–538, 1988.

[11] H. Hayashi and S. Ishizuka, 'Chaotic nature of bursting discharges of Onchidium pacemaker neuron,' *J. Theor. Biol.*, 156, pp. 269–291, 1992.

[12] K. Aihara, 'Chaos in neural responses and dynamical neural network models: Toward a new generation of analog computing,' in *Towards the Harnessing of Chaos*, M. Yamaguti ed., pp. 83–98, Elsevier Science, 1994.

[13] W. J. Freeman, 'Strange attractors that govern mammalian brain dynamics shown by trajectories of electroencephalograpic(EEG) potential,' *IEEE Trans. on Circuits Syst.*, vol. 35, pp. 781–784, 1988.

[14] H. Hayashi and S. Ishizuka, 'Chaotic activity in hippocampus neural network and interacranical self-stimulation,' in *Proc. Int. Conf. Fuzzy Logic and Neural Networks*, 2, pp. 583–586, Iizuka, Japan, July 1990.

[15] Y. Yao and W. J. Freeman, 'Model of biological pattern recognition with spatially chaotic dynamics,' *INNS Neural Networks*, vol. 3, pp. 153–170, 1990.

[16] Y. Yao and W. Freeman, 'Model of biological pattern recognition with spatially chaotic dynamics,' *Neural Networks*, vol. 3, pp. 153-170, 1990.

[17] W. J. Freeman, 'The physiology of perception,' *Scientific American*, 264, pp. 78–85, 1991.

[18] K. Aihara, 'Spatio-temporal dynamics and possible computation in chaotic neural networks,' in *Proc. INNS World Congress on Neural Networks*, vol. 1, pp. 262–265, Washington, D.C., July, 1995.

[19] I. Tsuda, 'Dynamic link of memory – Chaotic memory map in neoequilibrium neural networks,' *INNS Neural Networks*, vol. 5, no. 2, pp. 313–326, 1992.

[20] K. Judd and K. Aihara, 'Neural networks and chaos,' in *Proc. IEEE Int. Symp. Circuits Syst.* pp. 2765–2768, San Diego, CA, May, 1992.

[21] S. Nara, P. Davis, and H. Totsuji, 'Memory search using complex dynamics in a recurrent neural network model,' *INNS Neural Networks*, vol. 6, no. 7, pp. 963–973, 1993.

[22] K. Kaneko, 'Clustering, coding, switching hierarchical ordering and control in network of chaotic elements,' *Physica D*, vol. 41, pp. 137–172, 1990.

[23] K. Hatamoto, T. Ikeguchi, T. Matozaki, and K. Aihara, 'An analysis of associative dynamics in chaotic neural networks,' in *Proc. IEEE Joint Conf. Neural Networks*, vol. 1, pp. 603–608, Peking, China, Dec. 1992.

[24] M. Adachi and K. Aihara, 'An analysis of associative transient dynamics in a chaotic neural network,' in *Towards the Harnessing of Chaos*, M. Yamaguti ed., pp. 335–338, Elsevier Science, 1994.

[25] T. Ikeguchi and K. Aihara, 'An analysis of associative dynamics of chaotic neural networks with external stimuli,' in *Proc. Int. Conf. on Dynamical Systems and Chaos*, vol. 1, pp. 327–330, World Scientific, 1994.

[26] M. Adachi, K. Aihara, and M. Kotani, 'An analysis of associative dynamics in a chaotic neural network with external stimulation,' in *Proc. IEEE Joint Conf. Neural Networks*, pp. 409–412, Nagoya, Japan, Oct. 1993.

[27] Y. Horio and K. Suyama, 'Dynamical associative memory using integrated switched-capacitor chaotic neurons,' in *Proc. IEEE Int. Symp. Circuits Syst.* pp. 429–432, Seattle, WA, May 1995.

[28] M. Adachi and K. Aihara, 'Associative Dynamics in a Chaotic Neural Network,' *INNS Neural Networks*, in printing.

[29] H. Nozawa, 'A neural network model as a globally coupled map and applications based on chaos,' *American Institute of Physics, Chaos*, vol. 2, no. 3, pp. 377–386, 1992.

[30] T. Yamada, K. Aihara, and M. Kotani, 'Chaotic neural networks and the traveling salesman problem,' in *Proc. IEEE Joint Conf. Neural Networks*, pp. 1549–1552, Nagoya, Japan, Oct. 1993.

[31] Y. Horio, K. Suyama, A. Dec, and K. Aihara, 'Switched-capacitor chaotic neural networks for traveling salesman problem,' in *Proc. INNS World Congress on Neural Networks*, vol. 4, pp. 690–696, San Diego, CA, June 1994.

[32] Y. Horio, N. Kanou, and K. Suyama, 'Switched-capacitor and switched-current chaotic neural networks for combinatorial optimization problems,' *Proc. of International Symposium on Nonlinear Theory and Its Applications(NOLTA)*, Vol. 1, pp. 1–6, Las Vegas, Dec. 10–14, 1995.

[33] Y. Horio and K. Suyama, 'Switched-capacitor chaotic neuron for chaotic neural networks,' in *Proc. IEEE Int. Symp. Circuits Syst.* pp. 1018–1021, Chicago, IL, Aug. 1993.

[34] Y. Horio and K. Suyama, 'IC implementation of switched-capacitor chaotic neuron for chaotic neural networks,' in *Proc. IEEE Int. Symp. Circuits Syst.* vol. 6, pp. 97–100, London,May 1994.

[35] Foresight Fabrication Service, Orbit Semiconductor, Inc.

[36] T. Ikeguchi and K. Aihara, private communications, 1993.

[37] Y. Horio, and K. Suyama, 'Experimental observations of 2- and 3-neuron chaotic neural networks using switched-capacitor chaotic neuron IC chip,' *IEICE Trans. Fundamentals*, vol. E78-A, no. 4, pp. 529–535, 1995.

[38] Y. Tanaka, K. Horikawa, N. Imamura, and Y. Horio, 'An estimation algorithm of the Lyapunov exponents from noisy and quantized data,' *Proc. of International Symposium on Nonlinear Theory and Its Applications(NOLTA)*, Vol. 1, pp. 305–308, Katsura-Hama, Japan, Oct. 7–9, 1996.

[39] Y. Horio and K. Suyama, 'Experimental verification of signal transmission using synchronized SC chaotic neural networks,' *IEEE Trans. Circuits Syst. I*, vol. 42, no. 7, pp. 393–395, 1995.

[40] G. V. Wilson and G. S. Pawley, 'On the stability of the traveling salesman problem algorithm of Hopfield and Tank,' *Biol. Cybern.*, vol. 58, pp. 63–70, 1988.

A Dynamical Systems Approach to Build Cognitive Robots

Jun Tani
Sony Computer Science Laboratory Inc.
Takanawa Muse Building, 3-14-13 Higashi-gotanda, Shinagawa-ku,Tokyo, 141 JAPAN
Tel 3-5448-4380, Fax 3-5448-4273, Email tani@csl.sony.co.jp

Abstract

This paper [1] discusses advantages of dynamical systems approach for the robots to attain intrinsic descriptions in terms of their behaviors. We show qualitative differences between taking views of "external observer" and "internal observer" dealing with the descriptions for the robots. The conventional symbolic approach takes the view of the external observer, in which there exists an observer who looks over the descriptions and try to manipulate them. On the other hand, the dynamical systems approach provides that of the internal observer, in which the descriptions and their manipulations become an unseparable entity. We explain how robots built with the view of the internal observer can be descriptive without having explicit descriptions, and how their mental processes of "manipulating descriptions" can be naturally situated in the behavioral context. We describe the problems with reviewing our prior experiments of navigation learning (Tani 1996) which was conducted with using a real mobile robot.

Introduction

We speculate that the problems of cognitions commence when the robots attempt to acquire descriptions of the world in certain forms so that they can mentally simulate or plan their own behavior sequences. By this means, we may not consider that purely reactive-type robots in terms of simple sensori-motor reflex involve the problems of cognition. In talking about the descriptions, it is important to consider how the descriptions can be grounded to the physical environments and how the mental processes manipulating them can be situated in the behavioral contexts. This question addresses one of the observation problems in cognition which asks us where the observer, dealing with the descriptions, is positioned. We study these problems with focusing on tasks of the robot navigation learning.

In the traditional approach of the robot navigation problems, the robots are forced to acquire exact maps of the environment measured in the global coordinate systems. Such robots apparently use the external views to describe their environments, since the descriptions are made by assuming the global observation from the outside.

On the other hand, the recent approach based on the landmark-based navigation (Kuipers 1987; Mataric 1992) does not assume any global observations of the environments. In this approach, the observer sits inside the robot and it looks the outside through the sensory device and focusing on the coming events or landmarks. The observer collects the sequences of the landmark-types and try to build the chain representation of them in the form of finite state machines (FSM) as the topological map of the environment. Although it is true that this approach provides us much more successful results of the navigation comparing to the global map strategies, the symbolic representation of the FSM can cause the symbol grounding problems. The symbol grounding problem is a general problem, as discussed by Harnad (Harnad 1990), that discrepancies which occur between the objects in the physical environment and their symbolic representations in the system cannot be resolved autonomously through the system's own operations. Let us consider the situation where the robot navigates in the pre-learned environment by identifying the current position from trying to match the state transitions in the FSM. A problem can arise when the robot fails to recognize an oncoming landmark because of some noise. The robot will be lost because it has received an erroneous sensory input which is different from the one expected using the FSM. The FSM will simply halt upon receiving this illegal input. Although some may argue that this problem can be resolved by further development of the categorization schemes for landmark recognition, we consider that this approach leaves the underlying problem unsolved. We believe that the underlying problem exists in the position of the observers who look over the symbolic representations and try to manipulate them. The observer here is external to the descriptions. As long as such external observers are allowed for the robots, the robots would face the symbol grounding

[1] The contents of this paper was originally presented in AAAI Fall Symposium 96, Embodied Cognition and Action and a part was modified.

problems.

We have investigated this problem from the dynamical systems perspectives (Beer 1995; Jordan & Rumelhart 1992). We speculate that real number systems best represent the mental activities of robots. We expect that the chaotic dynamics may serve as a basis for the mental activities of robots, as the theories of symbolic dynamics (Crutchfield 1989; Pollack 1991; Tani & Fukumura 1995) have shown that chaos exhibits a certain linguistic complexity. When the internal dynamics, which describe the mental processes of the robot, and the environment dynamics are coupled together through the sensory-motor loop, those two dynamics would share the same metric space. We consider that the mental processes of the robots can be naturally situated to the environments as the coherence is achieved between those two dynamics interacting each other in the same phase space. An important objective here is to unify the two separate entities for "descriptions" and their "manipulations" in the systems into one entity within the framework of the time-development of dynamical systems. We speculate that the internal observer finally appears in the cognitive processes of robots if this objective is accomplished. The next section reviews our embodied work of the mobile robot learning of cognitive maps based on the dynamical systems approach.

Formulation

Firstly we introduce our navigation scheme which is applied to the *YAMABICO* mobile robot. *YAMABICO* can obtain the range image by a laser range finder in real-time. In our formulation, maneuvering commands are generated as the output of a composite system consisting of two levels. The control level generates a collision-free, smooth trajectory using a variant of the potential field method i.e. the robot simply proceeds towards a particular potential hill in the range profile (direction toward an open space). The navigation level focuses on the topological changes in the range profile as the robot moves. As the robot moves through a given workspace, the profile gradually changes until another local peak appears when the robot reaches a branching point. At this moment of branching the navigation level decides whether to transfer the focus to the new local peak or to remain with the current one. The navigation level functions only at branching points which appear in unconstructed environment. The importance here is that the navigation of the robot is made on the topological trajectories which are determined by the branching sequences. Hereafter, our discussions focus on how to learn and determine the branching sequences using neural learning schemes.

In the learning phase, the robot explores a given obstacle environment by randomly determining branching. Suppose that the robot comes to the n*th* branch point with receiving the sensory input (range image

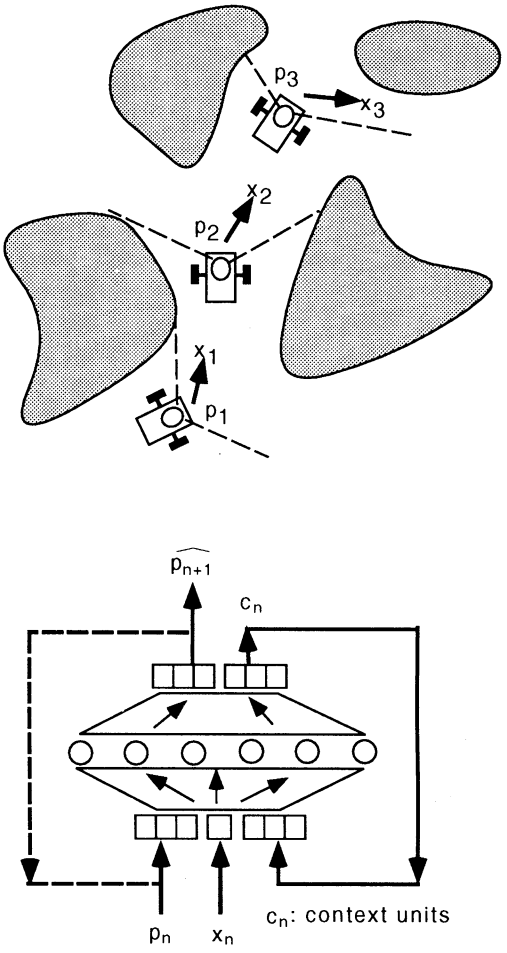


Figure 1: Sensory-motor sequence in branching and RNN architecture.

vector) p_n and randomly determine branching (0 or 1) as x_n, then it moves to the next branch point n+1*th* (see the left-hand side of Fig 1.) Through the entire exploratory travel, the robot acquire the sensory-motor sequence of (p_i, x_i). Using this sample of the sensory-motor sequence, a recurrent neural net (RNN) is trained so that it can predict the next sensory input p_{n+1} in terms of the current sensory input p_n and the branching motor command x_n (see the right-hand side of Fig 1). We employ the idea of the context reentry by Jordan (Jordan & Rumelhart 1992) which enables the network to represent the internal memory. The current context input c_n (a vector) is a copy of the context output in the previous time: by this means the context units remember the previous internal state. The navigation problem is an example of a so-called "hidden state problem" a given sensory input does not always represent a unique situation/position of the robot. Therefore, the current situation/position

is identifiable, not by the current sensory input, but by the memory of the sensory-motor sequence stored during travel. Such memory structure is self-organized through the learning process. We expect that the RNN can learn certain "grammatical" structure hidden in the obstacle environment as embedded in its intrinsic dynamical structure by utilizing the context re-entry. (As many have shown the capability of RNNs for grammar learning.) We employ back-propagation through time algorithm (Rumelhart, Hinton, & Williams 1986) for the RNN learning.

Experiment

Here, we review a part of our experiments of lookahead prediction. The robot explored a given workspace and the RNN were trained with 193 samples of the sensory-motor sequence. After this learning, the robot is started to travel from arbitrary positions. The robot maneuvers following an arbitrary motor program (branching sequence) and it tries to predict the coming sensory input of the next branch using the sensory input and a given motor command at each current branch. (This is one-step lookahead prediction.) Fig 2 shows an example of the result. The left-hand side shows the measured trajectory of the robot. The right-hand side shows the comparison between the actual sensory sequence and the predicted one. The figure shows the nine steps of the branching sequence, where five units in the most left are the sensory input, the next five units are its prediction, the next one unit is the motor command (0 or 1 of branching), and the most right four units are the context units. Initially the robot cannot predict correctly. It, however, becomes able to predict correctly after the 4th step. Since the context units are randomly set initially, the prediction fails at the very beginning. However as the robot continues to travel, the sequence of the sensory input "entrain" the context activations into the normal state transition sequence, thereafter the RNN becomes able to predict correctly. We repeated this experiment with various initial settings (positions and motor programs), which showed that robot always starts to predict correctly within 10 steps. Furthermore we found that although the context is easily lost when perturbed by large sensory noise (e.x., when the robot fails to detect a branch), the prediction can be always recovered as long as the robot continues to travel. This auto-recovery of the cognitive process is made in consequence that a sort of coherence is organized between the internal and the environmental dynamics in their interactions.

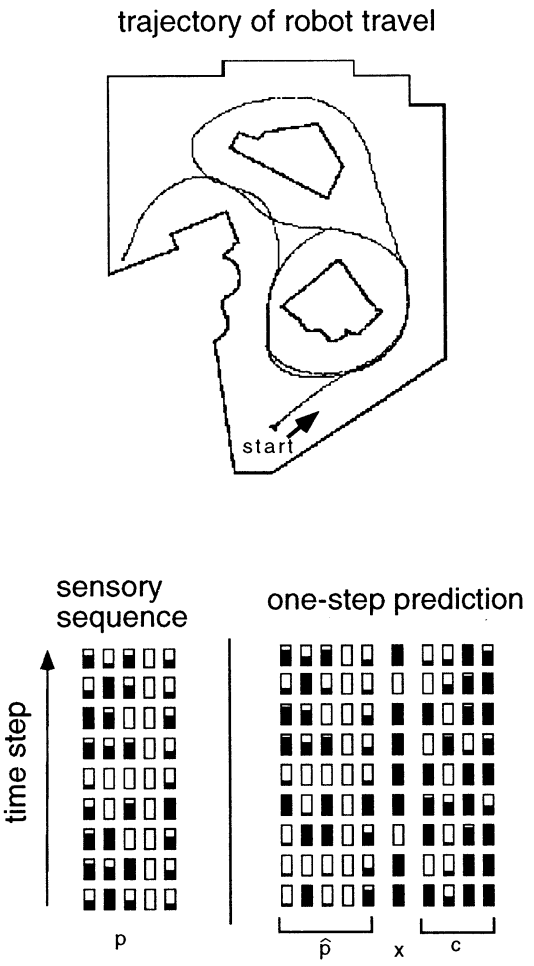


Figure 2: One-step prediction.

Analysis and Discussion

It is assumed that there exists an essential dynamical structure which can generate the coherence, as we have discussed. We conducted the state space analysis of the obtained RNN in order to see such structure. The re-entry loop is connected from the sensory outputs to

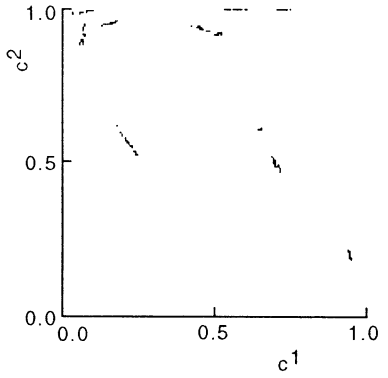


Figure 3: The global attractor appeared in the RNN dynamics.

the sensory inputs nodes so that the RNN can conduct lookahead predictions for arbitrary length of motor command sequences. Then the RNN was activated for 2000 steps with feeding randomly generated branching sequence of $x*$. (Here, the RNN conducts mental simulations for the random branching sequence.) The state space trajectory of the context units was plotted using the activation sequences of two context units (we took a 2-D projection of the entire state space). As the result we obtained a one-dimensional like invariant set as shown in Fig. 3. Our mathematical analysis shows that this invariant set is dense and topologically transitive. We also found that the invariant set is a global attractor since the trajectory always converges into the same one independent of the initial setting of the context values. In this plot, the invariant set shows the boundary of rationality/cognition for the mental processes of the robot. When the RNN is perturbed by receiving noisy inputs, the context values go out of the invariant set where the rationality in terms of predictability is lost. However, as the RNN continues its dynamical iterations, the context values always come back to the rational region i.e.–the invariant set, then the RNN becomes able to predict correctly again. This cognition boundary is determined solely from the system's own dynamical iterations, as stated by Maturana and Varela (Maturana & Varela 1980). Here, the dynamical systems approach shows that this inherent mechanism of the autonomy is indispensable to achieve the cognition of the robot.

Our further analysis of this invariant set revealed the fact that each line segment corresponds to each identical branching point. Each segment of these has two ways of transitions depending on the binary branching. And each segment is accessible from all other segments within finite steps of state transitions. (Remember that the invariant set is topologically transitive). What we see in the state space is a dynamical closure which some might interpret as equivalent to an FSM representation. However, the important remark is that the internal system itself neither sees the descriptions of FSM nor involves their manipulations. The internal system merely repeats its dynamical iterations as a mental process, something the outside observer may perceive as being equivalent to if symbols actually existed and were manipulated internally. In fact, all that exists is the dynamical structure and the resultant time-development of the system. Here, the descriptions and manipulations appear to be an unseparable entity in the dynamical system. Since there are no observers dealing with the descriptions, we finally find the internal observer in our robot. Consequently, there are no descriptions or symbols to cause the symbol grounding problem from the view of this internal observer.

References

Beer, R. 1995. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence* 72(1):173–215.

Crutchfield, J. 1989. Inferring statistical complexity. *Phys Rev Lett* 63:105–108.

Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335–346.

Jordan, M., and Rumelhart, D. 1992. Forward models: supervised learning with a distal teacher. *Cognitive Science* 16:307–354.

Kuipers, B. 1987. A qualitative approach to robot exploration and map learning. In *AAAI Workshop Spatial Reasoning and Multi-Sensor Fusion (Chicago)*, 774–779.

Mataric, M. 1992. Integration of representation into goal-driven behavior-based robot. *IEEE Trans. Robotics and Automation* 8(3):304–312.

Maturana, H., and Varela, F. 1980. *Autopoiesis and cognition: the realization of the living.* Boston: D. Riedel Publishing.

Pollack, J. 1991. The induction of dynamical recognizers. *Machine Learning* 7:227–252.

Rumelhart, D.; Hinton, G.; and Williams, R. 1986. Learning internal representations by error propagation. In Rumelhart, D., and Mclelland, J., eds., *Parallel Distributed Processing.* Cambridge, MA: MIT Press.

Tani, J., and Fukumura, N. 1995. Embedding a Grammatical Description in Deterministic Chaos: an Experiment in Recurrent Neural Learning. *Biological Cybernetics* 72:365–370.

Tani, J. 1996. Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective. *IEEE Trans. System, Man and Cybernetics Part B, Special issue on learning autonomous robots* 26(3):421–436.

Comparison of Local Reconstruction Methods

KOYAMA Masaya, TANIGUCHI Minako, IOKIBE Tadashi

System Development Section
System Technology Division
Meidensha Corporation
36-2, Nihonbashi Hakozakicho,
Chuo-ku, Tokyo 103 Japan
TEL: 03-5641-7507 FAX: 03-5641-9303

Keywords: Chaos, Time series, Prediction, The Gram-Schmidt orthogonal system method, The tesselation method, The local fuzzy reconstruction method

Abstract

In the field of nonlinear prediction of the chaotic time series, several local reconstruction methods are proposed. The Gram-Schmidt orthogonal system method and the tesselation method are proposed as a local linear reconstruction method, the local fuzzy reconstruction method is proposed as a local nonlinear reconstruction method. The authors applied these methods to the prediction of chaotic time series generated by the logistic equation and the Lorenz system, and examined the prediction performance and the computation time.

1 Introduction

In recent years, studies on the chaos have been drawing attention. And we can know that chaos has a potentiality of application in extensive new fields such as deterministic nonlinear prediction, identification and modeling of nonlinear system, memory generation and retrieval, compression and coding of information, pattern recognition, application to arts and generation of fluctuations.

By the way, time series prediction is an important subject in diverse fields including tap water demand, electric power demand, local weather forecast, stock price, exchange rate and merchandise sales and so on. For enhancing the accuracy of this prediction, a variety of methods have been developed.

As an attempt to formulate prediction models, a number of proposals have been made based on a linear stochastic process. They are represented by the ARIMA model which is applicable to the time series data having a seasonal change. It is an approach to a preliminary identification of model, estimation of model parameters and reasonableness diagnosis of model and prediction of future state. The conventional linear predictions represented by this method often involve difficulty in parameter identification due to modeling.

On the other hand, there is a different approach as an application of chaos to engineering, i.e., short-term prediction on the time series data having a chaotic behavior. This is based on the following standpoint. When the behavior of the observed time series data is chaotic and its dynamics can be estimated, the behavior of the data is predictable within the time period before the dynamics is lost. Concretely, the observed time series data is reconstructed in a multi-dimensional state space depending on the "Takens' theorem on embedding [1]" and a local reconstruction is carried out using the data vector including the latest data observed and the neighboring data vector, whereby the data in near future is predicted. There are a variety of local reconstruction methods such as "the Gram-Schmidt orthogonal system method" [2], "the tesselation method" [3], "the local fuzzy reconstruction method" [4] and others. This paper examines the differences in prediction performance and necessary calculation time among the above methods according to the result of applying each method to the prediction on typical chaotic time series.

2 Principle of deterministic prediction

When the behavior of the observed time series data is chaotic, it can be assumed that the behavior follows a certain deterministic regularity. Then, if the deterministic regularity can be estimated, the data in the near future till the deterministic causality is lost can be predicted from the observed data because chaos has a "sensitivity on initial condition."

In this prediction, estimation of the dynamics in a dynamical system is important, which is conducted in the following way. First, the observed time series data is reconstructed in an n-dimensional state space to form a trajectory by the Takens' theorem of embedding [1] while retaining the phase structure of that data. Next, the data vector including the latest observed time series data is plotted in the n-dimensional reconstructed state space. Then, the data vector which should be predicted is estimated by the local reconstruction method, using the neighboring data vectors of the data vector including the latest data observed and the data vectors at s steps ahead of each neighboring data vector. Finally, the

data vector estimated by the local reconstruction method is converted into the original time series, thereby obtaining the predicted value of the target time series data. This method of prediction is termed as a deterministic nonlinear prediction.

The advantage of this method lies in that extrapolation in the prediction of time series data is replaced with interpolation in the n-dimensional reconstructed state space by embedding the original time series data into the n-dimensional reconstructed state space. Problems on interpolation can be solved far easier than those on extrapolation.

Explained below are the method of embedding the time series data into the n-dimensional state space according to the Takens' theorem of embedding and the 3 local reconstruction methods.

2.1 Takens' theorem of embedding

The Takens' theorem of embedding can be summarized as follows. Generally, the state of a system is described by "m" state variables ("m" stands for the number of variables which affect the observed time series data) and can be represented as a trajectory in the m-dimensional space. However, the observable time series data is only a part usually. In this case, from the time series data of a single observed variable, its trajectory can be reconstructed in an n-dimensional space using delay time. Concretely, from the observed time series data y(t), the vector $\mathbf{x}(t)\equiv(y(t),y(t-t),\ldots,y(t-(n-1)t))$ is generated where n and τ represent embedding dimension and delay time. This vector indicates one point of an n-dimensional reconstructed state space. Therefore, a trajectory can be drawn in the n-dimensional reconstructed state space by changing t with τ fixed. The reconstructed trajectory can be said to be embedded in the reconstructed state space when embedding dimension n is sufficiently large. To be concrete, it has been proven by Takens[1] that the sufficient condition for the reconstructed trajectory to retain the phase structure in the original m-dimensional state space, i.e., the reconstructed trajectory to be embedded is as follows.

$$n \geq 2m + 1 \quad \ldots\ldots (1)$$

However, embedding can be established even when n is less than 2m + 1, depending on data.

2.2 Local reconstruction method

On the reconstructed manifold, data vector $\mathbf{z}(T)$ including the latest observed data is plotted. A sphere having radius ε with its center at $\mathbf{z}(T)$ is called "neighboring space" and the data vectors present in the neighboring space is called "neighboring data vector." These data vectors are represented by $\mathbf{x}(t_i)$ where i equals 0, 1, ...,k. Since data vectors $\mathbf{x}(t_i)$ is the past data, the state $\mathbf{x}(t_i + s)$ at s steps ahead of $\mathbf{x}(t_i)$is already known. Utilizing this, the trajectory of the near future of vector $\mathbf{z}(T)$ is assumed to determine the data vector $\mathbf{z}(T + s)$ at s steps ahead of $\mathbf{z}(T)$. The method for assuming the trajectory in the near future is called "local reconstruction method."

2.2.1 The Gram-Schmidt orthogonal system method

The neighboring data vectors are replaced here with $\mathbf{x}(t_i)$ (i = 0, 1, ... k) starting from the nearest data vector. Let us denote by ξ_i the deviations between centroid $\mathbf{R}$ of $\mathbf{x}(t_i)$. That is $\xi_i = \mathbf{x}(t_i) - \mathbf{R}$. and let also ξ_z be the deviations between $\mathbf{R}$ and $\mathbf{z}(T)$. That is $\xi_z = \mathbf{z}(T) - \mathbf{R}$. The coefficient C_i, which defines the optimal linear association of ξ_z in terms of ξ_i, is obtainable by projecting ξ_z on the The Gram-Schmidt orthogonal base $\mathbf{V}_i$ constructed out of ξ_i :

$$\mathbf{V}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}\ ,\ \mathbf{w}_i = \xi_i - \sum_{j=1}^{i-1}\left\{\langle \xi_j, \mathbf{V}_j\rangle \mathbf{V}_j\right\} \quad \ldots\ldots (2)$$
$$C_i = \langle \xi_z, \mathbf{V}_i\rangle$$

Let us predict the vector of s steps ahead of $\mathbf{z}(T)$, using a coefficient set $\{C_i\}$ obtainable for each $\mathbf{x}(t_i)$. Replacing the center of gravity for the data vector $\mathbf{x}(t_i + s)$ at s steps ahead of $\mathbf{x}(t_i)$ with $\widetilde{\mathbf{R}}$ and the deviation vector of $\mathbf{x}(t_i + s)$ and $\widetilde{\mathbf{R}}$ with $\widetilde{\xi}_i$, the Gram-Schmidt orthogonal base $\widetilde{\mathbf{V}}_i$ can be constructed in the same way as above.

$$\widetilde{\mathbf{V}}_i = \frac{\widetilde{\mathbf{w}}_i}{\|\widetilde{\mathbf{w}}_i\|}\ ,\ \widetilde{\mathbf{w}}_i = \widetilde{\xi}_i - \sum_{j=1}^{i-1}\left\{\langle \widetilde{\xi}_j, \widetilde{\mathbf{V}}_j\rangle \widetilde{\mathbf{V}}_j\right\} \quad \ldots\ldots (3)$$

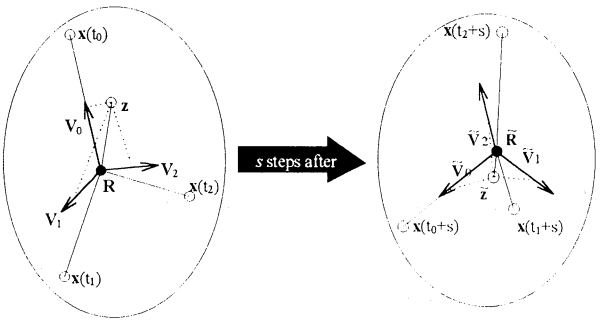


Fig. 1 Concept of Gram-Schmidt orthogonal system method

So far as the number of step s is rather small, the set of the coefficients $\{\widetilde{C}_i\}$, which defines the optimal linear association of $\widetilde{\xi}_z = \mathbf{z}(T+s) - \widetilde{\mathbf{R}}$ in terms of $\widetilde{\xi}_i$, can be thought to be approximate to $\{C_i\}$. Therefore, the predictive value $\widetilde{\mathbf{z}}(T+s)$ can be expressed as follows.

$$\widetilde{\xi}_z = \sum_{i=1}^{k} C_i \widetilde{\mathbf{V}}_i \quad \ldots\ldots (4)$$
$$\widetilde{\mathbf{z}}(T+s) = \widetilde{\xi}_z + \widetilde{\mathbf{R}}$$

2.2.2 The tesselation method

First, Voronoi Tesselation is constructed using the neighboring data vectors $\mathbf{x}(t_i)$ (i = 0, 1, ... k) of $\mathbf{z}(T)$. Each tile after Tesselation is replaced with $Vx(t_i)$(see Fig. 2(a)). Next, Voronoi Tesselation is reconstructed using $\mathbf{x}(t_i)$ and $\mathbf{z}(T)$ (See Fig. 2(b)). The resulting tile is replaced with Vz(T). Also, the Vz(T) sections divided by $Vx(t_i)$ are replaced with Vzi(T). Then, the predictive value $\tilde{\mathbf{z}}(T+s)$ at s steps ahead of $\mathbf{z}(T)$ is given by formula (5).

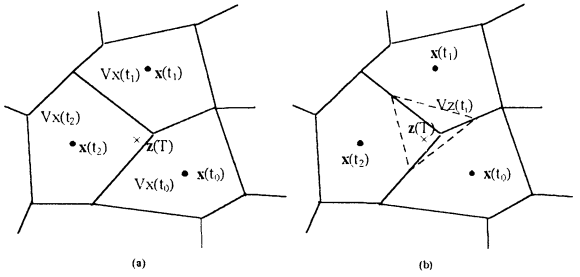


Fig. 2 Voronoi tesselation

$$\tilde{\mathbf{z}}(T+s)=\sum_{i=0}^{k}\lambda i\{\mathbf{z}(T)\}\mathbf{x}(i+s)$$

$$\lambda i\{\mathbf{z}(T)\}=\frac{\mu_n\{Vzi(T)\}}{\sum_{i=0}^{k}\mu_n\{Vzi(T)\}} \quad \ldots\ldots (5)$$

Where, μ_n stands for Lebesgue measure in R^n.

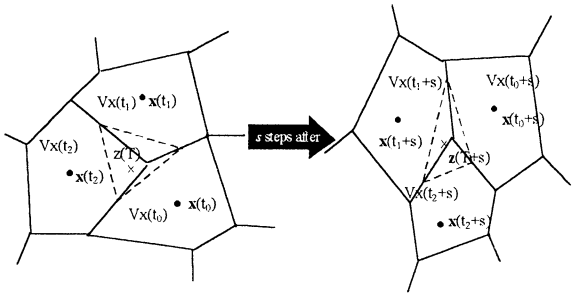


Fig. 3 Concept of the tesselation method

2.2.3 The local fuzzy reconstruction method

If the behavior of the observed time series data corresponds to deterministic chaos, the transition from state $\mathbf{x}(t_i)$ to state $\mathbf{x}(t_i + s)$ can be assumed to be dependent on the dynamics subjected to determinism. This dynamics can be linguistically expressed as follows, using $\mathbf{x}(t_i)$ and $\mathbf{x}(t_i + s)$.

IF $\mathbf{x}(T)$ is $\mathbf{x}(t_i)$ THEN $\mathbf{x}(T + s)$ is $\mathbf{x}(t_i + s)$
Where, $\mathbf{x}(T)$: Set of data vector neighboring to $\mathbf{z}(t)$
$\mathbf{x}(T + s)$: Set of data vector at s steps ahead of $\mathbf{x}(T)$

Because $\mathbf{x}(t_i)$ is the data vector neighboring to $\mathbf{z}(T)$, the dynamics from state $\mathbf{z}(T)$ to $\mathbf{z}(T + s)$ can be considered approximate to the dynamics from state $\mathbf{x}(t_i)$ to state $\mathbf{x}(t_i + s)$, and the trajectory from $\mathbf{z}(T)$ to $\mathbf{z}(T + s)$ is influenced by the distance from $\mathbf{z}(T)$ to $\mathbf{x}(t_i)$. That is, it can be thought that the trajectory from $\mathbf{x}(t_i)$ to $\mathbf{x}(t_i+s)$ nearer to $\mathbf{z}(T)$ exerts larger influence on the trajectory from $\mathbf{z}(T)$ to $\mathbf{z}(T + s)$ and that the trajectory farther to $\mathbf{z}(T)$ exerts smaller influence. Also, because the attractor embedded in the n-dimensional reconstructed state space is a smooth manifold, it receives nonlinear influence. The relevant relationship can be expressed in fuzzy function as follows.

IF $\mathbf{z}(T)$ is $\mathbf{x}(t_i)$ THEN $\mathbf{z}(T + s)$ is $\mathbf{x}(t_i + s)$

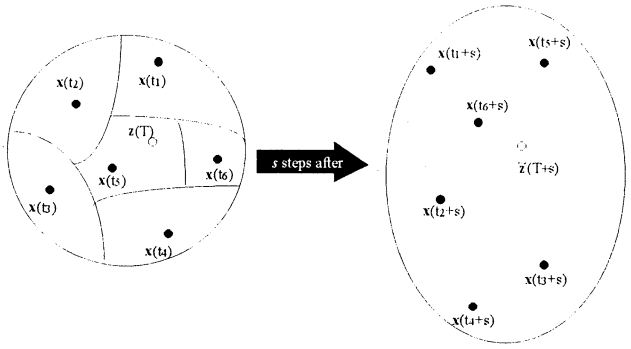


Fig. 4 Concept of the local fuzzy reconstruction method

3 Application to Chaotic Time series Prediction

In this section, we predict the time series generated by the typical chaos, i.e., logistic map and Lorenz system using the aforementioned local linear reconstruction methods "the Gram-Schmidt orthogonal system method" and "the tesselation method" and the local nonlinear reconstruction method "the local fuzzy reconstruction method." And performance is compared among these local reconstruction methods. The computer used was SUN SPARK 10/41.

3.1 Comparison of prediction performance

For comparison of prediction performance, the initial values of coefficient and variable were given as a = 4.0 and x_0 = 0.35 in formula (6) for the logistic map, and they were given as A = 10, B = 28, C = 8/3, X_0 = 0.05, Y_0 = 0 and Z_0 = 0 in formula (7) for the Lorenz system, and 6,000 data was generated in a pitch width dt of 0.1. In the Lorenz system, only X component was used. In all predicting conditions, the first half $\{y\}_{t=0}^{t=(n-1)/2}$ of the given time series data $\{y\}_{t=0}^{t=n-1}$ was used as the initial value for embedding and the data at s steps ahead was predicted. Then, $\{y\}_{t=((n-1)/2)+1}$ was added to embedding and the data at s steps ahead was predicted again. This procedure was repeated until the given data was all processed. The coefficient of correlation between the calculated time series data value and the predicted value was taken as an index of prediction performance. Note that the number of steps "s" was selected within a range where the coefficient of correlation became 0.5 or less in at least one prediction method.

3.1.1 Comparison for logistic map

$$x_{n+1} = f(x_n) = ax_n(1 - x_n) \quad \text{..... (6)}$$

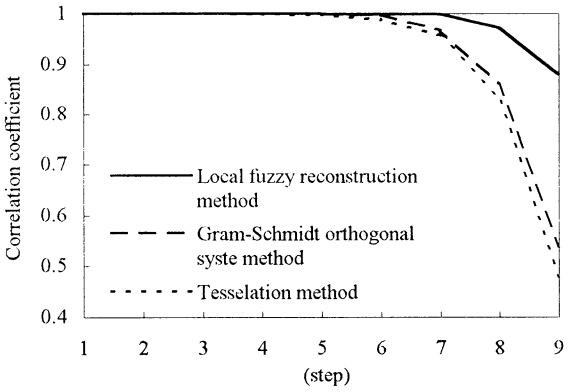


Fig. 5 Predictability in logistic map

3.1.2 Comparison for Lorenz system

$$dX/dt = -AX + AY$$
$$dY/dt = -XZ + BX - Y \quad \text{..... (7)}$$
$$dZ/dt = XY - CZ$$

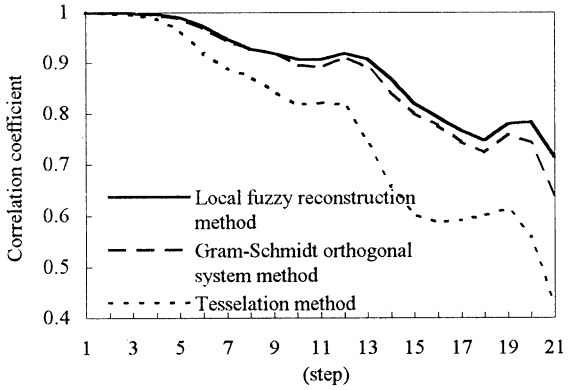


Fig. 6 Predictability in Lorenz system

3.2 Comparison of calculation time

With the logistic map, 1,000 data was generated under the same conditions as in 3.1. Using the data, the calculation time by each method was measured.

The processing of embedding time series data in multi-dimensional reconstruction state space and selecting the neighboring data vector used for prediction is common among all local reconstruction methods. Therefore, the difference in calculation time shown in Fig. 7 is attributable to the difference in production algorithm.

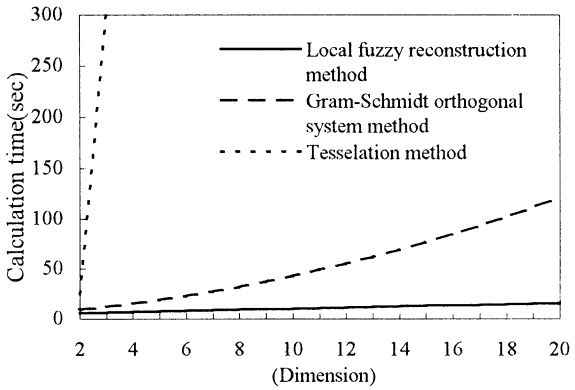


Fig. 7 Calculation time by local reconstruction method

4. CONCLUSION

Prediction performance is not so different in case prediction step is small or non-linearity of the target time series is inconspicuous. However, the local non-linear reconstruction method has a higher prediction performance when prediction step is large or non-linearity is conspicuous.

Calculation time is hardly affected by an increase in the number of embedding dimensions in case of the local fuzzy reconstruction method, and it is small in terms of absolute value. By contrast, the calculation time by the Gram-Schmidt orthogonal system method and the tesselation method are extended in response to an increase in the number of embedding dimension.

From the above results, the local fuzzy reconstruction method can be said to be optimum for prediction on chaotic time series.

References

[1] F. Takens: "Detecting Strange Attractors in Turbulance", in Dynamical Systems and Turbulence, (eds. D. A. Rand and L. S. Young), Springer-Verlag, Berlin, pp. 366-381 (1981)

[2] A.I. Mees: "Dynamical Systems and Tesselations: Detecting Determinism in Data", International Journal of Bifurcation and Chaos, Vol.1, No.4, pp.777-794 (1991)

[3] J. Jiménez, J.A. Moreno and G.J. Ruggeri: "Forecasting on Chaotic Time series : A Local Optimal Linear-reconstruction Method", Physical Review A, Vol.45, No.6, pp.3553-3558 (1992)

[4] T. Iokibe, M. Kanke, Y. Fujimoto and S. Suzuki: "The local fuzzy reconstruction method for short-term prediction on chaotic time series," Journal of Japan Society for Fuzzy Theory and Systems, Vol. 7, No. 1, pp. 186 - 194 (1995)

Forecasting Complex Time Series by Bell-Shaped Regularization Networks

Takaya Miyano
Device Technology Section
Advanced Technology Research Labs.
Sumitomo Metal Industries, Ltd.
1-8 Fusocho, Amagasaki
Hyogo 660
Japan

Kazuyuki Aihara
Department of Mathematical Engineering
Faculty of Engineering
University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo 113
Japan

Abstract

We examine the performance of generalized regularization networks consisting of $(1 + cosh\ x)^{-1}$ as the basis function through time series prediction about real-world data: thermal sequences observed in a blast furnace, global temperature variations observed from 1856 to 1991, and acoustic signals of Japanese vowel /a/. Stochastic gradient descent is exploited as learning rule for optimizing the networks. Nonlinearity in each data is estimated in terms of the difference in the prediction error generated by iterative forecasting with the regularization network as a nonlinear predictive model and with a linear autoregressive predictor as a linear predictive model.

1 Introduction

Generalized regularization networks are becoming a popular approach to approximation in diverse technological fields such as object recognition and time series prediction [1 − 6]. A standard component of the networks is Gaussian radial basis function whose shape is associated with the statistical properties of given examples. This nature of Gaussian function often facilitates searching the optimal structure of a network in that appropriate shape of the basis functions can be inferred at the initial stage of learning according to the statistical distribution of library examples. The use of bell-shaped regularization networks has been also encouraged by recent findings of the selective response of a population of neurons in the visual cortex that suggest the biological plausibility of the networks architecture [7]. However, Gaussian function is not a tractable component when one might attempt to hardware-implement regularization networks, say, on a silicon chip [8]. As for the feasibility of hardware implementation, we recently found that $(1 + cosh\ x)^{-1}$ as the basis function of the same class as Gaussian function is a tractable component of regularization networks in that its analytical expression can be achieved on a silicon chip using a simple analog circuit consisting of a differential amplifier followed by an analog multiplier with bipolar transistors [9].

In this paper we examine the performance of generalized regularization networks consisting of $(1 + cosh\ x)^{-1}$ as the basis function through application to complex time series prediction, demonstrating that the networks really work as well as Gaussian regularization networks. Three examples of real-world time series are taken: (1) temperature sequences observed in a blast furnace for iron-making [6], (2) a meteorological time series, and (3) speech signals of Japanese vowel /a/ [10, 11]. The meteorological data were kindly offered by Newell and his research group at the Massachusetts Institute of Technology [12]. An interesting issue on these data is whether or not deterministic chaos exists in the relevant dynamical systems [5, 6, 10, 11, 13 − 17], which is a second subject investigated in this paper. There has been a long controversy as to the existence of climatic attractors [13 − 16]. The possibility of vocal attractors in Japanese vowel /a/ was first suggested in [17]. In the present work we consider this problem in terms of degrees of nonlinearity in each data. To characterize the dynamical nature of each data, we resort to the following strategy. Regularization networks and linear autoregressive predictors are optimized using the first half of each data as library examples. We then make a comparison between the predictive errors as a function of the prediction-time interval generated by iterative forecasting with the nonlinear and the linear predictive models for the last half of each time series. In unstable dynamical systems a small change in an initial state of systems will grow exponentially with time through nonlinear dynamics. Such is not the case with linear dynamical systems. Hence unstable (nonlinear) dynamic behavior should be captured by nonlinear predictors better than by linear predictors when viewed on a characteristic timescale of the system. Nonlinear nature would be manifested in better predictive performance of the nonlinear predictor rather than the linear predictor.

2 Theoretical Preliminaries

Generalized regularization networks are of the general form [1, 2]:

$$y = f(\boldsymbol{x}) = \sum_{h=1}^{M} c_h\, G_h(\boldsymbol{x}) \qquad (1)$$

where G_h are the basis functions that have radial symmetry in many applications, M is the number of the basis functions, i.e., hidden nodes, and c_h are the parameters to be optimized through learning. Note that M is less than the number of library examples N_0 used for optimizing the network (usually $M << N_0$). Equation (1) describes an input-output mapping from vectors $\boldsymbol{x} \in R^p$ to vectors $\boldsymbol{y} \in R^q$. In application to time series prediction, given a time series $\{x(t)\}_{t=1}^{N}$, input vectors are generated by embedding as lagged sequences of data points equidistant in time:

$$\boldsymbol{x}(t) = (\, x(t), x(t-\Delta t), \cdots, x(t-(D-1)\Delta t)\,)$$

where D is the embedding dimension and Δt is an appropriately chosen time lag that should match the characteristic timescale of the system, and output vectors are scalar values $x(t + \tau\ \Delta t)$ where τ is the prediction-time step [4]. To measure the performance of the network, we introduce the risk functional defined to be

$$H[f] = \frac{1}{N_0} \sum_{t=1}^{N_0} \{ f[\boldsymbol{x}(t)] - x(t+\tau\ \Delta t) \}^2 + \lambda\ \psi[f] \quad (2)$$

$$\psi[f] = \| Pf \|^2$$

where λ is the regularization parameter, $\| \cdot \|$ is an L^2 norm, and P is an a priori given differential operator, referred to as the stabilizer, determining the smoothness of the approximating function. The first term of Eq.(2) gauges how well the network reproduces the library examples, while the second term gives penalty to the oscillating behavior of the approximating function. Let $Supp\ \psi = \{f \mid Pf \neq 0\}$ and $Ker\ \psi = \{f \mid Pf = 0\}$ be the support and the kernel (i.e., the null space) of the stabilizer, respectively. Then the approximating function belonging to $Supp\ \psi$ can be derived from

$$\frac{\delta H[f]}{\delta f} = 0 \qquad (3)$$

The approximating function belonging to $Ker\ \psi$ is usually given by algebraic polynomials of order $n-1$ or less for the stabilizer consisting of nth derivatives. The entire approximating function space is thus the direct sum of $Supp\ \psi$ and $Ker\ \psi$:

$$f[\boldsymbol{x}(t)] = \sum_{h=1}^{M} c_h\, G_h[\boldsymbol{x}(t)] + \sum_{i=0}^{n-1} \boldsymbol{d}_i \cdot \boldsymbol{x}^i(t) \qquad (4)$$

In the present work a particular form of regularization networks is invoked:

$$x(t+\tau\ \Delta t) = \sum_{h=1}^{M} \frac{c_h}{1 + cosh(-\beta_h | \boldsymbol{x}(t) - \theta_h |)} + E(\tau)$$

where $E(\tau)$ are random variables indicative of the prediction error, and c_h, θ_h, and β_h are network parameters to be optimized by stochastic gradient descent [5, 6] so as to minimize the empirical risk functional (i.e., $H[f]$ with $\lambda = 0$). Note that θ_h and β_h may be related to the statistical distribution of library examples. The basis function has the same class as Gaussian function. Hence the null space of the stabilizer is zero [1, 2]. This implies that no algebraic polynomials may be necessary. Nevertheless, we may add linear terms as a linear autoregressive (AR) predictor in parallel to the regularization network:

$$x(t+\tau\ \Delta t) = \sum_{h=1}^{M} \frac{c_h}{1 + cosh(-\beta_h | \boldsymbol{x}(t) - \theta_h |)} + \boldsymbol{a} \cdot \boldsymbol{x}(t) + d + E(\tau) \qquad (5)$$

where $\boldsymbol{a}$ and d are parameters to be determined in a least-mean-squared-error sense. In this networks architecture the optimization of the linear parameters is preceded to training the regularization network whose parameters are optimized so as to minimize the empirical risk functional consisting of the residuals of the linear AR predictor. Actual time series include a certain amount of stochastic ingredients such as the measurement error that are manifested in random noise. The dynamic behavior of random noise may be represented by a flat hyperplane in phase space (an embedding space). The linear terms added in parallel to the regularization network are thus expected to capture the dynamic behavior stemming from the observational noise. Chaotic ingredients, i.e., unstable (nonlinear) dynamic behavior should be captured by the regularization part of the network. This effect has been observed in time series prediction with Gaussian regularization networks in [5, 6]. In the present work the prediction error $E(\tau)$ is gauged by the the root-mean-squared-error between predicted and the corresponding actual values normalized by the standard deviation of library data.

3 Applications to Time Series Prediction

3.1 Temperature Sequence in a Blast Furnace

A first example is a time series of the temperature differences between input and output cooling water of

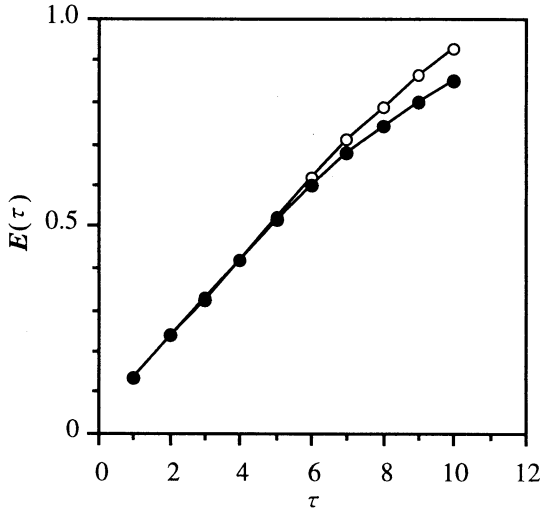


Figure 1. Plots of $E(\tau)$ versus τ estimated by an AR predictor (○) and a regularization network (Eq.(5)) (●) for the temperature data observed in a blast furnace.

water-cooling apparatus attached to the side wall of a blast furnace for iron-making [6]. The sampling time is 10 *min.* The data consist of 528 data points. During measurement no control actions were taken to the plant. From previous analysis, $D = 3$ and $\Delta t = 1$ were known to be appropriate [6]. Figure 1 shows plots of $E(\tau)$ versus τ estimated by an AR predictor and a regularization network ($M = 2$) including the AR predictor as Eq.(5). The predictors have been optimized using the library patterns generated from the first 300 data points. Although both predictors yield a similar prediction error at $\tau = 1$, the predictive performance of the nonlinear predictor surpasses that of the linear predictor for larger τ. These observations are considered to be the signature of nonlinear nature in the underlying dynamics. This agrees with the previous diagnosis by distinct algorithms that the temperature time series may represent low-dimensional chaos [6].

3.2 Global Temperature Variations

A second example is a time series of three months averages of global sea surface temperature anomalies over 544 periods from January of 1856 to December of 1991 [12]. From previous analysis, $D = 3$ and $\Delta t = 1$ were known to be appropriate [10]. Figure 2 shows plots of $E(\tau)$ versus τ estimated by an AR predictor and a regularization network ($M = 2$) including the AR predictor. The predictors have been optimized using the library patterns generated from the first half of the time series. In this case, there is no significant difference between the predictive performances of both predictors. Nonlinear nature is invisible. This implies that a climatic chaotic attractor is unlikely to exist, which is consistent with the previous analysis by distinct diagnostic algorithms [10].

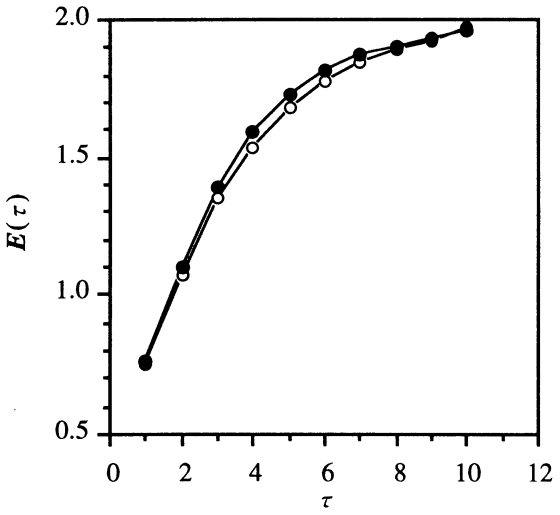


Figure 2. Plots of $E(\tau)$ versus τ estimated by an AR predictor (○) and a regularization network (●) for the meteorological time series.

3.3 Speech Signals of Japanese Vowel /a/

A last example is speech signals of Japanese vowel /a/ observed at a sampling rate of 20 *kHz* for one female and two male persons [11]. Each time series consists of 10000 data points. In this case the data contain regular and irregular ingredients: the regular ingredients correspond to the formants related to discrete power-law spectra, while the irregular ingredients are associated with continuous power-law spectra. We would like know whether or not the irregular ingredients represent chaos. Note that the regular signals incur no loss of information, i.e., no decay of predictability. From previous analysis, $D = 10$ and $\Delta t = 10$ were known to be appropriate [11]. Figures 3 and 4 illustrate plots of $E(\tau)$ versus τ estimated by an AR predictor and a regularization

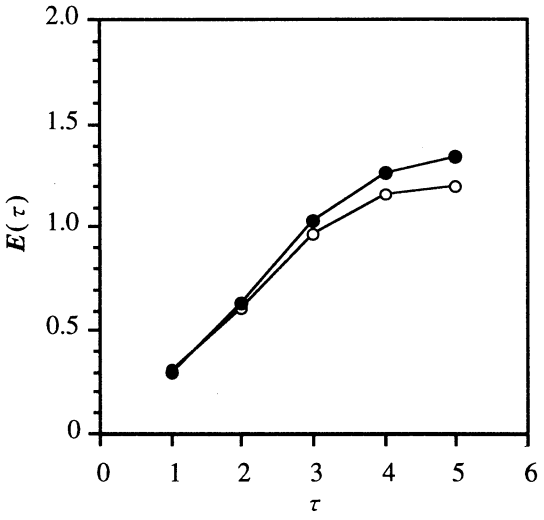


Figure 3. Plots of $E(\tau)$ versus τ estimated by an AR predictor (○) and a regularization network (●) for the female vocal time series.

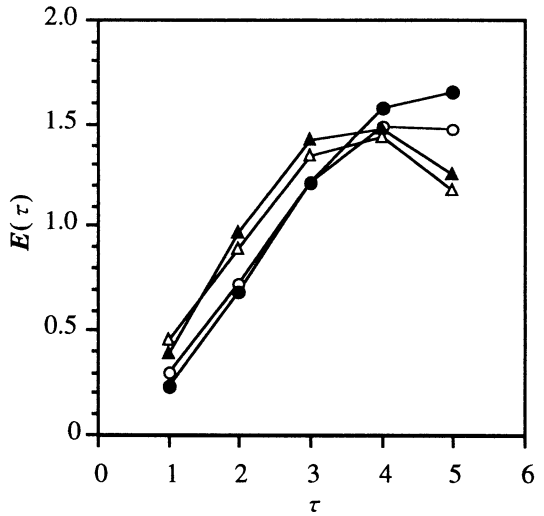


Figure 4. Plots of $E(\tau)$ versus τ estimated by AR predictors (○, △) and regularization networks (●, ▲) for the male vocal time series.

network (M = 6) including the AR predictor for the female and the male vocal time series, respectively. The predictors have been optimized using the library patterns generated from the first half of the time series. For the female data, the linear predictor achieves better predictive performance. For the male data, on the contrary, their dynamic behavior can be better captured by the nonlinear predictor at $\tau = 1$ and by the linear predictor at larger τ. The time series seem to have little nonlinearity. However, such interpretation may be problematic. Previous analysis based on the diversity of neighboring trajectories generated from the data in phase space indicated that determinism is less visible in the female data [11]. This implies that chaotic ingredients of the female data, if any, could be masked by stochastic noise. In fact, for the male data the nonlinear predictor better forecasts the dynamic behavior at $\tau = 1$, which could be the signature of chaos. Furthermore, the quick decay of predictability with time makes the estimates at larger τ very unreliable. We may not be able to make any conclusive interpretation on these results.

4 Conclusions

We have demonstrated that generalized regularization networks consisting of $(1 + cosh\ x)^{-1}$ really work, which encourages us the use of the basis function when attempting the hardware implementation of regularization networks. The comparison of the prediction error generated by linear and nonlinear predictors is a good way to detect nonlinear feature of complex time series. The present work suggests that chaos exists in the dynamic behavior associated with the chemical reaction in a blast furnace and that low-dimensional climatic chaos is unlikely to exist. The existence of vocal attractors is unclear in this work. One might raise a question as to the validity of the present diagnostic strategy because of the following reason: The representation capacity of the regularization network is larger than that of the AR predictor, hence the comparison of the prediction error generated by the linear and the nonlinear predictors with similar generalization error is unfair. To make the comparison fair, the predictors should be optimized in terms of information criterion such as Akaike's information criterion (AIC) [18]. We are not sure, however, that AIC can be a benchmark to gauge the representation capacity over linear and nonlinear estimators. This is a remaining issue to be explored.

Acknowledgment

We would appreciate Professor Reginald Newell and his research group at the Massachusetts Institute of Technology for offering the meteorological data.

References

[1] T. Poggio and F. Girosi, "Networks approximation and learning", *Proc. IEEE*, Vol. 78, pp. 357－381, 1990.

[2] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures", *Neural Computation*, Vol. 7, pp. 219－269, 1995.

[3] T. Poggio and S. Edelman, "A network that learns to recognize three-dimensional objects", *Nature*, Vol. 343, pp. 263－266, 1990.

[4] M. Casdagli, "Nonlinear prediction of chaotic time series", *Physica* D, Vol. 35, pp. 335－356, 1989.

[5] T. Miyano and F. Girosi, "Forecasting global temperature variations by neural networks", *A. I. Memo*, No. 1447, 1994.

[6] T. Miyano, H. Shibuta, K. Nakashima, and Y. Ikenaga, "Predicting chaotic sequences in a blast furnace by a generalized radial basis function networks", *Electronics and Communications in Japan*, Part 3, Vol. 79, pp. 1－10, 1996.

[7] N. K. Logothetis, J. Pauls, and T. Poggio, "Shape representation in the inferior cortex of monkeys", *Current Biology*, Vol. 5, pp. 552－ 563, 1995.

[8] J. Choi, B. Sheu, and J. Chang, "A Gaussian synapse circuit for analog VLSI neural networks", *IEEE Int. Symp. Circuits Syst.*, Vol. 6, pp 483－486, 1994.

[9] H. Inada, A. Nagami, and T. Miyano, "Generalized regularization network with $(1 + cosh\ x)^{-1}$ as the basis function", unpublished.

[10] T. Miyano, "Time series analysis of complex dynamical behavior contaminated with observational noise", *Int. J. Bifurcation and Chaos*, in print.

[11] T. Miyano, "Are Japanese vowels chaotic?", *Proc. 4th Int. Conference on Soft Computing*, Vol. 2, pp. 634－637, 1996.

[12] N. E. Newell, R. E. Newell, J. Hsiung, and Z. Zhongxiang, "Global marine temperature variation and the solar magnetic cycle", *Geophys. Res. Lett.*, Vol. 16, pp. 311－314, 1989.

[13] C. Nicolis and G. Nicolis, "Is there a climatic attractor?", *Nature*, Vol. 311, pp. 529－532, 1984.

[14] P. Grassberger, "Do climatic attractors exist?", *Nature*, Vol. 323, pp. 609－612, 1986.

[15] C. Essex, T. Lookman, and M. A. H. Nerenberg, "The climatic attractor over short timescales", *Nature*, Vol. 326, 64－66, 1987.

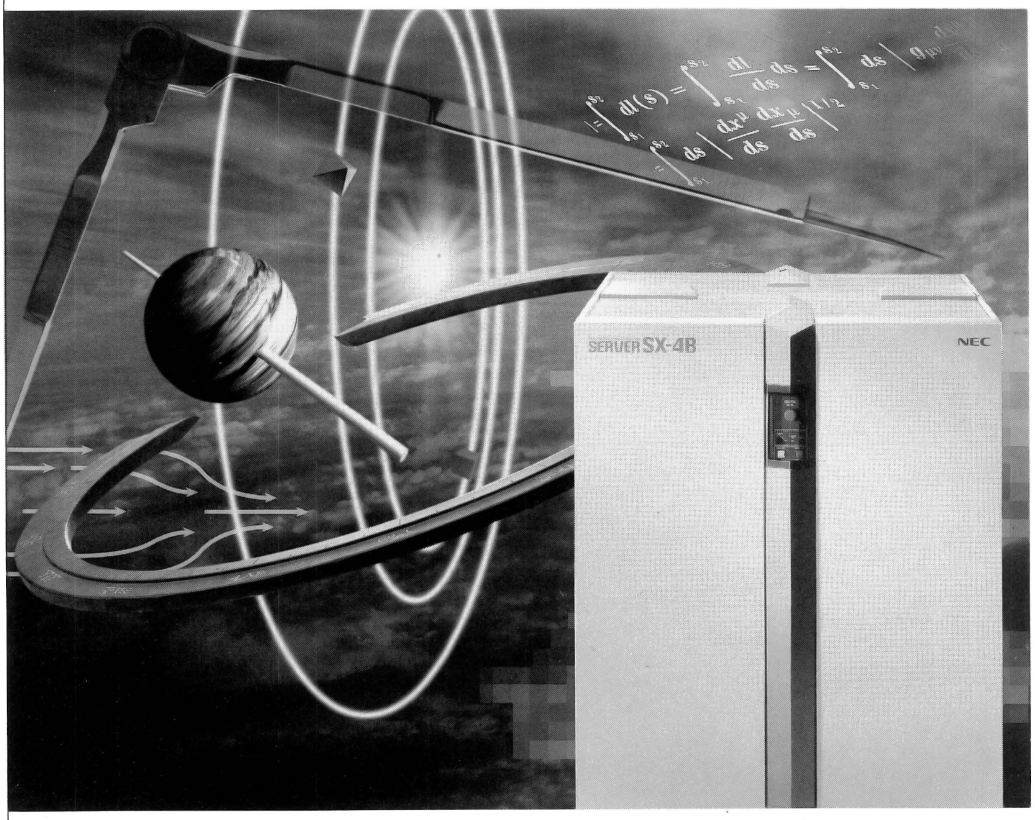
[16] E. N. Lorenz, "Dimension of weather and climate attractors", *Nature*, Vol. 353, 241－244, 1991.

[17] I. Tokuda, R. Tokunaga, and K. Aihara, "A simple geometrical structure underlying speech signals of the Japanese vowel /a/", *Int. J. Bifurcation and Chaos*, Vol. 6, pp. 149－160, 1996.

[18] M. Barahona and C. Poon, "Detection of nonlinear dynamics in short, noisy time series", *Nature*, Vol. 381, pp.215－217, 1996.

Author Index

SUPERCOMPUTER PERFORMANCE: THE SX-4B SERVER



Scientists may still be dreaming about personal supercomputers. But servers with supercomputer-class processing power have just taken a giant step toward affordability. NEC's SX-4B high-performance computing servers are economical enough for the exclusive use of an enterprise, laboratory department or project team.

The SX-4B includes three models with peak vector performance ranging from 0.9 GFLOPS to 3.6 GFLOPS. SX-4B servers feature high-speed CMOS technology. CPU performance ranges up to 1.8 GFLOPS. Synchronous SRAMs provide the main memory.

The SX-4B is compact and easy to install. It measures only 1m wide, 0.9m deep and 1.33m high.

Users of the new system work in a familiar UNIX environment. The server uses SUPER-UX, a mature, production-proven OS based on System V UNIX. The SX-4B supports a wide variety of third-party applications, including structural analysis, crash analysis and computational chemistry.

just imagine
NEC MULTIMEDIA

NEC

For further information, please contact: NEC Corporation, Supercomputers Marketing Promotion Division, 7-1, Shiba 5-chome, Minato-ku, Tokyo 108-01, Japan
Tel:+81-3-3798-9131.Fax:+81-3-3798-9132.e-mail:sx-4b@sxsmd.ho.nec.co.jp SX-4B information at http://www.nec.co.jp/english/product/computer/sx-4b/index.html