人工知能学会
第6回　SIG-Challenge 研究会

2000

RoboCup

2000年6月23,24日
公立はこだて未来大学
（第3回ロボカップジャパンオープン 2000）

目次

1. **RoboCup Challenge: Cognitive Developmental Robotics As a New Paradigm for the Design of Humanoid Robots,**
浅田 稔，カール マクドーマン (大阪大学大学院) ・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 1

2. 離散最適化問題としての走行誘導・経路計画と強化学習によるパラメータ決定法,
五十嵐治一 (近畿大学) ・・・ 7

3. 強化学習を用いたサッカーロボットの行動獲得,
大橋 健，福田 真人，榎田 修一，吉田 隆一，江島 俊朗 (九州工業大学) ・・・・・・・・・・・・・・・・ 13

4. 強化学習における **Fuzzy ART** を用いた状態空間の階層的分節化,
八谷 大岳，渥美 雅保 (創価大学) ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 19

5. マルチエージェント系における組織的学習効果についての一考察,
篠田 孝祐，國藤 進 (北陸先端科学技術大学院大学) ・・・・・・・・・・・・・・・・・・・・・・・・・・ 24

6. 優先度を考慮した多目的最適化手法としての適応的評価関数の提案,
柳瀬 正和，内部 英治，浅田 稔 (大阪大学大学院) ・・・・・・・・・・・・・・・・・・・・・・・・・・・ 30

7. 動画像処理によるロボカップのシーン解析,
須藤 智，郭 俸受，小竹 正裕，吉田 誠，小沢 慎治 (慶應義塾大学大学院) ・・・・・・・・・・・・・・・ 36

8. 全方位移動機構と全方位視覚を有する小型ロボットによるサッカー競技の実現–チーム**OMNI**の戦略–,
森 信人，家田 純一，松井 渉，臼井 智也，三宅 修，金 東杓，前田 哲裕，杉本 浩和，辰巳 優介，
藤本 良平，関森 大介，升谷 保博，宮崎 文夫 (大阪大学，明石工業高等専門学校) ・・・・・・・・・・ 42

9. **KU-Boxes2000**における画像処理と旋回性能の改良,
影山 茂，三吉 孝則，飯土居 修一，小末 将吾，五十嵐 治一 (近畿大学) ・・・・・・・・・・・・・・・・ 48

10. **KU-Boxes2000**における画像処理と旋回性能の改良,
影山 茂，三吉 孝則，飯土居 修一，小末 将吾，五十嵐 治一 (近畿大学) ・・・・・・・・・・・・・・・・ 48

11. エージェントの意思決定のための情報量基準による観測戦略,
光永 法明，浅田 稔，野原 達郎 (大阪大学大学院) ・・・・・・・・・・・・・・・・・・・・・・・・・・・ 54

RoboCup Challenge: Cognitive Developmental Robotics As a New Paradigm for the Design of Humanoid Robots

ロボカップチャレンジ：ヒューマノイドロボット設計の新たなパラダイムとしての認知発達ロボティクス

Minoru Asada[1] and Karl F. MacDorman[2]

浅田稔[1]，カール・マクドーマン[2]

[1] 大阪大学大学院工学研究科，[2] 大阪大学大学院基礎工学研究科

[1] **Graduate School of Engineering, and [2]Graduate School of Engineering Science, Osaka University**

asada@ams.eng.osaka-u.ac.jp

Abstract

This paper proposes *cognitive developmental robotics* as a new principle for the design of humanoid robots. This principle may provide new ways of understanding human beings that goes beyond the current level of explanation found in the natural and social sciences. Furthermore, a methodological emphasis on humanoid robots in the design of artificial creatures holds promise because they have many degrees of freedom and sense modalities and, thus, must face the challenge of scalability that is often side stepped in simpler domains. The potential of this new principle is examined, and future issues are given.

1 Introduction

We have known that many robot heroes and heroines in science fiction cartoons and movies like Star Wars in the US and Astor Boy in Japan have attracted us so much, which as a result motivated many robotics researchers. Unlike the special purpose machines, a robot has its own reason to exist as a multi-purpose intelligent machine performing a variety of complex tasks in the real world including the capability of communication with us. To realize such a machine, what is missing in the existing robotics? We advocate the need of *cognitive developmental robotics* (CDR), which aims to understand the cognitive development processes that an intelligent robot would require and how to realize them in a physical entity. However, cognitive developmental robotics has just started; therefore, its definition and methodology have not yet been established. In this paper, we discuss the potential and future issues of cognitive developmental robotics, which, we hope, stimulate many researchers not simply in robotics but also from other disciplines to discuss and tackle this controversial new paradigm.

The most significant meaning of CDR is its design principle. Existing approaches often explicitly implement a control structure in the robot's 'brain' that was derived from a designer's understanding of the robot's physics. On the other hand, CDR attempts to embed a structure into a robot which realizes such a process of understanding through the robot's interactions with its environment. Since both CDR and the traditional approach may lead to the same result, CDR may seem unnecessary if we evaluate it merely in terms of task performance. However, the traditional approach can break down if the robot's body or the environment are difficult to model or if they change unpredictably [1, 2].

Brooks et al. [3] proposed the methodology for alternative essences of intelligence as a humanoid design principle, which consists of parallel themes: development, social interaction, embodiment, and integration. Any of these themes seems essential for CDR, and we share very similar concepts. But, we emphasize more fundamental issues of cognitive development and propose a more constructive approach to CDR. Cognition and development have been key issues for human intelligence, and recent progress in these disciplines promoted a new area called *developmental cognitive neuroscience* (DCN) [4], which emerged at the interface between two of the most fundamental questions that challenge mankind. The first one concerns the relation between mind and body, and especially between the physical substance of the brain and the mental process it supports (cognitive neuroscience). The second concerns the origin of organized biological structure, such as highly complex structure of the adult human brain (development). Johnson claimed that light can be shed on these two questions by focusing on the relation between the postnatal development of the human brain and the cognitive process it supports.

The basic idea seems applicable to the approach of CDR since it has to deal with cognitive process during the development of a robot's brain. However, the dif-

ference between CDR and DCN is that CDR has the potential to verify its models by implementing them in humanoid robots. The cycle of fault diagnosis and reimplementation may iterate many times in order to refine the model [5]. This is precisely what a constructive approach is. The idea is that the model that results from this process of refinement — especially a model of interaction — might contribute as a useful model of human interaction. Since brain science is primarily concerned with structural details of human brains at the microscopic level, it may not be well suited for providing a comprehensive model of human interactions. Sometimes, however, the social sciences have attempted to understand human activities at a purely macroscopic level — without concern for the biological structure of individuals (for example, humans are sometimes treated as black boxes). CDR can take a role to bridge the two areas by providing a means for verification and suggesting other models. Rather, we expect that, through the process of designing and implementing humanoid robots, a new way of understanding human beings will develop that differs significantly from the ways in which humans are understood in the natural and social sciences.

As a concrete model, we propose an architecture consisting of two kinds design principles: the former is for an embedded structure capable of developing inside the robot brain, and the latter is for the issues of environmental design, which consists of spatio-temporal environment setups in order for the embedded robots to gradually adopt themselves to more complex tasks in more dynamic environments. It may include teaching by human or other robot instructor. Fig.?? shows typical methods of the embedded structure and environmental design issues. The rest of this article is organized as follows: First, we review "embodiment" as the least requirement for cognitive development, and then explain the design principles and the approaches of CDR. Finally, discussion and future issues are given.

2 Physical Embodiment and Interactions

Owing in part to the influence of a series of Brooks' publications (cf., [6, 7]), "having a physical body" has gradually become regarded as a necessary condition for designing the structure of intelligent artificial systems by AI and computer science researchers. A physical body enables an agent to interact with an environment, which we may expect could lead to the emergence of intelligent behavior and internal organization. Robotics researchers have never really disputed the need for having a physical body because it is essential to their research. Therefore, few have entered into a critical dialogue concerning the relationship between having a physical body and the emergence of intelligence. Here, we review the meaning of having a physical body (embodiment) [8].

1. Perception and action are not separable but tightly coupled.
2. Under resource-bounded conditions (memory, processing power, controller, etc.), an agent is able to learn a sensory-motor mapping based on experiences (interactions with an environment).
3. According to the increase of complexity of its task or environment, the agent is able to develop the consequence of its learning by reusing and modifying it in order to adapt itself to changes.

No one seems to have any objection to 1 and maybe 2. Pfeifer and Scheier explained "embodiment" in a variety of contexts in their book [9] based on the definition by Brooks [6]. However, they seem to put more emphasis on physical coupling than cognitive (and possibly physical body) development process. A typical example is passive dynamic walking [10] by exploiting the system dynamics. Here, we focus on cognitive development by adding 2 and 3 since current technology does not really support the realization of a growing, changing body.

3 Cognitive Science, Developmental Psychology, Neuroscience and Cognitive Developmental Robotics

"Developmental Cognitive Neuroscience" [4] has been emerged from cognitive science, developmental psychology, and neuroscience partially owing to recent progress of imaging technology in brain science. A fundamental controversy in cognitive science has been the relative importance of nature and nurture in determining the structure and behavior of individuals. One extreme is that gene coding has all kinds of information necessary for development. The other extreme is that much of the information involved in the formation of human mind comes from the environment. Both viewpoints are lacking. In the last decade new evidence has revealed that the complicated interactions between genes, developmental processes, and the environment lead to the emergence of structural organization and behavior at many levels [11].

The view that the conventional robotics methods corresponds to the nurture side since they embedded robot behaviors explicitly while CDR corresponds to the nature side is incorrect because both sides do not emerge new information as DCN pointed out. CDR aims at a constructive approach to realizing a mechanism that can adapt to complicated and dynamic changes in the environment based on its capacity for interaction.

4 Design Principles of CDR

From the standpoint of engineering, there are two sides to the design principle of CDR: (1) how to design a robot brain whose embedded structure can learn and develop;

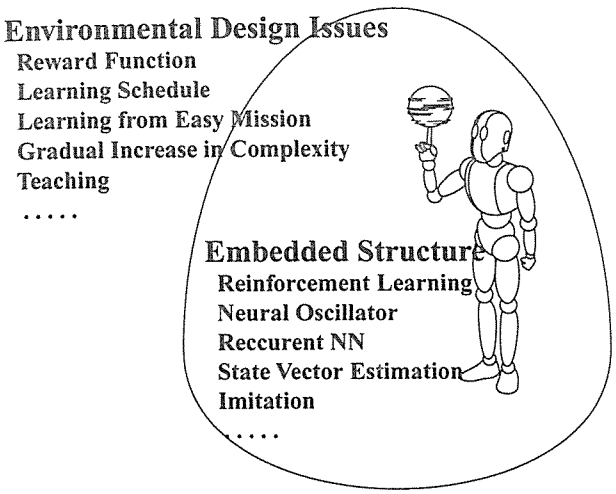


Table 1: Interaction between embedded structure and environment

and (2) how to create a social environment capable of supporting the development of cognitive processes.

4.1 embedded structure

The embedded structure is a mechanism which efficiently supports the interaction with an environment inside the robot. The information obtained through interaction will differ qualitatively depending on the size and organization of the robot's functional modules, which may range from the neural level to larger units such as visual and motor subsystems. However, a common feature is that new information emerges inside the robot. Reinforcement learning which map from sensory information to actuator outputs is a typical of functional module.

4.2 environmental design issues

The conventional robot design principle has put much more emphasis on the embedded structure than on environmental issues, although the produced behaviors seriously depend on both. Environmental design issues are essential for a robot with embedded structure to learn and develop so that it can gradually adapt itself to more complicated environments. Environmental design issues include all kinds of factors that come from outside the robot. Not simply the number and the configuration of stationary objects, but also other active agents should be considered in the case of multiagent learning, which is divided into several situations including only cooperative agents (rescue activities in a disaster situation), only competitive agents (a prey surrounded by predator), and both (game situations such as RoboCup [12]). Further, other agents can be coaches or teachers with any capability of communication with robots. From the viewpoint of development, Learning from easy missions (hereafter, LEM) [13], learning schedule [8], and gradual increase in complexity [14] are typical issues.

5 Approaches to CDR

Although a full scale implementation of a humanoid robot, built according to the principles of CDR, currently stands beyond our reach, for the time being, we can focus on essential issues in CDR, keeping the long-term goal in mind.

5.1 Development

Developmental issues have been examined within the reinforcement learning paradigm because it enables complex behavior to emerge through interaction without making many, often untenable assumptions about the structure and initial state of the internal mechanisms of cognition. Yet the flexibility of reinforcement learning has until recently also been its weakness; it results in a huge space of possible states and actions to explore. Only recently have researchers begun to develop powerful nonlinear algorithms that may be able to generalize across that space efficiently.

5.1.1 Guidance by starting with easy tasks

Although human beings live long enough for the various stages of cognitive development to unfold gradually [15], robots have not yet attained that level of reliability. *Robot shaping [16]* or *learning from easy missions* (LEM) [13] provide typical and intuitive methods for accelerating learning. In LEM the essential problem is how to define easy missions. One solution is that the robot starts from a situation that is close to the goal state and is gradually located further from the goal state as learning progresses. A distance measure is defined for the state space, and changes in the Q-values are used to determine when to shift to more difficult situations.

5.1.2 Environmental complexity control

Generally, the state space consists of multiple state axes, and therefore the question along which axis we should define the closeness to the goal state should be answered. This raises a more general issue. How do we define the complexity of the environment in terms of the developmental stage of the robot, and how do we adjust the environment to meet the robot's changing developmental needs? While at first it may seem that we are looking at the problem the wrong way around — adapting the task to the robot rather than the robot to the task — this is in fact what parent's do naturally in finding stimulating, age-appropriate ways of interacting with their children. Asada et al. [8] defined the complexity of the environment in terms of the relationship between the change of the sensory information and self-induced motor commands.

1. **Body of its own and static environment:** The body of its own or static environment can be defined in a way that notes the changes in the image plane that can be directly correlated with the self-induced

motor commands (e.g., looking at your hand showing voluntary motion, as does changing your gaze to observe the environment).

2. **Passive agents:** As a result of actions of the self or other agents, passive agents can be moving or still. A ball is a typical example. As long as they are stationary, they can be categorized into the static environment. But no simple correlation of motor commands with its body or the static environment can be expected when they are in motion.
3. **Other active agents:** Active agents do not have a simple and straightforward relationship with self motions. In the early stage, they are treated as noise or disturbance because they lack direct visual correlation with the self-induced motor commands. Later, they can be found from more complicated and higher order correlations (coordination, competition, etc.). The complexity is drastically increased.

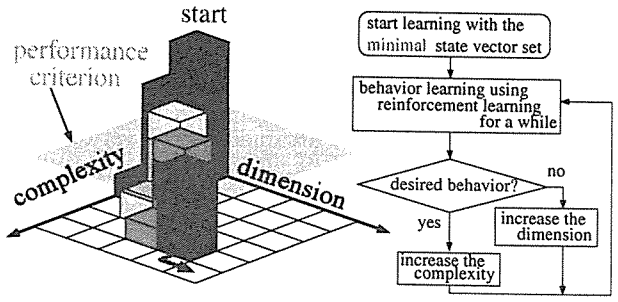


Table 2: Interaction between embedded structure and environment

According to the complexity of the environment, the internal representation of the robot should be more sophisticated and complex in order to generate various intelligent behaviors. Now, the problem is how to find such complexity. Uchibe et al. [17] invented an algorithm to estimate the state vectors of which dimensions corresponds to the complexity, and they applied the method to skill up the shooting behavior in a simplified one (defender) on one (shooter) soccer game [14]. Fig. 2 shows how the dimensionality of the state vector grows according to the increase in the environment's complexity (speed of the defender). As long as the performance (success rate) exceeds the threshold, we increase the complexity (speed). If the success rate drops below the performance criterion, the robot increases the dimension of the state vector.

The final performance by the environmental complexity control method reached the same or slightly better one than the the learning with full set of the state vector from the beginning with more than three times learning time. Since the robot is a physical entity, enormous amount of learning time is not realistic. Therefore, *starting small* [11] is important like child language learning [18].

5.1.3 Learning schedule in multiagent learning

In the last example [14], the number of the learner is just one, and the behavior of the other agent (defender) is controlled according to the skill level of the learner. What kind of control is possible if the both are learners? Generally, simultaneous learning in multiagent situations is difficult because the policy in the early stage of the learning can be viewed random to each other since the learning starts with much less exploitation phase but more exploration one, therefore the learning on the both sides may diverge. Then, learning schedule is introduced [8], in which the number of the learner is always one and the other agents have fixed policies during the learning. The learner changes to each other when the learner's skill reached some level of the pre-specified criterion. They applied a cooperative task in the context of RoboCup [12], that is, passing and shooting tasks. Mutual skill development which is generally difficult to reach in case of co-evolution [19] is successfully obtained by the learning schedule.

5.2 Social interaction and communication

The possible contribution of CDR to new ways of understanding human beings is to provide a model of the developmental process of communication and to verify the model using humanoid robots. Especially, from nonverbal communication to verbal one, that is, the process of symbol emergence and language acquisition. This is a considerably big issue (a missing link between primate and human species [20]), and therefore we are not able to give any through survey of the existing areas such as linguistics, philosophy, and sociology. Instead, we focus on the issue from a viewpoint of engineering design.

Over the last few decades language researchers seem to have reached a consensus that language is an innate ability, and human babies are born with such an ability like "language device" [21]. Broca and Wernicke areas seem to be such devices, but things are not so simple. Because, it seems difficult to say that all language abilities can be reduced to the activities in these areas, but there are many areas related to such activities implicitly. Also, from a viewpoint of biological continuity from primate to human species, the claim that only human beings have such a device seems difficult to accept. Without question, human brains come into the world specially equipped something for the language ability. So the problems facing with CDR are:

1. What kind of structure should be embedded inside robot brain? We do not care whether it is explicit or implicit, but expect to give a new explanation for the missing link.
2. From which level should we start? Gene or neuron? Since we do not intend to follow the whole biolog-

ical evolution process for a million of billion years (this might be able to be done by simulation, but this causes the similar question on which level and in what kind of environment), we should carefully select the level.

The current technology produced the speech recognition and generation systems useful in some contexts. These systems are supported by a huge amount of language knowledge, therefore their use is limited and difficult to say that the system understands the meaning of the language as a tool for communication like a Chinese room example [22]. In their book [11], Elman et al. showed the ability of grammar learning by artificial neural networks as a tool for verification. However, inputs and outputs are all symbols without any semantics. We intend to start how symbols emerge through the interactions with an environment [23]. Steels and Voget [24] implemented adaptive language games using robotic agents, and their approach seems closer to us, but they assumed a protocol to make robots communicate. Since CDR aims to have a new interpretation for the missing link, we should focus how such a protocol is generated between humanoid robots.

Schaal [25] surveyed imitation learning methods, and emphasized the importance of imitation as the route to humanoid robot focusing the efficient motor learning, the connection between action and perception, and modular motor control in form of movement primitives, and pointed out the open problem such as learning perceptual representations, movement primitives, movement recognition through movement generation, and understanding task goals. The third topic is related to the recent finding that some of neurons called "mirror neuron" were active both when the monkey grasps or manipulates objects and when it observes the experimenter making similar actions. Rizzolatti and Arbib [26] speculated that the ability of imitate actions and to understand them could have subserved the development of communication skills based on the fact that similar system includes Broca area (known to be related to speech generation) in human brain.

From a viewpoint of CDR as a humanoid robot design principle, such a system should be included because capabilities of both motion generation by imitation and motion understanding (matching with own motion repertory) seem necessary. There seems mainly two kinds of imitation pathways: *visual imitation:* (in other words, imitation learning by observation) and *audial imitation:* (in other words, imitation learning by listening).

The existing methods (ex. [27, 28, 29]) for the former often assume the global coordinate transformation from a god's eye viewpoint, however CDR should focus how such a transformation is emerged through the interactions between humanoid robots and/or humans like a baby and its parents. Asada et al. [30,

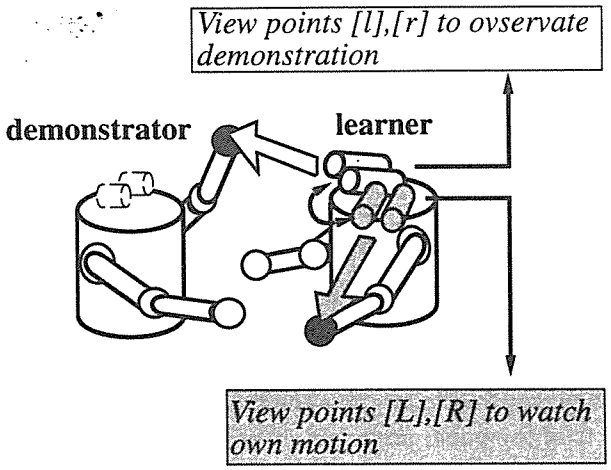


Table 3: An imitation system

?] proposed an imitation system which recovers the other agent view without any knowledge of global coordinate transformation but assuming that the other agent has the same body structure (see Fig 3), which they expect to route motion generation, motion understanding, and then mind reading ("the theory of mind" [31]).

Since the mechanical structure of the human speech generation system is quite complicated and the sophisticated integration of voluntary and involuntary muscle controls is necessary to generate sounds [20], the audial imitation has an essential problem at which level the robot should start to imitate. The imitation means not a simple copy but an ability to generate something new.

6 Conclusion

We have argued a variety of issues on CDR, most of which are far from maturation since CDR has just started. Among them, two issues seem essential for future arguments. The first one is environmental complexity definition which can be expected to have a role in realizing a developmental process by increasing the complexity. However, definition itself involves the contradiction, that is, before encountering a new environment the robot cannot define the complexity. In the example method [14], the complexity corresponds to the dimension of the estimated state vector which is obtained in off-line process, but actually it seems difficult to estimate the full dimension of the state vector before learning. One alternative is to develop an online method of state vector estimation.

The second one is imitation as social interaction between the learner and teacher. There are several levels of interactions, each of which has its own issues. If the teacher knows everything about the learner like a god, the teacher can guide the learning process optimally. Actually, however, both are independent agents, therefore, there is a limit to such knowledge. Then, issues are what

kind of communication lines (regardless whether it is explicit or not: vision, auditory, or any other specific lines with predetermined protocol) are available, and to what extent the teacher knows the learner's state. Imitation learning seems essential to develop the cognitive process both motion generation and language acquisition.

Acknowledgement

The authors like to express their appreciation to Dr. Yasuo Kuniyoshi, ETL, and Prof. Hiroshi Ishiguro, Wakayama University for their constructive discussions and suggestions to improve the draft of the paper. Some parts of the paper are based on the reference [32].

参考文献

[1] K. F. MacDorman. Grounding symbols through sensorimotor integration. *Journal of the Robotics Society of Japan*, 17(1):20–24, 1999.

[2] K. F. MacDorman. Responding to affordances. In *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, 2000.

[3] C. Breazeal (Ferrell) R. Irie C. Kemp M. Marjanovic B. Scassellat Brooks, R.A. and M. Williamson. Alternate essences of intelligence. In *AAAI-98*, 1999.

[4] Mark H. johnson. *Developmental Cognitive Neuroscience.* Blackwell Publisher Inc., Cambridge, Massachusetts 02142, USA, 1996.

[5] S. J. Cowley and K. F. MacDorman. Simulating conversations: The communion game. *AI & Society*, 9(3):116–137, 1995.

[6] R. A. Brooks. "A robust layered control system for a mobile robot". *IEEE J. Robotics and Automation*, RA-2:14–23, 1986.

[7] R. A. Brooks. "Elephants don't play chess". In P. Maes, editor, *Designing Autonomous Agents*, pages 3–15. MIT/Elsevier, 1991.

[8] Minoru Asada, Eiji Uchibe, and Koh Hosoda. Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence*, 110:275–292, 1999.

[9] Rolf Pfeifer and Christian Scheier99. *Understanding Intelligence.* The MIT Press, Cambridge, Massachusetts 02142, USA, 1999.

[10] T. McGeer. "passive walking with knees". In *Proc. of 1990 IEEE Int. Conf. on Robotics and Automation*, 1990.

[11] J. Elman, E. A. Bates, M. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. *Rethinking Innateness: A Connectionist Perspective on Development.* The MIT Press, Cambridge, Massachusetts 02142, USA, 1999.

[12] Minoru Asada and Hiroaki Kitano, editors. *RoboCup-98: Robot Soccer World Cup II.* Springer, Lecture Note in Artificail Intelligence 1604, 1999.

[13] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[14] Eiji Uchibe, Minoru Asada, and Koh Hosoda. Environmental complexity control for vision-based learning mobile robot. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 1865–1870, 1998.

[15] Jean Piaget. *Child's Conception of the World.* Littlefield Adams, 1990.

[16] Marco Dorigo. *Robot shaping: An experiment in behavior engineering.* MIT Press, Cambridge, MA, 1997.

[17] E. Uchibe, M. Asada, and K. Hosoda. "state space construction for behavior acquisition in multi agent environments with vision and action". In *Proc. of ICCV 98*, pages 870–875, 1998.

[18] E. L. Newport. Maturational constraints on language learning. *Cognitive Science*, pages 11–28, 1990.

[19] D. Floreano, S. Nolfi, and F. Mondada. Competitive co-evolutionary robotics: From theory to practice. In *Proc. of the 5th Int. Conf. on Simulation and Adaptive Behaviors - From animals to animats 5 -*, pages 515–524, 1998.

[20] Terrence W. Deacon. *The Symbolic Species: The co-evolution of language and the brain.* W. W. Norton & Company, New York, London, 1998.

[21] Norm Chomsky. *Language and Mind.* New York: Harcourt Brace Jovanovich, 1972.

[22] J. R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, pages 417–457, 1980.

[23] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[24] L. Steels and P. Voget. Grounding adaptive language games in robotic agents. In *Proc. of the 4th European Conf. on Artificial Life*, pages 474–482, 1997.

[25] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Science*, 1999.

[26] G. Rizzolatti and M. A. Arbib. Language within our grasp. *Trends Neuroscience*, 21:188–194, 1998.

[27] K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation. *IEEE Trans. on R&A*, 10:368–385, 1994.

[28] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching. *IEEE Trans. on R&A*, 10:799–822, 1994.

[29] M. Kawato F. Gandolfo, H. Gomi, , and Y. Wada. Teaching by showing in kendama based on optimization principle. In *Proc. of the International Conf. on ANN*, pages 601–606, 1994.

[30] M. Asada, Y. Yoshikawa, and K. Hosoda. Learning by observation without three-dimensional reconstruction. In *Proc. of the 6th International Conf. on Intelligent Autonomous Systems (IAS-6)*, page (to appear), 2000.

[31] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, pages 515–526, 1978.

[32] Minoru Asada, Hiroshi Ishiguro, and Yasuo Kuniyoshi. Toward cognitive robotics (in japanese). *Journal of the Robotics Society of Japan*, 17(1):2–6, 1999.

離散最適化問題としての走行誘導・経路計画と強化学習によるパラメータ決定法

Path Planning and Navigation of a Mobile Robot and Adjustment of Weight Parameters by Reinforcement Learning

五十嵐治一
Harukazu Igarashi

近畿大学工学部（広島県東広島市）
School of Engineering, Kinki University
igarashi@info.hiro.kindai.ac.jp

Abstract

In this paper, we propose a solution to path planning and navigation of a mobile robot. In our approach, we formulate the following two problems at each time step as discrete optimization problems: 1) estimation of position and direction of a robot, and 2) action decision.. To solve the optimization problems, we use objective functions consisting of several terms. This paper presents a theoretical method using reinforcement learning for adjusting the weight parameters in the objective functions.

1 はじめに

自律移動型ロボットの走行誘導方式に関しては，これまでに多くの提案がなされている[1]-[7]が，環境世界や使用するロボットに関する前提条件によって，対象とする問題の性格は大きく異なってくる．また，必要となる技術も問題によって異なってくる．

一般に，環境地図が与えられており，障害物の位置，形状，行動が既知である場合においては，走行誘導に必要とされる主な技術は，軌道(または経路)の計画と自己の位置・姿勢の推定である．一方，環境地図が未知であれば，地図作成などの環境学習や過去の行動履歴などを用いた行動学習の技術が重要な研究課題となってくる．

仮に対象を既知環境下における軌道計画問題に限定しても，障害物の移動の有無，ロボットのサイズ・形状，性能面の制約（速度や加速度の上限[7]など），床面の状況（weighted-region 問題[5]），ゴール到達時の時間窓制約，安全性（壁面からの距離制約など）などの点から，さまざまな種類の軌道計画問題が存在する．従来，軌道・経路計画の分野で提案されている，ポテンシャル法，スケルトン法，空間分割法といった手法[2][5]は，主として最短性と障害物回避との観点から軌道・経路計画を行っている．その結果，大局的に最短と思われる概略軌道を空間のスケルトン化とグラフ探索により求めておき，その上で詳細軌道をポテンシャル法で求めるという方式が最も妥当であると考えられている[1]．しかし，上記の制約・前提条件やユーザの要求仕様[6]の追加・変更に対しては，アルゴリズムや制御プログラムの大幅な修正を余儀なくされる場合もあり，我々は柔軟性という点で改善の余地があると考えた．

また，位置・姿勢推定の問題も，使用するセンサの種類，精度によって推定アルゴリズムが異なってくる．さらには，複数のセンサを用いる場合には，いかにそれらのセンサ情報を統合すべきかと言った問題[8]も生じてくる．極言すれば，センサの組合せの数だけ位置・姿勢推定アルゴリズムを考案する必要があると言える．

本論文では，既知環境下における自律移動型ロボットの走行誘導問題において，軌道・経路計画に要する行動決定と位置・姿勢推定との問題を，それぞれ別の離散的最適化問題として定式化することにより，問題の多様性に対応できる柔軟性のある手法を提案する．

2 走行誘導の基本処理サイクル

前章でも述べたように，自律移動型ロボットの走行誘導の問題には，たとえ，既知環境下であっても，制約条件，前提条件，ユーザの要求する仕様[6]等による多様性が存在する．その多様性が，走行誘導のアルゴリズムを複雑にしている要因と考えられる．そこで，図１に示すような走行誘導アルゴリズムの基本サイクルを考えた[9]．この基本サイクルは，i)センシング(sensing)，ii)位置・姿勢の推定(Estimating a robot's location)，iii)行動決定(planning)，iv)行動(action)，の４ステップからなる処理サイクルである．本方式では，時間を離散化し，各時刻においてこのサイクルを実行する．

以下，この基本サイクルの各ステップについて説明する．まず，ステップ i では，障害物までの距離を測定するための超音波センサ，赤外線センサ，レーザー光センサ，その他の距離センサ，あるいは，自走距離を計測するエンコーダや加速度センサ，さらには，視覚情報を取り込むための CCD カメラなどを用いて，壁や障害物などの環境情報を獲得する．

ステップ ii では，得られた距離データ，環境地図（壁や障害物の位置が正確に記載），直前での位置・姿勢と行動などを手がかりに現在のロボットの位置と姿勢（ロボットの正面前方の方向）とを推定する．以下では，ロボットの位置と姿勢とをあわせて配置(location)と称する．

さらに，ステップ iii では，ゴール地点までの距離・方向，衝突回避，ユーザの要求仕様などを考慮して，現時点で最適と思われる行動を決定する．具体的には，ロボットの速度ベクトルを決定する．ステップ iv では，その行動を実行する．

本方式では，上記のステップ ii（配置推定）とステップ iii（行動決定）とを，それぞれ別の離散最適化問題（組合せ最適化問題）として定式化する．離散最適化問題の目的関数としては，1.で述べたような問題の多様性を考慮して，目的や制約をペナルティ量の線形結合の形で表現する．すなわち，位置・姿勢推定と速度決定の問題とをそれぞれの目的関数の最小化問題に帰着させる．

このような定式化であれば，目的関数中の各目的と各制約の重要度は項の重み係数の値により表現することができる．さらに，考慮すべき目的や制約を追加，あるいは除去したい場合でも，それらに対応する項の追加や除去により容易に対応できる[6]．したがって，本方式は走行誘導問題の多様性に柔軟に対応できる方式である．

また，本方式のように各時刻で次の行動を逐次決定していく方式であれば，その時点における環境に関する最新情報をロボットの行動決定に反映させることができるので，走行前には部分的にしか環境情報が得られなかった場合や行動予測ができない移動障害物への対応がしやすいという利点がある．その上，走行中に発生する目的の変更（例えば，ゴール地点の変更など）への柔軟な対応も可能である．また，未知環境下での行動学習の問題に関しても，目的関数中のパラメータの学習により対応できる場合もある．

さらに，通常の軌道・経路計画問題の場合のように，スタート地点からゴール地点までのすべての時刻での行動の計画を走行前に予め立てておきたい場合には，図1の処理サイクルにおいてステップ i を省略し，ステップ ii で各時刻におけるロボットの真の配置を与えてやれば，容易にシミュレータ上で経路や軌道を立案することができる[10]．

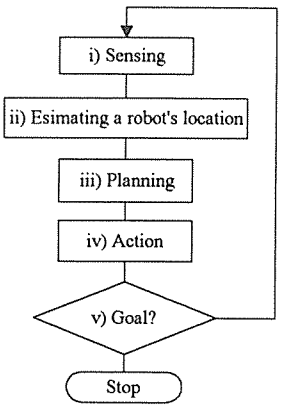


図1　走行誘導の基本処理サイクル

3　走行誘導の最適化

3.1 目的関数

前章では，自律移動型ロボットの走行誘導の問題を，配置推定問題と行動決定問題とに分離し，各時刻においてこの2つの問題を交互に解いていく方式を提案した．

本方式ではこの配置推定問題と行動決定問題とにおいて，それぞれ目的関数を定義し，その目的関数を最小化する解を求める．ここでは，それらの目的関数を以下のように表すことにする．

$$E_l(l_t) \equiv \sum_{i=1,\ldots,m} a_i E_{li}(l_t) \qquad (1)$$

$$E_v(v_t) \equiv \sum_{j=1,\ldots,n} b_j E_{vj}(v_t) \qquad (2)$$

$E_l(l_t)$は配置推定問題の目的関数であり，時刻 t における配置 l_t の関数である．また，$E_v(v_t)$は，行動決定問題の目的関数であり，時刻 t における行動 v_t の関数である．いずれの目的関数も，いくつかの項の線形和で表されていると仮定する．項の重み係数が，それぞれ$\{a_i\}(i=1,\ldots,m)$と$\{b_j\}(j=1,\ldots,n)$とであり，各項の重要性を表している．4章では具体的な目的関数の例を示す．ただし，本論文で述べる方式は，目的関数が上記のような線形和以外の関数であることや，パラメータとして重み係数以外のパラメータを含むことを適用範囲外とするものではない．

3.2 走行誘導の目的

ここまで，自律移動型ロボットの走行誘導問題を明確には定義してこなかった．本論文で対象とする(既知環境下での)走行誘導問題を以下のように定義する．

> [走行誘導問題]
> 環境地図が与えられている場合に，指定した 2 地点間を結び，制約条件とユーザの要求仕様とを満足する軌道上をロボットに走行させること．

走行軌道に対する制約条件やユーザの要求仕様としては，走行時間の最短化，安全性（壁面からの距離の確保や速度上限など），運動性能面での制約（加速度の上限，回転角の制約），タスク上の要求（特定地点への経由，接近など）等がある．

3.3 軌道計画

図1のステップii（配置推定）において，各時刻でのロボットの真の配置を与えることにより，シミュレータ上で軌道計画を行うことができる．この際，問題となるのは目的関数E_v中の重み係数$\{b_j\}$である．この値が適切に調整できないと，ユーザの要求仕様を満足する軌道を生成することはできない．従来，このような重み係数は，試行錯誤的に手動修正するのが殆どである．しかし，確率的な行動決定（確率的政策）と強化学習とを用いると，生成された軌道uに対してユーザが与える報酬R(u)の期待値V=E[R(u)]を最大化（正確には極大化）するように，次の学習則により重み係数を調節することができる[10].

$$\Delta b_k = +\varepsilon \frac{\partial V(\pi)}{\partial b_k}$$

$$= -\frac{\varepsilon}{T_v} E\left[R(u) \cdot \sum_{t=0}^{N(u)-1} \left\{ \frac{\partial E_v(v_t)}{\partial b_k} - \left\langle \frac{\partial E_v(v_t)}{\partial b_k} \right\rangle_{T_v,\{b_k\}} \right\} \right] \quad (3)$$

ここで，εは学習係数（正の定数），N(u)は軌道uの最終時刻，T_vは政策決定時の揺らぎの大きさを表すパラメータ，<...>は以下の期待値操作である．

$$\langle X \rangle_{T,\{b_k\}} \equiv \frac{\sum_v X \cdot e^{-E_v(v;\{b_k\})/T}}{\sum_v e^{-E_v(v;\{b_k\})/T}} \quad (4)$$

(4)で定義された期待値は，時刻tにおける行動v_tの探索範囲が離散化かつ局所的に限定されていれば，容易に計算することができる．ただし，ここでは確率的政策として以下のBoltzmann分布による政策$\pi(v_t)$を用いる．

$$\pi(v;\{b_k\},T) \equiv \frac{e^{-E_v(v)/T}}{\sum_v e^{-E_v(v)/T}} \quad (5)$$

このように，環境地図を与えてあらかじめシミュレータ上で重み係数を学習しておき，その重みを用いて，各時刻において$E_v(v_t)$の最小値（局所探索なので実際は極小値）を与える行動の列が一つの軌道を決定する．これは，(5)の確率的政策πにおいて，T=0と置くことに相当する(決定論的政策).

なお,(3)の右辺の期待値操作E[...]は必ずしも計算する必要はなく（確率的勾配法），{bj}は局所的な変数であっても (3)の学習則は使用することができる．また，環境が迷路のような複雑な場合で，各地点ごとに進むべき方向を学習したい場合や，あるいは，各地点ごとに取るべき行動を直接学習させたい場合には，$E_v(v_t)$に，例えば，

$$\sum_{r,v} Q(r,v)\delta_{v,v_t}\delta_{r,r_t} \quad (6)$$

という項を入れておけば良い．ここで，rは位置座標，r_tは時刻tにおける位置，vは行動を表し，ともに離散化されているものとする．δはクロネッカーのδ記号である．すなわち，位置rにおいて，行動vを選択したときの妥当性をパラメータQ(r,v)で表現しておけば，そのパラメータを(3)の学習則を用いて学習することができる．このパラメータは，Q学習でのQ値に対応すると考えることができる．

3.4 配置推定

ロボットの配置（位置，姿勢）を推定するには，センサ値と環境地図との照合，内界センサや過去の行動履歴からの推定（dead reckoning）などの方法がある．これらの推定法を目的関数$E_l(l_t)$の最小化問題として定式化する．もちろん，グローバルビジョン（例，天井カメラ）などを用いた配置推定結果l^{obs}も，例えば，$\Sigma_t(l-l^{obs})^2$などの項を重みをつけて，$E_l(l_t)$に入れておけば容易に考慮することができる．他の推定手段による推定結果も同様に取り入れることができる．

また，この際，重みをローカル変数とすれば，ある地点ではどの推定手段が有効であるかということも表現することができる．

さらに，項の種類を選ぶと，配置推定の理論的な取り扱い方式として提案され，実際のツアーガイド用ロボットに採用されているMarkov localization法[11]を近似することもできる[9].

ところが，ここで問題となるのは，重み係数$\{a_i\}$の値をどう定めるかということである．次節で述べるように，実は，この重み係数の値も(3)と同様な学習則により学習することができる．

3.5 軌道計画と配置推定との統合

本節では，3.3で述べた軌道計画法と3.4で述べた配置計画法とを統合した走行誘導法を提案する．すでに，2章で行動決定と配置推定とを交互に繰り返す走行誘導の基本処理サイクルを提案し(図1)，各処理をそれぞれ式(1),(2)で示された目的関数の(局所的な)最小化問題に帰着させていた[9].

この走行誘導方式は，以下の式で定義される確率P(u)の右辺において，$T_l \to 0$かつ$T_v \to 0$とすることにより得られる軌道uを求めることと等価である．

$$P(u) \equiv \prod_{t=1,\ldots,N(u)} Bel(l_t) P^{\pi}(v_t) \qquad (7)$$

この式で，Bel(*l*)はロボットの配置が*l*であるという主観的信念である．式(7)では，確率的政策πにより時刻tにおいて行動v_tが選択される確率を$P^{\pi}(v_t)$で表しており，

$$P^{\pi}(v) \equiv \frac{e^{-E_v(v)/T_v}}{\sum_v e^{-E_v(v)/T_v}} \qquad (8)$$

で定義されている．

さらに，配置推定にも次式で定義される確率的な推定を用いる．

$$Bel(l) \equiv \frac{e^{-E_l(l)/T_l}}{\sum_l e^{-E_l(l)/T_l}} \qquad (9)$$

ここで問題となるのは，重み係数$\{a_i\}$, $\{b_j\}$の値である．実機ロボットを，(8),(9)を用いて図1の基本処理サイクルに従って走行させ，得られる軌道uに報酬を与えて，この2種類の重み係数を同時に強化学習することもできるが，以下の手順で，別々に学習する方が学習時間の短縮を図ることができる．

[重み係数の学習法]

a)まず，シミュレータ上で，3.3で述べた方法により重み係数$\{b_j\}$の値を定める．

b)図1に示した基本処理サイクルに従いロボットを走行させる．ただし，ステップiiiで用いる重み係数$\{b_j\}$の値は，上記ステップaで得られた値を用い，決定論的に行動決定を行う(i.e. (8)でT_v=0とした場合に相当)．一方，配置推定には(9)の確率的推定法を用いる．

c)ステップbにより得られた軌道uに対して報酬R(u)を与え，次式により重み係数$\{a_i\}$を更新する．

$$\Delta a_j = +\varepsilon \frac{\partial E[R(u)]}{\partial a_j}$$

$$= -\frac{\varepsilon}{T_l} E\left[R(u) \cdot \sum_{t=0}^{N(u)-1} \left\{ \frac{\partial E_l(l_t)}{\partial a_j} - \left\langle \frac{\partial E_l(l_t)}{\partial a_j} \right\rangle_{T_l,\{a_j\}} \right\} \right] \qquad (10)$$

ただし，(10)の期待値操作E[...]は計算する必要はない（確率的勾配法が使える）．

d)ステップbとステップcとの操作を十分繰返す．

4 既知環境下における走行誘導の例

4.1 ロボットと環境

例題として，単数ロボットが静止障害物の間を通って，出発地点からゴール地点まで移動する問題を考える．簡単のために，ロボットには大きさはない(point robot)が，向きは定義可能で，近距離センサを周囲に搭載した円柱型ロボットと仮定する．その動特性は，スポット回転は可能であるが，並進移動は正面方向へのみ可能とする．また，障害物は多角形で表され，その正確な位置と形状を記した地図情報は使用可能とする．

4.2 配置推定問題における目的関数の例

式(1)に示した目的関数$E_l(l_t)$の例としては，以下のような関数が考えられる．

$$E_l(r_t, \alpha_t; R_t^{obs}, r_{t-1}, v_{t-1}) \equiv a_1 E_{data} + a_2 E_{cnst} + a_3 E_{prdt} \qquad (11)$$

ただし，配置*l*はロボットの位置r=(x, y)と，姿勢（ロボットの正面方向）αとからなる．(11)の右辺の各項の意味と定義は以下のとおりである[9]．

● $E_{data}(r_t, \alpha_t; R_t^{obs})$：データ項．時刻tで観測された距離データパターン$R_t^{obs}$と，推定された位置・姿勢において観測されるべきパターン$R_t(r_t, \alpha_t)$との相違量を表現する．

$$E_{data}(r_t, \alpha_t; R_t^{obs}) \equiv \left[R_t(r_t, \alpha_t) - R_t^{obs} \right]^2 \qquad (12)$$

● $E_{cnst}(r_t, \alpha_t; R_t^{obs})$：拘束項．移動ベクトル$\Delta r_t \equiv r_t - r_{t-1}$の方向$Dir(\Delta r_t)$がロボットの正面の向き$\alpha_t$であるという拘束条件を表現している．

$$E_{cnst}(r_t, \alpha_t; r_{t-1}) \equiv \begin{cases} [\alpha_t - Dir(r_t - r_{t-1})]^2 & if \quad r_t \neq r_{t-1} \\ 0 & if \quad r_t = r_{t-1} \end{cases} \qquad (13)$$

● $E_{prdt}(r_t, \alpha_t; r_{t-1}, v_{t-1})$：予測項．時刻t-1における推定結果$(r_{t-1}, \alpha_{t-1})$と行動$A_{t-1}$とから，時刻tにおける位置と姿勢が予測できる．この予測値を(r'_t, α'_t)で表す．行動A_tとは，時刻tにおける速度ベクトルの制御である．

$$E_{prdt}(r_t, \alpha_t; r'_t, \alpha'_t) \equiv (r_t - r'_t)^2 + (\alpha_t - \alpha'_t)^2 \qquad (14)$$

$$(r'_t, \alpha'_t) \equiv (v_{t-1}\Delta t + r_{t-1},\ Dir(v_{t-1})),\ v_{t-1} = A_{t-1}(r_{t-1}, v_{t-2})$$

(13), (14)からわかるように，時刻tにおける位置・姿勢の推定は，1サイクル前の時刻t-1の推定位置と行動とに依存している．したがって，ここでの配置推定決定過程は非マルコフ的な決定過程である．また，ここで定義した目的関数の最小化は，Markov localizationアルゴリズムと対応をつけることができる[9]．

4.3 行動決定問題における目的関数の例

式(2)に示した目的関数$E_v(v_t)$の例としては，以下のような関数が考えられる[9]．

$$E_v(v_t; r_t, v_{t-1}, r_{goal}) = b_1 E_{goal} + b_2 E_{smth} + b_3 E_{clsn} \quad (15)$$

ここで，v_tは時刻 t におけるロボットの速度ベクトル，r_{goal} はゴール地点の(地図上での)位置である．この目的関数は，時刻 t において速度ベクトル v_tの値を選択した場合の不適当な度合い(ペナルティの量)を表現している．(15)の右辺の各項の意味と定義とは以下のとおりである．

● $E_{goal}(v_t;r_t,r_{goal})$：引力項．ゴールへの到達を要請する．ただし，sgn( )は符号を，‖ ‖はベクトルの大きさ(ユークリッドノルム)を表す．

$$E_{goal}(v_t; r_t, r_{goal}) \equiv \mathrm{sgn}[G(v_t)] \cdot G(v_t)^2 \quad (16)$$

$$G(v_t) = \left\| r_{goal} - r'_{t+1}(v_t; r_t) \right\| - \left\| r_{goal} - r_t \right\| \quad (17)$$

● $E_{smth}(v_t;v_{t-1})$：滑らか項．速度ベクトルの変化を小さくし，ロボットの移動経路を滑らかにする．

$$E_{smth}(v_t; v_{t-1}) = \left\| v_t - v_{t-1} \right\|^2 \quad (18)$$

● $E_{clsn}(v_t;r_t)$：斥力項．障害物や壁面との衝突回避．

$$E_{clsn}(v_t; r_t) = \begin{cases} D_{clsn} & \text{if } Dist(r'_{t+1}) < 0 \\ -Dist(r'_{t+1})^2 & \text{if } 0 < Dist(r'_{t+1}) < R \\ -R^2 & \text{if } Dist(r'_{t+1}) > R \end{cases} \quad (19)$$

ここで，Dist(r)は位置 r から障害物や壁面までの最短距離．D_{clsn}, R は正の定数である．D_{clsn}はロボットが次時刻 t で障害物や壁面に衝突した場合のペナルティの大きさ，R は障害物や壁面からの斥力ポテンシャルが働き始める最小限の距離である．(19)は，次時刻における障害物からの距離，すなわち，安全性を評価して速度ベクトルを決定するための項である．

4.4 軌道計画の例

3.3 で述べたシミュレータ上での軌道計画の例を示す．ここでは，行動決定のための目的関数として(15)を用いる．また，軌道 u に対するユーザの報酬 R(u)としては，

$$R(u) \equiv c_1 R_{time}(u) + c_2 R_{dist}(u) \quad (20)$$

を用いた例を示す[10]．ここで，$R_{time}(u)$は到達時間ステップ数に関する希求水準を軌道 u がどの程度満足しているかという度合いを表しており，$R_{dist}(u)$は全時刻における障害物・壁までの距離の最小値に関する希求水準を軌道 u がどの程度満足しているかという度合いを表している．ユーザの希求水準に対する満足度を表したこれらの関数を，制約目標値関数(achievement function)と呼ぶ．今回は，制約目標値関数として，図 3 のような台形状のアナログ関数を用いた[10]．図 3 に示した制約目標値関数は，各項目に関するユーザの要求レベルを 0 から 10 の間の実数値で表現している（具体的な形状は文献[10]参照）．また，式(18)の右辺のパラメータ c_1 と c_2 とは，各項目の重要度を表現しており，実験では c_1=c_2=0.5 と設定した．なお，パラメータを全く含まない積の形の報酬関数も試したが，ほぼ同じ結果を得ている．

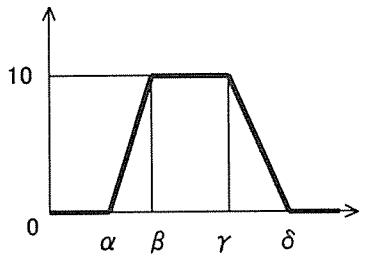


図 3　実験で用いた制約目標値関数

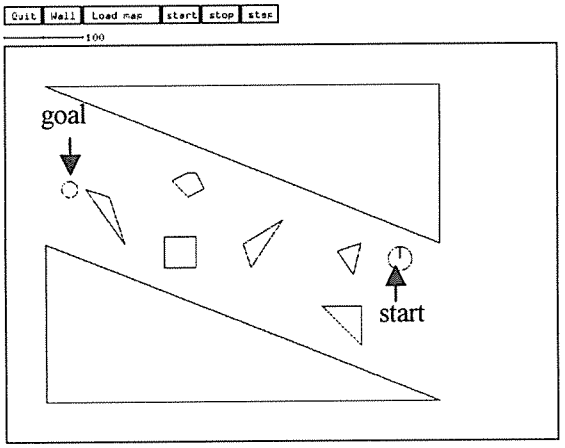


図 4　実験で用いた環境とスタート／ゴール地点

図 5 に報酬 R(u)の期待値（1000 回更新ごとの平均値）を，図 6 に$\{b_j\}$(j=1,2,3)の値の変化(b_1=b_2=b_3=1.0 に初期設定)を，図 7 には学習前の重み係数を用いた場合に決定論的に得られる軌道を，図 8 に学習後の重み係数を用いた場合に決定論的に得られる軌道の一例（25000 回更新）を示す．この例では，壁面・障害物からの斥力項の重み係数 b_3が学習により強化され，障害物から十分な距離を保った安全な軌道が得られているのがわかる．

5　今後の課題

現在のところ，軌道計画に処理時間が相当かかっている(SUN Ultra Sparc30 を使用して約 36 時間)．これは主としてシミュレータ作成上の問題であり，壁面・障害物とロボットとの距離計算等を高速化する必要がある．

また，配置推定に用いる目的関数の重み係数$\{a_i\}$の学

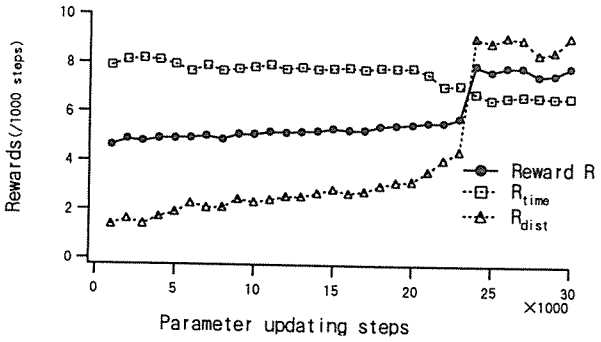


図5　報酬の期待値(1000 回の平均)の変化

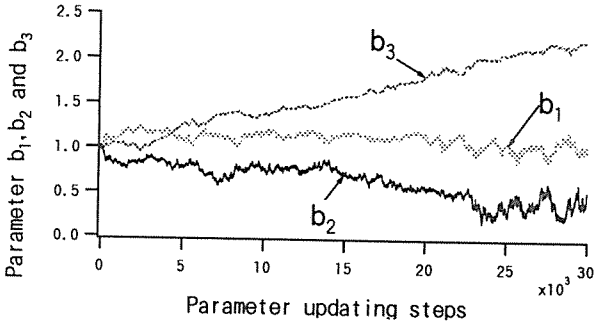


図6　重み係数$\{b_i\}$(i=1,2,3)の学習

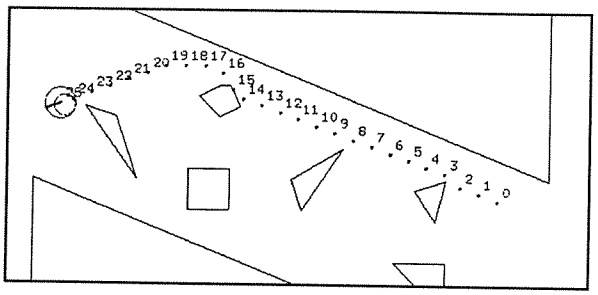


図7　$\{b_i\}$の学習前，決定論的に得られていた軌道

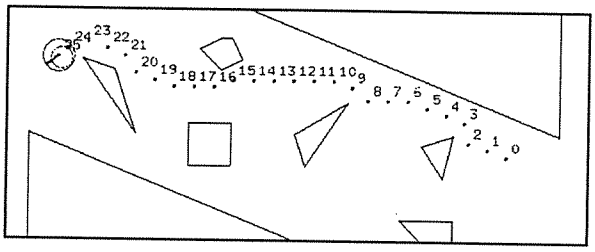


図8　$\{b_i\}$の学習後(25000 回更新)，決定論的に得られた軌道

習アルゴリズム(3.5 参照)を，シミュレーションや実機ロボットを用いて検証する必要がある．

謝辞:本研究に関してご討論いただいた，本学部五百井清助教授に感謝の意を表する．なお，本研究は，日本学術振興会より科学研究費補助金（C(2)，課題番号11680405）の助成を受けた．

参考文献

[1]. 小森谷　清，小谷内範穂，" 移動ロボットの知能，" 日本ロボット学会誌，Vol.9，No.1，pp.100-111，1991.

[2] 藤村希久雄，" 行動ストラテジーとアルゴリズム，" 日本ロボット学会誌，Vol.11，No.8，pp.1124-1129，1993.

[3] T. Arai, H. Ogata, and T. Suzuki, "Collision Avoidance Among Multiple Robots Using Virtual Impedance," IEEE/RSJ International Workshop on Intelligent Robots and Systems 9, Sep.4-6, Tsukuba, Japan, pp.479-485,1989.

[4] D.Kortenkamp, R.P.Bonasso, and R.Murphy(eds.), Aritificial Intelligence and Mobile Robots, AAAI Press/The MIT Press,1998.

[5] Y.K. Hwang and N.Ahuja, "Gross Motion Planning -A Survey," ACM Computing Surveys, VOl.24, No. 3, pp.219-292, 1992.

[6] 小方博之，新井民夫，太田順，" 時変環境でユーザ仕様を考慮した移動ロボットの軌道計画法，" 日本ロボット学会誌，Vol.12，No.6，pp.905-910，1994

[7] B.H.Krogh and C.E. Thorpe, " Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1664-1669, 1986.

[8] 浅田稔，" ロボットの環境モデリング，" センサフュージョン，山崎弘郎，石川正俊著，pp.215-226，コロナ社，東京，1992.

[9] 五十嵐治一，五百井清，" 最適化問題としての経路計画と走行誘導"，第 4 回ロボティクスシンポジア，予稿集 pp.269-274（1999.3，仙台）．

[10] 五十嵐治一，" 離散最適化問題としての自律移動型ロボットの経路計画，" 第 9 回インテリジェント・システム・シンポジウム，pp.408-413，Oct. 1999.

[11] S.Thrun, A.Bucken, W.Burgard, et al.," Map Learning and High-Speed Navigation in RHINO," in [4], pp.21-52.

[12] W.Burgard, D.Fox and S.Thrun," Active Mobile Robot Localization," Poc. of 15th Joint Conference on Artificial Intelligence, Vol.2, pp.1346-1352, 1997.

強化学習を用いたサッカーロボットの行動獲得

Behavior Acquisition of Soccer Robot using Reinforcement Learning

大橋健，福田真人，榎田修一，吉田隆一，江島俊朗

Takeshi OHASHI, Masato FUKUDA, Shuichi ENOKIDA,

Takaichi YOSHIDA, Toshiaki EJIMA

九州工業大学情報工学部

Kyushu Institute of Technorogy

{ohashi,masa,enokida,takaichi,toshi}@mickey.ai.kyutech.ac.jp

Abstract

Autonomous Robots, which are required to do complex tasks, should be flexible for small change of environment. The flexibility can be realized by learning through interactions between robots and environment to acquire knowledge about both environment and a given task. In this paper, we focus on the nested Q-learning, which is designed to calculate optimal sequences of actions for going through with the task as well as to emerge hierarchical structure reflecting the task structure simultaneously. Directly applying the nested Q-learning to a practical problem (for example, making a soccer robot to play smartly), however, cause the cost problem of time and state space. To overcome the problem, first, based on priori knowledge, we give a hierarchical structure reflecting the task. Then, Q-tables obtained through learning are properly integrated to make a suitable mapping from sensor inputs space to actions space of the given robot. The proposed algorithm is applied to design a soccer robot supposed to be an attacking player in a soccer team. As a result, the learning algorithm is effective to adjust deviations due to variation of camera position and/or the wheel and axle.

1 はじめに

強化学習は、自律型ロボットがタスクを達成させる手法を獲得する強力な手法である。そのようなロボットは、センサーからの入力により環境を認識し、自らの行動を計画する。もしそのロボットが強化学習を使うならば、環境から得られる強化信号により最適な行動政策を獲得することができる。強化信号は、それまでの行動に対する報酬または罰を表している。Q-learning[Watkins, 1989] は、最もよく用いられる強化学習アルゴリズムの１つである。この手法は、環境を離散的な状態で表わし、その状態における各行動に対する評価値を表の形にした Q-table を利用する。基本的に Q-learning は、反射的な行動の獲得に適しているが、Q-table は平坦な構造を持っているので、複雑なタスクに適応するのは難しい。ここでは難しいタスクとして、いくつかの簡単なタスクの組み合わせまたは連続するタスクを扱うことにする。

Degney B.L. らが提案した nested Q-learning[Degney, 1996][Degney, 1998]は、頻繁に訪ずれる状態を探し、その状態を新たなサブタスクと考えそれを Q-table で表し、Q-table の木構造をつくる手法である。これは、強化学習の能力を向上させる仕組として興味深い研究であるが、現実世界の問題に対しては適応し難しいと考えられる。なぜなら、どのようにセンサーからの入力を扱えばよいかという問題と、構造を創発させるにはどれだけの学習が必要であるかという問題があるからである。現実世界では、多くの場合にセンサーからの入力は連続値として与えられる。これを詳細に表現するためには、高い周波数の標本化と高解像度の量子化が必要であり、これにより計算コストは増大してしまう。これらを強化学習で扱うためには、センサー空間の適切な分節化が必要である[高橋, 1999][榎田, 1999]。また、学習により構造が創発できるかという点については、典型的な鶏と卵問題である。Q-table の構造は、学習が進行し Q-value の値が十分埋まらなければ創発できない。学習を進行させるためには、適切な構造を持っていないとゴールに辿りつけず、報酬が得られない。

我々は、現実世界のサッカー選手の複雑な行動を Q-learning を用いて獲得できるよにすることに焦点を当て、具体的に RoboCup[Kitano, 1998]中型機リーグのロボットを主な対象とした。ロボットはビジョンセンサーなどにより自律的に行動しなければならない。攻撃するプレーヤの主な行動は、自分のホームポジションに戻る、ボールを探す、ボールに接近する、相手ゴールに向ってシュートする、相手のロボットや壁などを避けるなどである。この論文において、これらの行動をサブタスクとして Q-table

で表現し、全体の攻撃行動をこれらのサブタスクを下位層として持つ階層構造で表現する。以下の章では、強化学習を簡単に紹介しどのようにこの階層構造を設計するかを検討する。そして、実際のロボットを用いて、実環境における学習実験を行い、その結果について考察する。

2 強化学習

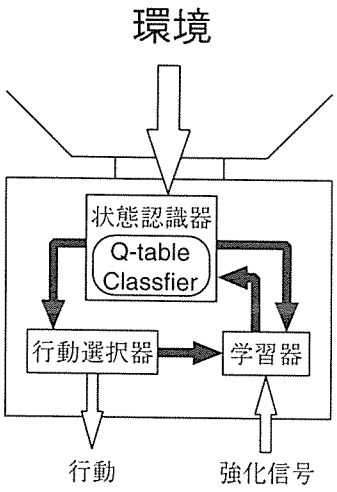


Figure 1: 強化学習の枠組

強化学習は、報酬や罰によって環境に適応する機械学習アルゴリズムである。自律型ロボットでは、環境を知覚するためにセンサーを用い、センサーからの入力に対して適切な行動を選択する。ロボットは、目的のゴールに到達したとき、一連の行動に対する強化信号を環境から受けとる。このような強化学習の概念を Figure 1 に示す。

自律型ロボットは、Figure1 のように、状態認識器、行動選択器、そして学習器から構成される。状態認識器は、環境に関する情報をセンサーから入力し、現在の状態における行動価値関数を出力する。行動選択器は、行動政策に従って適切な行動を決定する。学習器は、強化信号をもとに選択した行動の行動価値を更新する。

2.1 Q-learning

最もよく用いられる強化学習アルゴリズムが、Q-learning[Watkins, 1989] である。行動価値関数 $Q(s,a)$ は、現在の状態 $s \in \mathrm{S}$ における行動 $a \in \mathrm{A}$ の価値を表す。ロボットが状態 s において、行動 a を行い、状態 s' に遷移しとき、$Q(s,a)$ は式 (1) に従って更新される。

$$Q(s,a) \Leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{b \in A} Q(s',b)) \quad (1)$$

行動価値関数は、Figure2 のような Q-table として表現される。十分な学習の後は、行動関数は最適な行動が最大値を取るように収束する[Watkins, 1989]。通常、行動選択には式 (2) のような Boltzmann 分布に従って確率的選

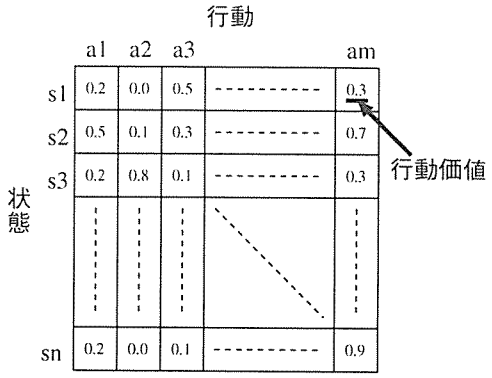


Figure 2: Q-Table 概念図

択器が用いられる。

$$P(s,a) = \frac{\exp(Q(s,a)/T)}{\sum_{a' \in \mathrm{A}} \exp(Q(s,a')/T)} \quad (2)$$

これにより、これまで選択されなかった行動にも選択される機会を与えつつ、なるべく行動価値を反映した選択を行うことができる。

3 サッカーロボットの行動獲得

3.1 サッカー行動の構造

サッカーにおいて適切に行動することは、ロボットだけでなく人間にとっても複雑なタスクである。人は基本的な動作を学習し、簡単な行動、総合的な行動と順を追って練習する。例えば、基本的な行動として、ドリブル、キック、ヘディングやランニングなどがあり、簡単な行動として、ボールを追いかける、ボールをシュートする、相手を避ける、ホームポジションに戻るなどがあり、これらを総合した攻撃行動や守備行動がある。これらの行動は、Figure3 のように階層的な構造で表現できる。

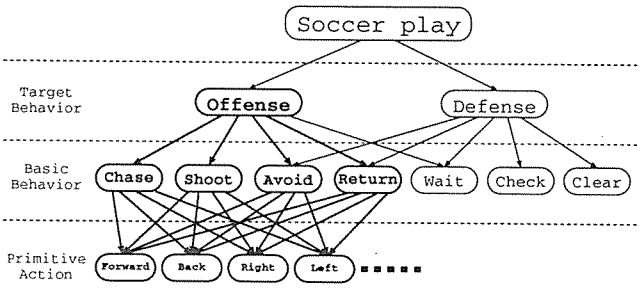


Figure 3: サッカータスクの階層の例

Nested Q-learning[Degney, 1996][Degney, 1998] は、自己組織化する仕組を導入した Q-learning である。この手法では、頻繁に訪れる状態を発見し、その状態をサブタスクとして新たな Q-table を作成し、行動を選択するのと同等にサブタスクの Q-table を選択できるようにしている。この仕組により、階層的な Q-table を作成している。これは、革新的なアプローチであるが、実際のロボットに適応するには 2 つの問題が考えられる。

1. 状態空間の大きさ：現実世界の問題は、膨大な状態空間が必要であり、このため最適な行動を学習するのが難しい。
2. 自己組織化：実際に役立ちそうなサブタスクを発見するのが困難であり、自己組織化が難しい。

最初の問題については、状態空間の大きさを適切に縮小する必要があり、状態空間を連続的な関数モデルで扱う手法や、似ている状態を統合する手法[高橋, 1999]などが提案されている。通常は、サブタスクで考慮すべきものは環境中のいくつかの対象物体に限られるので、全てを扱う必要はない。つまり、タスクに応じて、入力をフィルタリングすることが有効である。

二つ目の問題は、実環境における実ロボットにとって難しい問題である。実ロボットでは、目の前で起きることを全て予測することは不可能であり、動作を続けながら対応しなければならない場面が想定される。予測できていない事象はシミュレータで事前に学習することはできないので、実ロボットで学習可能であるかどうかは重要な課題である。Takahasi ら[Takahashi, 1999]は、階層的学習機構を用いて、複雑な問題に対応させているが、サッカーの攻撃行動全体の段階まで複雑な問題を実環境上で学習するのは難しいと思われる。この論文では、Figure3 に示すタスクの階層構造を用い、それぞれのサブタスクを実機のロボットにおいて、攻撃行動全体を獲得させることを検討した。学習方法の詳細は、第 3.5 章に示す。

3.2 実験対象とする環境について

実験対象とする環境は、RoboCup 中型機リーグであり、フィールド内の物体は認識しやすくするために色分けされている。ボールはオレンジ、ゴールは青と黄色、壁は白地に黒いロゴマーク、フィールドは緑でラインは白、そしてロボットは黒である。

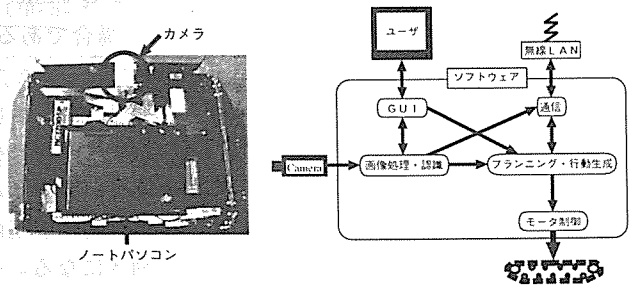


Figure 4: サッカーロボット (実機)

3.3 ロボットの機能

我々 KIRC のサッカーロボットとソフトウェアの構造を Figure4 に示す。ロボットは正面に向けたカメラを持ち、環境を認識する。そして、左右に独立したモータとギヤボックスを持ち、選択可能な行動は、図 5 に示す 9 通り

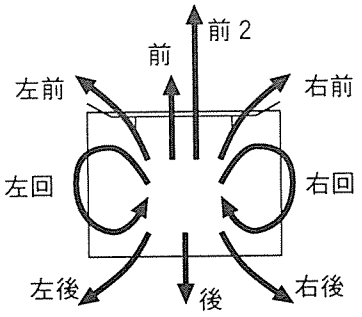


Figure 5: ロボットの行動

とする。

図 6 に示すように、カメラからの画像は画像処理により目的の物体ごとに抽出される。

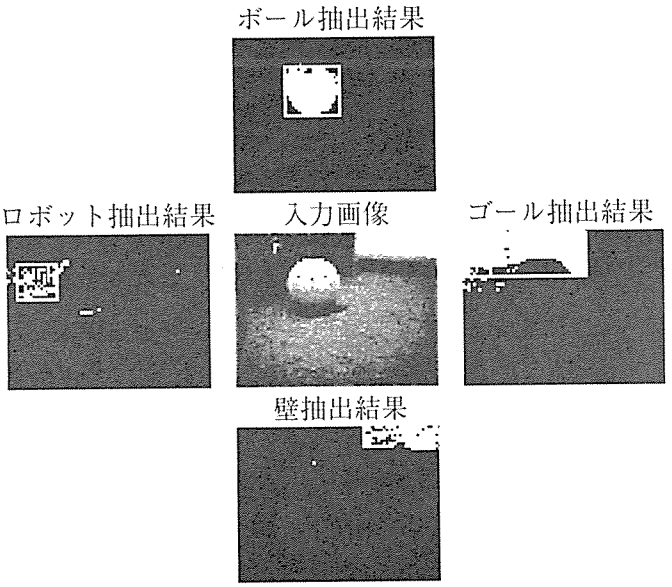


Figure 6: 入力画像と認識結果

3.4 環境に対する状態空間

それぞれの色を手掛りとして抽出しされた対象の物体は、状態空間を構成する情報に置き換えられる。ボールとロボットの状態、対象物体までの実際の距離と方向、壁の状態を Figure7 に示す。ボールやロボットは、距離 (3 状態) ×方向 (3 状態) ＋見失った状態 (左側と右側) の 11 状態として表現され、図中に示す数字は、画像の横幅に対する相対値である。壁については、距離と見失った状態のみを扱うこととする。

3.5 学習戦略

攻撃行動について、サブタスクを構成を検討する。状態空間は、それぞれのサブタスクに応じて適切に設計されなければならない。なぜなら、実際のロボットで学習するためには、空間が大きすぎると学習が収束しないからである。攻撃行動については、それぞれのサブタスクにおいて次

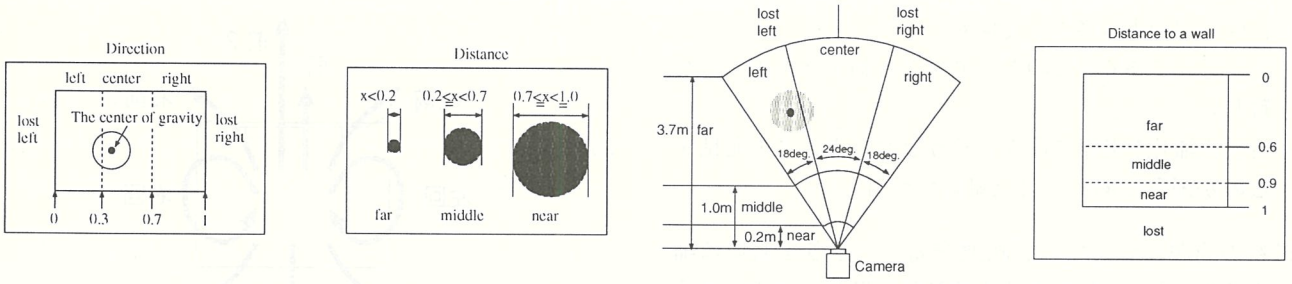


Figure 7: ボール，ロボットの状態と実際の距離と壁の状態

の物体のみを扱う空間を構成する。

1. Chase タスク：ボールを探し、それをキープする。
2. Shoot タスク：ボールを相手ゴールにシュートする。
3. Return タスク：自分のホームポジションに戻る。
4. Avoid タスク：相手ロボットや壁との衝突を避ける。

このとき、それぞれのサブタスクでは、以下の状態空間を用いればいよい。

- Chase タスクでは、ボールの状態のみの 11 状態。
- Shoot タスクでは、見失なった 2 状態を統合した 10 のボール状態と、10 のゴール状態、これら全体で 100 状態。
- Return タスクでは、ボールの 11 状態と、相手ゴールが見えているかいないかの 2 状態、5 つの味方ゴールの状態、壁の 4 状態、これら全体で 440 状態。
- Avoid タスクでは、ロボットの 10 状態と、壁の 4 状態、これら全体で 40 状態。

もし、これらのサブタスクに分けずに攻撃行動全体を表わすとすると、合計 4400 状態となり、これを用いて実際のロボットで学習するのは困難である。なぜなら、Q-learning では、ある程度ゴールに到達して環境から報酬が与えられ、学習が進行するまでは、効果的な行動がほとんど選択されないため、学習の進行が遅いからである。その一方、本手法のように簡潔なサブタスクに分ければ、実ロボットでの学習も可能である。攻撃行動全体は、これらのサブタスクの組み合わせとして表わし、衝突回避を優先し、ボールを探してシュートするようにサブタスクの切替はプログラムしたものを用いる。各サブタスクは、Q-learning でそれぞれ実ロボットを用いて学習し、このときの学習率 $\alpha = 0.25$、減衰率 $\gamma = 0.99$、温度定数 $T = 0.1$ を用いる。

3.6 実ロボットによる学習実験

3.6.1 実ロボットによる学習実験

各サブタスクの学習は、Q-learning を用いて独立に行う。1 回の学習の終了条件は、サブタスクのゴールに到達するか同じ行動を 30 ステップ以上選択しつづけるかである。各サブタスクのゴールと学習回数を表 2 に示す。学習の効果を評価するために、各サブタスクに対応する Q-table を手動で設定したものと学習により獲得したものを用いて比較する。例として、手動で設定した Chase タスクの Q-table を Table1 に示す。全体の行動は以下の流れにそって行う。

Table 2: サブタスクのゴールと学習回数

タスク	ゴール状態	学習回数
Chase	ボールが近くて中心にある	400
Shoot	ボールとゴールが近くて中心にある	100
Return	ボールが見えず味方ゴールが近い	300
Avoid	ロボットか壁が近い (負の目的)	100

1. 近くに障害物があれば、Avoid タスクを実行する。
2. それ以外で、ボールが近ければ Shoot タスクを行い、相手ゴール付近であれば Return タスクを行い、そうでなければ、Cchase タスクを実行する。
3. 以上を繰り返す。

各サブタスクと全体の行動の実験結果を Table3 に示す。このときの、試行回数は 100 回であり、**Q-learning** は、Q-learning により獲得した Q-table を用いた実験の結果であり、**Manual** は、手動による Q-table を用いた場合、**Manual+Q-learning** は、手動による Q-table をもとに Q-learning で 50 回の再学習を行った場合である。

3.6.2 障害発生時の学習実験

階層的に Q-table を構成した場合の適応能力を調べるために、左のモータに障害が発生して回転速度が減少した場合を想定した実験を行う。このとき、前進行動は、左前進に近い動きになり、後退は左後退に近い動きになる。その一方、右前進と右後退はほぼ真っ直の動きとなる。これは、実際のロボットに不意の事故が発生したときを模擬している。障害発生後の Chase タスクの実験結果を Table4 に示す。ここで、障害発生後に適応させるための追加学習の回数は 100 回である。障害により成功率は、86 から 45 に減少しているが、追加学習の後は、82 まで回復している。学習後の動作を解析すると、右前進を前進の代りとして用いている。このことから、Q-learning は環境の変化

Table 1: 手動で設定した Chase タスクの Q-table

	行動								
状態	前進	速い前進	後退	左に後退	右に後退	右回転	左回転	右前進	左前進
0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	1.0
1	0.5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.5	0.0	1.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.5
4	1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.5	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0

ボールの状態		左	中央	右	
遠い	9:左に見失った	0	1	2	10: 右に見失った
中位		3	4	5	
近い		6	7	8	

ここで、状態 7 はゴール状態である。

Table 3: 実験結果

サブタスク	成功回数	状態遷移回数
Chase[Q-learning]	49	13.2
Chase[Manual]	81	17.3
Chase[Manual + Q-learning]	86	16.9
Shoot[Q-learning]	46	8.3
Shoot[Manual]	74	16.5
Shoot[Manual + Q-learning]	69	12.6
Return[Q-learning]	42	21.4
Return[Manual]	52	25.3
Return[Manual + Q-learning]	60	22.4
Avoid[Q-learning]	74	(13.5)
Avoid[Manual]	92	(13.8)
Avoid[Manual+ Q-learning]	76	(8.2)
Offense[Q-learning]	42	25.2
Offense[Manual]	52	33.0
Offense[Manual + Q-learning]	51	25.9

ここで、状態遷移回数は、成功した試行の状態遷移回数の平均回数である。() 内の数字は、衝突するまでの平均回数である。

Table 4: 障害発生時の実験結果

サブタスク	成功回数	状態遷移回数
Chase[障害発生前]	86	16.9
Chase[追加学習前]	45	23.3
Chase[追加学習後]	82	20.7

に対する適応能力があり、実機のロボットに対して有効であることが確認できた。

4 議論

Table3 の実験結果より、状態遷移の回数と動作の効率が関係していることが分る。これらの実験は、シミュレータを用いた実験ではなく、実際のロボットを用いた学習実験である。学習の回数は、これらのタスクの達成させるに十分な回数ではなかったので、成功回数は手動の場合より劣っているが、効率は高いことが分る。つまり、これらの結果は、手動による設定と学習と間に Figue8 の関係があることを示している。

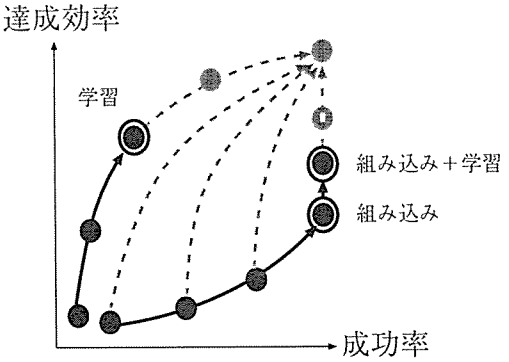


Figure 8: 手動と学習の関係

Chase、Shoot、Return の各タスクの結果から、50 回たらずの追加学習により成功率が上昇している。このことから、期待される学習による性能の向上は Figure8 の点線のようになると予想できる。

実際のロボットは、常に実環境の変化に適応しなければならない。もし、事故が発生した場合は、追加学習の能

力が重要である。左のモータの動作を減少させた実験の結果では、学習方法に手を加えなくても追加学習の効果があることを示している。追加学習後における、ボールが前方にあるときの右前進のQ値は上昇し、前進の代りに用いられている。これにより、追加学習後は障害発生前の成功率に近い状態まで改善している。

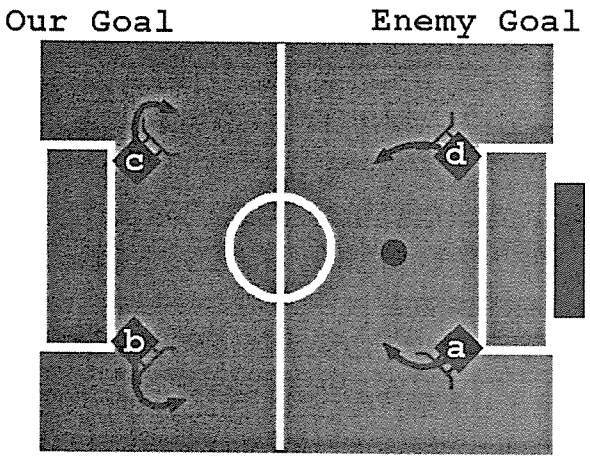


Figure 9: Around タスク

最後に、不完全な知覚における学習の困難さについて議論する。今回の実験では、攻撃行動に必要なサブタスクとしてReturnタスクを用意したが、その代りにAroundタスクを用意し学習する予定であった。Aroundタスクとは、ボールと敵ゴールが正面に見えるように回り込むタスクである。このAroundタスクのために用意した状態は、Returnタスクで用意したものと同じである。そして、Aroundタスクのゴール状態を“ボールと敵ゴールが正面に見える”状態として設定した。このような設定でAroundタスクに対してQ-learningを行ったが、全くゴール状態に近づく行動を獲得できなかった。その理由としては、ロボットの獲得している状態が，環境に対して一意でないということが考えられる。例えば，Figure9に示すようなロボットの位置に対する状態を考えた場合、aとbもしくはcとdについて，ロボットは同じ状態として知覚してしまう。しかし、aとb、cとdにおいてAroundタスクのゴール状態に近づく行動は、反対向きの行動である。このように、ロボットの獲得している状態が、環境を十分表現できない場合には、学習できない。これを改善するためには、知覚できるセンサー空間をより詳細にすることや内部状態を利用することが考えられるが、これらは学習時間の増大を招くことが考えられ、実ロボットで実現するためには検討が必要である。

5 まとめ

この論文では、複雑な行動を実際のロボットで学習することを調査した。このために、複雑なタスクをいくつかのサブタスクに分け、サブタスクごとにQ-tableを構成し、階層的なQ-tableを作成した。そして、各サブタスクごとにQ-learningによる学習を実ロボット上で行い、実験を行った。実ロボットによる実験結果より、この手法により全体の行動を学習により獲得できることと環境の変化に適応できることが明らかとなった。また、実ロボットにおいて、知覚が不完全な場合の学習の困難さも浮上した。最後に、実ロボットを用いて学習実験を行うのは、学習者のかける負担がとても大きい。単に現実をシミュレートするだけでなく、現実世界の変化を知覚してシミュレータ内部の世界を適応させて、実ロボットを用いた実験と同等の実験を行えるようなシミュレータの開発を予定している。

参考文献

[Watkins, 1989] C.J.C.H. Watkins , “**Learning from delayed rewards**,” PhD thesis ,University of Cambridge, 1989.

[Degney, 1996] Degney, B.L. , “**Learning and Shaping in Emergent Hierarchical Control System**,” Space'96 and Robots for Challenging Environments II, June 1996, Albuquerque, New Mexico

[Degney, 1998] Degney, B. L. , “**Learning Hierarchical Control Structures for Multiple Tasks and Changing Environments**,” From animals to animats 5 : SAB 98, 1998.

[Takahashi, 1999] Yasutake Takahashi and Minoru Asada, “**Behavior Acquisition by Multi-Layered Reinforcement Learning**,” Proceeding of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, pp 716 – 721, 1999

[高橋, 1999] 高橋泰岳，浅田稔，“実ロボットによる行動獲得のための状態空間の漸近的構成”，日本ロボット学会誌, Vol.17, No.1, pp 118 - 124, 1999

[榎田, 1999] 榎田修一，大橋健，吉田隆一，江島俊朗，“行動確率場モデルに基づく強化学習 –拡張Q-学習–”，情報処理学会誌論文誌：数理モデルと応用，Vo.40, SIG9(TOM2)，pp.72-80, 12月，1999.

[Kitano, 1998] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara, “**RoboCup:A Challenge Problem for AI and Robotics**,” Lecture Notes in Artificial Intelligence Vol.1395, pp .1–19, 1998.

強化学習における Fuzzy ART を用いた状態空間の階層的分節化

Fuzzy ART-Based Hierarchical Segmentation of the State Space for Reinforcement Learning

八谷 大岳　　渥美 雅保
創価大学工学部情報システム学科

Abstract

This paper deals with an application of a Fuzzy-ART neural network to adaptive segmentation of the sensor space. Our idea is to connect hierarchically among Fuzzy-ART. It makes possible adjustment of segmentation needed individually. We present experimental results to demonstrate the efficacy of our approach through RoboCup simulation task.

1. はじめに

強化学習では，限りある学習者の記憶容量を考慮して状態空間を構成しなければならない．一般的には連続的なセンサー空間を分割して構成する．しかし，適当な分割の程度（分割度）は一様ではなく，センサー空間を等分割して状態空間を構成した場合には，学習において幾つかの問題が生じるのである．すなわち，分割度が細かい領域（細領域）とそうではない領域（粗領域）が混合していることにより，細領域に合わせて等分割した場合は，全体的に細かく分割されるので，粗領域では無意味な分割が行われ学習者の記憶容量を圧迫し，結果的に多大の学習時間を要することになる．また逆に粗領域に合わせて等分割した場合，全体的に粗く分割されるので，細領域では，1状態に複数の適切な行動が存在し，学習ができなくなるのである．

これらの問題を解決するためには，等分割を行わず細領域では細かく粗領域では粗く分割するといった適応的な分割が必要となる．この場合，どのような手法を用いて適応的分割を実現するのか，また細領域か粗領域かなどの分割度を決定する方法等が研究の対象となる．Dubrawski と Reignier [1]は，Fuzzy ART [2]を用いて，期待する強化値が得られない場合のみ新しい状態の追加と現状態の代表ベクトルの更新を行うことによって実現している．村尾，北川，北村 [3]は，Q 学習を拡張し，行動に利用された Q 値と結果として得られた強化信号との差を用いて，現状態を分割する必要があるかどうかを判断し，行動をするに従って実現していく．

本研究では，Fuzzy ART を階層的に連結することによって適応的分割を実現する，また分割度の決定には，報酬を用いている．まず，第 2 章で本研究のエージェントアーキテクチャを紹介し，第3章で RoboCup における適応的分割を必要とするタスクの例を紹介し，第4章で，階層的分節化システムを説明して，第5章では，シミュレーションにより本手法の有効性を確かめる．第6章で結論と今後の課題について述べる．

2. エージェントアーキテクチャ

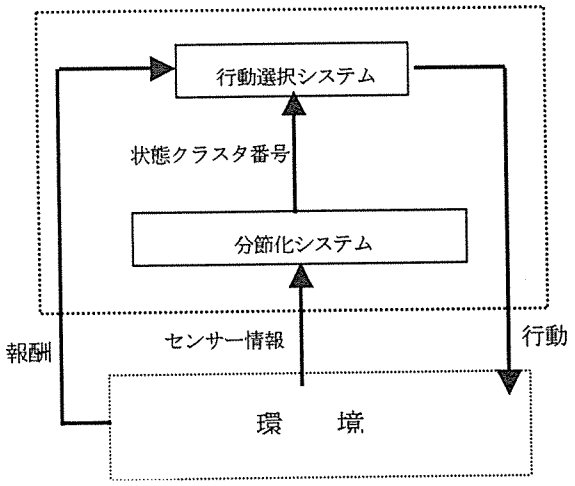


図1 エージェントアーキテクチャ

エージェントアーキテクチャは図1に示す通りである．以下に詳細を説明する．

・<u>センサー情報</u>

各オブジェクトから距離と角度の実数値を得る．

・<u>分節化システム</u>

センサー情報を入力しFuzzy ARTを用いてどのクラスタに属するかを調べる，どれにも属さない場合は，新しいクラスタを追加する．そしてクラスタ番号を出力する．

・<u>行動選択システム</u>

クラスタ番号を入力しQテーブルに基づき行動を選択し，行動を出力する．結果として環境から報酬を受けるとり，次の式(1)，(2)を用いてQ値の更新を行う[4]．

$$Q(x,a) \leftarrow Q(x,a) + \alpha(r + \gamma V(y) - Q(x,a)) \quad (1)$$

$$V(x) = \max_{b \in actions} Q(x,b) \quad (2)$$

r:報酬値，　$\gamma(0 \leqq \gamma < 1)$:割引率，$(0 < \alpha < 1)$:学習定数

3. RoboCup適応的分割タスク

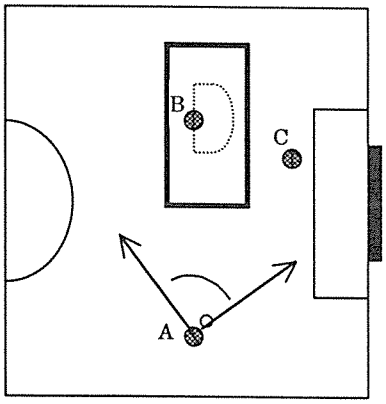


図2 タスクの概観

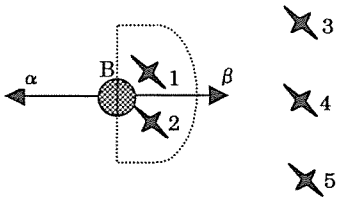


図3 ×はCの配置パターンを意味する

概要

キッカー(A)とフォワード(B)とデェフェンダー(C)から構成され，AがBに適切なパスを出すことを目的とする問題である．学習者はAのみで上記のエージェントアーキテクチャで構成されている．行動はkick(*power*, *dir*)のみで*power*はBとの距離に応じて自動的に計算され，*dir*は0,10,20,30,40,50,60,70,80,90の10個から選択することができる．※*dir*は時計反対回りに正である．

太線はBがランダムに配置される範囲（50×100）を表し，点線はBがAからのパスに対してシュート可能な範囲（半径30の半円）を表す．また，この中にボールが入った時，Aに報酬1が与えられる．Cは図3のようにBに対して相対的に5つのパターンで表1の確率で配置され，パターン1，2は点線内のB近く，パターン3，4，5は点線外である（参照 表2）．Bはパターン1のときβの行動dash(100)をとり，パターン2のときαの行動dash(-100)をとる．パターン3，4，5のときは動かない．

パターン	1	2	3	4	5
確率	1/3	1/3	1/6	1/6	1/6

表1　パターンの確率

パターン	1	2	3	4	5
座標	(10, 10)	(10, -10)	(70, 40)	(70, 0)	(70, -40)

表2　座標はBを原点としたものである

Aのパスが成功（点線内に入る）するまでを1エピソードとし，Aのパスが失敗した場合は，リセット（B,Cの配置を変える）して繰り返す．

解説

パターン1, 2とパターン3, 4, 5では分割度が異なる．パターン3, 4, 5の時Bは動かないので，3, 4, 5を分割する必要はなく，分割度は点線の大きさ程度でよく，比較的粗い分割で十分である．パターン1,2の時Bはそれぞれ異なる行動をとるので，1と2を分割する必要がある，よって分割度は，1と2を区別できる程度と，比較的細かい分割が要求される．また，パターン1, 2は1/2の確率で現れるので，分割するか，しないかでは学習曲線に大きな差がでることは間違いない．

4. 階層的分節化システム

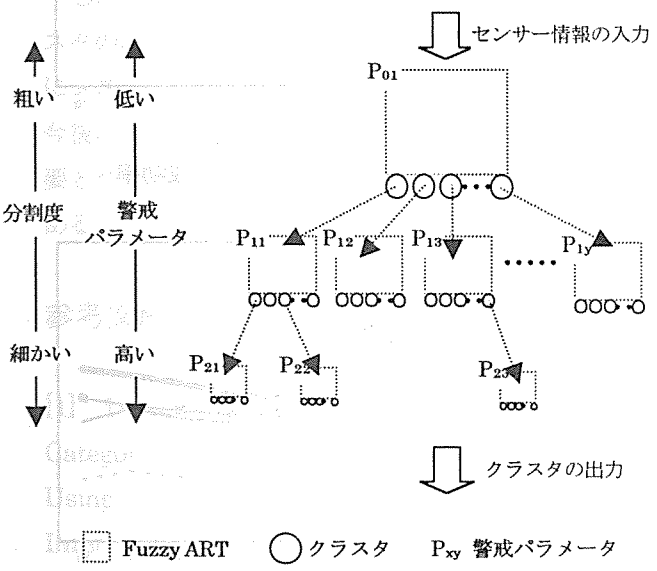


図4　階層分節化システムの概観

概観

階層分節化システムは，図4ようにFuzzy ARTを階層的に連結して構築される．それぞれのFuzzy ARTはセンサー情報の入力を受け最も近似するクラスタを一つ選択する．もし近似の度合いが警戒パラメータによる条件を満たさない場合は，新しいクラスタを追加する．選択されたクラスタが末端の場合は，そのクラスタ番号を出力し，末端ではない場合は，下位のFuzzy ARTへセンサー情報入力する．

Fuzzy ART ごとに警戒パラメータを設定することができるので，各領域の分割度に合わせた分割が可能になる．実際には，Fuzzy ART ごとには設定せず層ごとに設定する．

手順

Step 1: 初期化

各層の警戒パラメータとクラスタを評価する時間を設定する．1つのクラスタをもつFuzzy ARTを生成する．

Step 2: 粗い等分割と行動の学習

Step 1で生成したFuzzy ARTにより粗く分割を行い，各クラスタに対する行動を学習する．

Step 3: クラスタの評価と階層化

全てのクラスタに対し報酬に基づき分割度が十分かどうかの評価を行う．十分ではない場合，つまり細かい分割が必要な場合は，下位の Fuzzy ART を生成し，Q-Table の継承を行う．

Step 4: 適応的分割と学習

Step 3 にて構築された階層化に基づき分割を行い，各クラスタに対する行動を学習しStep 3へ移動．

クラスタの評価

クラスタiで行動をとった時に得られる報酬の平均をr_i，評価すべきクラスタ数をnとすると式(3)を満たす場合，分割度が不十分と判断し，階層化を行う．

$$r_i \le \frac{\sum_{i=0}^{n-1} r_i}{n} \qquad (3)$$

Q-Table の継承

下位クラスタにてQ-Tableを初期（すべてを0.1に設定）から学習するよりは，上位クラスタにて学習したQ-Tableを継承する方が効率的である．しかし，そのまま継承した場合，つまりコピーした場合，サブクラスタにて新しい行動が選択される可能性が少なくなる．式(4)のようにQ値間の差を小さくしたものを継承することにする．クラスタの行動iに対するQ値をq_i，下位クラスタの行動iに対するQ値をq_i'，行動数をnとする．

$$q_i' = \frac{q_i}{\sum_{i=0}^{n-1} q_i} \qquad (4)$$

5. 実験

実験の目的と方法

階層的分節化システムの有効性を確かめるために次の２点に着目し第３章で取り上げたタスクを用いて実験を行う.

(1) 階層化した場合どのような効果があるのか
(2) どの配置パターンが階層化されたのか

(1)に関して調べるために, 次の４つのケースに対してそれぞれ10000エピソードを５回繰り返しその平均から, 学習曲線, パターン1,2とパターン3,4,5の成功率をグラフにして比較する.

1. 警戒パラメータを0.975に設定した等分割
2. 警戒パラメータを0.980に設定した等分割
3. 警戒パラメータを0.985に設定した等分割
4. 階層的分節化システムによる適応的分割
   警戒パラメータは,
   1層：0.975, 2層：0.98, 3層：0.985
   クラスタを評価する時は,
   2000,3000,4000エピソードに設定した.

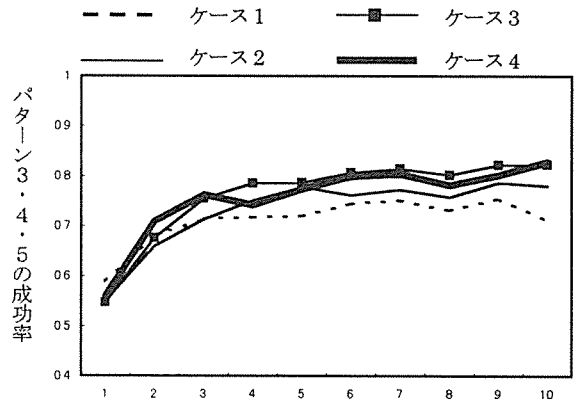
実験の結果と考察

どのケースもパターン 3,4,5 では, 高い成功率を得ている（グラフ１参照）, 一方, パターン1,2では非常に細かい分割が要求されているのでケース１は対応できていない. ケース1,2,3に関してはクラスタ数, 警戒パラメータに比例して高い成功率を得ているのがわかる.(グラフ２を参照)

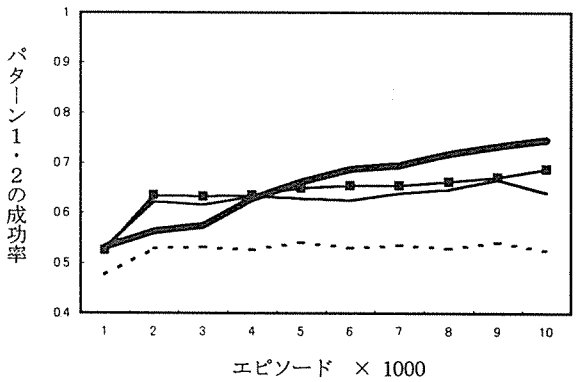
しかし, ケース３よりクラスタ数の少ないケース４が最も成功率を上げている. 分割度に合った分割が可能であるからだと言える. つまり, ケース３は, パターン 3,4,5 に対しても細かく分割しているので, クラスタ数が多くなっている. 肝心なパターン 1,2 ではケース４の方が, クラスタ数が多い.

ケース	1	2	3	4
クラスタ数	50	92	150	131

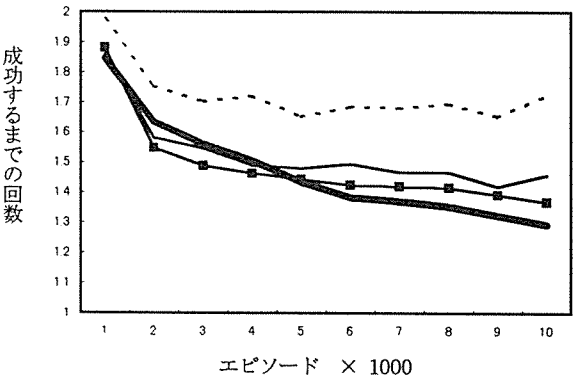
表３



グラフ１ パターン3,4,5の成功率



グラフ２ パターン1,2の成功率



グラフ３ 学習曲線

(2)に関して調べるために，ケース 4 にて階層化が行われたパターンの割合を計算したところ，パターン 1,2 が約 70%，パターン 3,4,5 が約 30%だった．このことからわかるように本システムは，適応的分割が可能である．

6. むすび

本論文では，連続的なセンサー情報を適応的に分割して状態空間を構成する手法として階層的分節化システムを提案した．そして，RoboCup シミュレーションのあるタスクを例にその有効性を調べた．

しかし，本システムは，層の警戒パラメータとクラスタの評価する時間を設定する必要があり，その設定にシステムの良し悪しは大きく依存している．よって，今後の課題として考えられるのは，それらの設定を必要とせずアーキテクチャ内で補えるようにすることである．

参考文献

[1] Dubrawski A., Reignier P.: Learning to Categorize Perceptual Space of a Mobile Robot Using Fuzzy-ART Neural Network, IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'94, Munich, Germany, September 1994.

[2] Gail. A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H, Reynolds, and David B. Rosen: Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps, IEEE TRANSACTIONS ON NEWURAL NETWORKS, VOL. 3, NO. 5, SEPTEMBER 1992

[3] 村尾元，北川郁，北村新三: Q 学習のための状態の適応的分割手法，第24回　知能システムシンポジウム 1997 年 3 月

[4] 荒井幸代，宮崎和光，小林重信: マルチエージェント強化学習の方法論，人工知能学会誌，*Vol. 13, No. 4*, pp.609-617(1998).

マルチエージェント系における組織的学習効果についての一考察

A Consideration of Effect of Organizationally Learning in Multi-Agent System

篠田 孝祐　國藤 進

Kosuke SHINODA, Susumu KUNIFUJI

北陸先端科学技術大学院大学 知識科学研究科

石川県能美郡辰口町旭台 1-1

School of Knowledge Science, Japan Advanced Institude of Science and Technology, Hokuriku

1-1 Asahidai, Tatsunokuchi, Nomi, Ishikawa,923-1211 JAPAN

{kshinoda, kuni }@jaist.ac.jp

Abstract

Recently, there is a variety of kind of research concerning the study of learning for agent. However, many of their studies does not enough to use effectively in the system by which it should reflect the real world like a social support system and GroupWare etc, as much as possible. The reason why the society consists of various organizations and the person belongs to its organization in order to solve the problem cooperating with other people. Therefore, it can be said that the person is able not only to improve own problem solving ability but also to learn the problem solving method of using the organization. That is what should be considered when the agent is used for social support system which especially reflects our society. Moreover, in the logic of organization, it is assumed that the learning process of the group has four processes as follow: 1) the single-loop learning in the individual level, 2) the double-loop learning in the individual level, 3) the single-loop learning in the organization level, 4) the double-loop learning in the organization level. Machine learning and cooperated learning, which researched in the artificial intelligence field are chiefly 1) and 2) of above assumption. But, we think that 3) and 4) are not enough to study, now.

In this paper, We consider the generation of cooperated behavior and the action of cooperated learning of the agent under assumption of designed a social support system with agent technology. Moreover, we will call the learning behavior of the organization "Organizationally Learning". Furthermore, we explain "Organizationally Learning" and describe the concrete effect with "Organizationally Learning".

1　はじめに

エージェント技術を利用した社会支援システムやグループウェアでは，それぞれのエージェントが個人またはグループの代理人として存在し活動する．それらエージェントにとって，利用者の利益を代理人として守りながらシステム全体の利益もたもつ行動すは理想的であるいえる．現在のマルチエージェントの研究では，その理想的なエージェントを設計するのに必要となる基礎技術とし，単体エージェントの行動最適化や意思決定モデル，そしてエージェントの集団における行動最適化などが主に行われてきた．しかしながら，社会支援システムなどでエージェントを利用するには，エージェントの利用者もしくはシミュレートされる人間が現実社会で「組織」に属しているという事が十分に反映されているとは言えない．つまり,「組織」には機能志向システムとしての面と目的志向システムとしての面があるが従来の研究では一方のみを考慮し他方考慮してないためである．

現実社会において人間は何かしらの組織に属している．社会支援システムやグループウェアのような人間が用いるシステムにおいてエージェントを利用するには，利用者の組織やその組織における立場などを踏まえた上で動作する意思決定や協調行動のための適切なフレームワークが必要であるといえる．当然のことのように、コミュニケーションネットワークのデザイン[Balch95]やチーム/グループ行動のフレームワーク[Collinot96]が存在する．また組織 (集団) のおける学習行動を研究対象とした組織学習型分類子システム[Takadama98]も存在する．しかし，そこでの研究の対象となっている集団/組織は単一の組織であるために，RoboCup-Soccer[Soccer98]のチームのような集団では利用できても，RoboCup-Rescue ないし現実的な社会支援システムなどでは，そこにある組織構造は複雑であり，組織は環境の要求にしたがってその形を変化させるなどの理由からそのままでの利用は困難である．そ

こでレスキューシステム[Rescue00a, Rescue00b]や社会支援システムでも利用できる適切なエージェントモデルが確立が必要となる．

そこで，我々は様々な組織に属してるエージェント同士による動的な組織形成モデル及び組織を主体とした協調行動，学習行為のためのモデルの確立を目標としている．ただし，本研究では全ての環境を研究対象とするのは困難であるために，複数の組織が存在する空間において一時的に形成されたた集団における組織形成と組織間の学習行動を研究の対象とする．また，そのようなすでに属する組織をもつエージェントが一時的な組織形成を行こないながら学習する行為を我々は“組織的学習”と呼び、本論文ではそれがエージェントの集団に与える効果について考察する．

以下，2章においては組織とは何であるかを見ると同時にエージェントにおける組織とその学習行為について述べる．そして3章にて，組織的学習とは何であるかを組織における学習行為に基づいて説明し，4章ではその組織的学習を用いることによって期待できる効果をサッカーを例に取り上げならが具体的に説明する．

2 エージェントにおける組織と学習行為

2.1 組織とは

“組織”とは明示的な目標を達成するために合理的に分配され整合化された人間諸力ないし活動であると規定される[佐藤 72]．このように規定することで“組織”を実体概念でなく，あくまでも目的意識的な機能関係を持った機能概念であると捉えることが出来る．こうすることで，それぞれの個体同士は直接的，全体的そして感性的な関係ではなく，目標を媒介として組織形成をする．逆にいえば，一定の行動目標を持った個体同士が集まった集団では“組織”として振舞ったほうがよい結果が期待できるという暗黙的仮定があるといえる．つまり，1) サッカーでのチームはもちろん，2) 偶然事故に遭遇した集団や 3) レスキュー活動で同じ災害現場に居合わせた複数の組織 (消防，警察，自衛隊) に属する個体により構成されている集団でも，組織として行動したほうがよりよい結果が期待できるといえる．また，4) 複数の組織が関係する環境問題においても，同一の目標を持つ組織は協力し一つの“組織”として行動したほうがよいといえる．

上であげた4つの組織の例をそれぞれ抽象的に表現すると，1) は狭い範囲で単一組織に属する個体が形成されている場合，2) は狭い範囲で一時的な組織が複数組織に属する個体によって偶然に形成された場合，3) は広い範囲で一時的な組織が複数組織に属する個体によって形成された場合，4) は広い範囲で長期的な組織が組織間で形成された場合を示している．これら4つの“組織”はそれぞれ Table 1 のような特徴を持っている．

Table 1: 1)-4) の組織における特徴比較

	構成単位	拘束性	所属組織の影響	集団の大きさ
1)	個体	強い	多少あり	全体
2)	個体	弱い	希薄	一部
3)	個体, 集団	やや強い	あり	一部
4)	組織	やや強い	—	全体

Table 1 では，組織を構成する要素と形成された組織の構成要素に対する拘束力としての“拘束性”，つまり構成要素がどれくらい強い意志で同じ目標を達成するつもりであるかということ，そして構成要素が属する組織が新しく形成された組織に対してどの程度影響があるのか，最後に集団が持つ大きさの4つで特徴が記してある．

研究で対象とする組織モデルとは，3) のレスキュー活動での一時的に形成された集団のようなケースや 1) でも FW，MF，DF のように明確な組織構成が存在する集団である．

2.2 組織における学習の関連研究

Table tab:sosiki01 における 1)，2)，4) のような組織における協調行動や協調学習に関しては，いくつかの先行研究[Balch95, Collinot96, Takadama98]がすでに存在する．しかしながら本研究にて対象とする 3) のレスキュー活動のような複数の組織が同時に存在するような環境で状況に適した一時的部分的な集団形成に関する研究や組織を対象とした学習に関しての研究はほとんどない．その理由とは，これまでのエージェントの協調行動及び学習行為は主体として個体のみを対象としためである．特に 1) のような組織においては，個体は自身以外の存在は受動的な姿勢で全て外部環境として捕らえ，内部状態の変化により意思決定モデルを形成していたためである．また，2) のような集団による協調学習では，他の個体を完全な外部環境ではなく能動的に変化させることの可能な内部環境の一部であるとみなすことで，情報の交換，知識の相互補助などを行うが，組織という点ではその構成要素はあくまでも個体である．また，これまでの集団における学習行為は，組織を構成する固体を没個性的に扱うことで目的意識をあらかじめ固定することで組織の持つ機能志向的側面のみが扱われていたためである．さらに，組織がもつ知識とは何であるかが明確にされていないのもその原因と思われる．

2.3 集団における学習行為

人間によって形成される組織についての議論は，社会科学における組織論の中で行われている．学習行為や協調

システムとしての組織については，その組織論の組織学習[Argyis78]や組織間関係[山倉 93]において様々な研究がなされている．組織学習とは個人では目標の達成が困難な問題を組織全体としての問題解決能力を向上させながら解決の糸口を創出するための組織的活動である．この組織学習の中では組織には次の４種類の学習が存在すると示唆している[Argyis78]. ただし，これらはあくまでも仮定であり現象について述べているが，学習過程やそのメカニズムに関しての議論がなされているわけではないので必ずしも厳密な規定はされていない．

- **個体のシングルループ学習**

  個体の持つ規範の中で，個体の問題解決能力を向上させる

- **個体のダブルループ学習**

  個体の持つ規範を変えながら，個体の問題解決能力を向上させる．

- **組織のシングルループ学習**

  組織のもつ規範の中で，組織の問題解決能力を向上させる

- **組織のダブルループ学習**組織のもつ規範を変えながら，組織としてのパフォーマンスを向上させる．

ここでいうところの規範とは，学習の主体である個体もしくは組織がもつ役割 (能力として持てる範囲) だと考えればよい．サッカーを例とするならば，規範が固定ということは，プレイヤはボールを蹴ることしか出来ないとすることであり，シングルループによる問題解決能力の向上とは正確にボールを蹴る能力，蹴ることに関する判断能力を向上させる学習のみであるためにシングルループ学習となる．また，ダブルループ学習のときには規範を帰ることが出来るために規範そのものの学習行動が必要となるためにダブルループ学習となる． 本研究では，これら４つの学習行為が同一の集団に複合的に現れる場合を対象とする．

2.4 エージェント組織の知識とは

エージェント個体がもつ知識と，組織の持つ知識との違いはどこにあるのであろうか．組織指向型分類子システム[Takadama98]では，個体知識を個体が独立にもつ知識，組織知識を，個体が共有してもつ知識であり個体知識の和によって実現されるものとしている．だが，我々の対象とする環境では，組織は複数存在し，かつエージェントは複数の組織に属することが可能で，時として集団から一時的な組織が形成されるという状況であるために，組織知識を個体知識の和とすると組織が必要だといえない知識まで蓄えることになる．そこで，本論文で個体知識及び組織知識を以下のように規定する．

- **組織知識：**

  組織に属する個体が利用できるルールの集合．

- **個体知識：**

  組織知識にあるルールに必要なパラメータの集合

こうすることで，エージェントは自由に自身の属する組織を変えることが可能となる．ただし，ここで問題となるのは組織知識はどこに蓄えるのかということになるが，現時点では暫定的にその組織のリーダとしかいいようがない．

3 組織的学習

3.1 組織的学習とは

本研究でいうところの “組織的学習” は，組織を構成する要素は個体と組織，２つの要素から成り立つ．また対象となる集団とは，すでに組織に属している個体の一時的に形成された集団である．前節の組織の４つの学習に基づいて分けると，個体は一時的に形成された集団の中で自身の役割を求めて “個体のダブルループ学習” を行う．一方，組織は個体が属している機能志向型組織での学習行為と，状況にあわせた目的指向型組織の組織形成に関する学習ループを持っている．つまり，“組織的学習” では，対象とする組織に属する個体は，個体のダブルループ学習を行うと同時に，組織形成と組織知識蓄積のためのダブルループ学習を行なっている．そのために，エージェントは機能志向型組織における組織学習と目的指向型組織における組織形成に関して獲得する必要がある..

“組織的学習” とは以下に述べるような行為において行われる学習行動のことである．

1. 一時的に集まった集団から組織を形成する．
2. 組織のもつ機能的システムとしての利点を利用し問題解決方法を探る
3. 個体は分担されたタスクを的確にこなす方法を探し，実行する
4. 個体の属する組織で利用できる知識を，組織の知識として蓄える．

つまり，エージェント単体が行うべき学習プロセスは以下のとおりである．

- 問題解決に必要な行動ルール及び知識の獲得 (個人のシングルループ学習)
- 協調行動獲得のための学習 (個人のダブルループ学習)
- 集団が形成されてたときの状況に適切な組織形成モデルの獲得 (組織のシングルループ学習)

- 個体が獲得した経験や知識を組織で利用できる知識に変換する．(組織のダブルループ学習)

前節でも述べたように，これまでのエージェントの学習に関する研究ではでは主に個人を主体とした学習行動及び意思決定モデルが主体であった．しかしながら，我々がエージェントの利用を想定しているのは社会支援システムなど複数の組織が混在する世界である．そのため本研究で研究の対象とするのは，後半の２つの組織のシングルループ学習と組織のダブルループ学習の部分である．以下は，これら４つの学習の違いや特徴などを述べ，サッカーを例題にあげて示してある４つの図などを参考に具体的にどのような状況を想定して，本研究の中心的な課題となる組織的学習を研究していくべきであるのかを明らかにする．

3.2 組織的学習における個体のシングルループ学習

Fig. 1は，個体のシングルループ学習を示したものである．図では個体であるサッカープレイヤエージェントが，環境の状態を示す情報として他のプレイヤエージェントやサッカーボールの位置情報などをもとに，ボールを蹴るや追いかけるというような自身の振る舞いについて学習する．つまり，個体は環境から情報を獲得し，それを利用して自身の問題解決能力を向上させる．このような学習に関しては，機械学習などの分野で研究されている．

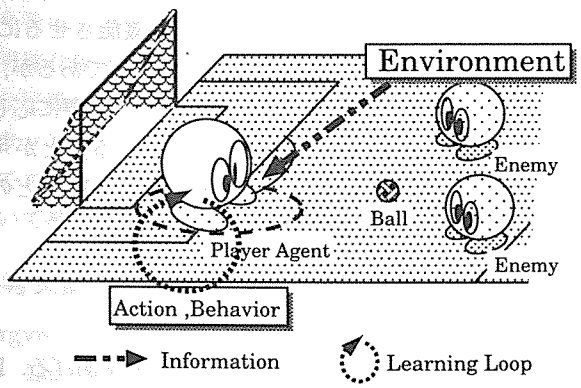


Figure 1: 個体のシングルループ学習

3.3 組織的学習における個体のダブルループ学習

Fig. 2は，個体のダブルループ学習を示したものである．図では個体であるプレイヤが味方との協調行動に必要な自身の役割を模索し，その際のパスや位置取りなどを学習する．つまり，協調グループ内での役割を学習するループとその協調行動に必要な問題解決能力を学習するループの２つ学習ループが存在する．このような学習に関しても協調行動獲得のような分野ですでに研究されている．

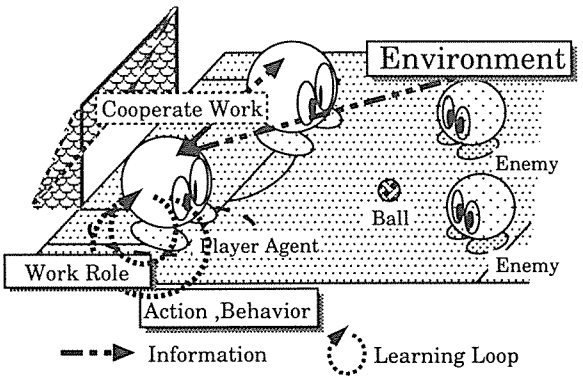


Figure 2: 個体のダブルループ学習

3.4 組織的学習における組織のシングルループ学習

Fig. 3は，組織のシングルループ学習を示したものである．図では個体であるプレイヤは，何かしらの役割を持ったグループ (たとえばディフェンスをするとか) のなかで，どれくらいのプレイヤ数が必要なのかやどのような組織を形成する必要があるのかなどについて学習する．つまり，ある目的を達成するために必要な組織学習するループを組織が主体となり持つ．このような研究は，固定した集団の中で，個体が学習の主体となってパスや位置取りを行う学習，言い換えれば個体のダブルループ学習は行われているが．組織構造そのものを学習の対象としたものは少ない．ただし，コミュニケーションモデルに関する研究の中で，コミュニケーションネットワークの変遷などによる社会形成の変遷を調べる研究では，構造的な社会形成がされることがあるという結果が出ている．そのため，π-calculus のような動的なコミュニケションモデルを記述できる言語を利用することで記述可能であると考えている．

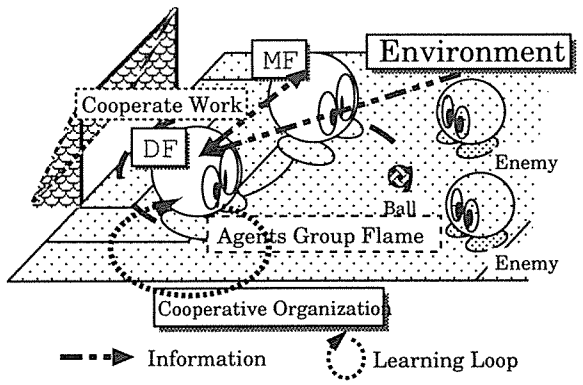


Figure 3: 組織のシングルループ学習

3.5 組織的学習における組織のダブルループ学習

Fig. 4では，組織のダブルループ学習を示したものである．図では個体であるプレイヤは，常に属している集団 (組織) と強い目的志向から形成される組織の２つの組織に属している．その中で，強い目的志向から属している組

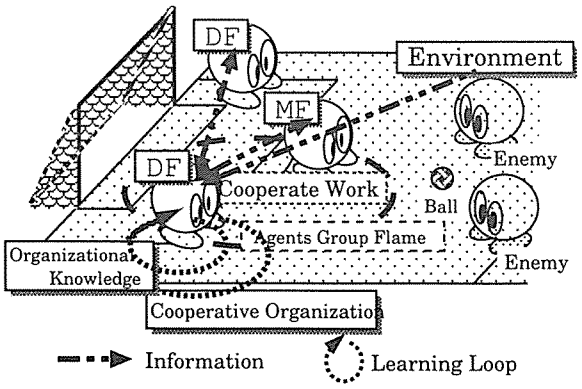


Figure 4: 組織のダブルルループ学習

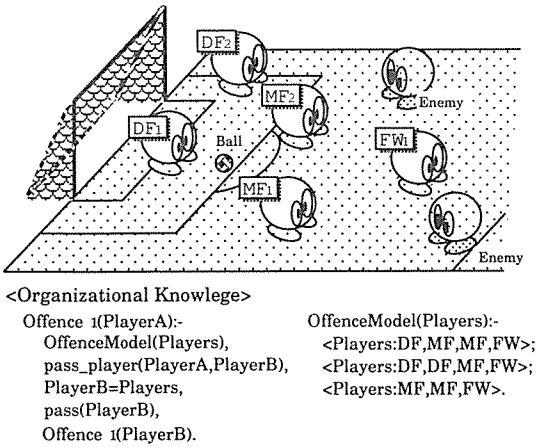


Figure 5: オフェンスの場合の例

織は目的達成に必要な組織構造や組織としての役割を学習し，一方でそこで学習した知識などをもとに一方の組織で知識と蓄えるための学習を行う．つまり，目的志向で形成された組織と機能志向で形成された組織の両方で学習ループを持つことになる．この学習行為も学習機構の構成要素である組織を個体と同等とみなせば難しくはないが，最終的な学習を行うのはあくまでも個体であるために難しい．この組織のダブルループ学習は上記の組織のシングルループ学習ともに組織が学習機構の構成要素であるが，実際に学習を行うのは個体であるために，エージェントの組織が蓄える知識の定義など重要な課題が多々ある．

3.6 組織的学習の特徴

ここまで，組織的学習を組織に見られる学習に分けてみてきた．その中で明らかであったことは，これまでの学習において学習を行なう主体とは，サッカーで言うならばプレイヤのような最終的な意思決定を行なう最小単位であった．また，集団を対象とするような学習についても，集団を機能的システムとみなすことで没個性的な個体の集団としていたために，そこでの意思決定を行なう最小単位は組織であるといえる．つまり，学習のメカニズムは個体であっても組織であっても大きな違いはない．それに対して，我々の提案する組織的学習では学習の主体を個体と組織という形で構造的にすることで，動的に環境が変化するシステムに対して適用できるのではないかと期待する．

以下に，この組織的学習の特徴について記す．

- 学習行為の主体は，個体と組織である
- 学習は，個体の能力向上と状況に応じた組織形成の２点が主なものである
- 機能的志向システムとしての組織，目的志向システムとしての組織，両方の組織を学習モデルに含むことが可能である

このような特徴を踏まえて，次節ではこの組織学習を取り入れることでどのような効果が期待出るのかを，具体例と共に説明したいと思う．

4 組織的学習により期待できる効果

4.1 例１：オフェンスの時

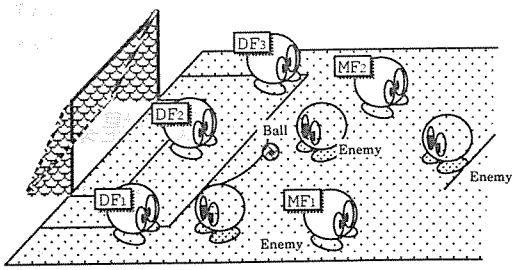
組織学習により期待できる効果として，１つ目の例はサッカーの攻撃時における行動をあげる．Fig. 5 では，簡単な攻撃の例としてパスをまわすルールが記述してある．ただ，ここで記述してある例ではチームとしての組織知識の例である．

Fig. 5 のルールでは，パスをまわすグループ分けを 3 通り用意している．このルールを実際に機能させるには，グループへの参加を問うプロセスなどが必要であるが，パスを出す側の判断でグループを決定したり，必要に応じて攻撃に参加するメンバーの数や構成を変えることが可能である．ただし，チームの中の一部である DF などの組織に適した行動ルールを用意する必要がある．

4.2 例２：ディフェンスの時

続いて，サッカーの守備時における行動の例をあげる．Fig. 6 では，簡単な攻撃を例としてあげてある．ここでも，図の中に書いてるルールは守備の時のグループ構成とそれに対しての簡単な振る舞いを記述したルールである．また，これはチームに対してのルールとなっているため，ポジションごとのルールはまた別に記述する必要がある．

Fig. 6 のルールでは，攻撃側の状況に合わせて守備に必要なグループを決定する．攻撃の時ほどグループを自由に変更できないが，状況によって守備の陣容は変更できるので，守備に参加していない味方は攻撃に必要な準備をすすめることが出来る．また，これも攻撃と同様にポジションごとのルールを別に記述することで，守備のバリエーションを増やすことが可能となる．



<Organizational Knowlege>

```
Defence(PlayerA,OffenceNum):-
  DefenceNum is OffenceNum +1,
  DefenceModel(DefenceNum,Players),
  (NerestBall(Player),Defence(Player));
  (MarkPlayer(Player,Target),
  Mark(Target)).

DefenceModel(DefenceNum,Players):-
  ( DefenceNum = 2,
  <Players:DF,DF>);
  (DefenceNum = 3,
  <Players:DF,MF,MF>;
  <Players:DF,DF,MF>;
  <Players:DF,DF,DF>).
```

Figure 6: ディフェンスの場合の例

5 まとめ

本研究では，社会支援システムのような現実社会を反映したシステムでのマルチエージェント技術の利用を前提にしている．そのエージェントの集団を，人間の組織論における組織の定義から，単なる集団ではなく適切な命令系統を与えた組織へとすることで問題解決能力の向上が期待できると考えた．本論文ではその仮定をもとに，組織の学習行為について説明し，複数の組織に属するエージェントの集団における学習行為に関して“組織的学習”と呼び，それに関しての考察を行なった．また，組織的学習よって期待できる効果についてサッカーゲームを用いて説明した．

今後は，エージェントにおける組織知識について言及し明確な定義を与えると共に，組織的学習に関する理論的な定義をしていくつもりでいる．また，それと同時にロボカップサッカーもしくはロボカップレスキューを題際としてエージェントの実装を行い，組織的学習の有効性について検証していく．

参考文献

[Argyis78] C. Argyris and D. A. Schon: “Organization Larning” Addison-Wesley, 1978.

[Balch95] T.R.Balch and R.C.Arkin: “Communication in reactive multiagent robotics system”, *Autonomous Robots*, Vol.1, No.1, pp.27-52, 1995.

[Collinot96] A. Collinot, A. Drogoul, and P.Benhamou: “Agent Oriented Design of Soccer Robot Team”, *The Second International Conference on Multi-Agent System* pp. 41-47, 1996.

[Takadama98] K. Takadama, S. Nakasuku, T. Terano: “Printed Circuit Board Design via Organizational-Larning Agents” *Applied Intelligence* Vol.9, No.1, pp.25-37, 1998.

[佐藤 72] 佐藤 慶平 現代組織の論理と行動 御茶の水書房, 1972.

[山倉 93] 山倉 健嗣 組織間関係 有斐閣, 1993.

[加護野 88] 加護野 忠男: 組織認識論. 千倉書房, 1986.

[Soccer98] I. Noda,etc. Soccerserver manual ver.4 rev00. *http://www.robocup.org/ Technical report*, November 1998.

[Rescue00a] 高橋 友一, 松野 文俊, 田所 諭 RoboCup-Rescue 技術委員会: RoboCup-Rescue シミュレータの構成 *Bit* **共立出版** Vol.32, No.3 2000.

[Rescue00b] RoboCup-Rescue 技術委員会, 田所 諭: ロボカップレスキュー 大規模災害救助への挑戦 **共立出版** 2000.

優先度を考慮した多目的最適化手法としての適応的評価関数の提案

Adaptive Fitness Function for Multiobjective Optimization Considering Priorities

柳瀬 正和, 内部 英治, 浅田 稔

Masakazu Yanase, Eiji Uchibe, Minoru Asada

大阪大学大学院 工学研究科 知能・機能創成工学専攻

Adaptive Machine Systems, Graduate School of Engineering, Osaka University

yanase@er.ams.eng.osaka-u.ac.jp

Abstract

We have to prepare the evaluation (fitness) function to evaluate the performance of the robot when we apply the machine learning techniques to the robot application. In many cases, the fitness function is composed of several aspects. Simple implementation to cope with the multiple fitness function is a weighted summation. This paper presents an adaptive fitness function for the evolutionary computation to obtain the purposive behaviors through changing the weights for the fitness function. As an example task, a shooting behavior in a simplified soccer game is selected to show the validity of the proposed method. Simulation results and real experiments are shown, and a discussion is given.

1 はじめに

自律移動ロボットにとって，環境の変動に対する頑健性が要求される．しかしながら全ての状況における行動を記述するのは困難である．近年，経験に基づいた行動を獲得する進化的手法として遺伝的プログラミング (GP)[3] が注目をあびている．GP は個体集団により形成された世代において，遺伝的操作により評価の高い個体の子孫を優先的に次の世代に残すことにより進化する手法である．

個体の評価関数に関しては，多目的最適化を考える場合が多い．多目的最適化手法について多くの研究がなされてきた．例えば，重みパラメータ法として各目的関数 f_i の重みつき線形和

$$f = \sum_{i=1}^{n} w_i f_i \qquad (1)$$

を用いる方法が考えられる．しかし，事前に適切な重み係数を設定することが非常に困難であり，不適切な重み係数が設定されると局所解に陥りやすくなる．また，タスクが複雑な場合には，最初から複雑なタスクを獲得させようとしても，非常に時間がかかり実用的ではない．浅田らは Learning from Easy Mission と呼ばれる簡単な状況から学習をはじめるというパラダイムを提唱している [2]．従来，初期配置 [2] や初期速度 [4, 7] などを適切に設定することにより，学習が高速化されることが示されてきた．ただしこれらの研究では評価関数が一定であり，学習の初期に適切な評価関数を設定しなければ，ロボットにとってタスクを達成するのが非常に困難となる [5, 6]．そこで，学習に応じて段階的に評価関数を設定することができれば，複雑なタスクが比較的容易に獲得できると考えられる．

本研究では目的関数に優先順位をつけ，進化の過程や評価値の勾配を考慮して，適応的に目的関数間の重み係数を変更することを提案する．行動獲得の手法として GP を採用する．提案手法を用いた GP を簡単なサッカーゲームにおけるシュート行動の獲得タスクに適用する．さらに提案した手法についてシミュレーション及び実機で考察する．

2 適応的な評価関数の提案

進化の程度や評価値の勾配を考慮して，適応的に目的関数間の重み係数を変更することにより評価基準を変更し，その世代に合った評価を行うことが可能になる．最小化問題の評価関数の重みの変更の指針を示す (**Fig. 1**)．まず全ての目的関数 f_j について優先順位を決め，優先順位の高いものから順番に並べる．つまり，f_1 が最も優先度が高く，以下 f_2，f_3 と順に優先度が低くなるものとする．次に以下の操作を世代交代のたびに最終世代まで繰り返す．

1. 最小二乗法により目的関数の平均変化率 $\Delta \bar{f}_j (1 \leq j \leq n)$ を求め，f_i と f_j の相関係数

$r_{f_i f_j}(1 \leqq i \leqq n)(1 \leqq j \leqq n)$ を求める．

ベクトル $\boldsymbol{f}$ を

$$\boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \tag{2}$$

とすると相関係数は，

$$R_{ff} = \frac{1}{n} \sum \boldsymbol{f}\boldsymbol{f}^T \tag{3}$$

により求まる．ただしベクトル $\boldsymbol{f}$ は分散が 1 になるように標準化したベクトルである．行列にすると

$$R_{ff} = \begin{pmatrix} 1 & r_{f_1 f_2} & \cdots & r_{f_1 f_n} \\ r_{f_1 f_2} & 1 & \cdots & r_{f_2 f_n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{f_1 f_n} & r_{f_2 f_n} & \cdots & 1 \end{pmatrix} \tag{4}$$

のように対角成分は全て 1 となり，各値は -1 から $+1$ の範囲の値をとる．

2. まず，優先度の最も高い目的関数 f_1 から順番に着目する．優先度の高い目的関数 $f_j(1 \leqq j \leqq n)$ の平均が増加している場合は以下の操作を繰り返す．減少している場合は 3 に進む．
   (a) 残りの目的関数 $f_i(j < i \leqq n)$ の中で優先度の高い順番に，f_j と相関があるものを 1 つ選択する．f_i と f_j の関係が正の相関のときはその重みを増加し，負の相関のときはその重み係数を減少する．
   (b) 全ての目的関数 f_i と f_j が無相関の場合には，f_j の重み w_j を増加する．
3. 残りの目的関数 f_j について，優先度の高い順番に 2 の操作を繰り返す．
4. 全ての目的関数 $f_j(1 \leqq j \leqq n)$ について以上の操作が終ったら，次の世代に進む．

これにより，f_j と正の相関のある目的関数の重みを増加することで間接的に f_j を強化し，f_j と負の相関のある目的関数の重みを減少することで間接的に f_j の劣化を抑制することが可能である．f_j が全ての目的関数と相関がないときは f_j の重みを増加することにより直接 f_j を強化することができる．また，重みが大きく変化すると評価基準が大きく違うものとなり前の世代までの学習の意味がなくなるで，少しずつ変化させることにする．

3 タスクの設定

1 対 1 の簡単なサッカーゲームを想定する．タスクは守備側ロボットとの衝突を回避しながらシュート行動を獲得することである．学習者は攻撃側 1 台とし，守備側の行動は簡単に記述する．フィールドサイズはロボカップ [1] の

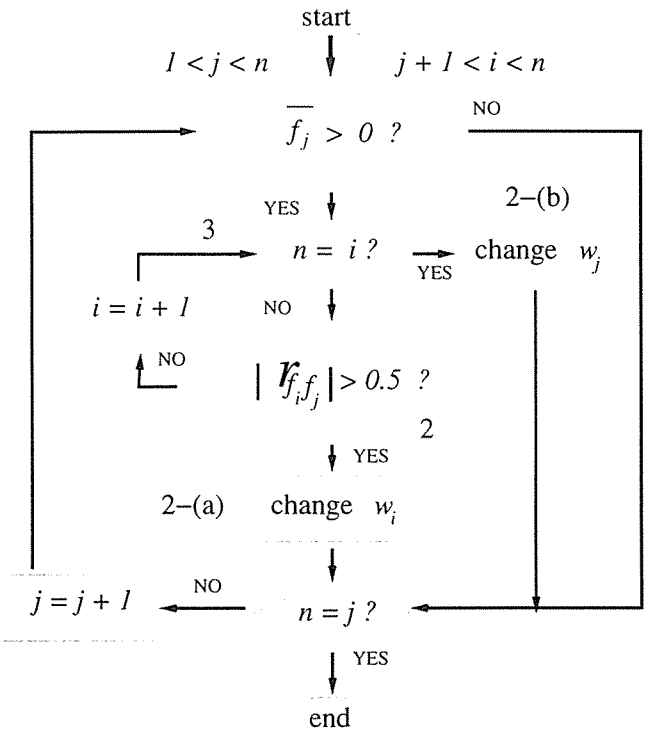


Figure 1: Flowchart of the adaptive fitness function

中型リーグ (**Fig. 2**) と同じである．環境内には攻撃側ロボット，守備側ロボット，ボール，2 つのゴールのみが存在し，回りは壁で囲まれているためにロボットやボールがフィールドの外に出ることはない．

学習ロボット (**Fig. 3**) は，観測のためのカメラと移動機構しか持っておらず，相手の内部情報を知るための通信機能や，ボールを蹴る機能は持っていない．観測のためのカメラはロボットの前方中央に配置されており，そのカメラから得られる限られた画像情報のみで対象物の特徴を獲得する．カメラより取り込んだ画像を日立製の画像処理ボード IP-5000 に入力し，制御に必要な観測値を得る．それぞれのロボットは，ボール，自陣ゴールと敵陣ゴール，他のロボットを識別可能であるとする．環境を取り囲む壁は，直接カメラで識別できないものとする．ロボットは環境中の物体を色によって識別する．本実験ではボールを赤，ゴールを青と黄，相手ロボットを水色とする．観測画像に対して色抽出を行い，得られた色領域の重心座標を求める．また，各ロボットの移動機構は左右の駆動輪が独立に駆動する左右独立駆動機構である．

個体数 150，1 個体 30 試行とし，各試行はボールがゴールに入るか制限時間になると終了する．各個体は 2 つの独立な木構造を持ち，それぞれ左右輪の回転速度を出力する．入力はセンサ情報とする．センサ情報には環境内に存在するボール，敵ロボット，2 つのゴールの各重心座標について，現在の状態について 8 種類と 1 ステップ前の状態について 8 種類あり，全 16 種類である．関数セッ

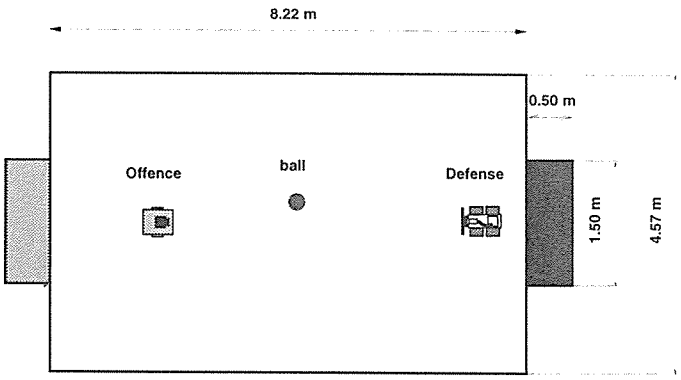


Figure 2: Environment

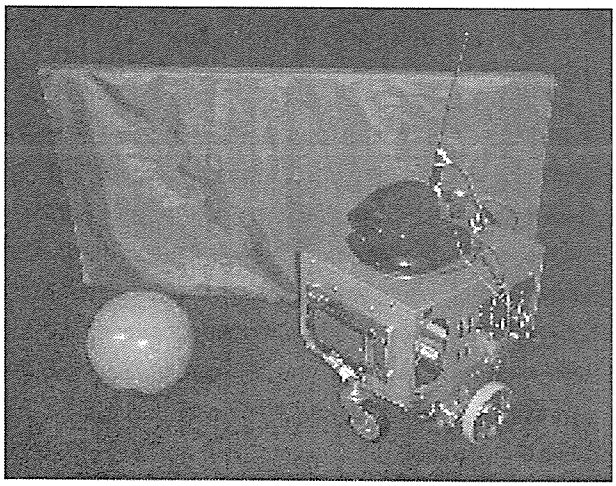


Figure 3: Agent

トとしては，四則演算などを用いた．適応的評価関数に関しては優先度の高い順に目的関数を挙げると，得点数，失点数，画像上の配置に関するオーバーラップ数，ボールを蹴った回数，守備側との衝突回数，ステップ数 (**Table 1**, case A) とした．重みの変化率は 2%とする．

$$
\begin{aligned}
f \;=\; & w_{my} \times f_{my} \\
& + w_{op} \times f_{op} \\
& + w_{ov} \times f_{ov} \\
& + w_{kick} \times f_{kick} \\
& + w_{co} \times f_{co} \\
& + w_{step} \times f_{step}
\end{aligned}
\tag{5}
$$

ただし，f_{my} は得点数，f_{op} は失点数，f_{ov} はオーバーラップ数，f_{kick} はキック数，f_{co} は衝突回数，f_{step} はステップ数に関する正規化された目的関数である．

ここで画像上の配置に関するオーバーラップ数というのは **Fig. 4** に示すようにゴールの重心が画像の中心線よりも右に見えるとき，ゴールの左端と敵の左端の間にボールの重心が見えるということである．同様にゴールの重心が画像の中心線よりも左に見えるとき，ゴールの右端と敵の右端の間にボールの重心が見えるということである．この条件を満たしている状態をオーバーラップと呼ぶことにする．

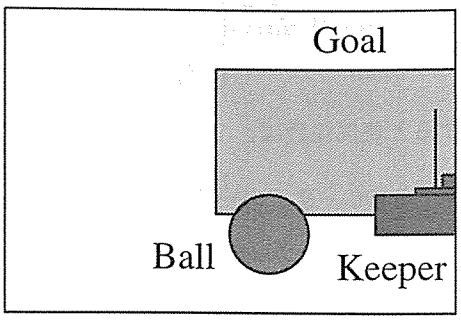


Figure 4: One example of overlapped states

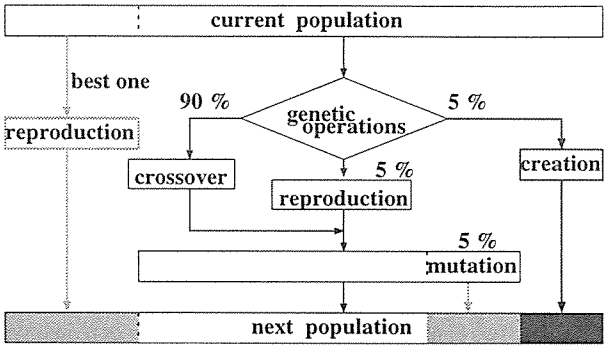


Figure 5: Flowchart of GP

Fig. 5 は，GP における世代交代である．遺伝的操作には，選択 (selection)，交叉 (crossover)，突然変異 (mutation) がある．本研究では，最良の個体はそのまま次の世代に残す．交叉のために必要な個体の選択はトーナメント方式により行う．各個体の最大深さは 25 とする．**Fig. 5** の確率に従って 200 世代まで進化を繰り返す．

Table 1: Fitness measures and the priorities

	case A	case B	case C	case D
f_{my}	1	1	1	1
f_{op}	2	2	6	2
f_{ov}	3	5	3	3
f_{kick}	4	3	4	6
f_{co}	5	4	5	5
f_{step}	6	6	2	4

4 実験結果

初期重みのパターンを複数設定し実験を行った．その一例を **Table 2** に示す．得点数に関する重み w_{my} の値が常に

Table 2: Initial weight

	case 1	case 2	case 3	case 4	case 5
w_{my}	9.0	9.0	9.0	9.0	4.0
w_{op}	9.0	9.0	8.0	1.0	5.0
w_{ov}	2.0	4.0	4.0	1.0	9.0
w_{kick}	8.0	2.0	7.0	1.0	6.0
w_{co}	4.0	2.0	6.0	1.0	7.0
w_{step}	2.0	4.0	5.0	1.0	8.0

大きいという制約を全ての場合に適用した．ただし case 5 のみは w_{my} の値が最も小さくなるように設定し，その他の場合と比較した．それぞれのパターンについて，提案手法と重み固定の場合について複数回シミュレーションを行った．

4.1 提案手法と重み固定法との比較

重みを固定した評価関数を用いた場合と適応的な評価関数を用いた場合について世代ごとの得点数の平均を比較した．横軸に世代数をとり，縦軸に得点数の平均をとっている．

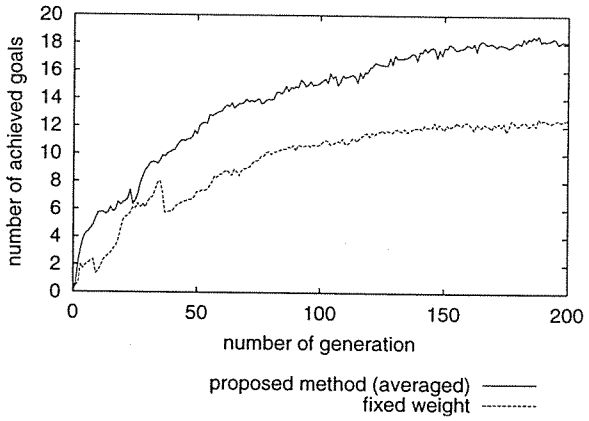


Figure 6: Average of the number of achieved goals in the first experiment with the stationary opponent

Fig. 6 は守備側エージェントが静止しているときの結果である．重みを固定したものは 80 世代以降進化がほとんど見られないが，提案手法を適用したものは確実に進化している．

Fig. 7 は守備側エージェントが移動するときの結果である．守備側エージェントが移動するために，より複雑に変化する環境における行動獲得についての比較である．守備側エージェントの行動指針は，ボールを追従し，見失うと停止するものとした．この実験における初期個体群として，守備側エージェントが静止している場合において，提案手法を適用して獲得された最良の個体群を用い

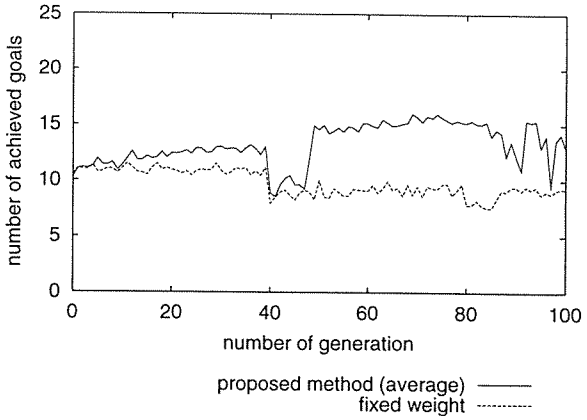


Figure 7: Average of the number of achieved goals in the first experiment with the moving opponent

た．40 世代と 80 世代で守備側エージェントの最大速度を段階的に速くした．

重みを固定した場合は，守備側の最大速度が速くなると得点数が減少している．しかし重みを変化させた場合は，最大速度があがっても得点数の減少がほとんどなく，適応的な評価関数を用いた方が良い成績を示している．

Fig. 8 に，守備側エージェントが移動する場合における実験について，提案手法を用いた場合のシミュレーションで獲得した行動の一例を示す．上のエージェントが守備側であり，下の攻撃側エージェントが上にある敵陣ゴールに向かってシュートをしている．左上の初期配置から始まっている．

実際に獲得した行動を比較する．重み固定の場合は，守備側エージェントがボールに先に触れたり，攻撃側エージェントがキックしたボールが守備側エージェントに触れるなどして，制限時間内にシュートする回数が比較的少ない．また，重み固定の場合と提案手法の場合で，得点数や失点数が同じような値であっても，重み固定の場合は得点する際にも守備側ロボットとの衝突が多かったり，シュートするまで時間がかかるなど，重みを変化させた場合と比較するとスムーズな行動の獲得が出来なかった．重みを変化させた場合は **Fig. 8** のような衝突回避行動を獲得できた．

重みを固定した場合では，以下のような問題点が挙げられる．初期世代ではキック回数が多く，失点が少なく，得点が多い方が評価される．しかし，世代が進むにつれて，シュート行動が獲得されても衝突回避やシュートまでのステップ数に関する重みが少ないため，スムーズなシュート行動の獲得は非常に困難となる．他方，提案手法の場合には，世代が進むにつれて衝突回避やステップ数に関する重みが変化するためにスムーズなシュート行動を獲得することが可能となる．

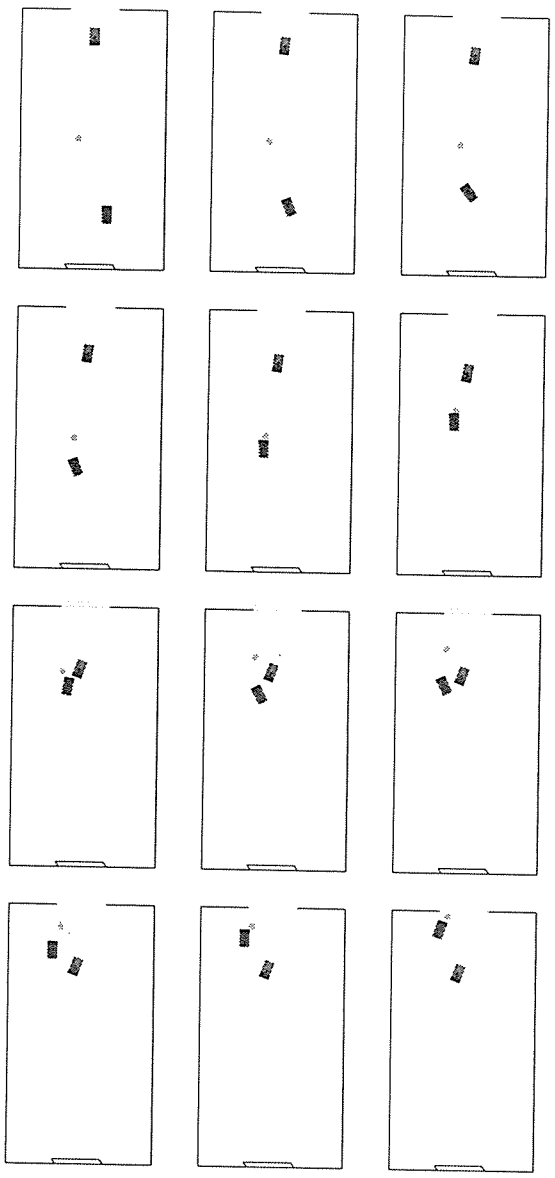
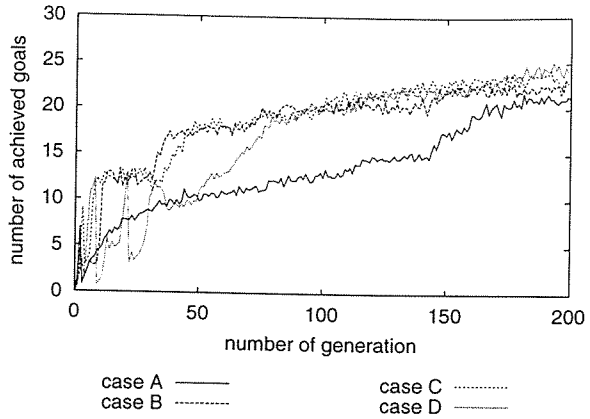
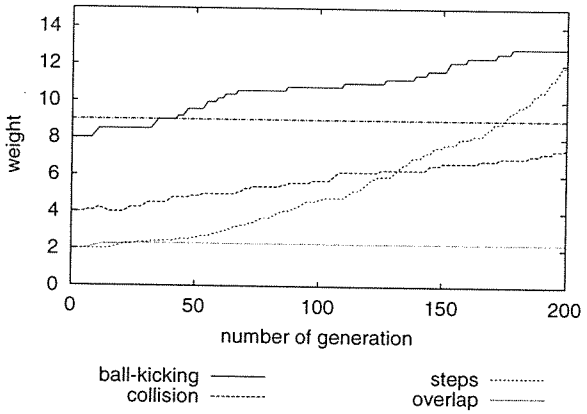


Figure 8: Typical shooting and avoiding behavior in computer simulation



(a) achieved goals



(b) weight based on case C

Figure 9: Comparison among four priority functions

4.2 初期重みの依存性

重みの初期値が **Table 2** の case 1 から case 4 までは性格的には少々異なるものの，最終世代における結果はほぼ同様で **Fig. 6**，**Fig. 7** のような結果となった．しかし，**Table 2** の case 5 の場合は適切なシュート行動を獲得するには至らず，行動としては，衝突を避けてボールに向かいキックを行うものとなった．

以上により，初期重みに関する厳しい制約は無いと言える．今回の場合は得点数に関する重みを最も大きく設定すると，相対的な重みが少々異なった場合でも重みが適応的に変化することが有効に働き，結果として適切な行動を獲得できた．

4.3 優先順位が違う場合の比較

次に優先順位の与える影響について検討する．**Table 1** に挙げる 4 つの場合について比較する．ここで，初期重みは **Table 2** の case1 を用いた．**Fig. 9**(a) にあるように，進化の途中におけるグラフの様子は，4 つの場合それぞれが違うカーブを描いているが，200 世代では全てがほぼ同じ値を示している．これらは全て **Fig. 6** に示す，重み固定の場合の 200 世代における得点数の平均と比較すると，優れた値であることが分かる．これらにより，優先順位に対する依存性は大きくないと考えられる．

Fig. 9(b) に，**Table 1, case C** の場合の重みの変化の様子を示す．初期世代における重みの大きさは，得点数 = 失点数 > キック数 > 衝突回数 > ステップ数 = オーバラップ数の順番であったが，200 世代では，キック数 > ステップ数 > 得点数 = 失点数 > 衝突回数 > オーバラップ数の順番となっている．重みの大きさの観点からすると，オーバラップ数に関する目的関数の重要度は小さいと考

えられる．

4.4 実機による実験

シミュレーション結果を実機に適用して，獲得された行動について検討を行う．**Fig. 10** に学習者がボールをゴールにシュートする様子を示す．**Fig. 10** には敵ロボットは存在しないが，まずは簡単な状態での実機への適用を示す．左上の配置において，カメラ画像ではボールが左に，敵陣ゴールが右に見える状態である．ロボットはボールと敵陣ゴールが一直線上に見える状態にまで回り込み，最終的にボールを敵陣ゴールにシュートすることに成功している．また，獲得された行動は本研究室における従来研究 [2, 6] を用いた場合と比較して，非常にスムーズな行動である．

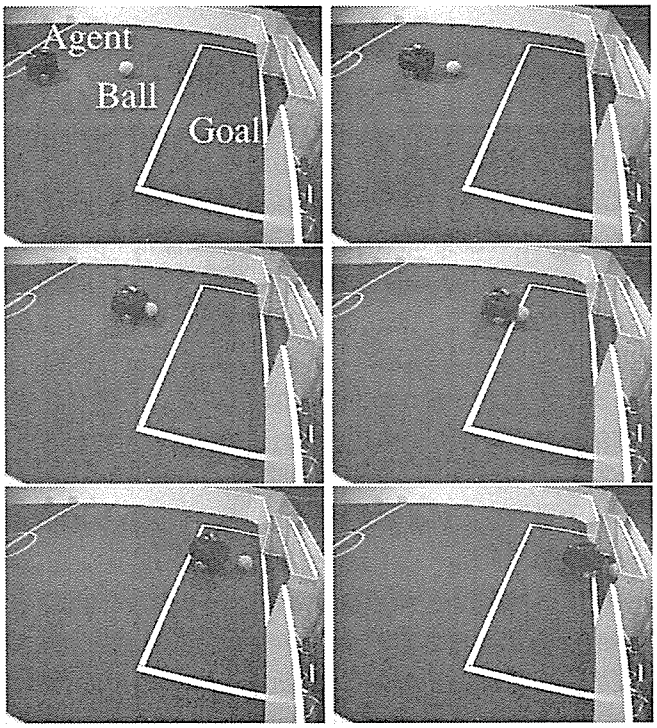


Figure 10: Typical shooting behavior in the real environment

5 おわりに

本報告では，学習による移動ロボットの行動獲得を行う際の適応的な評価関数を提案した．実験により適応的な評価関数を用いた方が，重みを固定した評価関数を用いた場合より優れた行動を生成することが分かった．また，目的関数の優先順位が異なる場合について，進化の過程は少し異なるが，最終的な結果はほぼ同様であり，優れた行動を獲得することができた．以上により，本手法の有効性が示された．

今後の課題としては，重み固定の評価関数を用いた，GP に基づくマルチエージェント環境における共進化の研究 [6] に対して提案手法を適用することにより，検討を行う．また，複雑な行動を単一の学習機構により獲得することは学習機構的にも学習時間的にも困難である．そこで機能のモジュール化による進化的手法を用いた階層型学習機構に提案手法を適用し，検討を行う．

謝辞

本研究は，日本学術振興会 未来開拓学術研究推進事業「分散協調視覚による動的 3 次元状況理解」プロジェクト (課題番号 JSPS-RFTF96P00501) の補助を受けた．

参考文献

[1] http://www.robocup.org/.

[2] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[3] J. R. Koza. *Genetic Programming I: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[4] T. Omata. Learning with assistance based on evolutionary computation. In *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2180–2186, 1998.

[5] E. Uchibe, M. Asada, and K. Hosoda. Environmental complexity control for vision-based learning mobile robot. In *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 1406–1413, 1999.

[6] E. Uchibe, M. Nakamura, and M. Asada. Cooperative and competitive behavior acquisition for mobile robots through co-evolution. In *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 2538–2544, 1995.

[7] B.-H. Yang and H. Asada. Progressive learning for robotic assembly : Learning impedance with an excitation scheduling method. In *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2538–2544, 1995.

動画像処理によるロボカップのシーン解析

Scene Analysis of Robocup Game by using image processing

須藤　智，郭　倖受，小竹　正裕，吉田　誠，小沢　慎治

Satoshi SUDO, Bong Soo KWAG, Masahiro KOTAKE, Makoto YOSHIDA, Shinji OZAWA

慶應義塾大学大学院理工学研究科

Faculty of Science and Technology, Keio University

{ satoshi,kwag,kotake,makoto,ozawa } @ozawa.ics.keio.ac.jp

Abstract

In recent years, RoboCup is paid attention as a standard problem of distributed collaboration system. The purpose of RoboCup is to make robots play soccer.In order to make robots cooperate through playing soccer, we need the various technologies, which are autonomous agents, multi-agent collaboration, strategy acquisition, real-time processing, and sensor-fusion. RoboCup project includes object recognition, localization recognition, moving object tracking, and so on.In the soccer field, the situation changes dynamically,so tracking moving objects is very effective to estimate the situation.

Now we propose the methods to recognize robots localization and track moving objects, by processing real image sequence of RoboCup game.

1　はじめに

近年，"RoboCup"が分散協調システムの標準問題として注目を浴びている．このRoboCupの目的は，ロボットにサッカーをさせる事だけではなく，非常に広範囲の技術を統合したり，実験したりすることが可能な標準問題を提供することによって，人工知能と知能ロボットに関する研究を促進するために始めた，サッカーを題材にした研究である．ロボットにより構成されたチームによってサッカーが実際に実行されるためには，自律エージェント，マルチエージェントによる協調，戦略の獲得，実時間処理，そしてセンサー技術などといった多様な技術を組み入れる必要がある．

RoboCupは物体の認識，ロボットの位置同定，移動物体の追跡，移動機構，ボールの操作機構，ロボット間の意志伝達などの問題を含んでいる．ここでロボットの位置同定については，ロボットの視線から見た画像とサッカーフィールドの環境モデルから作成した画像のマッチングを取って、ロボットの位置同定を行う研究[望月, 1999]がある．実際の試合では動的に環境が変化しており，移動物体の位置を同定し追跡を行うことは，試合の流れを記述し，予測する上で極めて有効性が高い．そしてこの移動物体追跡の処理を経て，戦術解析の処理へと後を委ねることになる．

ロボカップの研究が始まる前から，スポーツのシーン解析の研究も盛んに行われており，テレビ中継にてカラー情報より同じチームの識別を行ったり[Vandenbroucke, 1997]，抽出された選手・ボールの位置情報から，選手の優勢領域を判定してスペースというあいまいな対象に対して，定量的な指標を定義したり[瀧, 1998]している．

シーンを解析するにはまずはじめに各選手の位置を特定する必要があり，そのためには正しい追跡法の確立が求められている．

そこで本研究では，天井に固定カメラを設置してRoboCupの試合を撮影した画像に対して各チームの各ロボットの位置を同定し，追跡を行う手法を提案する．

2　提案する処理の概要

図1に提案手法の流れを示す．本手法では，まずはじめに短時間(1秒間)の移動物体の運動を表すSTOP(Short Term Object Pattern)と呼ばれるパターン図形を作成する．さらにSTOPを時間的に接続していくことにより軌跡を抽出する．このようなパターンを作成することにより，移動物体領域の抽出の際の分裂等の不安定さの影響を軽減することができる．また，オクルージョンが発生した際は前後の運動情報からその部分の動きを推測することができる．フィールドスポーツにおいては，見かけ上の交錯・接

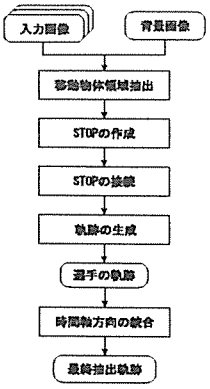


Table 1: 処理全体の流れ

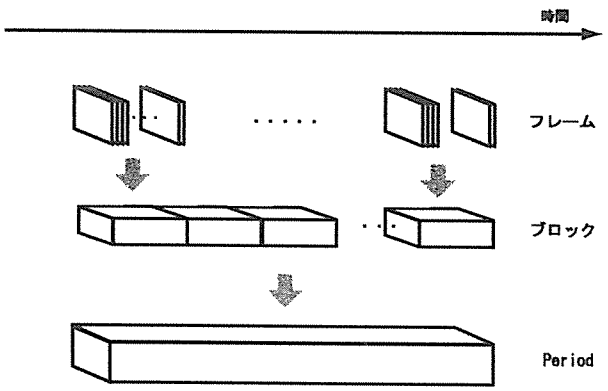


Table 2: 時間の区切りの定義

触は頻繁に起こってしまうため，その部分を画像上で詳しく解析するよりも前後の運動情報よりオクルージョン時の動きを推定するほうが有用であると考えられる．

ここで，時間の区切り方の呼び名を定義しておく．図2に示すように，画像列があるとする．今回使用した画像は30フレーム/秒である．このとき1秒間の画像列，すなわち今回では30フレームごとにブロックとして区切る．そしてこのブロック毎に連続フレーム間の探索を行いSTOPを作成する．このブロックが15区間続いたものをピリオドと呼び，このピリオド内で選手の運動軌跡を抽出する．このピリオドを連続して設定し，それらを接続していくことにより，長時間の移動物体の抽出を可能にする．

3 選手の追跡

3.1 撮影環境

図3に示すような，フィールドの真上に設置されたカメラからの画像を入力とする．

入力画像例は図4に示すようになり，フィールド全体が一つの画像内に収まるようにセットされている．このときロボットの大きさは直径が18cm以下のため，画像上では15画素程度になり，追跡するためには十分な解像度であると考えられる．真上から見ているため，ロボットの影に他のロボットが隠れて見えなくなってしまうことはなく，すべてのロボットは同時に画像上に映っているのがわかる．

3.2 カメラキャリブレーション

ロボットの運動軌跡を再現するために，フィールドモデルを定義する．高さ方向にYとし，フィールドはXZ平面にのっているものとする．ロボカップのフィールド上ではロボットは常にグラウンド上にいるため，高さYは常に一定となり，したがって，画像上の座標とフィールド上の座標との変換は2次元同士の変換となるため，双方向で一意の変換が可能である．なお，撮影環境にての入力画像を見てもわかるように，カメラはほぼ鉛直方向を向いている撮影されているため，ロボットの高さ方向の補正は考えなくて良い．

このワールド座標系 (X,Y,Z)(フィールド上) とカメラ座標系 (x,y) の間の関係をあらわす変換行列を式 (1) に示す．

$$\begin{pmatrix} Hx \\ Hy \\ H \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

この変換行列は3次元情報が既知の点を利用して，パラメータを較正する．ロボカップのフィールドが規定通りに引かれているとして，ラインの交点等の6つの基準点を用いて，最小自乗法によりカメラパラメータを算出する．

カメラは固定であるので，あらかじめパラメータは求めておき，以後，変換はこの行列を用いる．

3.3 移動物体抽出

撮影環境で述べたように，本手法では固定カメラを用いて撮影を行うため，移動物体抽出は背景差分により行う．ロボカップは卓球の台の上で行われるため一台のカメラで上方から撮影しても，解像度はそれほど低くはならない．ロボットの大きさは直径18cm以内と決められており，最大の大きさを取るときには画像を256x240画素でディジタイズしたときには15画素四方程度になるため，抽出する

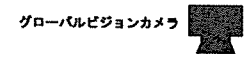


Table 3: カメラの設置

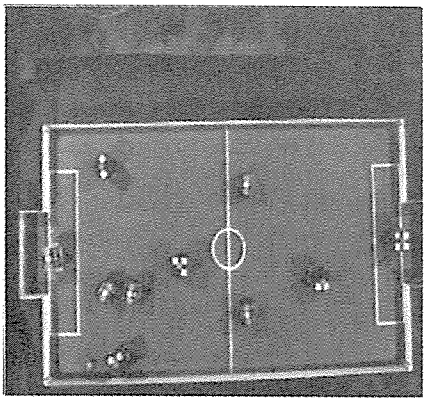


Table 4: 入力画像例

には十分の大きさであるといえる．しかし，領域を抽出する際に2値化のしきい値等の問題により，1台のロボットであっても複数の領域に分裂してしまう可能性がある．そこで，背景差分画像に対し，同一フレーム内で領域の再統合を行う．フレーム内にて面積の比較的大きい移動物体領域に対し，その周りの小さな移動物体領域を探索し統合した結果が単一の動物体になりうるかどうかを判定する．これにより，分裂による影響を軽減するが，ここで誤った接続をしてしまうと，分裂は難しいため，比較的厳しい条件のもとに結合を行っている．以上の処理を行うことにより，移動物体領域を抽出する．

3.4 STOPの作成

ここでは，物体の短期間の運動を表すパターン図形(STOP)の作成の方法について述べる．これはある一定時間の移動物体領域の論理和を取ることにより作成する．これを作ることにより，動領域の分裂を防ぎ安定して追跡ができると考えられる．以下に実際の手順を述べる．

(1) ラベルの対応付け
連続する2フレーム間において，各ラベルのユークリッド距離を計算し閾値以下のものを対応するラベルとして登録する．この処理を時間的に昇順，降順の双方向で行う．双方向で行うことにより，ひとつの物体が一時的に複数のラベルに分裂してしまっても，時間がたった後に再び一つのラベルに接続されると，途中で分裂してしまったラベルもすべて，同じ物体としてのラベルであると判定できる．長い間物体が同じような複数のラベルに分割してしまうことは稀であり，通常では一つのラベルとして表現されるフレームの方が長い時間あると考えられる．そこで、対応付けには双方向の探索を用いて，分裂時も考慮するようにする．

(2) STOPの生成
(1)にて対応付けられたラベルを1秒間(1ブロック)

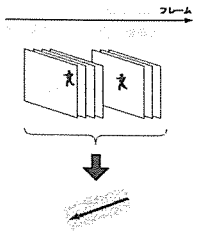


Table 5: STOPの生成

分加算した画像を図5のように生成する．このときに生成された加算画像をSTOP(Short Term Object Pattern)と呼ぶ．これは，1秒間の物体の動きをあらわしたものであり，通常はひとつの物体の動きを一つのSTOPが表している．STOPの属性としては，各フレームにおける物体の重心位置と物体の移動方向を持つものとする．

以後，ここで求めたSTOPを移動物体領域の最小単位として追跡を進めていく．

3.5 STOPの接続

求められたSTOPに対して，方向性も考慮して接続していく．各STOPをそれぞれフィールド座標に変換し，STOPの終点と次のブロックでの始点とのユークリッド距離を計算し，もっとも距離の近いものを対応付けする．この処理を昇順，降順と双方向で行う．双方向ともに同じものが対応付けられたSTOP同士はセグメントとして連結される．一方，複数の接続の可能性がある場合には，そこにノードというものを発生させ，接続の可能性を保持しておく(図6)．サッカー等のフィールドスポーツにおいては、選手同士が激しく入り乱れ，見かけ上の交錯も多々起こり画像上で接触するが，ロボット同士のサッカーにおいては接触する機会は人間に比べて格段に多く，一意に接続を決定するのは困難である．そこで，接続の可能性があるものはすべて保存しておき，前後の動きからその部分を推定するほうが，よりロバストな接続が可能となる．以後は，セグメントは一つの単位として，それ以上は分解せずに，ノードにおいてセグメント同士がどのように接続するのかを推定していく．

3.6 軌跡の生成

(1) 軌跡候補の生成
ここでは，3．5にて生成したセグメントをノードを元につなぎ合わせる処理を行う．セグメントとセグメントの間にはノードが発生しており，ここには複数の接続の可能性が保存されている．そこで，すべての可能性を網羅するようにセグメントの列の組み合わせを探し，これらすべてを軌跡候補とする．軌跡候補の中には正しい軌跡も含まれるが，誤った接続

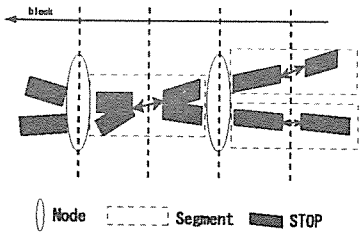


Table 6: セグメントとノード

をされているものも存在しているため，軌跡候補の中から正しい軌跡を探索する．

(2) 軌跡候補のグループ化
複数の移動物体が互いに接触しながら運動している場合，接触が起こった場合には誤った追跡をしてしまう場合がある．そのときにはお互いに入れ替わって追跡してしまうと考えられる．そのため，これらの互いに関与する物体も考慮に入れて接続を推定するほうがより正しい接続が行われると考えられる．そこで，(1)で生成された軌跡候補において，次にグループ化を行う．すべての軌跡候補の中で，同一のセグメントを共有している軌跡候補同士を集めてきてグループを生成する．直接的に共有している軌跡候補から、ある軌跡を経由して間接的に共有しているものまですべて集めてきて，一つの大きなグループを生成する．

(3) 軌跡数の推定
(2)で生成された軌跡候補グループの中から存在すると考えられる軌跡の数を推定する．一般的にグループ内には複数の物体の運動軌跡が混在していると考えられる．そのため，グループ内にいくつ移動物体が存在するかを推定する必要がある．そこで，長く続いているセグメントを基準セグメントとして，グループ内のすべての軌跡候補に対し，各ブロックに基準セグメントが何個存在するかをカウントする．すべてのブロックでの最大カウント数をそのグループ内での軌跡の数と推定する．

(4) グループ内の軌跡抽出
推定された数だけの軌跡をグループ内から抽出する．探索はすべて基準セグメントから行うものとする．ノードにおける接続の決定は，連続する二つのセグメントの中でノードに接しているSTOPの移動ベクトルを計算する．ノード部分は時間的な区切りのため，幅を持たない．そのため，前後のSTOPにおいて，移動ベクトルの変化は少ないはずである．そこで，ノードに接しているSTOPのうち，最も似かよったものを接続する．ただし，セグメントの選択の際には同じグループ内の他の軌跡の選択結果を考慮し，複数の組み合わせがある場合には，すべての軌跡の組み合わせの中で，最適な組み合わせを選択する．これにより，軌跡候補の中から軌跡を抽出し，物体の運動軌跡として表現する．

4 実験および結果

実際のロボカップの試合の映像を256x240画素，グレースケール256階調，フレーム間隔30 frame/secの時系列画像にディジタイズし，これを入力とする．ロボット領域の抽出，追跡を行い，最終結果として，ワールド座標上のフィールドモデルに射影した軌跡により表現する．追跡結果例を図7,8に示す．またそれぞれの軌跡を図9，10に示す．

ロボットの領域を抽出する際に回りのノイズを拾ってしまって誤った形になってしまったものが存在した．今回は動領域の抽出に背景差分を利用したが，このとき背景画像と原画像が微妙にずれてしまっていると比較的明度値の高い直線部分などがノイズとして動領域として残ってしまう．また、光が4方向から照らされており，それらの影が出てしまった関係で，ロボット同士が接触していないにもかかわらず，差分画像ではくっついて一つの領域と抽出されてしまうこともあった．

人間のサッカーに比べロボットのほうが，フィールド上にできるあいているスペースが少なく，ロボットが占める割合が多かった．そのため，頻繁にロボット同士の接触が起こってしまい，動領域抽出の際に，すでに一つの物体としてなってしまっていることが多々あった．このため，STOPを作成する段階で複数のロボットの動きを一つで表しているSTOPが数多くできてしまい，接続の段階で，さまざまな可能性が出てしまい，選択する段階で誤ってしまっているものもあった．また，今回は軌跡の抽出の際に長いセグメントを持つ軌跡候補の中で軌跡を絞り込んでいったため，短いセグメントしかない軌跡候補は選択されていない．また，一度軌跡が途切れてしまうと，次に出てきたときには別の物体と認識してしまっていた．

5 おわりに

実際のロボカップの試合を撮影した天井カメラから，フィールド上のすべてのロボットを抽出し，追跡する手法を提案した．この手法では15秒間の追跡結果が出るが，処理区間をオーバーラップさせることにより，それらの連続した区間内の情報の連結ができ，1試合通しての解析は可能となる．

選手の運動を表すパターン図形(STOP)を作成することにより，移動物体領域抽出の不安定さを軽減でき，また，オクルージョン時においては，その前後の運動からその時点での動きを推定できる．

しかし，完全に二つの物体が接触して運動を止めてしまうと，分離は難しく，高解像度の画像を利用してテクスチャ解析をする必要があると考えられるが，ある程度の修正機構を組み込み，オペレーターにより操作すれば，高い信頼度のシステムが出来上がると考えられる．また、その場合には分離後の動き方により，選手を同定することが考えられる．

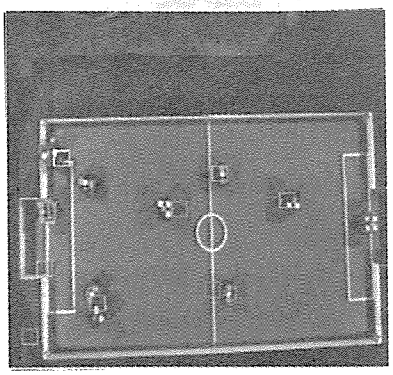
今回は動領域抽出には濃淡画像の背景差分を利用したが，ロボットには固有のID番号が触れるように表面にシールを張ることになっているため，その情報を元にして，動領域を抽出することができれば，さらに追跡は精度良いものになることが考えられる．

参考文献

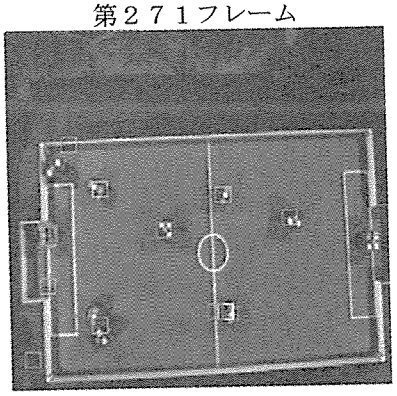
[望月, 1999] 望月恒治，上野敦志，武田英明，西田豊明：”RoboCup Robotにおける画像を利用した位置同定と敵の発見手法の提案”，人口知能学会研究会資料，pp.1-6，1999

[Vandenbroucke, 1997]
N.Vandenbroucke, L.Macaire, J.G.Postaire: Soccer players recognition by pixels classification in an hybrid color space”, Multispectral and hyperspectral imagery III (SPIE proceedings n° 3071), pages 23-33, avril 1997.

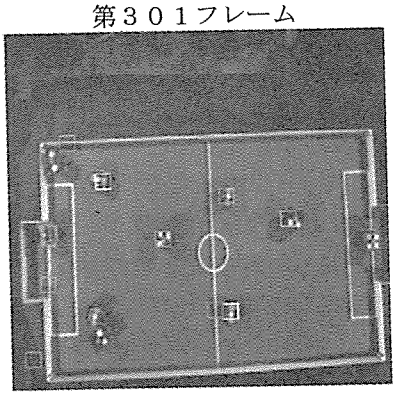
[瀧, 1998] 瀧 剛志，松本 貴之，長谷川 純一：チームスポーツにおける集団行動解析のための特徴量とその応用”，信学論 (D-II)，**J81-D-II**，8，pp.1802-1811(1998-8)



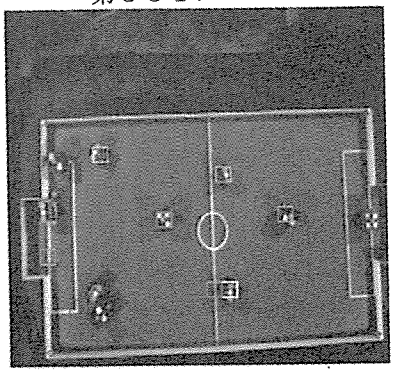
第２７１フレーム



第３０１フレーム

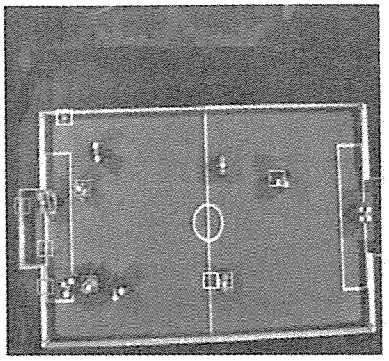


第３３１フレーム

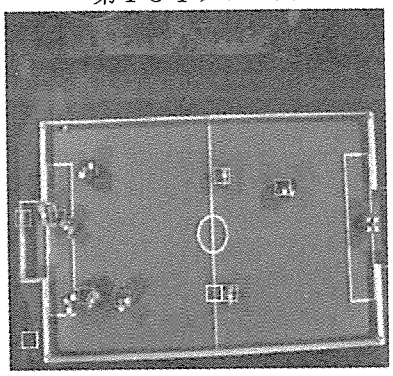


第３６１フレーム

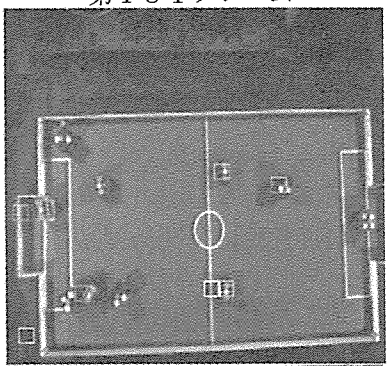
Table 7: 追跡結果例 (Scene1)



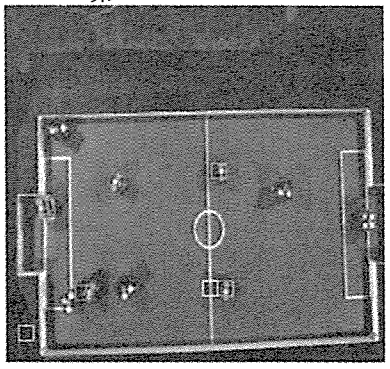
第１５１フレーム



第１８１フレーム



第２１１フレーム



第２４１フレーム

Table 8: 追跡結果例 (Scene2)

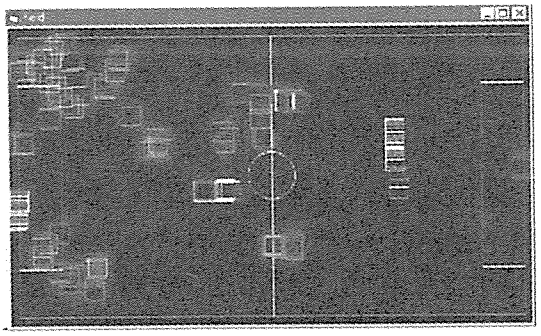


Table 9: 最終結果 (Scene1)

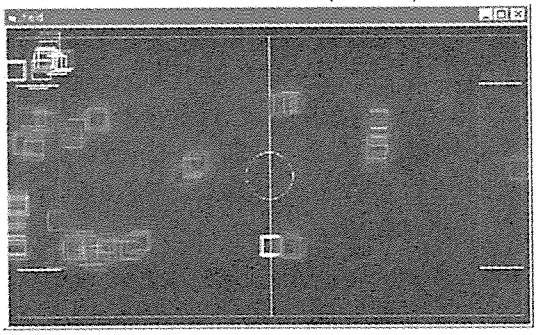


Table 10: 最終結果 (Scene2)

全方向移動機構と全方位視覚を有する小型ロボットによるサッカー競技の実現 — チーム OMNI の戦略 —

Realization of Soccer Game with Small Size Robots which have Omni-directional Mobile Mechanism and Omni-directional Vision — Strategies of Team OMNI —

森 信人† 家田 純一† 松井 渉† 臼井 智也† 三宅 修† 金 東杓† 前田 哲裕†
杉本 浩和‡ 辰巳 優介‡ 藤本 良平‡ 関森 大介‡ 升谷 保博† 宮崎 文夫†

Nobuhito MORI†, Junichi IEDA†, Wataru MATSUI†, Tomoya USUI† Osamu Miyake†
Tonpo KIN†, Tetsuhiro MAEDA†, Hirokazu SUGIMOTO‡, Yusuke TATSUMI‡
Ryouhei FUJIMOTO‡, Daisuke SEKIMORI‡, Yasuhiro MASUTANI†, Fumio MIYAZAKI†

†大阪大学 ‡明石工業高等専門学校

†Osaka University, ‡Akashi College of Technology

http://robotics.me.es.osaka-u.ac.jp/MiyazakiLab/Research/soccer/

Abstract

We, the robotic foot ball team "OMNI", are developing a soccer robot system with omni-directional mobile mechanism and omni-directional vision according to the rule of RoboCup small-size league. The system consists of the actual robot and the main computer which are connected by two radio links. In this paper, we describe the system configuration, omni-directional mobile mechanism, omni-directional vision, strategies, and simulator for our robot system.

1 はじめに

我々は，RoboCup 小型機部門のルールに準じたサッカーロボットシステムとして，全方向移動機構と全方位視覚を有する小型ロボットシステムの開発を行なってきた[1]. 我々のチーム名である "OMNI" は全方向移動 (omni-directional mobility) と全方位視覚 (omni-directional vision) に由来している．現在，我々はこのロボットシステムを複数台に拡張し，サッカータスクの実現に必要な制御アルゴリズムや戦略の研究を進めている[2][3].

本研究で開発したロボットシステムは，実際にフィールドを動き回るロボット部と画像処理や行動決定を行なうメインコンピュータ部から構成されている (**Fig.1**参照). 現在の RoboCup の小型機部門では，天井カメラの画像を用いるグローバルビジョン方式が主流—であるが，我々は自律分散型知能の研究を重視する立場から，ロボットに搭載されたカメラの画像のみを用いる完全ローカルビジョン方式を採用している．

本稿では，本ロボットシステムのハードウェア構成，全方向移動機構，全方位視覚，ソフトウェア構成，基本タスクの実現の方法について述べる．さらに，本システムに対応したシミュレータについても紹介する．

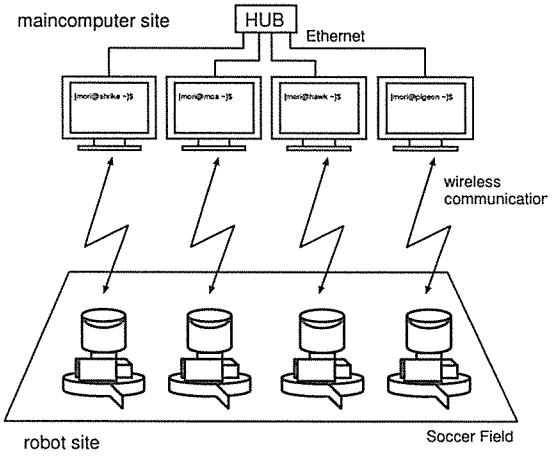


Fig.1 Concept of soccer robot system

2 ハードウェア構成

本ロボットシステムは大きく分けて，ロボット部とメインコンピュータ部から構成されており，両者は無線通信で結合されている．また，サッカーロボット同士の通信はそれぞれのメインコンピュータ部が Ethernet を介して行なっている．**Fig.2**にシステムの構成図を示す．

2.1 ロボット部

黒色カバーを外したロボット部の外観を **Fig.3**に示す．ロボット部は円形の車体とその周りを回転できるアウタリングから構成されており，CPU(NEC V55, クロック 16MHz) ボード，I/O ボード，CCD カメラ，全方位ミラー，駆動用モータ 2 個，アウタリング用モータが搭載されている．車体の直径は 140[mm] で高さは 225[mm] である．

アウタリングにはフィンが取り付けられており，これを利用してボールの操作を行なう．**Fig.4**に現在用いている 2 種類のフィンの形状を示す．このフィンは，ボール

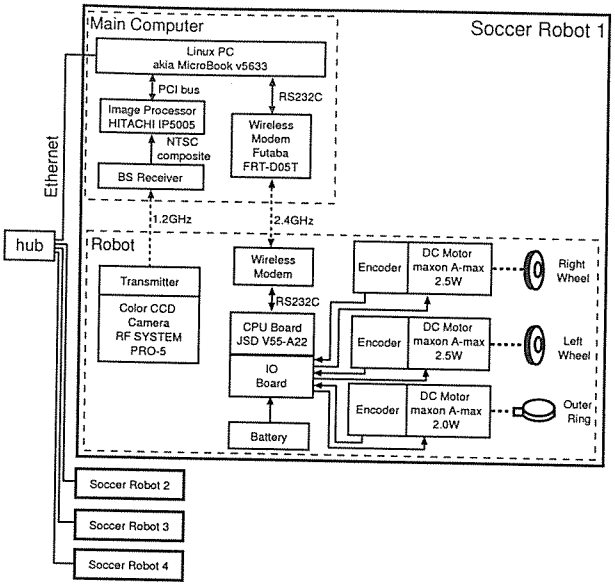


Fig.2 System configuration

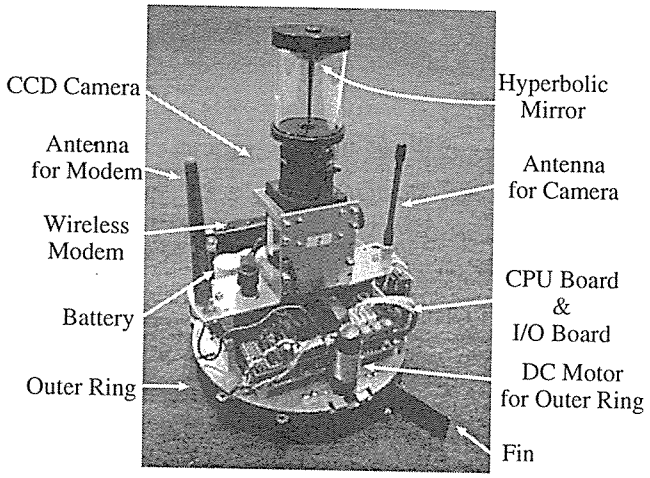


Fig.3 Overview of the robot

を操作するための板にボールの運動を停めるためのトラップクッション (衝撃吸収材) が取り付けられている．このクッションの導入により，ボールをロボットの足下に停止させてから次の行動を選択することができ，より知的なプレーが実現できると考えている．なお，このフィンを含めたロボットの底面積を凸包で求めると，それぞれ 17534, 17560[mm^2] となる．また形状は，ホールディングのルールを考慮して設計されている[5].

車体上部には無線モデムが搭載されており，メインコンピュータから送信された指令を基に CPU が左右の車輪用モータとアウタリング回転用モータの制御を行なう．また，各モータの回転量を積算してデッドレコニング値の演算も行なっている．

CPU ボードと I/O ボードの上部に CCD カメラが上向きに設置されており，レンズの先にはアコウル製の双曲面ミラーが取り付けられている．これらによって，全方位画像を得ることができる．得られた画像はカメラに内蔵されている送信機でメインコンピュータ部に送られる．

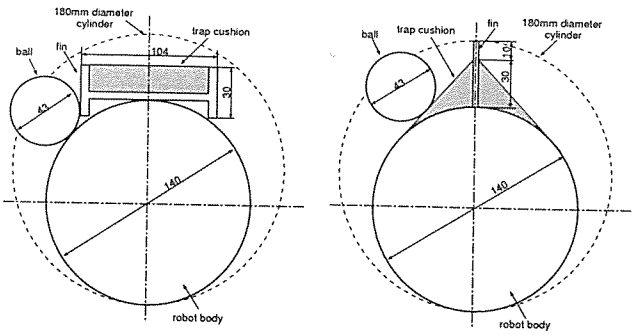


Fig.4 2 types of fin with trap cushion

2.2 メインコンピュータ部

メインコンピュータ部には，CPU に PentiumII 333MHz を用いた一般的な PC を使用し，画像処理ボード (日立 IP5005) および無線モデムが接続されている．さらに他のサッカーロボットのメインコンピュータ部と 100Mbps の Ethernet を用いた LAN を構成している．OS は Linux を用いている．

3 全方向移動機能

サッカーロボットには時々刻々と変化する動環境に素早く対応するために全方向へのスムーズな移動能力が求められる．そこで，本研究では和田らによって提案されているアクティブ単輪キャスタ機構[6]を参考に，左右独立駆動方式による全方向移動機構の開発を行なった．

Fig.5に示すように，ワールド座標系を Σ_w とし，車体中心に取りつけられた座標系を Σ_r とする．X_w 軸と X_r 軸のなす角を車体の姿勢角 θ とする．このとき，ワールド座標系 Σ_w に対して，ϕ 方向へ速度 $\boldsymbol{v}_r$ を発生させる場合を考える．車体の姿勢角 θ はデッドレコニング値から得られるため，ロボット座標系 Σ_r における速度 $\boldsymbol{v}_r$ の X_r, Y_r 成分は次のように与えられる．

$$v_x = |\boldsymbol{v}_r|\cos(\phi-\theta),\ \ v_y = |\boldsymbol{v}_r|\sin(\phi-\theta) \qquad (1)$$

よって，左右の駆動輪に与えるべき速度 v_L, v_R は以下の式で求めることができる．

$$v_L = v_x - \frac{d}{s}v_y,\ \ v_R = v_x + \frac{d}{s}v_y \qquad (2)$$

ここで，s はオフセット距離，d は車輪間距離の 1/2 を表す．

アウタリングの回転速度は，**Fig.6**のように，指令速度に車体の回転を打ち消す回転速度を加えることで実現する．そのために車体に対するアウタリングの位置指令 R_O を以下のように与える．

$$R_O = R_w + k_c(C_L - C_R) + k_f(v_L - v_R) \qquad (3)$$

ここで，R_w はワールド座標系における目標位置，第 2 項は車体の回転分を打ち消す角度，第 3 項は過度特性分を

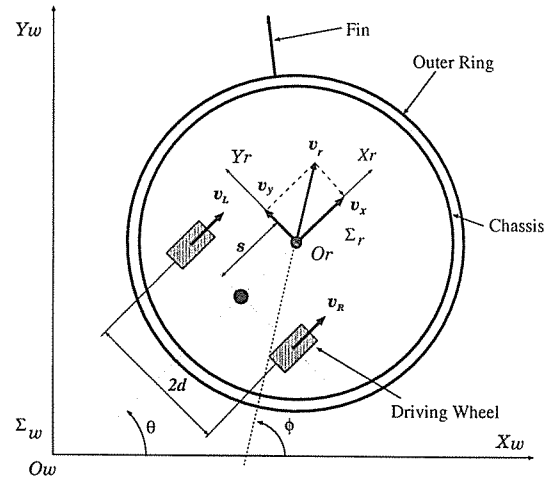


Fig.5 Model of the omni-directional mobile mech.

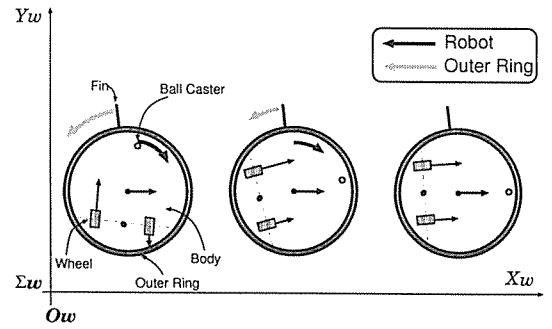


Fig.6 Principle of omni-directional mobility

改善するフィードフォワード項である．C_L, C_R は左右駆動輪の積算回転数，k_c はキャンセルのための定数，k_f はフィードフォワードゲインを表わす．本研究では，以上を基本として，駆動輪のスリップを抑制するための車体の加速度の最大値を制限する速度指令法や，高速旋回中の車体の揺動現象を抑制するためのフィードフォーワード補償を提案している[2][3].

4 全方位視覚

本研究では広範囲の視野を獲得するために，ロボットの視覚システムに双曲面ミラーを用いた全方位視覚システムを採用している．**Fig.7**に得られた画像の一例を示す．

画像処理ボードの機能を利用して，全方位視覚によって得られた画像から特定の色領域を抽出し，ボールやゴール領域の重心座標を得る．なお，画像メモリの解像度は256×220[pixel]としている．

このとき **Fig.8**のようにロボット座標系 Σ_r におけるボールやゴールの方向 ϕ は全方位画像上の方向 Φ と一致すると見なす．一方，画像上の距離 R[pixel] から床面上の実距離 r[mm] への変換式として，ボールには式 (4)，ゴールには式 (5) のモデルを考える．

$$r = \frac{R}{A - B\sqrt{C + R^2}} \quad (4)$$

$$r = D\tan(ER) + F \quad (5)$$

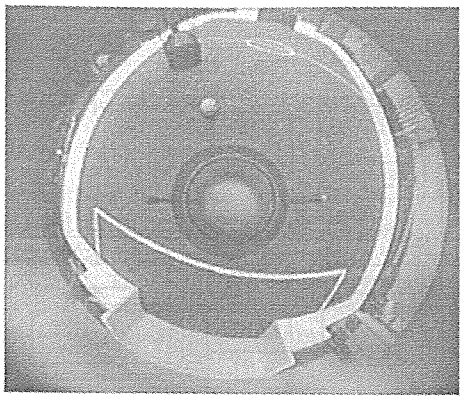


Fig.7 Omni-directional image

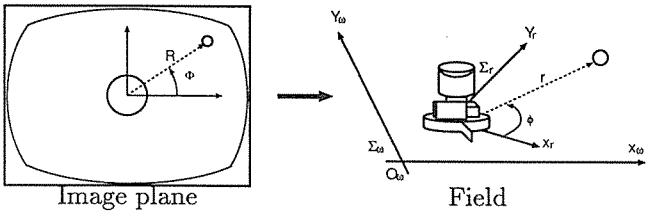


Fig.8 Distance and direction of ball

実測値から最小二乗法を用いて，パラメータ A, B, C および D, E, F を以下のように決定した．

$$A = 0.796,\ B = 0.00563,\ C = 6820$$
$$D = 169,\quad E = 0.0143,\quad F = 27.9$$

この全方位視覚システムを用いて，位置同定実験を行なったところ，直径 43[mm] のオレンジ色のゴルフボールに関しては，カメラ中心から約 1400[mm] 程度の範囲内で同定でき，両ゴールに関しては，フィールドのほぼ全域で同定することができた．

5 ソフトウェア構成

5.1 ロボット部

ロボット部のプログラムは，MS-DOS 用の C 言語 (Turbo C++) を用いて開発しており，実行プログラムが EEPROM に書き込まれている．プログラムは，メインコンピュータ部からのコマンドを解釈し，左右の駆動輪とアウターリングの制御を行なう．そのため，メインコンピュータ部は，モータの制御や下位の運動生成を行なう必要がなく，全方向移動，円弧運動，フィンの操作，デッドレコニング値の問い合わせなど，マクロな指令を送ることができる．メインコンピュータ部とロボット部の間の通信には，1byte のコマンドと数個の 2byte または 4byte のパラメータが用いられている．

5.2 メインコンピュータ部

メインコンピュータ部では，POSIX 1003.1c 準拠の Linux Threads ライブラリ[4]を用いたマルチスレッドなプロ

グラムをC言語で記述している．Fig.9にスレッドの構成を示す．図中の楕円が1つのスレッドを表しており，これらが並行処理されている．スレッド間では大域変数を介してデータを授受している．

ip5kでは，画像処理ボードに対して，画像の取り込み，フィールド，ボール，両ゴールの色領域の抽出，領域の重心計算などを指示する．

communicateでは，kernelで決定されたコマンドを，無線モデムを介してロボット部へ送る．また，ロボット部で積算しているデッドレコニング値を問合わせて，その返答をスレッドball，locationに送る．

ballでは，間欠的に得られるデッドレコニング情報と全方位視覚のボール位置情報に基づいて，ボールの運動の推定を行う．

locationでは，間欠的に得られるデッドレコニング情報と全方位視覚のゴール位置情報に基づいて，自己位置の推定を行う．

fspaceでは，ip5kで抽出されたフィールドの情報をもとにロボットの周囲の移動可能領域を決定する．

kernelでは，TCPによるソケット通信で他のロボットと通信しながら，センサ情報に基づいて行動を決定する．

playerNetでは，他の全てのロボットからの送られてくる情報を更新する．

guiでは，ロボットの自己位置，ボールの推定位置，移動可能領域などをXサーバに描画させる．

5.3 ロボット間通信

本システムでは，ロボット間通信の実現にはTCPを用いている．ロボット間の送信部分は直接行動決定処理(kernel)に組み込み，受信部分はスレッド(playerNet)処理することで，常に他のロボットからの最新情報を得る．

各ロボットは，行動の一単位ごとに，他の全てのロボットへ，自己位置，ボールの位置，実行中のタスクの識別番号，他ロボットに対するリクエストなどを送信する．

また，本システムでは，ロボット間通信以外に各ロボットに審判の指示を伝達するプログラム(伝令プログラムと呼ぶ)を競技中にメインコンピュータのうちの1台で実行している．現在は，審判からの指示がコンピュータに理解しやすい形式にはなっていないため，その機能の一部(審判の音声や身振りの認識)を人間が代替している．

6 基本タスク

6.1 自己位置の同定

ロボットの自己位置同定は，全方位画像とデッドレコニング値を融合することで決定するが，ここでは全方位画像から得られたゴール位置に基づく方法について説明する．

Fig.10に示すように，ワールド座標系をΣ_wを取り，ロボット座標系をΣ_rとする．まず，得られた画像を処理し，両方のゴール領域を抽出し，各ゴールの方向ϕ_A, ϕ_Bと距離r_A, r_Bを求める．ここで，$c = 1/\tan(\frac{\phi_2-\phi_1}{2}) = \tan(\frac{\pi}{2} - \frac{\phi_2-\phi_1}{2})$と置けば，ワールド座標系に対するロボットの位置$x_w, y_w$，姿勢$\theta$は以下の式で求めることができる．

$$x_w = L - \frac{2r_A^2c^2 + r_A(1-c^2)\sqrt{L(1+c^2)^2 - r_A^2c^2}}{L(1+c^2)^2} \quad (6)$$

$$y_w = \frac{r_Ac(r_A - r_Ac^2 - 2\sqrt{L(1+c^2)^2 - r_A^2c^2}}{L(1+c^2)^2} \quad (7)$$

$$\theta = -\phi_A - \mathrm{atan2}(y_w, L - x_w) \quad (8)$$

ここで，Lはフィールド中央からゴールまでの距離を表す．

実際のフィールド上の20箇所で自己位置同定の実験を行なったところ，実際の位置から最悪でも300[mm]の誤差で位置を同定することができた．

6.2 移動可能領域の検出

全方位視覚から得られる画像に基づいてロボット周囲の移動可能領域を導出する．

まず，サッカーフィールドの緑色を抽出して2値画像を得る．次に，数回の膨張・圧縮を繰り返してノイズの除去を行なう(**Fig.11**(a),(b)参照)．ここで，予め作成しておいたテンプレートとAND演算を行ないフィールド部を極座標における細かな領域に分割する(**Fig.11**(c),(d)参照)．

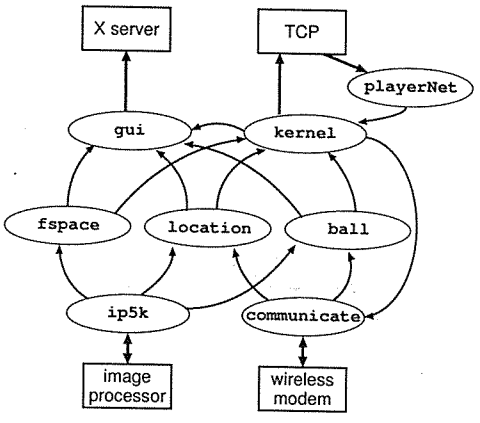


Fig.9 Configuration of multi-threads

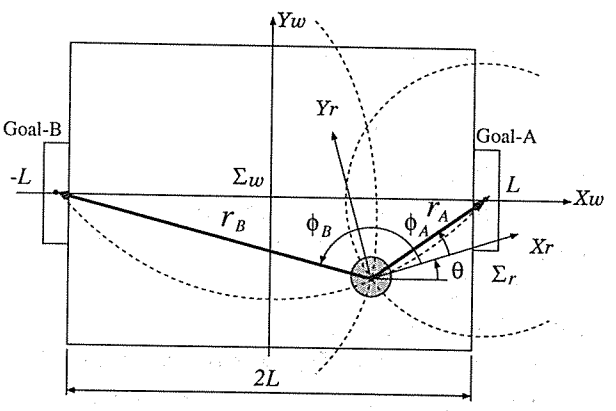
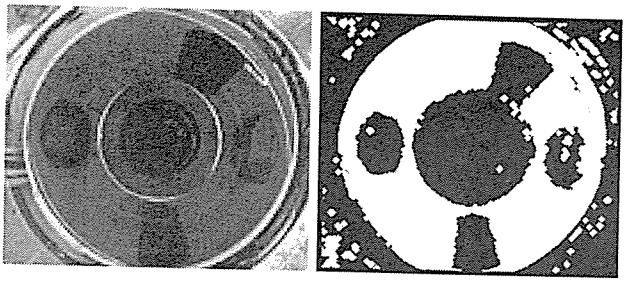
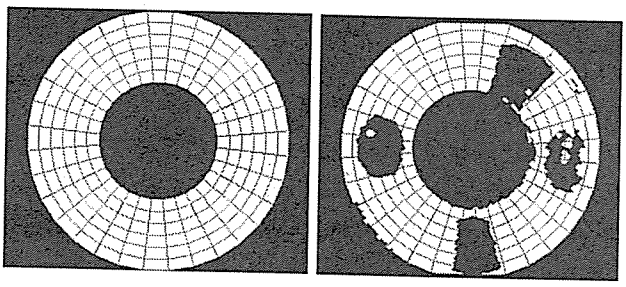


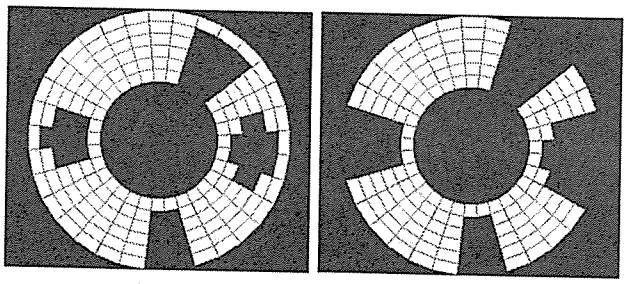
Fig.10 Position Identification



(a)Camera image (b)Binarization



(c)Template (d)After AND operation of (b) and (c)



(e)After labeling and integrating (f)Detected free space

Fig.11 Example of method of detecting free space

IP5005 のラベリングの機能を利用し，各領域毎に，フィールドの比率を算出し，その領域の障害物の有無を判定する (**Fig.11**(e) 参照)．さらに，障害物より遠い部分 (半径方向) を障害物なしの候補から外す．(**Fig.11**(f) 参照)

以上の処理により，ロボットの周囲の各方向の最も近い障害物までの距離が離散化された一次元データとして得られる．これを利用して，障害物を回避しながら，他のタスクを実行している．

6.3 ボールの運動推定・トラップ

転がってくるボールをトラップするためには，その動きを予測する必要がある．RoboCup の小型機部門ではボールの速度は 1000～2000mm/s にも及び，一方，本システムのロボットがボールを認識できる最遠距離は 1200～1400mm 程度である．従って，トラップを実現するには，ボールが見え始めてから 0.2～0.5s で安定して速度推定できる必要がある．

短期的には，ボールは静止座標系で等速直線運動しているとモデル化すれば十分である．静止座標系におけるボールの位置を知るには，全方位画像から得られるロボット座標系におけるボールの位置と，デッドレコニングから得られるロボットの位置姿勢の情報を統合しなければならない．

ところが，本システムでは，画像処理とデッドレコニングは非同期に行なわれている．また，画像から得られた情報には画像の量子化に起因する誤差が含まれている．そこで，全てのセンサ情報には取得時刻を付加して管理し，過去 10 時刻のデッドレコニング値を保持して，補間処理後に画像情報との統合を行なうようにした．また，量子化誤差を克服するために，等速直線運動のモデルに基づき，局所最小二乗法やカルマンフィルタを用いた運動推定を検討している[3]．

6.4 ボールに対する回り込み

ボールを目標の方向へ移動させるためには，ロボットはボールに対してその方向の反対側に回り込む必要がある．ここでは，ゴールを目標方向として考える．ロボット座標系 Σ_r におけるボールの位置を r_B, ϕ_B，ゴールの位置を r_G, ϕ_G とする (**Fig.12**参照)．ボールを原点，ロボットを x 軸上の $x > 0$ にとる座標系 Σ_O を考えると，Σ_O におけるゴールの位置 x_G, y_G は以下のように表され，それを用いて，ロボットが回り込むべき方向 α を求めることができる．

$$x_G = r_B - r_G \cos(\phi_B - \phi_G) \quad (9)$$

$$y_G = r_G \sin(\phi_B - \phi_G) \quad (10)$$

$$\alpha = \mathrm{atan2}(-y_G, -x_G) \quad (11)$$

メインコンピュータ部からは，半径 r_B と回転量 α だけを指定すれば，ロボット部は円弧運動機能によって，回り込み位置への移動を行なう．移動が完了後，ゴール方向へ直進し，シュートやドリブル等の次のタスクに移る．

7 シミュレータ

サッカーロボットシステムを開発するに当たり，開発の度に実機を用いて，条件をそろえ，数多くの実験を行なうことは大変な労力が必要とされる．戦略アルゴリズム等

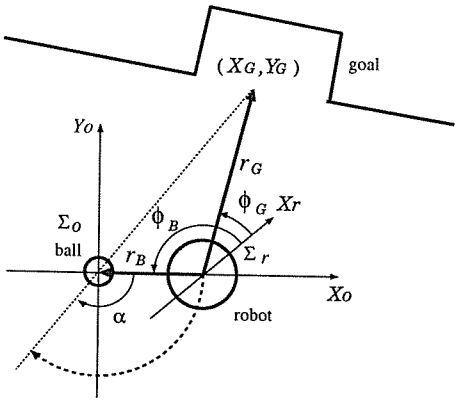


Fig.12 Turning around the ball

を様々な条件で数多く試行するために，本ロボットシステムに対応したシミュレータの開発を進めている．本シミュレータの開発コンセプトは次の通りである．

1. 複数のロボットが作業環境を共有し，相互干渉が可能なこと．
2. 実機用のプログラムの多くの部分をそのまま使うことができること．
3. 実機のハードウェアに依存する処理時間 (遅れ) や処理周期や CPU 負荷を再現できること．
4. ボールの反発，摩擦，ロボットの加減速，画像の量子化等は実際の現象を適切に再現できること．

以上を実現するために次のような実装を行なっている．

1. 複数のロボットが環境を共有する枠組みとして，RoboCup のシミュレーション部門で使われているサッカーサーバ[7]を利用する．
2. 実機用のメインコンピュータ部のプログラムはマルチスレッドで構成されているので，その中のハードウェアに依存するスレッドをシミュレータ用に差し換え，その他のスレッドは実機用のものをそのまま用いる (**Fig.13**参照)．
3. 実機用のプログラムのスレッドの実行頻度や処理の遅れを分析し，シミュレータにおいてそれらが近くなるように usleep() や空ループで調整する．
4. オリジナルのサッカーサーバでは，現実の物理現象を十分にモデル化しているとは言い難い．そこで，サッカーサーバのモデルに手を加え，サーバに対するコマンドも追加した．
   - 全方向移動: ロボットの姿勢に関わらず指定の方向に移動できるコマンドを追加した．その際の過渡的な姿勢の変化も再現した．また，実ロボットの加速度制限にも対応している．
   - フィン: 車体の回りを回転するリングとフィンをモデル化し，目標位置か目標速度を与えるコマンドを追加した．
   - 衝突: 車体とボール，ボールとフィンの衝突判定を厳密に行ない，ニュートンの反発則に基づくモデル化を行なった．kick コマンドは使わずに，車体やフィンとボールとの衝突によってキックやドリブルを再現する．
   - デッドレコニング: 各ロボットの状態として，真の位置姿勢と，デッドレコニングを想定した位置姿勢の 2 種類を保持している．
   - 全方位視覚: 視野を 360 度に固定している．頻度や分解能は，実ロボットに合わせて調整している．
   - 壁とボールの衝突: 傾斜壁との衝突のモデル化を検討中である．

実際のハードウェアの代替は，サッカーサーバとの間に入るスレッドが受け持つ．実ロボット用のコマンドは，サッカーサーバへのコマンドへ変換される．

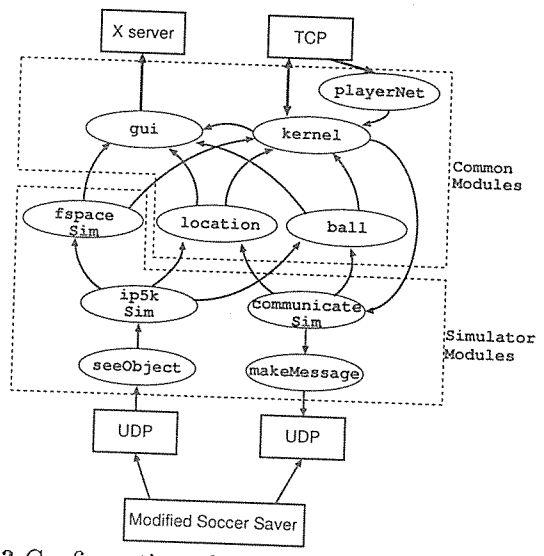


Fig.13 Configuration of multi-threads of the simulator

また，実時間でシミュレーションを行なう際のコンピュータの負荷を考慮して，カメラ画像の生成までは行なっていない．しかし，サッカーサーバから送られてくる幾何情報に基づいて，物体の大きさを考慮した隠れや量子化を考慮した処理の導入を検討している．

8 おわりに

本稿では，チーム OMNI のハードウェア構成，全方向移動機構，全方位視覚，ソフトウェア構成，基本タスク，シミュレータについて述べた．今後は，各基本タスクの精度やシミュレータの有用性を向上させ，協調行動やチーム戦略のさらなる研究を進め，高度なサッカータスクの実現を目指す．

参考文献

[1] 森信人 他: “全方位視覚と全方向移動能力を有する小型サッカーロボット”, 第 17 回日本ロボット学会学術講演会予稿集, No.2, pp.553-554, 1999

[2] 関森大介 他: “左右独立駆動型全方向移動機構の制御アルゴリズム”, ロボティクス・メカトロニクス'00 講演会講演論文集, 2PI-41-056, 2000

[3] 森信人 他: “搭載カメラを用いた小型サッカーロボットのボールの運動推定”, ロボティクス・メカトロニクス'00 講演会講演論文集, 2P2-27-030, 2000

[4] The LinuxThreads library
http://pauillac.inria.fr/~xleroy/linuxthreads/

[5] RoboCup 小型機部門のルールの WWW
http://parrotfish.coral.cs.cmu.edu/robocup-small/rules/

[6] 和田 正義, 森 俊二, “ホロノミック全方向移動ロボットの開発”, 日本ロボット学会誌, Vol.15, No.8, pp.1139-1146, 1997.

[7] サッカーサーバの WWW,
http://ci.etl.go.jp/~noda/soccer/server/index.html

KU-Boxes2000 における画像処理と旋回性能の改良

Improvements in Image Processing of Off-board System and Turning Ability of Robots in Team KU-Boxes 2000

影山茂，三吉孝則，飯土井修一，小末将吾，五十嵐治一
Shigeru Kageyama, Takanori Miyoshi, Shuichi Iidoi,
Shogo Kosue, Harukazu Igarashi

近畿大学工学部（広島県東広島市）
School of Engineering, Kinki University
igarashi@info.hiro.kindai.ac.jp

Abstract

This paper describes improvements to our robot system developed under the JP/S-II Project. There are three major improvements. First, The image processing time was shortened by prefetching the next frame image during processing the current image and by restricting the processing area. Our experiment showed that the processing speed for recognizing the ball was increased to , at least, 10 fps, which is doubly faster than that before the improvement. Second, we corrected distorted images caused by a wide-angle lens of our video camera. In our experiment, the maximum recognition error was reduced from 109mm to 11mm. Third, we built a 16bit software counter in the on-board CPU program to control the pulse frequency for motor drivers instead of a hardware 8bit counter in CTC. That made it possible for our robots to run at slower speed and make more sharply turns.

1 はじめに

動的環境におけるマルチエージェントシステムの標準問題としてロボットによるサッカー競技（RoboCup）が取り上げられている．我々は，'97年9月から実機小型部門用の共通プラットホーム作成を目的とする JP/S-Ⅱプロジェクトを立ち上げている[1-5]．このプロジェクトについては，すでに，'00 年 3 月に東京で開催された RoboCup Spring Camp Symposium 2000 において，一応の総括を行った[6]．今回は，そこで指摘した本プロジェクトの問題点の改良方法について述べる．

2 システム構成の概要

本プロジェクトのシステムは，ロボット本体(on-board システム)とロボットを制御するシステム(off-board システム)とからなっている．システ

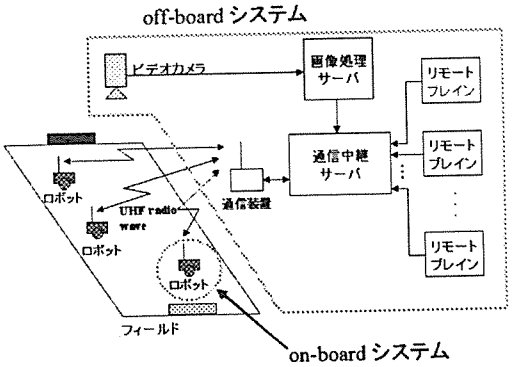


図1 システムの全体構成

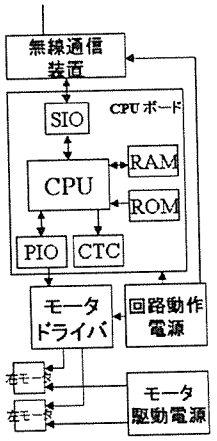


図2 on-boardシステムの構成

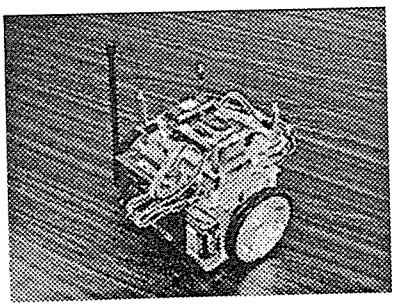


図3 ロボットの外観

ムの全体構成を図1に示す．また，on-board システムの構成を図2に，ロボット本体の外観を図3に示す．図3でロボット本体の大きさは約W150×D100×H120mm（アンテナを除く）である．

3 現行システムの問題点

本プロジェクトにおいて作成した現行システムには以下の問題点がある．

(1) 画像処理速度が遅く，5 flame／sec という値にとどまっている．
(2) 広角レンズを用いた場合，画像の歪みが大きい．
(3) 両輪走行時にロボットの小回りが効かない．
(4) ロボット自体のスピードが遅い．

今回は(1),(2),(3)についての改良を行った．以下，4章では(1)の画像処理速度の向上について，5章では(2)の画像歪みの補正方法について，6章では(3)の旋回性能向上のための改良点について述べる．

4 画像処理の高速化

4.1 画像処理アルゴリズム

ボールとロボットの認識のために，本システムでは以下の手順で画像処理を行っている[3]．

1)画像処理領域の設定
2)色の抽出，8近傍膨張，4近傍収縮，ラベリング，降順ソート，座標変換
3)ロボット識別用マーカーの認識

上記2)の色の抽出から降順ソートまでの処理にはホスト PC 内蔵の画像処理専用ボード（日立 IP5005）を用いている．

4.2 ウィンドウ設定とプリフェッチ機能

従来，本システムではボールやロボットの位置を抽出する際，画面全体を処理していた．本研究では局所的な画像処理ウィンドウ（中心の位置は対象物体の速度により予測する）を設定し，処理領域を制限することによりボールに関する画像処理の高速化を図った．以下に手順を示す．

1)ウインドウの中で 4.1 の 2)の色抽出から座標変換までを行う．
2)ウィンドウの中心を1フレーム前のボールの重心位置に移動させる．
3)上記 1)と 2)とを繰り返す．

また，4.1 の 1)～3)の処理と並行して画像を取り込むことで，全体の処理時間を減少させた（画像処理ボードのプリフェッチ機能を利用）．

4.3 画像処理時間の測定

ウィンドウの設定とプリフェッチ機能との性能を評価するために，画像処理の時間を測定した．測定結果を表1に示す．測定時間は 10ms 単位で，それ以下の値は切り捨てられている．表1の実験では，画像の取り込み要求を 100ms 間隔に固定してある．また，認識対象はボールだけである．

画像取り込み時には NTSC 信号(30fps)と同期を取る必要があり，33－66ms の待ち時間が必要である．実験1の結果からわかるように，この待ち時間を引くと，ボールの画像認識に要する時間は約 20－30ms である．実際，実験2でプリフェッチ機能を用いて，画像の取り込みと認識処理とを並列実行すると，この待ち時間は不要となり，認識処理だけに要する処理時間が20－30ms 程度と観測され，これは実験1の結果と一致する．

さらに，実験3の結果(30-60ms)から，待ち時間を引くと，ウインドウの設定時の認識処理時間は，10ms 以下と推定されるが，これは実験4でさらにプリフェッチ機能を用いて待ち時間をゼロにした場合の結果（処理時間が0と表示された）と一致している．したがって，画像取り込み間隔が 100ms 間隔であれば，プリフェッチ機能により待ち時間がゼロとなり，ウインドウの設定により認識処理時間が10ms 以下に短縮されている．これは，100ms ごとに1フレーム処理する（すなわち，10fps の処理速度）のは全く問題がないことを表している．

表1　ボールに関する画像処理時間

実験NO.	1	2	3	4
プリフェッチ機能	×	○	×	○
ウィンドウの設定	×	×	○	○
1回の処理時間* [ms]	50-90	20	30-60	0

* 測定は 10ms 単位で切り捨て値．

5 画像歪みの補正

5.1 画像歪み

現在の我々の開発環境では，部屋の天井が低く（約 2m60cm），天井に設置してあるビデオカメラとロボットフィールドとの距離が近い．そのため，フィールド全体の画像を一台のカメラで取り込む

ためには広角レンズの使用が不可欠である．しかし，広角レンズを使用すると取り込んだ画像が樽型に大きくひずんでしまう（画像歪み）[6][7].

これまでの予備実験では，ロボットの位置については最大約 9cm，姿勢（方向）については最大約 14.3 度の誤差を観測している[6][7]．そこで，この画像歪みの問題を解決するために，次節の方法による画像補正を考案した．

5.2 画像補正のアルゴリズム

RoboCup 小型部門競技フィールドの 4 分の 1 の領域に，図 4 のように 30 個の参照点を設置する(図 4 では□で表示)．すなわち，フィールドの中央を原点とし，図 4 に示すように，長辺方向に X 軸を，短辺方向に Y 軸をとる．次に，15cm 間隔で X 軸に平行な直線を 6 本引き，それらを $L_i(i=1,...6)$とし，30cm 間隔で Y 軸に平行な直線を 5 本引き，それらを $M_j(j=1,...5)$とする．直線 L_i と直線 M_j との交点を $r_0(i,j)=(x_0(i,j),y_0(i,j))$で表し，画像補正のための参照点と称する．各参照点には，4×4cm のオレンジ色のマーカーを置く．

次に，画像補正を以下の手順で行う．

1)画像処理により参照点の位置認識を行う．参照点 $r_0(i,j)$の認識結果を $r'_0(i,j)=(x'_0(i,j),y'_0(i,j))$で表す．

2)各 L_i 上の参照点（5 個）の認識結果について，横軸に X 方向の測定値$\{x'_0(i,j)\}(j=1,...,5)$を，縦軸にその認識誤差$\{x'_0(i,j)-x_0(i,j)\}$をとり，この 5 点を近似する 3 次関数 $f_{xi}(x')$を最小 2 乗法により作成する．この関数を補正関数と称する．3 次関数で画像歪みは十分近似できる[8].

3)各 M_j 上の参照点(6 個)の認識結果についても，2)と同様に Y 方向の補正関数 $f_{yj}(y')$を作成する．

4)上記2)と3)とで求めた補正関数$\{f_{xi}(x')\}$ (i=1,...,6), $\{f_{yj}(y')\}$ (j=1,...,5)を用いて，任意の地点 r=(x,y)の認識結果 r'=(x',y')の画像補正を行い，推定値 r''=(x'',y'')を算出する．

上記のステップ 4)での推定値の算出法としては，種々の方法が考えられる．実時間性を考慮した簡便な計算法が望ましい．

5.3 画像補正の実験と結果

前節で述べた画像歪みの補正方法の検証実験を行った．検証用の測定点として，図 5 や図 6 に示すように，参照点を配置した長方格子の裏格子点上に測定点（計 20 個，■で表示）を設定した．測定点上には，4×4cm の青色のマーカーを置いた．

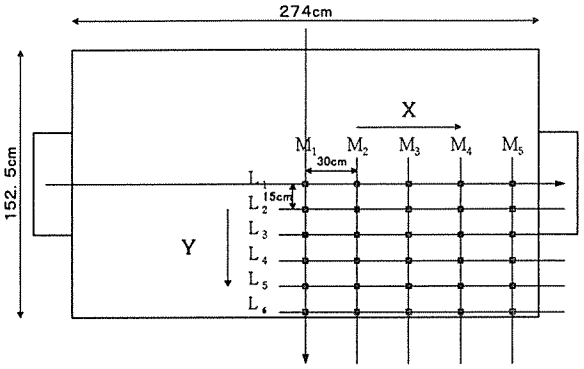


図 4　参照点の位置．記号□で示されている．

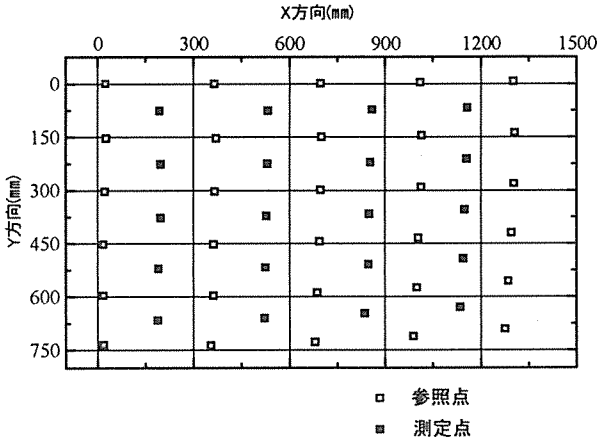


図 5　補正前の認識結果．

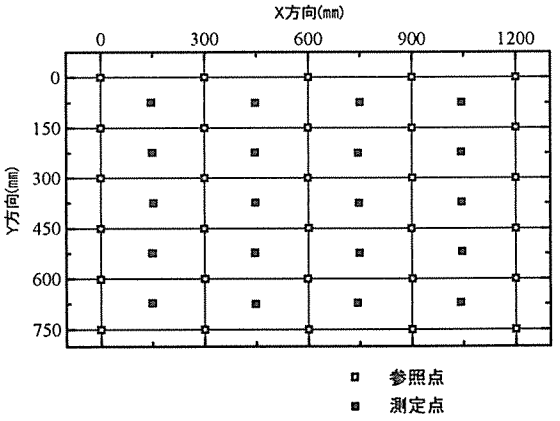


図 6　補正後の推定結果

最初に，参照点上に置かれた位置認識を 10 回行い，その後，測定点の位置認識を 1 回行った．認識結果を図 5 に示す．ただし，参照点に関しては，10 回の測定値の平均値を計算し，図 5 に示してある．補正方法のステップ 1)での $r'_0(i,j)$としては，この平均値を用いた．また，ステップ 2)の補正関数の例として，直線 L_1 上（正確には X 軸上への射影上）での X 方向の補正関数 $f_{x1}(x')$の結果を図 7 に示す．

今回の実験では，ステップ 4)の推定値の算出法

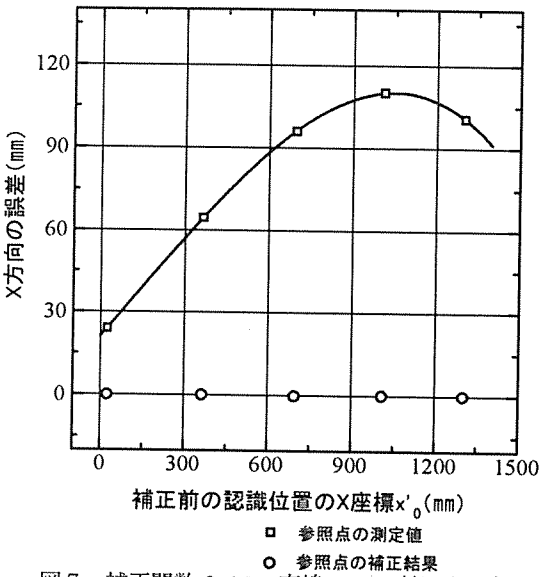


図7　補正関数 $f_{x,1}(x')$．直線 L_1（X 軸）上の参照点に関する X 方向の補正関数．

として，参照点の推定値を $r''_0(i,j)=(x''_0(i,j),y''_0(i,j))$，測定点の推定値を $r''=(x'',y'')$ とおくと，

$$x''_0(i,j)=x'_0(i,j)-f_{x,i}(x'_0(i,j)) \quad (1)$$

$$y''_0(i,j)=y'_0(i,j)-f_{y,j}(y'_0(i,j)) \quad (2)$$

$$x''=x'-\{f_{x,m}(x')-f_{x,m+1}(x')\}/2 \quad (3)$$

$$y''=y'-\{f_{y,n}(y')-f_{y,n+1}(y')\}/2 \quad (4)$$

により近似的に計算した．ただし，測定点(x',y')は，図5で直線 L_m,L_{m+1},M_n,M_{n+1} に囲まれている測定点とする．式(1)-(4)による推定値の算出法は，参照点と測定点とが元のどの格子点や裏格子点に属しているかという情報を利用しており，測定値のみから推定値を計算しているわけではなく，一般性に欠けているが，今回は画像補正アルゴリズムのおおよその精度を見積もるためにこの算出法を用いた．推定結果を，図6に示す．これらの推定値の誤差の最大値，平均値を参照点と測定点とに分けて表2に記した．この表から，測定点で見られた最大約 109mm，平均約 81mm の誤差が画像補正により最大約 11mm，平均約 4.8mm にまで減少していることがわかる．

表2　測定誤差の最大値と平均値

	参照点（計 30 ヶ所）		測定点（計 20 ヶ所）	
[mm]	最大値	平均値	最大値	平均値
補正前	114	77	109	81
補正後	2	0.5	11	4.8

6　On-board システムの改良

3章で述べた問題点のうち，(3)の両輪走行時に小回りが効かないのは，車輪の低速回転ができていないからである．本研究では，低速回転を可能とすることにより半径の小さい旋回を実現した．

6.1 速度制御のメカニズム

本システムにおけるロボットの速度制御には，カウンタ(CTC 内)と基準クロックとが用いられている．速度指定時にカウンタの値は対応する値(時間定数）に初期化され，基準クロックが出力されるたびに1ずつ減算される．カウンタがゼロになった時にモータドライバに CTC からパルス波(速度クロック)が出力され，このパルス波の周波数に比例した速度でステッピングモータが回転する．したがって，カウンタの時間定数が大きければ大きいほど，回転速度を小さくすることができる．

6.2 ソフトウェアによる速度制御

従来，本システムでは，基準クロックと CPU ボートに内蔵されていたCTC内の8bitカウンタとを用いてパルス波を発生していたが，基準クロックによるタイマ割り込みと ROM 上のプログラムとによりパルス波を PIO から出力するように改めた．その結果，発生するパルス周波数の幅を広げることが可能となった．今回は，16bit のサイズのカウンタを用意し，低速回転の実現を図った．改良前のパルス発生機構を図8に、改良後のそれを図9に示す．

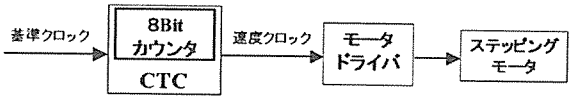


図8　改良前の速度制御

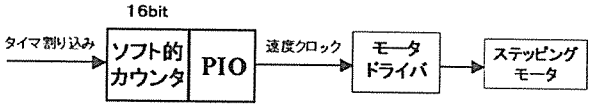


図9　改良後の速度制御

6.3 カウンタ用プログラム

カウンタ用プログラムは，アセンブラ言語で記述されている．メインプログラムからは一定時間毎に“タイマ割り込み“により呼び出され，ソフトウェア上のカウンタ値が1ずつ減算される．カウンタ値がゼロになった時に PIO からモータドラ

イバにパルス波が出力され，時間定数(図 10 の COUNTER 値)が再びロードされる．カウンタ用プログラムの処理の流れを図 10 に示す．

6.4 移動速度と旋回半径の理論式

本ロボットにおける移動速度は以下の式で理論的に表される．

$$V = \frac{9600\pi d}{n \cdot m} \qquad (5)$$

ただし，V は移動速度[mm/sec]，d は車輪の直径(=68[mm])，n はカウンタの時間定数(≦65535~16bit)，m はモータの 1 回転当たりのステップ数(=200)である．

また，車輪の移動速度 V_1(内輪)，V_2(外輪)と両輪走行時の旋回半径 R との間には，

$$R = \frac{L(V_1 + V_2)}{2(V_2 - V_1)} \qquad (6)$$

という関係がある．ただし，R は旋回半径[mm]，L は車輪間隔（=138[mm]）である．したがって，今回の改良により，これらの式から，理論的には，最低速度は約 0.16mm/sec，両輪走行時の最小旋回半径は約 69mm であると推定できる．これは片輪走行時の半径とほぼ同じであり，両輪走行時においても片輪走行時と同程度の小さな半径走行の旋回が可能であることがわかる．

6.5 走行実験

本ロボットにおける両輪走行時のこれまでの最低速度は 46mm/sec(理論値)で，両輪走行時の最小旋回半径は 120mm(理論値)であったが，走行実験を行った結果，それ以下の最低速度および旋回半径で走行することを確認した．

7 今後の課題

off-board システムについては，今回，ボールのみを処理対象とするときに，画像処理速度が 10 フレーム／秒以上は可能であることを確認したに過ぎない．これ以上の処理速度が，プリフェッチ機能と処理ウインドウの設定とでどこまで可能であるかを実験していきたい．また，現在はボール 1 個に対する画像処理の高速化にとどまっているが，これ以外に，敵・味方ロボットと味方の第二マーク（2 色）とに対する処理速度の高速化も行う必要がある．

on-board システムについては，問題点の(4)である並進移動速度の向上が必要である．現在，モータ駆動用バッテリーの電圧を上げることや，定電流方式によるステッピングモータの制御方式を検討中である．

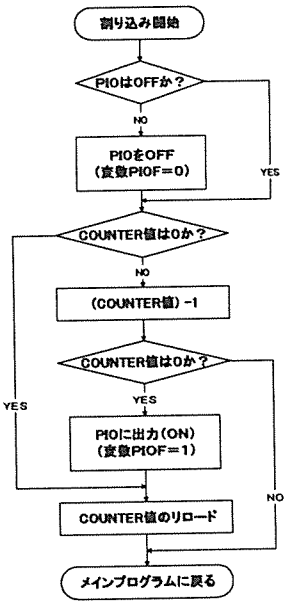


図 10 カウンタ用プログラムにおける処理の流れ

謝 辞

JP/S-II プロジェクトの遂行にあたり，画像処理アルゴリズムにご協力いただいた財団法人京都高度技術研究所，朝岡忠氏，徳島大学大学院博士課程在学中の田中一基氏，ロボット設計や画像処理など全般にわたって貴重なアドバイスをいただいた，本学部機械システム工学科 五百井清 助教授，RoboCup 小型部門参加者の方々，学会等で討論いただいた関係諸氏に感謝の意を表する．

なお，本研究は，日本学術振興会より科学研究費補助金（C2，課題番号 11680405）の助成を受けた．

参考文献

[1] 小末将吾，五十嵐治一，黒瀬能聿，" RoboCup 小型部門用ロボットシステムーJP/S-II グループの現状ー" ('98.3 ホットトピックスと並列人工知能研究会資料 SIG-HOT/PPAI-9702，pp.1-3)

[2] 小末将吾，五十嵐治一，黒瀬能聿，" RoboCup 小型部門用ロボットシステムの開発ーJP/S-II グループー" ('98.4 第 3 回 JSME ロボメカ・シンポジア論文集，pp.21-24)

[3] 田中一基，朝岡忠，小末将吾，五十嵐治一，

黒瀬能聿," RoboCup 小型部門用共通ロボットシステムにおける画像処理サーバ - JP/S-II プロジェクト - "('99.3 第4回AIチャレンジ研究会, 人工知能学会研究会資料 SIG-Challenge-9804, pp.1-4)

[4] 小末将吾, 五十嵐治一, 黒瀬能聿, 田中一基, 朝岡忠," RoboCup 小型部門用共通ロボットシステムの開発 - JP/S-II グループ - ", 第4回ロボティクスシンポジア予稿集 pp.87-92 ('99.3.30-31, 仙台)

[5] 小末将吾, 五十嵐治一, 三吉孝則, 飯土井修一, 黒瀬能聿, "JP/S-II ロボットシステムの現状と問題点", 第 6 回 AI チャレンジ研究会資料 SIG-Challenge-9906, pp.58-63('99.10 生駒市)

[6] 五十嵐治一, 小末将吾, 田中一基, 黒瀬能聿, 五百井清, "JP/S-II プロジェクトの総括と今後の展開", ロボカップスプリングシンポジウム 2000 ('00, 3 月, 東京), 人工知能学会研究会資料 SIG-HOT/PPAI-9909, pp.1-5.

[7] 小末将吾, "グローバルビジョンを用いた分散協調型移動ロボットシステム", '99 年度近畿大学大学院工業技術研究科修士論文, http://www.ip.info.hiro.kindai.ac.jp/kenkyu2/kenkyu2.html

[8] 左貝潤一," 光学の基礎", コロナ社, 1997 年, pp.87-95.

エージェントの意思決定のための情報量基準による観測戦略

Observation strategy for decision making of legged robot based on information criterion

光永 法明　浅田 稔　野原 達郎

Noriaki MITSUNAGA　Minoru ASADA　Tatsuro NOHARA

大阪大学大学院工学研究科

Graduate School of Engineering, Osaka University

{mitchy, asada, nohala}@er.ams.eng.osaka-u.ac.jp

Abstract

This paper proposes a method of constructing a decision tree and prediction trees of the landmarks that enable a robot with a limited visual angle to make decisions without self-localization in the environment. Since global positioning from the 3-D reconstruction of landmarks is generally time-consuming and prone to errors, the robot makes decisions depending on the appearance of landmarks. By using the decision and the prediction trees based on information criterion, the robot can achieve the task efficiently.

1　はじめに

移動ロボットは様々な場所へ移動し，場所に応じた行動をとることが期待される．そのため意思決定には，場所の認識が重要な役割を果たすと考えられる．場所の表現としては定量幾何学的あるいは位相幾何学表現 (トポロジー) が広く用いられているが，ロボットに利用可能なセンサの制約のため，複数回の観測の統合が必要となることが多い．

一般に観測が多いほど自己位置の確度は高くなるが，必要となる観測は少ないことが望ましい．そこで効率的な観測を行うための手法が提案されている．文ら [1] は障害物を回避するナビゲーションの問題で，高速に移動するため，場所の確認観測点 (view point) 計画を行っている．できるだけデッドレコニングを用い，視覚による場所の確認を減らしている．彼らの方法ではあらかじめ地図と経路がわかっている必要があり，反射的な行動などを行う場合には利用できない．Burgard et al. [2] は，Bayes 推定を利用した占有格子 を計算する自己位置推定の手法を基に，曖昧さの減少する行動をとる手法を提案している．この手法は自己位置の同定が目的であり，それ以上のタスクは考慮されていない．

これらの手法は場所を幾何学的表現に基づいて記述している．しかしながら，一般には，場所を区別することはできるが通常の位置の表現形態をとらない記述も可能である．視覚を持つ移動ロボットを考えた場合，視覚情報から幾何学的な位置を求めるには，計算コストだけでなく，ロボット自身と環境についてのモデルやパラメータといった知識が必要となる．一方，視覚情報をそのまま用いる場合には，そういった知識は必要ない．また観測を位置の計算とそれ以外で区別することなく扱うことができる．そういった特長から，観測の統合にリカレントニューラルネットワークによる予測を用いた研究 [3] [4] などが行われている．これらは受動的に観測を行っているが，Tani et al. [5] は観測の能動的な切り替えを行っている．この実験では場所によらず，ロボットが必要とする情報は壁と視覚目標物の見え方であり，意思決定に必要な情報は一定である．そのため，状況に応じて観測を効率的に行う手法とはなってない．

本研究では，移動ロボットの行動決定に関して，効率的な観測を行う行動決定法を提案する．情報量基準により生成した決定木による，視覚情報の予測と行動決定を行うことで，意思決定に必要な観測のみを行い，観測を効率化する．あらかじめデッドレコニングモデルを用意することは行わない．提案手法は，視野角の限られたセンサを持ち，意思決定に必要な情報の量が変化する環境内を行動する移動ロボットに特に有効である．

以下では，まずタスクと仮定について述べ，提案手法について説明する．そして実機を利用した実験結果を示し，最後にまとめと今後の課題を述べる．

2　タスクと仮定

ロボットや環境，与えられるデータ等に関して以下を仮定する．1) ロボットの視野角が限られており，行動決定

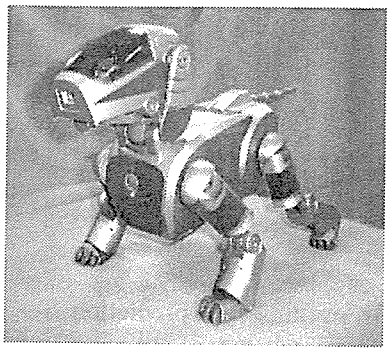


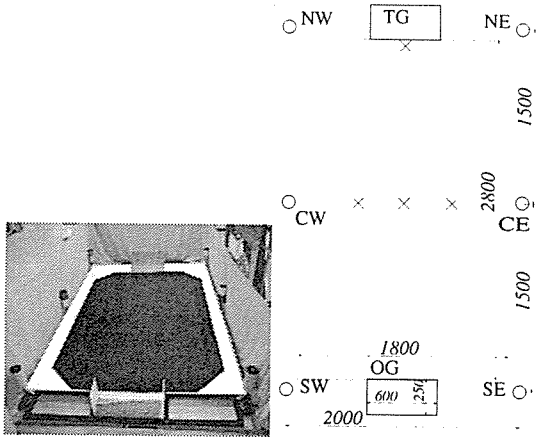
Figure 1: The SONY legged robot for RoboCup 99 SONY legged robot league.

に必要な情報が瞬時には得られない．2) ランドマークが配置されており，カメラを振ることで視野を拡大することにより，行動決定に十分な情報が得られる．3) 行動決定に必要な情報は一定ではなく，状況に応じて変化する．4) 決定木作成のため，行動，視覚情報は離散化されており，決定木を生成するために十分なデータが用意されている．

以下では決定木のうち，視覚情報の予測を行う木を予測木と呼ぶが，行動を決定する木と生成法は同じである．ロボットは予測木を用いた視覚情報の予測値，現在の視覚情報から行動決定木を使って，過去の経験から行動候補を決定する．複数の行動候補から一つに絞りきれない場合には視覚情報が不十分であるとし，カメラを振る．各木のノードは論理センサを，ノードをつなぐ枝は論理センサの値を示し，葉は枝をたどったセンサ値の場合の行動あるいは予測される論理センサの値である．本研究の実験では，ロボットとして歩行移動ロボットを利用し，決定木生成のためのデータを用意するため教示を用いた．また説明を簡単にするため，以下の決定木生成の説明ではセンサ値としてランドマークの方位のみを扱うがそれ以外も同様に扱うことができる．

ロボットとしては，RoboCup 99 SONY 脚式ロボットリーグのロボット (Fig. 1) を用いた．カメラの画角は横 53 度，縦 41 度，画素数はそれぞれ 88, 59 である．脚は各 3 自由度，首は 3 自由度 (パン，チルト，視線周り) あるが，ランドマークを見る際には脚の角度と，首のチルト，視線周りの角度を固定し，パン軸のみを利用した．パン軸はロボット正面に対して，-90 度から 90 度が可動範囲である．

環境を Fig.2 に示す．ランドマークは 8 個あり，ボールが一つある．それぞれ，敵ゴール (TG), 自陣ゴール (OG), 北西 (NW), 北東 (NE), 中央西 (CW), 中央東 (CE), 南西 (SW), 南東 (SE) ポールとする．すべてのランドマークとボールは色により識別される．ロボットがボールを TG に入れることをタスクとする．タスクの実現には場所に応じたボールへの回り込みなどが必要となる．



(a) Photo of the field. (b) Size of the field

Figure 2: Experimental field (same as the one for RoboCup SONY legged league). Cross marks are for the first experiment.

3 手法

3.1 決定木，予測木の生成

行動の種類を r，ランドマークの方位の分割数 (見えない場合を含む) を q，ランドマークの種類の数を m，トレーニングデータの数を n とする．まず各行動 $k = 1, ..., r$ の生起確率 p_k を求める．行動 k をとった回数を n_k とすると，

$$p_k = \frac{n_k}{n} \tag{1}$$

このときの p の情報量 I_0 は，

$$I_0 = -\sum_k p_k \log_2 p_k \tag{2}$$

である．次に，ランドマークあるいは視覚目標物の見え方がわかった場合の事後生起確率を求める．ランドマーク i が j に見えたときに行動 k をとった回数を n_{ijk} とすると，

$$p_{ijk} = \frac{n_{ijk}}{\sum_k n_{ijk}} \tag{3}$$

これらを知ったときの，情報量期待値を計算すると，

$$I_i = -\sum_j \{ \frac{\sum_k n_{ijk}}{\sum_j \sum_k n_{ijk}} \sum_k (p_{ijk} \log_2 p_{ijk}) \} \tag{4}$$

となる．ここで I_0 より情報量が最も減少するランドマークから順に木の上位に置き木を生成する．同じ見え方で異なる行動をとったトレーニングデータに関しては，それぞれの行動をとった確率を計算し別の葉として生成する．ランドマークの見え方予測木も同様に生成する．ここで用いている情報量基準は ID3 [6] と同じであり，ID3 での分類クラス，属性，属性値はそれぞれ，行動，ランドマー

Table 1: An example of teaching data.

Landmark 1	Landmark 2	Landmark 3	action
1	1	1	1
2	1	1	2
3	2	1	2
1	3	1	3
1	2	2	3

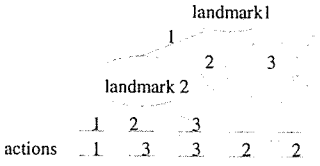


Figure 3: An example of an action decision tree.

ク，ランドマークの見え方に相当する．ノードの順位が固定であるので，計算をやり直すID3とは生成される木が多少異なる．

Table 1の教示データが得られた場合の計算は次のようになる．まず，$p_1 = 0.2$, $p_2 = 0.4$, $p_3 = 0.4$ より，$I_0 = 1.52$ となる．p_{ijk} を計算し，情報量期待値を計算すると $I_1 = 0.551$, $I_2 = 0.8$, $I_3 = 1.2$ となるので，決定木でのランドマークの順位を 1,2,3 とし，木を生成するとFig.3となる．

3.2 確率分布の計算

時刻 t での各ランドマーク i がどの方向 j に観測されるかの確率を $p^L_{ij}(t)$ $(i = 1, ..., m, j = 1, ..., q)$ とし，時刻 t で行動 k をとった確率を $p^a_k(t)$ $(k = 1, ..., r)$ とする．過去の経験による時刻 t でとるべき行動が k である確率を $\hat{p}^a_k(t)$ $(k = 1, ..., r)$ とする．

確率分布の計算は次のように行う．現在画像上で観測されているランドマーク i については，その方向 J の確率を $p^L_{iJ}(t) = 1$ とし，それ以外を $p^L_{ij}(t) = 0 (j \neq J)$ とする．1時刻前の行動については，実際にとった行動 K を $p^a_K = 1$ とし，それ以外を $p^a_k = 0 (k \neq K)$ とする．観測されていないランドマークについては，1時刻前のそれぞれの確率分布 $p^L_{ij}(t-1)$ から予測木を使って計算する．カメラを振ることによりランドマーク探索した場合に見えない場合にのみ，見えていない確率を 1 とし，残りの方位を 0 とする．

予測木は次のように用いる．ランドマーク i の木の根から葉までたどると，時刻 $(t-1)$ の各ランドマークの見え方と行動の論理積を満たした場合の，ランドマーク i の時刻 t での見え方が記述されている．そこですべての葉について，論理積をその見え方であった(行動をとった)確率の積に置き換え，その葉に到達する確率を計算を行う．複数の葉に同じ見え方が現れるので，それらの和を時刻

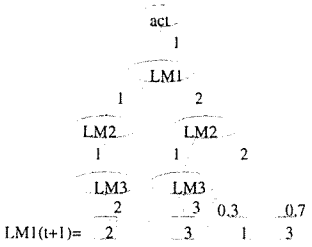


Figure 4: An example of a prediction tree for landmark 1 (LM means landmark) .

t での，その見え方をとる確率と見なす．例えば，ランドマーク1の予測木がFig.4であれば，

$$
\begin{aligned}
p^L_{11}(t) &= p^a_1(t-1)p^L_{12}(t-1)p^L_{22}(t-1) \times 0.3 \\
p^L_{12}(t) &= p^a_1(t-1)p^L_{11}(t-1)p^L_{21}(t-1)p^L_{32}(t-1) \\
p^L_{13}(t) &= p^a_1(t-1)p^L_{12}(t-1)p^L_{21}(t-1)p^L_{33}(t-1) \\
&\quad + p^a_1(t-1)p^L_{12}(t-1)p^L_{22}(t-1) \times 0.7 \quad (5)
\end{aligned}
$$

となる．

これらから得られた時刻 t でのランドマークの見え方の確率分布 $p^L_{ij}(t)$ を用いて，行動決定木を同様にたどり，行動の確率 $\hat{p}^a_k(t)$ を計算する．

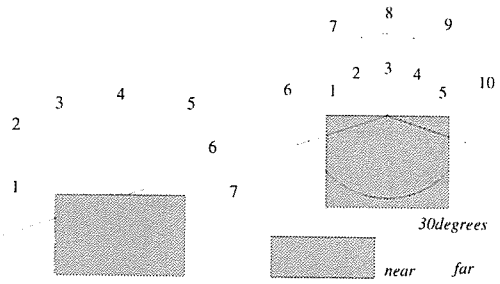
3.3 行動決定

行動の確率分布の計算後，行動を決定する．確率分布の中で，ある行動の確率が，特に高い山になっていればその行動をとればよい．そうでなければ，行動確率の山が十分に高くなるまで，行動決定木の上から順にランドマークの確率分布を調べ，山が低いランドマークについて再観測を繰り返す．すなわち情報量基準で順に再観測の必要なランドマークを調べる．また再観測の際，ランドマークの確率分布の山(方位)を優先的に調べることで，再観測時間を軽減できると期待される．

4 実験結果

各ランドマークの見え方は，ロボットに対して前方を 0 度として，$(, -65)$, $[-65, -40)$, $[-40, -15)$, $[-15, 15)$, $[15, 40)$, $[40, 65)$, $[65,)$ の 7 方位に分割し，見えない場合を含めて見え方は8通りとした(Fig. 5)．ボールの見え方は，$(, -45)$, $[-45, -12)$, $[-12, 12)$, $[12, 45)$, $[45,)$ の 5 方位に分割し，さらにロボットから遠い近いの2通り(水平に対して下向30度で分割)に分け，見えない場合を含めて見え方は11通りとした．ボールは，ロボットの行動と一時刻前のボールの見え方のみに依存する特殊なランドマークとして扱い予測木を生成した．

行動決定木や各予測木をたどって確率分布を計算する際，各木を生成する時に含まれなかった見え方により，確率分布の合計 $\sum_{i=1}^{N} p_i$ が1にならない場合がある．ここでは，



(a) for landmarks (b) for the ball

Figure 5: Quantization for landmarks and the ball.

合計が1となるよう $(1-\sum_{i=1}^{N} p_i)/N$ を $p_i(i=1,...,N)$ に加えた．N は分布の大きさである．

生成された木を使って行動を決定する (以下教示再生と呼ぶ) 際には，行動の確率分布の最大値が 0.6 以上であればその行動をとり，そうでなければ首を振って見えていないランドマークとボールを再探索することとした．教示中，教示再生時ともにランドマーク再探索時以外は，ボールを追跡視あるいは探索するようにした．また，行動確率分布の山が平らなときには，ランドマークの見え方分布の山も平らであったため，再探索の際には予測した確率分布の山を利用しなかった．

4.1 実験 1

まず Fig.2 のフィールドにおいて，ゴール前 (図の×印) にボールをおき，フィールドの中央3点 (フィールド中央の×印) から開始してボールをゴールにいれるタスクを行った．行動は我々の開発した歩行プログラムを用い，前進，左右大回りの3つとした．行動の継続時間は，2.4 秒とした．これは 4 歩行周期で，直進の場合約 0.45[m] の移動となり，ほぼ一度の行動で見え方が変化する．教示は中央3点から各5回ずつ行い，80 のデータを得た．このデータから生成した決定木の一部を Fig.4.1 に示す．ボールが左前方 (2 の方向) に見えた場合には，回り込みのため敵ゴールの方向 (3,4,6) に応じてとるべき行動が変化することが分かる．決定木と予測木の大きさと情報量による順序を Table 2 と 3 に示す．Table 2 は行動決定木と，予測木の大きさと深さを示し，表中 # of leaves は葉の数を表し，min dep., mean dep., max dep. は，それぞれ木の最小深さ，平均深さ，最大深さを表す．Table 3 は行動決定木と，それぞれの予測木の情報量による根からの視覚情報の順位を表し，左が最も根に近く右が葉に近い．表中 ball, act は一時刻前のボールの見え方，行動を，TG, OG, NW, NE, CW, CE, SW, SE はそれぞれのランドマークの一時刻前の見え方である．

次に教示再生を行った場合の予測と実際にとった行動例

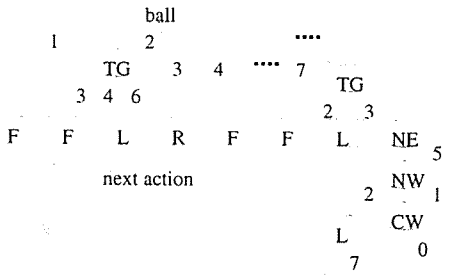


Figure 6: Part of the action decision tree (experiment 1). F, L, and R mean forward, left forward, and right forward respectively.

Table 2: Depth and size of the action decision tree and the prediction trees (experiment 1).

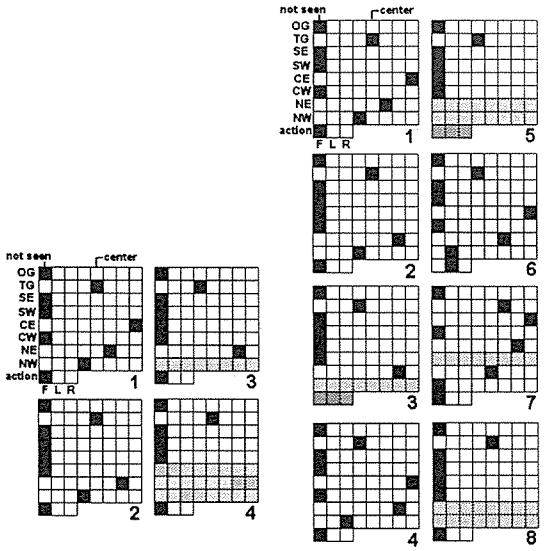
tree for	# of leaves	min dep.	mean dep.	max dep.
action	43	1	4.91	8
ball	52	2	2	2
OG	13	1	4.23	8
TG	44	1	5.39	8
SE	6	1	2	3
SW	1	0	0	0
CE	28	2	4.69	8
CW	11	1	3.91	8
NE	51	1	5.96	8
NW	54	2	5.91	8

を示す．まずフィールド中心から開始した場合に，ロボットは，前進→前進→前進→前進という行動をとった．このときボールと TG は常に見えており，それ以外のランドマークの予測木をつかった確率分布と過去にとった行動の確率分布は Fig.7(a) のようになった．OG, TG, SE, SW, CE, CW, NE, NW はそれぞれのランドマークの見え方の予測あるいは観測による確率分布であり，action は行動決定木による確率分布である．

同じフィールド中心から開始した場合でも，別の行動をとることもあった．これは，初期のロボットからのランドマークの見え方や歩行の結果が必ずしも同一ではないからである．この例では，初期のランドマークの見え方は一致しているが，歩行の結果が一致しなかった．ロボットは，前進→前進→ランドマーク確認→前進→ランドマーク

Table 3: The order of information for the action decision tree and prediction trees (experiment 1).

tree for	1	2	3	4	5	6	7	8
action	ball	TG	NE	NW	CW	CE	OG	SE
ball	ball	act						
OG	act	NE	TG	NW	CW	CE	OG	SE
TG	TG	act	NE	NW	CE	OG	CW	SE
SE	act	CE	NE	OG	NW	TG	CW	SE
SW	-							
CE	act	NE	TG	CE	NW	CW	OG	SE
CW	TG	act	NE	NW	CE	CW	OG	SE
NE	NE	act	NW	TG	CE	CW	OG	SE
NW	act	NE	TG	NW	CE	OG	SE	CW



(a) experiment 1-1 (b) experiment 1-2

Figure 7: Probability distribution in experiment 1 (The gray scale of each box indicates the probability, black 1 and white 0) .

確認→左回転→前進→前進という行動をとった．ボールとTGは常に見えており，それ以外のランドマークの予測木をつかった確率分布と行動決定木の分布はFig.7(b) のようになった．

次にフィールドの右側から開始した場合の行動例を示す．ロボットは，左回転→ランドマーク確認→前進→前進→左回転という行動をとった．ボールとTGは常に見えており，それ以外のランドマークの予測木をつかった確率分布と行動決定木の分布はFig.8のようになった．

教示再生中の再観測回数をTable 4に示す．左から順に試行開始場所，試行回数，合計行動回数，合計再観測回数，再観測率である．再観測回数が半分程度に減少していることが分かる．

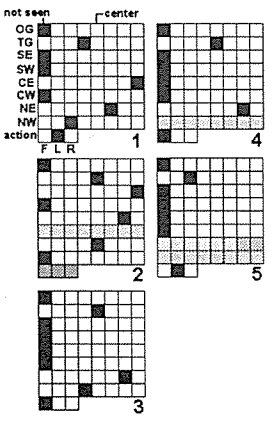


Figure 8: Probability distribution in experiment 1-3.

Table 4: The number of needed observation (experiment 1).

begin from	# of trials	# of total steps	# of re-observations	rate of re-observation
center	12	35	18	.51
left	12	31	15	.48
right	16	64	38	.59

Table 5: Depth and size of the action decision tree and the prediction trees (experiment 2).

tree for	# of leaves	min dep.	mean dep.	max dep.
action	586	2	5.89	9
ball	403	2	2	2
OG	958	2	7.58	9
TG	1050	2	7.67	9
SE	845	2	7.35	9
SW	901	2	7.41	9
CE	901	2	7.13	9
CW	873	2	7.37	9
NE	1031	2	7.60	9
NW	980	2	7.55	9

4.2 実験 2

同じフィールド上でRoboCup 99の試合を考慮した教示を行った．教示の負担を減らすため，行動は，前進，左右大回り，左右小回り，ボール追跡の6つとした．行動の継続時間は，前の実験と同じ2.4秒とした．この教示により，1364のデータを得た．このうち不適切な教示を除き，856を行動決定木の生成に，1364すべてを予測木の生成に用いた．各木の大きさ，情報量による順序をTable 5と6に示す．

このデータを実際に，RoboCup 99で用いたところ，ロボットは教示者の期待した行動を行った．しかしながらランドマークの確認が頻繁に行われた．これはランドマークの予測木生成に用いたデータが少なかったためと思われる．

Table 6: The order of information for the action decision tree and the prediction trees (experiment 2).

tree for	1	2	3	4	5	6	7	8	9
action	ball	TG	OG	SW	SE	NW	NE	CE	CW
ball	ball	act							
OG	OG	SE	SW	TG	NW	CW	NE	CE	act
TG	TG	OG	SE	SW	NW	NE	CW	CE	act
SE	SE	OG	TG	SW	CE	NE	NW	CW	act
SW	SW	OG	CW	SE	TG	NW	NE	CE	act
CE	CE	SE	OG	TG	NE	SW	NW	act	CW
CW	CW	SW	OG	TG	NW	SE	NE	CE	act
NE	TG	NE	OG	SE	CE	NW	SW	CW	act
NW	NW	TG	OG	SW	CW	SE	NE	CE	act

Table 7: Comparison of the numbers of average observed directions (experiment 3).

fixed	5.28	5.18	4.80	4.75	5.38	5.22	5.36	5.21
info.	4.64	4.43	4.41	4.42	4.49	4.58	4.53	4.45

4.3 実験 3

次に，各ランドマークを観測した場合の情報量ではなく，ある方位を観測した場合の情報量を計算し，その情報量が最も大きい方位を順に観測した実験結果を示す．ここではボールに関しても Fig.5(a) の方位に分割し，遠近は無視した．データとしては実験 2 と同じものを用いシミュレーション上で教示を行い，予測木は生成しなかった．教示中にカメラを全体にパンするのではなく，過去の教示データから行動が決定出来るまで各方位を順に観測し，その行動が現在教示された行動と一致する場合には，それ以上の観測を行わないことにした．一致しない場合には，決められた順に各方位の観測を増加するものとした．教示は用意したデータを順に与え，全てのデータを与えた時点で観測方位が増加しなくなるまで，同じ教示を繰り返した．

教示終了後に行動を決定するのに要した観測した方位数は，用意した方位の観測順に依存した．8 通りの観測順に対する，教示データについての平均観測方位数を Table 7 に示す．fixed はそれぞれに予め決めた順に観測した場合で，info. は情報量による順に観測した場合の平均観測方位数である．いずれの場合にも，観測方位数は減少するが，最初に与えた観測順に依存することが分かる．

5 まとめと今後の課題

実験 1 を見ると，行動確率の分布はほぼ 1 の高い山があるか，ほとんど平らな確率分布かのいずれかとなっている．行動確率分布が平らな場合，情報量の高いランドマークの確率分布も平らになっている．このためランドマークの山を優先的に見ることはできない．これは，予測木の作成に用いたトレーニングデータが少なかったためと思われる．そのため，行動確率が低い際にはランドマークを見直すという戦略は正しかったと思われる．

実験 1, 実験 2 のランドマーク予測木を比較すると，実験 1 では行動が上位に来ており，実験 2 では行動が下位になっている．実験 2 の結果のように場所に依存して見え方の異なるランドマークの予測は，ランドマークにより大まかな位置が判明してから行った方がよい．実験 1 でこれが見られないのは，教示データが少ないためであると思われる．

実験 3 において，観測対象ではなく観測方位について行動決定に関する情報量を計算したが，平均観測方位が比較的多かった．ところが，ある方位の観測により何が観測されるかを予測する予測木を生成すると，観測方位の分割を多くするにつれて予測木の数が増加するという問題がある．また実験 1, 2 を見ると観測方向の予測はさほどよくない．こういったことから，予測に関しては提案した予測木より精度のよい手法を用い，観測対象について行動決定に関する情報量を計算し，行動決定木を生成するのがよいと考える．

ここでは，決定木の圧縮はさほど行っていない．圧縮を行うとトレーニングデータにない状況への対応が期待される一方，ランドマークの再確認を行うべき状況において確認を行わないことが増加すると思われる．また各ランドマーク，ボールの見え方の離散化方法をここではあらかじめ決定している．しかし用意した離散化法が最適であるとはいえない．行動決定木の生成に C4.5[6] などの連続値を扱え情報量基準を用いた離散化を同時に行える手法を用いることで，自律的な離散化を行える可能性がある．

この手法は，to look or to move また，what to look の解にはなると思われるが，行動の切り方 (ここでは 2.4 秒で固定)， when to look の問題は残っている．さらに，一度見渡せば場所に関して十分な情報を得られると仮定しているが，仮定の成立しない場合への対処が必要である．また，行動中の注視対象をボールに限ったが，これも情報量基準で選択することが望ましい．ここでの実験では行動の確率分布から行動を決定する際の閾値は実験的に決定したが，確率分布からの行動決定法とともに閾値の適切な決定法は今後の課題である．

参考文献

[1] 文仁赫, 三浦純, 白井良明. 不確かさを考慮した観測位置と移動のオンライン計画手法. 日本ロボット学会誌, Vol. 17, No. 8, pp. 1107–1113, 1999.

[2] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann, San Mateo, CA, 1997.

[3] Jun Tani. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. on System, Man and Cybernetics Part B (Special Issue on Robot Learning)*, Vol. 26, No. 3, pp. 421–436, 1996.

[4] 鈴木育男, 横井浩史, 嘉数侑昇. 連想記憶を用いたロボットナビゲーションシステムに関する基礎研究. 第 17 回日本ロボット学会学術講演会講演論文集, pp. 613–614. 日本ロボット学会, 1999.

[5] Jun Tani, Jun Yamamoto, and Hiro Nishi. Dynamical interactions between learing, visual attention, and behavior: An experiment with a vision-based mobile robot. In Phil Husbands and Inman Harvey, editors, *Fourth European Conference on Artificial Life*, pp. 309–317. The MIT Press, 1997.

[6] J. Ross Quinlan. *C4.5: PROGRAMS FOR MACHINE LEARNING*. Morgan Kaufmann Publishers, 1993.

〒162-0821 東京都新宿区津久戸町 4-7 OS ビル 402 号室 03-5261-3401 Fax: 03-5261-3402

(本研究会についてのお問い合わせは下記にお願いします.)

AIチャレンジ研究会
主 査
奥乃 博
東京理科大学 理工学部 情報科学科/
科学技術振興事業団 ERATO
北野共生システムプロジェクト
〒150-0001 東京都渋谷区神宮前 6-31-15
マンション31, 6A室
03-5468-1661 Fax: 03-5468-1664
okuno@nue.org

担 当 幹 事
浅田 稔
大阪大学大学院 工学研究科
知能・機能創成工学専攻 創発ロボット工学講座
〒565-0871 大阪府吹田市山田丘 2-1
06-879-7347 Fax: 06-879-7348
asada@ams.eng.osaka-u.ac.jp

Executive Committee
Chair
Hiroshi G. Okuno
Dept. of Information Science,
Science University of Tokyo/
Kitano Symbiotic Systems Project,
ERATO, JST
Manshon 31, Room 6A
6-31-15 Jingumae, Shibuya, Tokyo
150-0001 JAPAN

Secretary in Charge
Minoru Asada
Dept. of Adaptive Machine Systems
Graduate School of Engineering
Osaka University
2-1 Yamadagaoka, Suita,
Osaka 565-0871, JAPAN

SIG-AI-Challenges home page (WWW): `http://www.nue.org/SIG-Challenge/`