Article

Generation of Evaluation Function for Life-time Learning of An Intelligent Robot

Setsuo HASHIMOTO
Faculty of Economics, Kyoto Gakuen University
1-1 Ootani, Nanjyo, Sogabe-cho, Kameoka, Kyoto, 621-8551 JAPAN
setsuo-h@kyotogakuen.ac.jp

Fumio KOJIMA
Dept. of Mechanical and Systems Engineering, Kobe University
1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, JAPAN
kojima@cs.kobe-u.ac.jp

Naoyuki KUBOTA
Dept. of System Design, Tokyo Metropolitan University
1-1 Minami-Osawa, Hachioji, Tokyo 192-0397, JAPAN
kubota@comp.metro-u.ac.jp
&
"Interaction and Intelligence," PRESTO,
Japan Science and Technology Corporation (JST)

Abstract

This paper deals with a mobile robot with structured intelligence. The robot interacts with a dynamic environment. The evaluation criteria or functions are the strategy for the behavior acquisition. Generally, it is difficult for human operators to describe internal models of the robot because the organization of the robot is quite different from that of a human. In the optimization, the evaluation function is generally given by human operators beforehand. It is easy to give the evaluation functions if the environmental condition is easy and fixed. But the robot must interact with dynamic, uncertain and unknown environments or human operators. Therefore, the robot should generate the evaluation criteria by itself based on its embodiment. A human improves its behavior by using and changing its evaluation criteria as adaptive processes. The robot also has to acquire their evaluation criteria through life-time learning. Therefore, we apply genetic programming (GP) for generating evaluation functions. The result of computer simulation shows that GP can generate the evaluation function suitable to the facing environments, the given tasks, and the robot.

Keywords: Behavior learning, Evaluation function, Genetic programming, Evolutionary computation, Intelligent robot

1. Introduction

Computational intelligence including fuzzy, neural, and evolutionary computing has been applied to manufacturing systems and robotics. To realize high intelligence on robotics, the emerging synthesis of various techniques is required as a whole system. Neuro-fuzzy computing has been developed for overcoming their disadvantages [3-6]. In general, the neural network part is used for its learning and classifying, while the fuzzy logic part is used for inference. Evolutionary computation (EC) is used for optimizing the structure and parameters of NN and FS [3,4,6].

Especially, these kinds of intelligent techniques have been applied to realize the next generation of intelligent robotic systems. An intelligent robot perceives, and makes decision, and takes action in unknown and dynamic environments. Behavior-based robotics was proposed to avoid the frame problem [1]. The behavior-based robotics considers not only the robot its own but also the environmental condition. But the methodology has no internal models and only uses raw sensing data and simple reactive motions because it lays stress on the relationship between sensing data and reactive motions. Generally, the robot can not perform a given task effectively if it uses only reactive motion. The perceptual mechanism plays the important role to build intelligent robots, because the robots closely interact with their external environments. Furthermore, decision making depends on the quality of the perceptual information translated by its perception mechanism. We have proposed the concept of perception-based robotics that emphasizes the importance of the perception [18]. Zadeh also emphasizes the importance of perception-based computing [22]. The perception-based robotics has internal models to translate perceptual information. A perception translates raw sensing data as quantitative information into internal descriptions as qualitative information. The flexible translation of the perceptual information can be done by fusion or restraint of sensing data. And it enables the robot to reconstruct its search space. By using perceptual information for the learning or adaptation, the perception and the action of the robot can be acquired effectively.

It is difficult for a robot to generate strategies in unknown environment beforehand. Furthermore, the evaluation functions and criteria are generally given to robots by human operators. However, if the robot can generate evaluation functions suitable to the facing environment, the robot can generate strategies and acquire behaviors. By reconstructing the evaluation criteria, the robot can search the direction of problem solving. Therefore, the robot should perform lifetime learning of evaluation functions and behavioral rules at the same time. In this paper, we apply GP to generate evaluation functions for a robot. Furthermore, the robot uses the functions to evaluate behavioral rules written by fuzzy rules for a given task in an unknown environment. The optimization of behavioral rules is done by a genetic algorithm (GA).

This paper is organized as follows. Section 2 describes the proposed method including GP, GA, and fuzzy rules for the life-time learning of the mobile robot. Section 3 shows computer simulations. Section 4 concludes this paper and describes future works.

2. Life-time Learning of An Intelligent Robot

A human improves its behavior by using and changing its evaluation criteria as adaptive processes. And the interpretation of the perceptual information is also changing as time goes by. These processes are continuously done in the life-time. In many cases, the learning processes of a human are endless. the robot interacts with a dynamic environment or a human operator. Therefore, it is generally difficult to describe internal models of the robot from the viewpoint of the human operator. Internal models of the robot should be acquired by the robot itself. Consequently, the robot also should learn continuously in the life-time like a human.

As another approach, lifelong learning is proposed so far [22,23]. This approach lays stress on the shortening of learning process in other environment by the accumulation of behavioral rules. On the other hand, the life-time learning is the mechanism that the robot continuously learn in its life-time. The behavioral rules can be acquired if the robot has evaluation criteria which is suitable to facing environments or given tasks. Accordingly, The accumulation of the evaluation criteria is more important than the accumulation of the behavioral rules for the robot. The life-time learning and lifelong learning is different each other in this point.

Figure 1 shows collision avoiding and target tracing behaviors. These behaviors are generally evaluated by the moving distance and time, the degree of danger. Therefore, the optimization of robotic behaviors is defined as a multiobjective optimization problem. However, the importance of each evaluation factor depends on the facing environment. This indicates it is difficult to determine evaluation functions in an unknown environment beforehand. Therefore, evaluation functions should be also optimized through the acquisition of behaviors. Consequently, the robot generates evaluation functions for behaviors in order to reach the given target, and furthermore, the behaviors are optimized

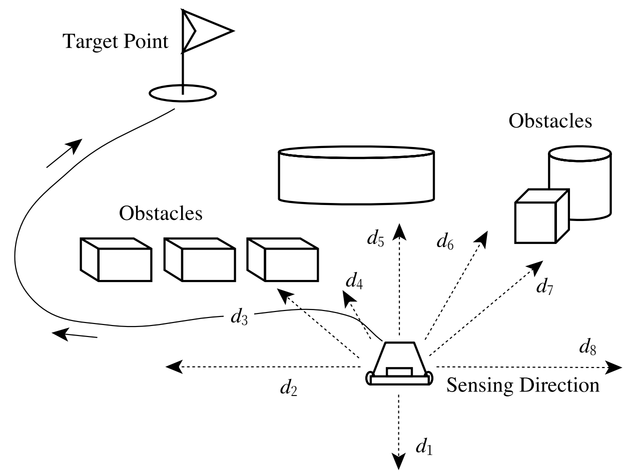


Fig. 1 Collision avoiding and target tracing behaviors of a mobile robot with 8 range

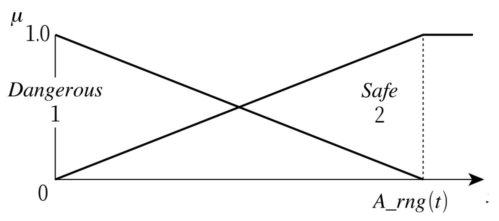


Fig. 2 The coding of membership functions corresponding to linguistic values

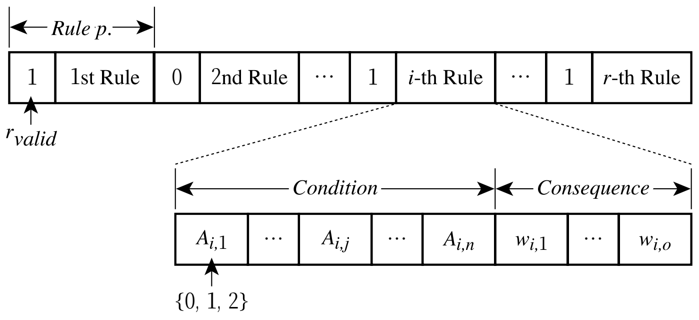


Fig. 3 A candidate solution representing fuzzy rules.

according to the generated evaluation functions. The robots can acquire its behavior if they have their evaluation criteria. What is important for the robots is not to accumulate behavior rules, but to accumulate the evaluation criteria. In this case, we only evaluate whether the robot reaches the target, or not (success or failure). Since the robot receives only the information of success and failure, this optimization can be regarded as reinforcement learning. In the following, this optimization process is divided into two phases of behavior acquisition (Phase A) and generation of evaluation function (Phase B) (see Fig.4).

2.1 Fuzzy Controller for Robotic Behaviors

Fuzzy theory provides us with the linguistic representation such as 'slow' and 'fast'. Fuzzy logic [5] expresses a degree of truth, which is represented as a grade of a membership function. FS implements mapping from its input space to output space by a number of fuzzy if-then rules.

In this paper, we use a simplified fuzzy inference method for decision making in the collision avoidance behavior. The input variables are the measured distance (see Fig.1), and the output variables are steering angle and speed of the mobile robot. A triangular membership function is generally described as,

$$\mu_{A_{i,j}}(x_j) = \begin{cases} 1 - \dfrac{|x_j - a_{i,j}|}{b_{i,j}} & |x_j - a_{i,j}| \\ 0 & otherwise \end{cases} \tag{1}$$

where $a_{i,j}$ and $b_{i,j}$ are the central value and the width of the membership function, $A_{i,j}$, respectively. Here we use two membership functions shown in Fig. 2. The activation degree (firing strength) of the i-th rule $(i=1,2,\ldots,r)$ is calculated by

$$\mu_i=\prod_{j=1}^{n}\mu_{A_{i,j}}(x_j) \tag{2}$$

Next, we obtain the j-th resulting output $(j=1,2,\ldots,o)$ by weighted average as follows,

$$z_j=\frac{\sum_{i=1}^{r}\mu_i\cdot w_{i,j}}{\sum_{i=1}^{r}\mu_i} \tag{3}$$

A fuzzy controller can not be tuned by delta rule, since the teaching data can not be generated by the robot itself. Therefore, this paper applies a steady-state genetic algorithm (SSGA) for optimization of fuzzy controllers.

2.2 SSGA for Fuzzy Controllers

EC is a field of simulating evolution on a computer, and the evolutionary process is based on iterative generation and alternation operating on a set of candidate solutions, which is called a population. All the population evolves toward better candidate solutions by selection operation and genetic operators such as crossover and mutation. It is experimentally known that the GA can obtain near or approximately optimal solutions with less computational cost.

The SSGA is used for optimization of the structure and parameters of fuzzy rules (Phase A) (Fig.4). The SSGA simulates the continuous model of the generation, which eliminates and generates a few individuals in a generation (iteration) [20]. A fuzzy controller for collision avoiding behavior is represented in Fig.3 where r_{valid} indicate the validity of the corresponding fuzzy rule, $A_{i,j}$ indicates the membership functions shown in Fig.2, and the invalidity of the i-th input of a fuzzy rule is represented as “0” (see Fig.3). Each consequence $(w_{i,j})$ is represented as real numbers. The SSGA eliminates the worst individual from the population and randomly reproduces an individual from the population. Next, the reproduced individual inherits partial genetic information from a randomly selected individual by a one-point crossover. Furthermore, a simple mutation is performed, which exchanges a gene with other genes in condition parts. Since the output parameters of each fuzzy rule (consequence parts) are described as real numbers, we use the following adaptive mutation,

$$w_{i,j}(t+1)=w_{i,j}(t)+\left(\alpha_j\cdot\frac{f-\min f}{\max f-\min f}+\beta_j\right)\cdot N(0,1) \tag{4}$$

where f is the fitness value of the i-th individual, $\max f$ and $\min f$ are the maximum and minimum of fitness values in the population, $\alpha_{i,j}$ and $\beta_{i,j}$ are the coefficient and

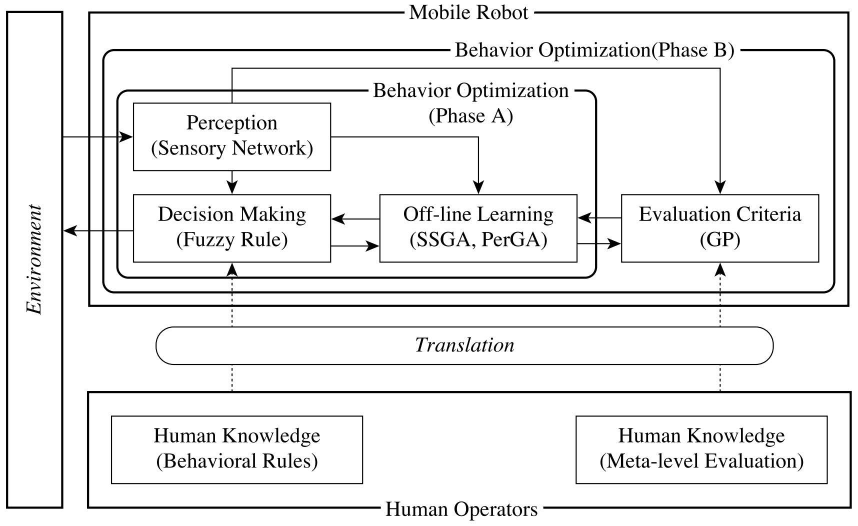

Mobile Robot
Behavior Optimization(Phase B)
Behavior Optimization (Phase A)
Perception (Sensory Network)
Decision Making (Fuzzy Rule)
Off-line Learning (SSGA, PerGA)
Evaluation Criteria (GP)
Environment
Translation
Human Knowledge (Behavioral Rules)
Human Knowledge (Meta-level Evaluation)
Human Operators

Fig. 4 The total architecture of optimization for the behavior and evaluation (fitness) functions

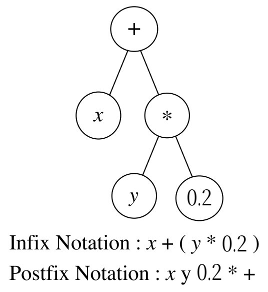

+
x
*
y
0.2
Infix Notation : x + (y * 0.2)
Postfix Notation : x y 0.2 * +

Fig.5 A syntax tree based on postfix notation

Table 1 Genotype and operators

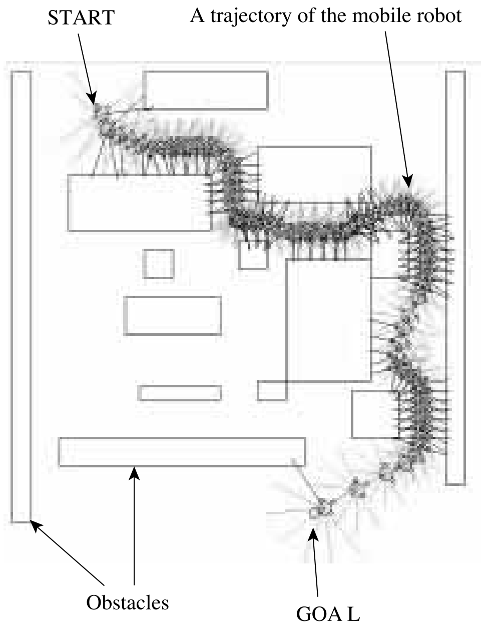
Genotype	Operators	
1	+	Binary Operators
2	−	
3	*	
4	max	
5	min	

offset, respectively. In the adaptive mutation, the variance of the normal random number is changed according to the fitness values of the population. The fitness value of the individual is calculated by the evaluation function f_k generated by GP. If the evaluation function is suitable to the facing environment, the individuals of fuzzy controllers are easily and quickly optimized. Figure 4 shows the total architecture of the optimization of behaviors and evaluation functions.

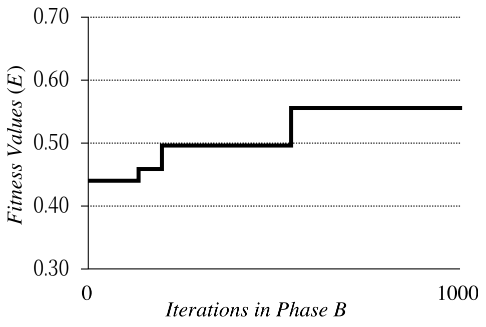
2.3 GP for Evaluation Functions

GP is an extension of GA using a structural coding method, which was proposed by Koza [9-11,17-19]. The GP can deal with tree structures, and have been applied for generating computer programs and functions.

As the human knowledge about the behavioral rules, fuzzy rules for fuzzy controller is basically given as the initial condition by the human operator (Fig.4). But these rules are

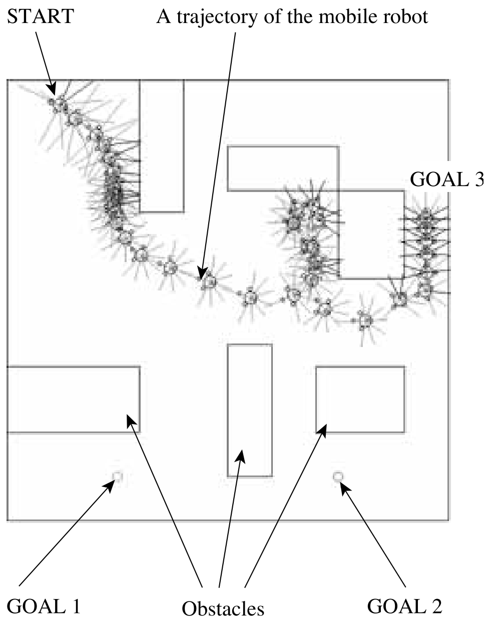


(a) A trajectory of the mobile robot

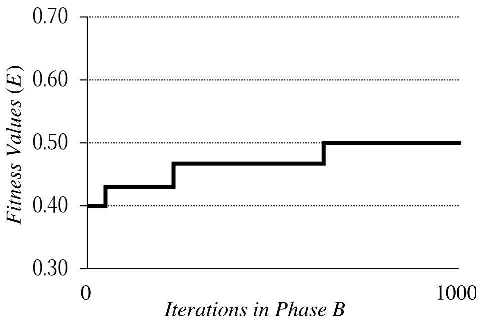


(b) Fitness values of the best individual in GP (Phase B)

Fig. 6 Simulation results (ex.I)



(a) A trajectory of the mobile robot



(b) Fitness values of the best individual in GP (Phase B)

Fig. 7 Simulation results (ex.II)

described from the viewpoint of human operators. Therefore, it is necessary for the robot to optimize these rules based on its embodiment. Fuzzy controllers are evaluated by evaluation functions generated by GP (Phase B). The total procedure is as follows:

```
Initialization
  repeat
    Behavior_SSGA(Phase B)
    repeat
      Function_GP(Phase A)
    until Termination_condition_in_Phase_A = True
  until Termination_condition_in_Phase_B = True
end.
```

An evaluation function is represented as Fig.5. A function is represented by the postfix notation. By using the postfix notation, we can deal with one-dimensional array of string. It includes constants, unary and binary operators, and variables. Variables for functions are the time to goal, the direction to goal, the distance to goal, and the measured distance of each sensor. Because the robot can perceive only partial information through its sensors, the evaluation criteria for the robot should be composed of perceptual information. If the robot acquires good fuzzy controller, many robots can reach the target point through the optimization of the robotic behavior by SSGA. That is, if the evaluation function is more suitable to the facing environment, the number of success is higher. Therefore, the evaluation functions can be evaluated by the number of success (E_k),

$$E_k=\frac{1}{p}\sum_{i=1}^{p}S_i \qquad S_i=\{0,1\} \tag{5}$$

where p is the population size in SSGA (Phase A), S_i indicates the success (1) or failure (0). This value corresponds to the meta-level evaluation from human operators in Fig.4. The crossover operator randomly decides two subtrees and exchanges them between two candidate solutions. The mutation operator exchanges the structure of function by replacing, shortening, and extending. The unary (binary) operator is replaced with the other unary (binary) operator.

3. Computer Simulations

This section shows a simulation result of the mobile robot. First, we describe the architecture of the mobile robot scaled according to the prototype of our developed mobile robot [9]. The robot has 8 range sensors to measure distances to obstacles (see Fig.1). The task is to reach the target point (Goal) while avoiding obstacles. Let us assume that the mobile robot can recognize the self-location and goal point in the environment, and therefore, the robot can calculate the moving direction toward the goal point. The size of environment is 500*500, when the radius of the mobile robot depicted as circle is 7. The sensing range is 60. The steering angle is restricted between 30° and -30°. The maximum velocity is 5 (the same size as the mobile robot). The setting parameters of fuzzy controllers, SSGA, and GP are as follows. The number of membership functions is 2. The maximum number of fuzzy rules are 10. The number of candidate solutions in the SSGA is 50. The number of iterations is 500. And the number of candidate solutions in the GP is 150. The number of iterations is 1000.

i) Experiment I

Figure 6(a) shows the trajectory of the mobile robot using the best fuzzy controllers. The sensing range is depicted as the broken line. When the mobile robot detects a obstacle in the sensing, it is depicted as the full line. The mobile robot can reach the goal without colliding with obstacles. In addition, when the mobile robot finds obstacles, it moves to the opposite direction to the obstacles. Consequently, the mobile robot acquired a collision avoiding behavior through interaction with the given environments by the proposed method. Figure 6(b) shows the change of fitness values (E) in GP (Phase B). By using evaluation functions generated by GP, the number of individuals which can reach its goal is gradually increasing. From this result, the evaluation function generated by GP plays the role of the guideline for the acquisition of a collision avoiding and a target tracing behavior. The behavior optimization process was done based on the evaluation criteria generated by GP. These simulation results also show that the mobile robot can acquire its behavior by using the evaluation functions. Consequently, the evaluation functions generated by GP is suitable to the environment and satisfy given tasks. And it can be also considered that the evaluation functions are based on the embodiment of the robot.

ii) Experiment II

In experiment II, we consider the task which is difficult to optimize behavioral rules of the robot with conventional approaches. The task is to reach one of the target points (Goals) while avoiding obstacles. Usually, evaluation functions for this behavior can be described as minimizing problem of moving time, moving distance, degree of danger and so on[3]. But we can't apply such evaluation functions if the environment has more than one goal. Figure 7(a) shows the simulation result. The task for the mobile robot is to reach one of the three goals (GOAL 3). By using the GP, the robot generate evaluation functions to accomplish the given task by itself. And figure 7(b) shows the change of evaluation value for the evaluation functions (Phase B). The evaluation, that is the number of individuals which can reach goal, is gradually increasing. These results show that the GP can generate the evaluation function suitable to not only the facing environment, but also given tasks and properties of the robot itself.

4. Conclusions

This paper applied genetic programming to generate evaluation functions for the life-time learning of an intelligent robot. The mobile robot should acquire its evaluation criteria through its life-time. Simulation results show that the GP can generate evaluation functions used in SSGA for collision avoidance. And it can be also said that the evaluation criteria was suitable for the facing environment, the given task, and the robot itself.

As future works, we intend to discuss the characteristics of the functions generated by GP and mapping capability as a total inference system from the measured data. And we should discuss about effective evaluation of the evaluation functions from meta-level. Furthermore, we have to discuss about the accumulation of the evaluation criteria.

5. References

[1] R. A. Brooks (1986), "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of*

Robotics and Automation, Vol.2, No.1, pp.14-23.
[2] S. J. Russell, P. Norvig (1995), *Artificial Intelligence*, Prentice-Hall, Inc.
[3] J. A. Anderson, E. Rosenfeld (1986), *Neurocomputing-Foundations of Research*, The MIT Press.
[4] J. M. Zurada, R. J. Marks II, C. J. Robinson (1994), *Computational Intelligence-Imitating Life*, IEEE Press.
[5] M. Palaniswami, Y. Attikiouzel, R. J. Marks II, D. Fogel, T. Fukuda (1995), *Computational Intelligence-A Dynamic System Perspective*, IEEE Press.
[6] L. A. Zadeh (1965), "Fuzzy Sets", *Information and Control*, Vol.8, pp.338-353.
[7] J.-S. R.Jang, C.-T. Sun, and E. Mizutani (1997), *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Inc.
[8] D. E. Goldberg (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Welsey.
[9] D. B. Fogel (1995), *Evolutionary Computation*, IEEE Press.
[10] J. R. Koza (1992), *Genetic Programming On the Programming of Computers by Means of Natural Selection*, The MIT Press.
[11] J. R. Koza (1994), *Genetic Programming II Automatic Discovery of Reusable Subprograms*, The MIT Press.
[12] J. R. Koza et. al (1999), *Genetic Programming III Darwinian Invention and Problem Solving*, The MIT Press.
[13] S. V. Kartalopoulos (1996), *Understanding Neural Networks and Fuzzy Logic*, IEEE Press.
[14] J. A. Anderson, and E. Rosenfeld (1988), *Neurocomputing-Foundations of Research*, The MIT Press.
[15] C. G. Langton (1995), *Artificial Life-An Overview*, The MIT Press.
[16] J. C. Bezdek (1994), "*What is Computational Intelligence ?,*" *in Computational Intelligence-Imitating Life* (J. M. Zurada, R. J. Marks II, C. J. Robinson (eds.)), IEEE Press, pp.1-12.
[17] T. Fukuda, and N. Kubota (1999), "An Intelligent Robotic System Based on A Fuzzy Approach", *Proceedings of The IEEE*, Vol. 87, No. 9, pp.1448-1470.
[18] N. Kubota, F. Kojima, S. Hashimoto, and T. Fukuda (1999), "Information Transformation by Virus-Evolutionary Genetic Programming", *Proc. of the 4th Int'l Symp. on Artificial Life and Robotics*, Vol. 2, pp.504-507.
[19] S. Hashimoto, N. Kubota, F. Kojima, and T. Fukuda (2000), "Genetic Programming for Perception-Based Robotics", *Proc. of The Fourth Asian Fuzzy Systems Symposium*, pp.674-679.
[20] J. R. Sherrah, R. E. Bogner, A. Bouzerdoum (1997), "The Evolutionary Pre-Processor", *Proc. of Second Annual Conference on Genetic Programming*, pp.304-312.
[21] G. Syswerda (1986), "A Study on Reproduction in Generational and Steady-State Genetic Algorithms", *In Foundations of Genetic Algorithms, San Mateo, Morgan Kaufmann Publishers, Inc*.
[22] L. A. Zadeh (1999), "A New Direction in Fuzzy Logic - Toward Automation of Reasoning with Perceptions", *Proceedings of IEEE International Conference on Fuzzy System*, Vol. 1, pp.1-5.
[23] S. Thrun (1996), *Explanatoin-Based Neural Network Learning: A Lifelong Learning Approach*, Kluwer Academic Publishers.
[24] F. Tanaka and M. Yamamura (1997), "An approach to lifelong reinforcement learning through multiple environments", *6th European Workshop on Learning Robots*, pp.93-99