An Evolutionary Fuzzy Behaviour Controller using Genetic Algorithm in RoboCup Soccer Game

Jong Yih Kuo and Yuan Cheng Ou
Department of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, Taiwan, R.O.C.
jykuo@ntut.edu.tw

***Abstract*—The problem of an effective behavior learning of autonomous robots is one of the most important tasks of the modern robotics. In fact, it is well known that the learning to optimize actions of autonomous agents in a dynamic environment is one of the most complex challenges of the intelligent system design. In this paper, we propose a hybrid approach integrating fuzzy logic system with genetic algorithm for high-level skills learning of robots within the RoboCup simulation soccer domain. Through the experiments, we found that the proposed method has good property of computation efficiency and also has a good advantage applied to the environment of RoboCup.**

***Keywords*—Fuzzy locic control, Genetic algorithm, Intelligent control, RoboCup**

I. INTRODUCTION

RoboCup is an international joint project to promote AI, robotics, and related field. It is an attempt to foster AI and intelligent robotics research by providing a standard problem: A soccer game as a central topic of research. As the nature of soccer game, autonomous robots participating in RoboCup soccer should have individual ability such as moving and kicking the ball, cooperative ability such as coordinating with teammates and the ability to deal with dynamic environment.

The problem of an effective behavior learning of autonomous robots is one of the most important tasks of the modern robotics. In fact, it is well known that the learning to optimize actions of autonomous agents in a dynamic environment is one of the most complex challenges of the intelligent system design.

Many machine-learning methods have been applied in RoboCup, reinforcement learning is a common one because it is appropriate for dealing with sequential decision making problems [1][2]. However, the application of reinforcement learning to practical problems is limited by the presentation of look-up table. It is difficult to scale up to continuous state space problems because the exponential growth of states in the number of state variables. This is a serious problem for robot agent has limited computational ability in real time process.

In this paper, we propose a hybrid approach integrating fuzzy logic system with genetic algorithm for high-level skills learning of robots within the RoboCup simulation soccer domain. Through the experiments, we found that the proposed method has good property of computation efficiency and also has a good advantage applied to the environment of RoboCup.

This paper is organized as follows: Section II introduces the related work about fuzzy system, genetic algorithm, and RoboCup Simulation League. Section III proposed our hybrid approach supported by fuzzy system and genetic algorithm. Section IV is the case study of the RoboCup game. Finally the conclusion is presented in section V.

II. RELATED WORK

A. Fuzzy System

Fuzzy logic is a powerful methodology for the control of the complex system operating under dynamic environment. Fuzzy logic is easy to express approximate knowledge and quick to implement reaction, because of its characteristic: knowing the expert's control rules formulated by words from natural language and people wants to produce a precise control strategy.

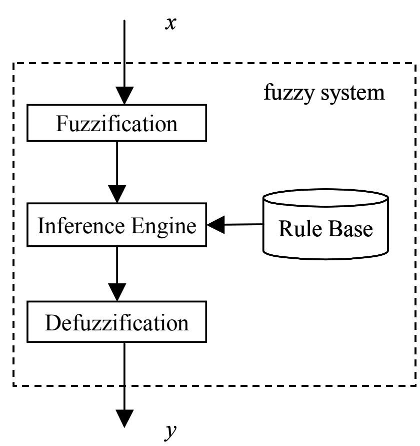


Figure 1: Block diagram of a fuzzy system.

Figure 1 shows the contents of a fuzzy system and described by the four major modules:

- Fuzzification - The input signals x are crisp values, which are transformed into fuzzy sets in this block.

978-0-7695-3745-0/09 $25.00 © 2009 IEEE
DOI 10.1109/HIS.2009.63

- Rule Base - the rule base contain as a whole the modeling information about the system, which is processed by the inference engine block.
- Inference Engine - The core section of a fuzzy system, which combines the facts obtained from the fuzzification with the rule base and conducts the fuzzy reasoning process.
- Defuzzification - The output y comes out directly from the defuzzification block, which transforms an output fuzzy set back to a crisp value.

Fuzzy control is the methodology that transforms the informal expert control rules into a precise control strategy. The idea was first proposed by Zadeh [3], and the methodology itself was first proposed and applied by Mamdani [4].

B. *Genetic Algorithm*

Genetic algorithms are very powerful for searching optimal solutions in a large search space. GA is very similar to the Charles Darwin's theory of natural selection; the 'fittest' individuals will be passed on while the 'worst' individuals will cease to exist. In general, GA is used to resolve the multi-parameter optimization problems, using strings to represent chromosomes can successfully act biological processes such as reproduction, cross-over and mutation and it has the ability of all fields optimization.

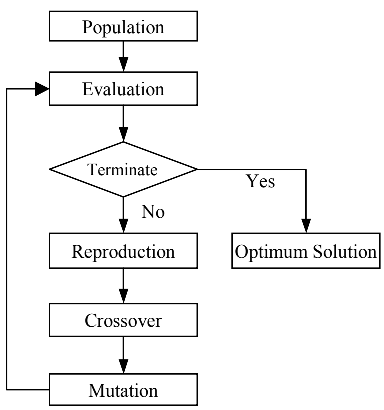


Figure 2: The basic process of Genetic Algorithm.

Figure 2 shows the basic steps of GA. An initial population of chromosomes can be created using randomly selected values for each gene in the solution space. The population is selected as the staring point of the process. In the evaluation stage, all chromosomes are evaluated by fitness function. The evaluation is achieved through the definition of an appropriate problem specific fitness function. The better performing individuals will get higher fitness value; on the contrary, the worse performing individuals will get lower fitness value. Each of the iterations in GA is known as a 'generation'. The iterations will be terminated if the stop condition is satisfied. Otherwise, the process goes to the reproduction stage. The better performing individuals will be chosen for Reproduction, GA has a selection operator which is responsible for selecting the fittest chromosomes into the crossover pool for crossover operation. In the crossover stage, it will reproduce new offspring chromosomes between two randomly selected chromosomes from crossover pool, the chromosomes are swapped at a particular point to generate offspring, and the goal of crossover is forming generational differences of chromosomes. In mutation stage, a number of the population is randomly chosen and a randomly chosen bit is changed in its bit string representation. And this mechanism is hoped to form a new chromosome that may have better performance evaluated by fitness function.

The usage of GA provides a technique that has a good efficiency for finding new solutions in achieving the final goal. GA can also be employed to generate and evolve computer programs as in genetic programming [5][6].

C. *RoboCup Simulation League*

As mentioned in Chapter I, RoboCup has become a standard problem for the artificial intelligence, robotics, and related communities. In particular, RoboCup provided the soccer simulator for the Simulation League. It created a more realistic soccer environment that allows researchers to deal with the higher-level tasks without worrying about lower-level issues such as robot movement, limited battery life and communications. The advantages of the soccer simulator make people who want to join in RoboCup research in an effective way.

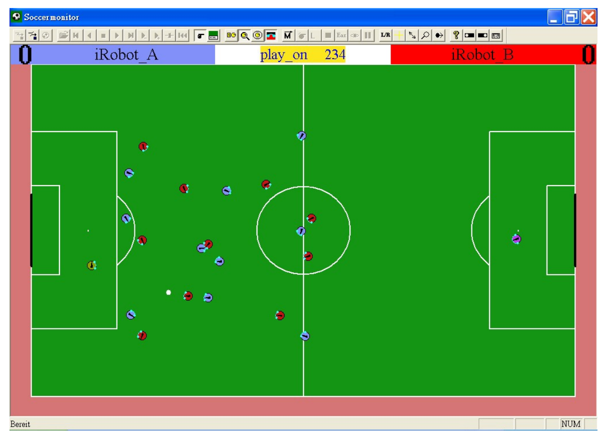


Figure 3: A snapshot during a RoboCup simulation soccer match.

In the simulation competition, there are 11 autonomous software agents per side play against each other using the RoboCup soccer simulator server. Games are played over the soccer server, which enables 22 client programs to connect through UDP/IP sockets. Each client player can send action commands to the server and the server is responsible for updating the game status and returns sensor information messages to each client regarding their current environment status. A Soccer Monitor is also available to allow users to see the current status of the match via a simple GUI window. Figure 3 shows a snapshot with 22 autonomous players in a soccer match.

The RoboCup Simulation League could be characterized as a fully distributed, multi-agent domain. Moreover, perceptions and actions cycles are asynchronous, communication is limited, and agent's action selection must be done in real-time. These characteristics made RoboCup Simulation League a challenging domain.

III. HYBRID APPROACH GA-FUZZY SYSTEM

A. Structure of the hybrid approach

Although the fuzzy logic has the advantage for the control of the complex system, but it is difficult to construct a well-define rule-base for the complex system, while genetic algorithm has strong searching abilities but it is weak at expressing rule-base knowledge. Hence, there are great advantages to using them collaboratively.

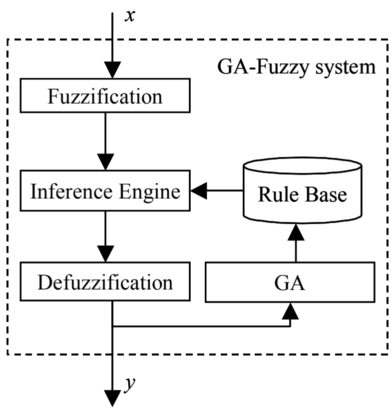


Figure 4: The structure of hybrid approach.

Figure 4 shows the hybrid approach for a GA-Fuzzy System. GA module will search and adjust the rule-base of the fuzzy system by GA operations. After that, the fuzzy system will get a better strategy for the control of the complex system.

A fuzzy system contains a number of parameters that can be modified to change the system performance, such as fuzzy rules and membership functions of linguistic values. In order to use GA for optimization of a fuzzy system, how to encode fuzzy rules and membership functions in binary strings (chromosomes) is the first thing for consideration.

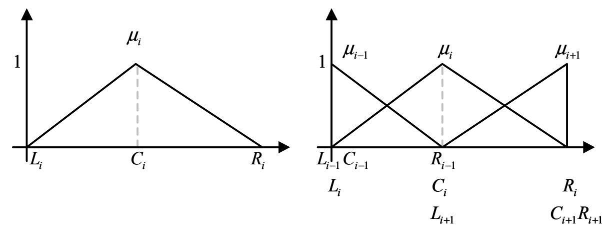
B. Genetic Coding

The first step in the design of GA-Fuzzy system is the definition of input and output variables. Each input or output variable can be treated as a linguistic variable in a fuzzy system. A linguistic variable can have a number of fuzzy sets (linguistic values) for representation and every linguistic value can have a membership function for determination. Hence, the number of the fuzzy sets of one linguistic variable and the parameters of the membership function can encode in genetic string.

In GA-Fuzzy system, all membership functions of linguistic values are in a triangular form defined by (1).

$$\mu_i = (L_i, C_i, R_i) \quad (1)$$

where L_i , C_i , R_i denotes the left boundary, center, right boundary point of the triangular with $L_i < C_i < R_i$, and i denotes the i-th membership function of a linguistic variable. Figure 5 (a) shows an example for a triangular membership function.



(a) Left boundary, center, right boundary point of a membership function.
(b) Relationships between membership functions.

Figure 5: Characteristics of membership functions in GA-Fuzzy system.

In order to reduce the computational complexity, some characteristics are applied to each membership function as in the following statement.

- Let μ_i is a membership function between μ_{i-1} and μ_{i+1} with $C_{i-1} < C_i < C_{i+1}$ and $C_i = R_{i-1} = L_{i+1}$, $C_{i-1} = L_i$, $C_{i+1} = R_i$. (An example in Figure 5 (b))
- A linguistic variable at least has two membership functions in each boundary side in the universe of discourse. L_{i-1} and C_{i-1} are at low bound position for the left-boundary triangular membership function. R_{i+1} and C_{i+1} are at up bound position for the right-boundary triangular membership function. (An example in Figure 5 (b))
- The universe of discourse was divided by 17 equidistance segments (16 points). The center position (C_i) of a triangular membership function can select anyone of them for adjustment.

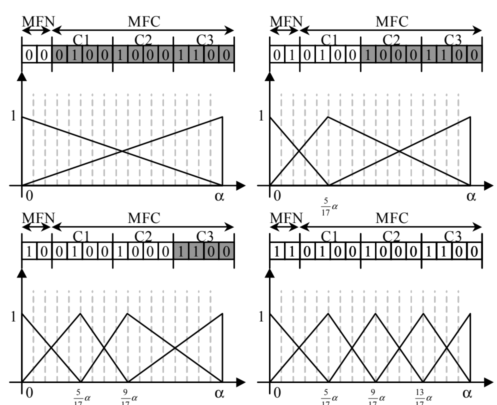


Figure 6: Using genetic bit string to change characteristics of membership functions.

This system uses a 2-bit column (MFN) to represent the number of linguistic values of a linguistic variable, and three

4-bit columns (MFC) for represent the center points of three triangular membership functions.

Figure 6 shows the relationship between genetic bit strings and characteristics of membership functions. By the coding characteristics, for each input linguistic variable, at least two membership functions (the left-boundary and right-boundary triangles) are always present, while using 2-bit MFN column can select the number of membership functions from two to five. The gray columns were ignored when the number of membership functions was below five.

A fuzzy rule base contains a number of IF-THEN rules. It is very similar to the human decision rules. The IF-THEN rule is denoted as the following form:

$$R^i : \text{ IF } x_1 \text{ is } v_1^i \text{ AND} \ldots x_n \text{ is } v_n^i$$

$$\text{THAN } y_1 \text{ is } w_1^i \text{ AND} \ldots y_m \text{ is } w_m^i \qquad (2)$$

where R^i represents the *i*-th rule in the rule base, v_j^i represents fuzzy linguistic value for input variable x_j $(j \in \{1,2,...,n\})$, w_j^i represents fuzzy linguistic value for output variable y_j $(j \in \{1,2,...,m\})$, and *i* is the number of rules.

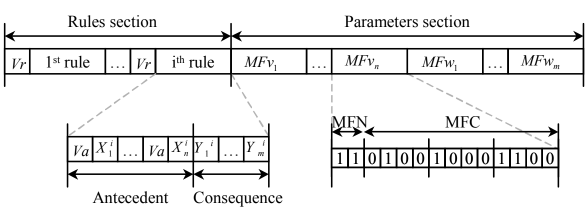


Figure 7: A candidate chromosome representing fuzzy rule.

Once the type of a rule in rule base is determined, and membership functions of a linguistic variable are also represented in genetic bit string, the representation of the encoding schema of a chromosome can define as shown in Figure 7.

A chromosome is composed of rules section and parameters section. In rules section, the *Vr* bit is used for representing the validity of one rule. In parameters section, the parameters of membership functions corresponding to inputs and outputs linguistic variables of the fuzzy system are stored here. A rule is composed of antecedent and consequence parts. The *Va* bit is used for representing the validity of one antecedent part in a rule. Each antecedent (X_n^i) and consequence (Y_m^i) variable is represented in real numbers. The remainder of antecedent/ consequence variable divided by (MFN+2) is representing the *i*-th membership function for this linguistic variable will be used.

Since the fuzzy rule base is fully encoding in a chromosome bit string, all of the genetic operations can be apply, such as crossover and mutation. And the most important is that the chromosome after genetic operations still be meaningful.

C. The Process of GA-Fuzzy System

The procedure of the GA evolutionary method for searching best rule base in GA-Fuzzy system is written as follows:

Step1: Initialization. A pre-specified number of chromosomes (individuals) of fixed length are generated by randomly assigning an integer value from [0, 1] for each bit.

Step2: Evaluation. All chromosomes are evaluated by fitness function. The chromosome with better performance will rate by higher score, the worse chromosome will get lower one.

Step3: Termination of the procedure. If pre-specified termination conditions are satisfied, stop the procedure. Otherwise go to Step 4.

Step4: Generation of new offspring chromosomes. A specified number of the best performance chromosomes will be selected into crossover pool by a roulette wheel method. Repeat the mating process by crossing over randomly selected chromosomes in the crossover pool until the number of new offspring chromosomes is reached. The chromosomes in the crossover pool will swapped at a particular point to generate offspring with a specified possibility. The new offspring chromosomes will also have mutation with a specified possibility.

Step5: Iteration. Return to evaluate the chromosomes performance (Step 2). This constitutes one generation.

In the Fuzzy system, the input signals are mapping to fuzzy sets (linguistic values) with appropriate membership functions first. Then, these fuzzy sets (antecedent parts) are used in the fuzzy rule base within an inference engine to result in new fuzzy sets (consequent part). The inference method is using Mamdani [4] implication and fuzzy relationships are synthesized by max-min principle. Finally, in the defuzzification step, the outputs of all rules in the rule base are synthesized into a single value. The center average defuzzifier is used because it is efficient in computation. The center average defuzzifier is indicated by (3).

$$y^* = \frac{\sum_{i=1}^{n} w_i \mu_i}{\sum_{i=1}^{n} \mu_i} \qquad (3)$$

Where w_i is the center of consequence fuzzy set of the *i*-th rule and μ_i represents its degree (height).

In this Chapter, a common representation in genetic bit string of a fuzzy rule base has constructed and the evolutionary method was also proposed. It can be easily apply into any fuzzy system.

IV. CASE STUDY

In the RoboCup soccer game, autonomous robots have to face many uncertain situations and have to adapt dynamic environments. In the beginning, it is hard to train the robot agent with a particular environment. It also will not be efficiency and not make good actions. We applied our GA-Fuzzy system for autonomous robot agents in different environment situations. We have study some situations of the RoboCup soccer games, and to choose one situation for the case study. This situation is that a robot will dribble with the ball toward the opponent side.

A. Problem Statement

In a soccer game, one of the most important skills is dribble with a ball. This action makes a robot agent to move with the ball while keeping it within a certain distance. A robot is repeatedly kicking the ball at a certain speed into a desired direction and then intercepting it again, and a robot will choose the direction and distance of the ball destination carefully to avoid the ball intercepted by the opponent team. This skill is an important way to break the defense of the opponent team that every robot should have learns.

Figure 8: A training case for a robot agent makes dribble toward the opponent side.

Figure 8 shows the training environment. We choose a robot agent makes dribble with the ball toward the opponent side, in the meanwhile, there are five opponents trying to intercept it. The robot agent's mission is dribbling with the ball as long as possible toward the opponent side.

B. System Implementation

1) The definition of input and output variables: Everytime when a robot agent can preform the dribble action, there are 5 environment facts need to be consideration. These are considered as input variables of GA-Fuzzy system:

x_1 : The direction of an opponent.

x_2 : The distance of an opponent.

x_3 : The direction of opponent goal.

x_4 : The direction of a sideline.

x_5 : The distance of a sideline.

And the system outputs variables are:

y_1 : The direction for the robot agent kicks the ball.

y_2 : The power for the robot agent kicks the ball.

2) Define the universe of discourse of a linguistic variable: Since this case is discussed in the soccer game, it is reasonable for using the same universe of discourse when linguistic variable is the same. Once the universe of discourse is determined, the membership functions of each linguistic variable can also be configured. Figure 9 shows one possibility of membership function for linguistic variables: distance, power and direction. There is an exception in membership functions of direction. It is because the direction is the standard in the world, such as 0° represents the North (μ_d^4), 90° represents the East (μ_d^6). Hence, the parameter of the direction membership function is not need to adjust by GA in anyway.

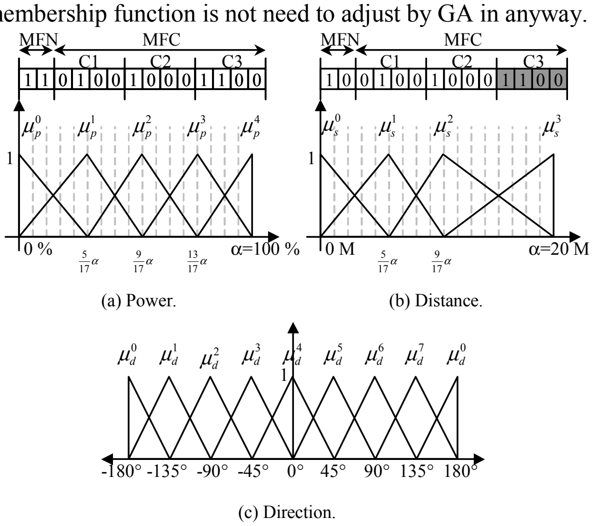


Figure 9: An example for membership functions of linguistic variables.

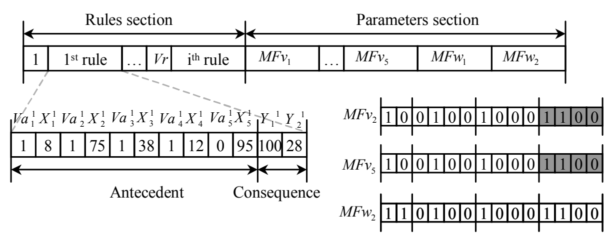


Figure 10: A part of a candidate chromosome.

3) Encoding and decoding chromosomes: The chromosome can be generated randomly by using encoding schema. Therefore, the first rule in a candidate chromosome in Figure 10 can be decoded as follows (also see Figure 9):

IF ($X_1^1 = \mu_d^0$) and ($X_2^1 = \mu_s^3$) and ($X_3^1 = \mu_d^6$) and ($X_4^1 = \mu_d^4$)
THEN ($Y_1^1 = \mu_d^4$) and ($Y_2^1 = \mu_p^3$)

4) Fitness function: To use GA for the evolution of fuzzy rule base for the robot agent, a fitness function is defined by (3).

$$F = \alpha(A) + \beta(B) + \gamma(\frac{1}{C}) \qquad (3)$$

where $\alpha = 1$, $\beta = 0.5$, $\gamma = 1000$.

A : The number of times the ball kicked.

B : Distance from the start point to the last position of the robot kicks the ball.

C : Distance from the last position of the robot kicks the ball to the opponent goal.

5) GA parameters settings: GA with population of 100 individuals is used in the experiment. Initial population was created randomly. The maximum number of rules in each individual is 20. Each individual is encoded in binary chromosome with 147 gene sets. 100 gene sets (5x20) are responsible for the input variables, and 40 (2x20) are responsible for the output variables, and others 7 are responsible for the parameters of membership functions of input and output variables. The best 10 chromosomes will be selected into crossover pool, and generates new offspring by using roulette wheel principle. Crossover was implemented using a single randomly selected crossover point with the probability 1.0. The probability of mutation for each gene sets is 1/147. The stop condition is set when the 200-th generation was reached.

C. *Experiment Result*

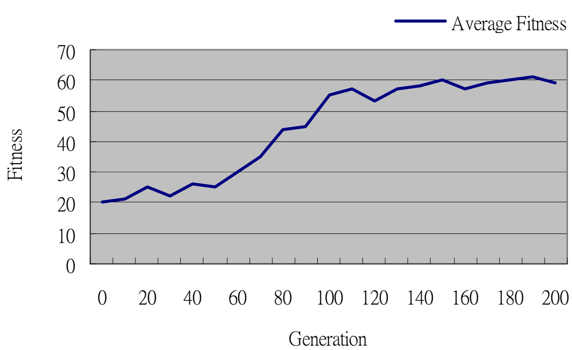


Figure 11: Fitness evolves by generation.

The main measurement is using fitness value that shows quality of the robot dribble action. Figure 11 shows the average fitness value of the population of 100 individuals from each generation. In the beginning, we can see that the value of the average fitness is relatively low, but it is improving during the evolutionary method. Finally, the better dribble skill can be evolved.

V. Conclusion

An effective behavior learning of autonomous robot in dynamic environment is a very difficult problem, main reason is that it is uncertainty to sense unknown environment for autonomous robots. GA-Fuzzy System is an effective method to solve uncertain controlling based on the sensing information. This paper presented a genetic fuzzy logic system according to the characteristic of the performance of an autonomous robot in the RoboCup soccer game. It can effectively be used to control an autonomous robot making good action based on the different environment information, namely, the direction of an opponent, the distance of an opponent, the goal direction, and so on. In addition, genetic algorithm is used to evolve the best rule base of the Fuzzy system; the GA-Fuzzy logic system has advantages such as higher learning speed, better flexibility. Results of the simulation in the RoboCup soccer game, autonomous robots show the advantages and the validity of the proposed method.

Acknowledgment

This work was supported by the National Science Council under grant NSC-97-2752-E-008-001-PAE and NSC-97-2218-E-027-004.

References

[1] J. Peters and S. Schaal, "Reinforcement Learning for Operational Space Control," in Proceedings of IEEE International Conference on Robotics and Automation, pp. 2111-2116, Roma, April 2007.

[2] H. Q. Min, J. A. Zeng, J. Chen, and J. H. Zhu, "A Study of Reinforcement Learning in a New Multiagent Domain," in Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 2, pp. 154-161, Sydney, NSW, Dec. 2008.

[3] L. A. Zadeh, "Outline of a new approach to the analysis of complex system and decision processed," IEEE Transactions on Systems, Man and Cybernetics, Vol. 3, No. 1, pp. 28-44, 1973.

[4] E. H. Mamdani, "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," IEEE Transactions on Computers, Vol. C-26, No. 12, pp. 1182-1191, 1977.

[5] T. Nakashima, M. Takatani, H. Ishibuchi, and M. Nii, "The Effect of Using Match History on the Evolution of RoboCup Soccer Team Strategies," in Proceedings of IEEE Symposium on Computational Intelligence and Games, pp. 60-66, Reno, NV, May 2006.

[6] H. Okada and T. Takagi, "Evaluation of Multi-objective Genetic Algorithm for RoboCupSoccer Team Evolution," in Proceedings of SICE Annual Conference, pp. 151-154, Tokyo, Aug. 2008.

[7] G. Narvydas, R. Simutis, and V. Raudonis, "Autonomous Mobile Robot Control Using Fuzzy Logic and Genetic Algorithm," in Proceedings of 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 460-464, Dortmund, Sept. 2007.

[8] E. A. Fares, M. Elbardiny, and M. M. Sharaf, "Adaptive Fuzzy Logic Controller of Visual Servoing Robot System by membership optimization using Genetic Algorithms," in Proceedings of IEEE International Conference on Computer Engineering and Systems, pp. 15-20, Cairo, Nov. 2007.