Knowledge Base Tuning using Genetic Algorithm for Fuzzy Behavior-based Autonomous Mobile Robot

Andi Adriansyah[1] Shamsudin H. M. Amin[2]
Centre of Artificial Intelligence and Robotics (CAIRO), Faculty of Electrical Engineering,
Universiti Teknologi Malaysia, Skudai, 81310, Johor Bahru, Malaysia
Phone: +607-5535319, Fax: +607-5566272,
E-mail: [1]andi_a3@hotmail.com [2]sham@fke.utm.my

Abstract

The attractive research in the field of robotics as a main alternative to conventional robot in recent years is Behavior-based mobile robot. This control architecture should generate perfect behavior action and able to handle conflicting actions that are seemingly irreconcilable, those are known as Behaviour Design Problem and Action Selection Problem. This paper presents a new schema to overcome behavior-based problems based on Fuzzy Logic Controller (FLC) where the fuzzy knowledge bases are tuned automatically by Genetic Algorithm (GAs), known as Genetic Fuzzy System (GFS). The behaviors are controlled by GFS to generate individual command action. Later, a Context-Dependent Blending (DBD) based on meta fuzzy rules coordinates the commands to produce final control action. The scheme is validated using parameters of MagellanPro mobile robot and tested by simulation using MATLAB/SIMULINK. Simulation results show that the proposed model offers hopeful advantages and has improved performance.

1. Introduction

A field of robotics now being attractive and has been well-known as a main alternative to conventional robot in recent years is Behavior-based controls [1]. This architecture is a bottom-up approach inspired by biology, in which several behaviors act in parallel accomplishing tasks [2,3]. However, several problems have been encountered with behavior-based approach. The problems are in two fold, namely, problem in *behavior design* and *action selection* problem. Behavior design problem means the robot needs to interact in dynamic unpredictable environment. At the side of that, behaviors-based control may produce conflicting actions that are seemingly irreconcilable in one particular time, which is known as action selection problem [4,5]. The problem arises because it is necessary to decide which behavior(s) should control the mobile robot at any given time to select among the action that most satisfied the system object.

In order to overcome the behavior design problems, a number of control systems have been provided, including simple and conventional logic up to intelligent control, such as Fuzzy Logic, Neural Network, Genetic Algorithm, and Evolutionary Programming [5, 6].

This paper will present an approach to overcome behavior-based problems aforementioned based on Fuzzy Logic Controller (FLC) where the fuzzy parameters, e.g. Fuzzy Membership Functions and Fuzzy Rule Bases are tuned by Genetic Algorithm (GAs) known as Genetic Fuzzy System (GFS). The organization of this paper is as follow. The Fuzzy Behavior-based and GFS are discussed briefly after this section. Later, the proposed design is explained. Afterward, some experimental results will be shown. Finally, this paper will be concluded in last section.

2. Fuzzy Behavior-based Robot

Behavior-based approaches were founded on the Subsumption Architecture [2], which is a methodology for designing autonomous mobile robot [1,2,3]. It imposes a general biologically inspired, distributed, bottom-up philosophy, allowing for a certain freedom of interpretation. In this approach, the robot task is decomposed into several modules, called *behaviors*, which the robot must accomplish and execute concurrently. The basic structure consists of all behaviors taking inputs from the robot sensors and sending outputs to the robot actuators. However, behaviors must completely be independent of each other. The parallel structure of simple behaviors allows a real-time response with low computational cost. Behavior-based method has demonstrated its reliable performance in standard robotic activities, such as navigation, obstacle avoiding, wall following, goal seeking, etc.

Consequently, more than one behavior could be generated in one particularly time. In this situation, behaviors with different objectives may produce conflicting actions. The objective of one behavior might be in contrast to the objectives of others. Therefore, it needs *behavior coordination* to come to a decision which behaviors should control the mobile robot at any given time to select among the action that most satisfied the system objective.

A fuzzy behavior-based robot means the using of fuzzy control technique to over come behavior-based problems, as mentioned before [4,7,8]. Behavioral design can be implemented using fuzzy system and behavior coordination, as well, as shown in Fig. 1.

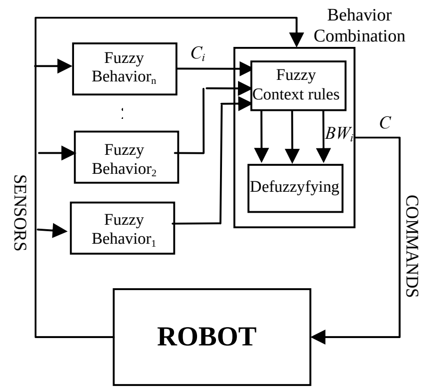

Fig. 1. Fuzzy Behavior-based Control Architecture

There are some considerable advantages of fuzzy approach to behavior-based mobile robot control [7]. First, due to its approximate reasoning capabilities, fuzzy logic produces controllers that are robust to uncertainty and imprecision (sensory noise, perturbations, etc.). Second, fuzzy behaviors can be conveniently synthesized by a set of IF-THEN rules using easy-to-understand linguistic terms to encode expert knowledge. Finally, the interpolative nature of fuzzy system has the ability to express partial and concurrent activations of behaviors, and the smooth transitions between behaviors.

The first and most common application of fuzzy logic techniques in the domain of autonomous mobile robot is the use of fuzzy control to implement individual behavior units. In most cases, attention was focused on the same two fundamental tasks: following environmental features (walls, edges, or other) and avoiding obstacles. More extensively developed autonomous robots are equipped with different behaviors, covering all the elementary sub-tasks that they need to perform.

Afterward, the general form of behavior combination that can be realized using fuzzy logic is obtained by using both *fuzzy context rules* to represent the arbitration policy, and *fuzzy combination* to perform command fusion. This form is called context-dependent blending (CDB) [6,7], as depicted in Fig. 1. There are two stages of CDB, i.e. generated *preferences* from fuzzy system; and represented *context* by formula in fuzzy logic. General form of meta fuzzy rule for representing context is like this:

IF *context* THEN *behavior*

meaning that *behavior* should be activated with a strength given by the truth value of *context*, a formula in fuzzy logic. When more than one behavior is activated, their outputs will have to be fused. Saffiotti [7] gave a formula for fusing the commands preferences, as follows:

$$C = \frac{\sum_i (BW_i * C_i)}{\sum_i BW_i} \tag{1}$$

where C is the final control action, i represent the active behavior activated by rules, BW_i is the behavior weight preferences and C_i is the behavior command output.

However, there are difficulties and trial-and-error work to set Membership Function as well as Fuzzy Rule Base in Fuzzy System properly. Also, increasing the number of input variables in fuzzy system, the number of rules is increased exponentially, which makes it more difficult to define the rule set for a good system performance. Training and learning is needed to tune those parameters. Therefore, most of the works in the field generated a certain interest for the study of fuzzy systems with added learning or tuning capabilities. Neuro-fuzzy systems are one of the most successful and visible direction of that effort. A different approach to hybridization leads to genetic fuzzy systems (GFS) [9], Fuzzy Genetic Programming, etc. GFS will be described briefly in next section.

3. Genetic Fuzzy System

GAs are general purpose search algorithms, based on natural genetics, that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problem requiring efficient and effective search process [10,11]. The basic idea is to maintain a population of chromosomes that evolves over time through a process of competition and controlled variation. A *chromosome* is representing candidate solutions to the concrete problem being solved.

A GA starts with a *population* of randomly generated chromosomes, and advance towards better chromosomes by applying genetic operators modeled on the genetic process occurring in nature. The population undergoes evolution in a form of natural selection. During successive iterations, called *generation*, chromosomes in the population are rated for their adaptation as solutions, and on the basic of these evaluation, a new population of chromosomes is formed using a selection mechanism and specific genetic operator such as *crossover* and *mutation*. A *fitness function* must be devised for each problem to be solved. Given a particular chromosome, the fitness function returns a single numerical value, which is supposed to be proportional to the utility or adaptation of the solution represented by that chromosome.

As previously stated, GFS is basically a fuzzy system augmented by a learning process based on a genetic algorithm (GA). In GFS, GAs operates to search an appropriate Knowledge Base (KB) of a fuzzy system for a particular problem and to make sure those parameter values that are optimal with respect to the design criteria. The KB parameters constitute the optimization space, which is transformed into suitable genetic representation on which the search process operates.

The KB is composed by membership functions (MF) and fuzzy rule base (RB), as mentioned before. So, there are some options to design Genetic Fuzzy System, e.g. tuning or learning membership functions, or fuzzy rule base or both of them, sequentially or concurrently [12]. When tuning membership functions, an individual population represents parameters of the membership function shapes at which fuzzy rule base is predefined in advance. In contrast, if be desired to tune fuzzy rules base, the population represents all of fuzzy rules possibility that membership functions is assumed before. Fig. 2 shows these conceptions.

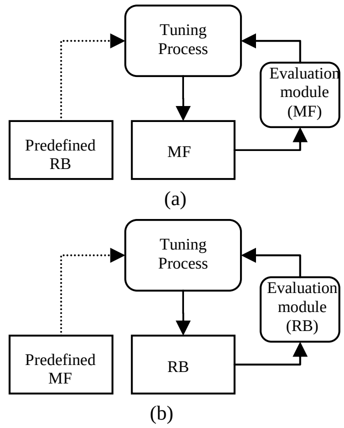

Fig. 2. Some designs of Genetic Fuzzy Systems

Recently, there are some successful applications of GFS to real world problems, e.g. control, robotics, manufacturing, consumer products, transportation, modeling and decision making [13]. In the next section, this paper will describe application a GFS in behavior-based mobile robot.

4. Proposed Design

The control architecture of behavior-based mobile robot proposed in this paper is based on GFS, as shown in Fig. 3. The mobile robot consists of four behaviors, namely; goal seeking, left wall following, right wall following and obstacle avoiding behavior. Each fuzzy behavior has full access to all sensor readings and processes its own command to control direction of the mobile robot. Every knowledge bases of fuzzy behavior are tuned by GA. Later, commands from each behavior are coordinated to generate a final command for best action according to particular situation using Context-Dependent Blending (CBD).

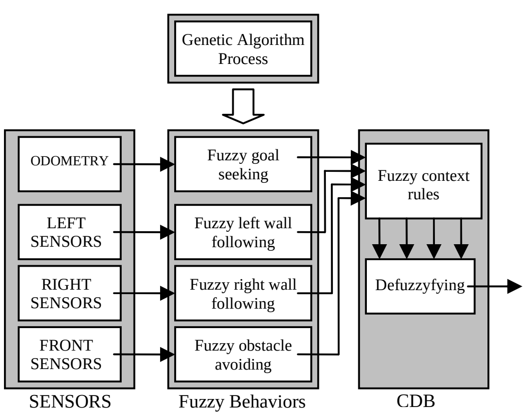

Fig. 3. Control Architecture of mobile robot

2.1 Model of Mobile Robot

The experiment used MagellanPro robot as a model. The MagellanPro is a circular mobile robot from iRobot. The robot has dimensions such as follows: D = 40.6 cm, H = 25.4 cm, r = 5.7 cm and W = 36 cm, where D is diameter, H is height, r is the radius of wheels and W is distance between two wheels, as depicted in Fig. 4(a). The robot is located in a two dimensional Cartesian workspace, in which a global coordinate $\{X,O,Y\}$ is defined. A local coordinate $\{X',C,Y'\}$ is attached to the robot with the origin at point C, the middle point of two wheels which is the guide point of this mobile robot. The robot has three degrees of freedom that are represented by a posture $p_c = (x_c, y_c, \theta_c)$, where (x_c, y_c) indicate the spatial position of the robot guide point in the global coordinate system and θ_c is the heading angle of the robot.

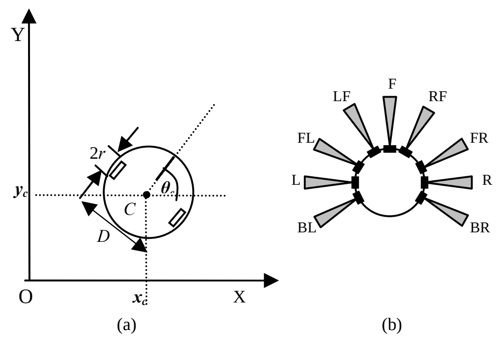

Fig. 4. Model of mobile robot: (a) Mobile robot dimension and position, (b) Mobile robot sensors configuration

2.2 Fuzzy Behavior-based System Design

The fuzzy system is based on trapezoid and triangular membership functions. The input of individual behavior module receives inputs from sensor signals ($x_1, x_2, \ldots, x_n$), and the control action ($y_1, y_2, \ldots, y_n$) is obtained from the output.

The output generated by applying the correlation-product inference and the centroid defuzzification scheme, as:

$$y_o = \frac{\sum_{l \in P_o} y_l . C_{ox} D_{ox}}{\sum_{l \in P_o} y_l . D_{ox}} \quad (2)$$

where C_{ox} and D_{ox} are the parameters of center and width of output membership functions, y_l are product of the membership functions for each rule input, and l is the total number of rules.

Most of behaviors have two inputs, except obstacle avoiding behavior has three inputs. In goal seeking behavior, the inputs are *target distance* (d) and *target angle* (θ). In the other hand, *front left distance (FL)* and *back left distance (BL)* are used for left wall following, also *front right distance (FR)* and *back right distance (BR)* are used for right wall following, respectively, Finally, in obstacle avoiding the input namely are *left front distance (LF), front distance (F),* and *right front* *distance (RF).* Every input has three membership functions, as a minimal number of membership functions. All of the behaviors have same two inputs, namely *translational speed (v)* and *rotational speed (ω)*, and have three membership functions too. All of fuzzy membership functions are depicted in Fig. 5.

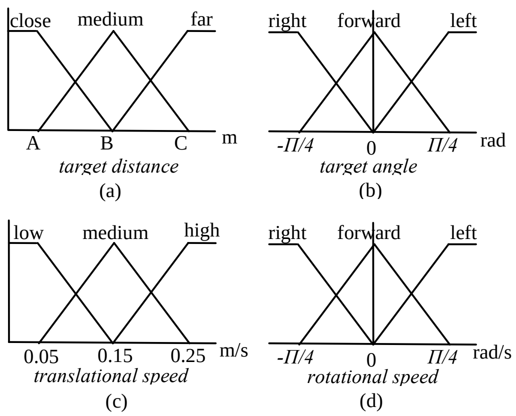

Fig. 5 Membership functions of (a) distances, (b) angle for input, and (c) translational and rotational speed for output

Obstacle avoiding behavior presented here as an example of fuzzy behavior design. With this fuzzy knowledge base, there are 3 membership functions for each input. Every membership function need 2 parameters should be adjusted, so, totally 18 (3_{input} x 3_{mf} x 2) parameters. Also, there are 27 fuzzy rule bases for each of output, thus a total of 54 (2_{output} x 3_{mf} x 3_{mf} x 3_{mf}) parameters are needed.

Finally, the meta fuzzy rules are designed based on the CDB mentioned before, as follows:

IF *obstacle_distance_close* THEN *obstacle_avoiding*

IF ~(*obstacle_distance_close*) AND *left_distance_close* THEN *left_following*

IF ~(*obstacle_distance_close*) AND *right_distance_close* THEN *right_following*

IF ~(*obstacle_distance_close*) AND ~(*left_distance_close*) AND ~(*right_distance_close*) THEN *goal_seeking*

with using (1) to perform command fusion if two or more behaviors are activated.

Afterwards, a GFS is then applied as a searching algorithm to tune the value of knowledge bases that described in next subsection.

2.3 Genetic Fuzzy System Design

As GA's deal with coded parameters, all parameters that need to be tuned must be encoded into a finite length of string or gene. The encoded genes are concatenated to form a complete chromosome.

For the tuning of membership function the following equations were defined:

$$C_x = C_x + k_i$$
$$W_x = W_x + j_i \qquad (3)$$

where k_i, j_i are adjustment coefficient, C_x, and W_x are set of centre and width of each fuzzy membership function. It means k_i makes each center of membership function move to the right or left and j_i makes them wider or sharper, as shown in Fig. 6. After that, the adjustment coefficients are encoded to form the population, as presented in (4).

Gene |1|,…,|5|,|6|,|7|,..|11|,|12|,|13|,…,|17|,|18|
Chromosome |*subchrom*$_1$|,|*subchrom*$_2$|,| *sumchrom*$_3$ |
Chromosome |....$k_{1i}j_{1i}$.....|,|....$k_{2i}j_{2i}$.......|,|.....$k_{3i}j_{3i}$......|
Parameter | ....(MF_1)...|,|...(MF_2)|…|,|...(MF_3)……| (4)

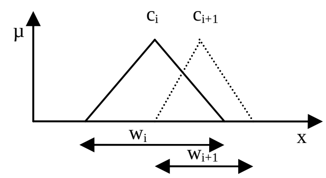

Fig. 6 Principle in tuning of membership function

In another side, for rule base searching, each of the parameter is encoded into integer codes that are based on number of output membership function. It means there are '1', '2', and '3' for RIGHT, FORWARD, and LEFT for *translational speed* output, and LO, MEDIUM, and HI for *rotational speed* output, respectively.

The coded parameters are arranged as show in the following equation to form chromosome of the population, as shown in (5).

Gene |1|, |2|, … ,|26|,|27|, |28|,|29|, …, |53|, |54|
Chromosome |*sub-chromosome*$_1$|, | *sub-chromosome*$_2$ |
Parameter | ……..(RB_1)…....|, | ……..(RB_2)|…… .| (5)

GA's process starts with randomly generated initial populations. Then, all chromosomes are evaluated and associated base on fitness function with linear ranking method to determine the members of the new generation population. The fitness function for obstacle avoiding as:

$$f = \sum_{i=0}^{I}[\sum_{k=0}^{K}(100\omega^2(k) + 0.5/v(k) + 100c(k)) \qquad (6)$$

where I is the total number of start position, K is the number of step simulation for each start position, $\omega(k)$, and $v(k)$ are the rotational speed and the translational speed at k, respectively, and c is constant for collision check, 0 if there is no collision and 1 if there is one. This function is minimized in order to achieve the condition than the robot move by obstacle avoiding, higher speed, and mostly straight direction.

After that, three operators of GA are carried out, namely *recombination*, *crossover* and *mutation*, with fixed crossover probability rate (P_c) and probability mutation rate (P_m), that are 0.7, and 0.7/parameter numbers, respectively. The number of new generation is adjusted by Generation Gap constant (*GGAP*), which is

0.9. The procedure is repeated until the termination condition is reached.

5. Results

Computer simulation for tuning fuzzy knowledge base of individual behavior and animation of mobile robot movement was developed using MATLAB Version 6.5 Release 13.

To find best parameters of knowledge base for obstacle avoiding behavior, some GA process have been done. There are four experiments provided, e.g. tuning membership functions with predefined rule base, tuning learning rule base with predefined membership functions, tuning membership function with the best rule base founded, and learning rule bases with the best membership function obtained. However, in each of experiment, the population consists of 25 chromosomes, 40 generation, 40 step simulations for each position, and 15 different start positions. Table 1 present results of the experiments. The table shows that tuning MF with RB resulted before from previous learning obtained minimum value for the fitness function. It means best knowledge base for fuzzy obstacle avoiding behavior obtained by learning the RB in advance and then tuning the MF after.

Table 1 Comparison of fitness value for some GA processes

GA process	Fitness Value
Tuning MF with predefined RB	1.9369×10^3
Tuning RB with predefined MF	2.1261×10^3
Tuning MF with RB resulted before	1.8715×10^3
Tuning RB with MF resulted before	1.8759×10^3

Fuzzy knowledge base tuning process for obstacle avoiding behavior is depicted in Fig. 7. The figure shows the evolution of the average fitness of each generation and the fitness of the best individual of each generation. It can be appreciated that, in learning of RB has big range for searching the best rule and converged in around 25th generation. But, in tuning MF, the range is not big and has not converged after 40th generation.

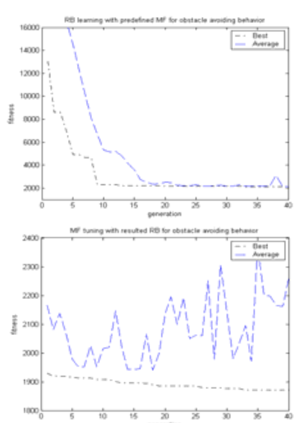

Fig. 7. Fuzzy rule base tuning process for obstacle avoiding behavior

The rule bases obtained is shown in Table 1. This was found using GA parameters previously described. From that table, it deals properly with the entropy problem: the behavior try to avoid obstacle with minimizing steering and maximizing speed. Besides, the speed of robot adjusted low when the robot make turning left or right, especially when the distance is close or medium.

Table 2 The best rule base for obstacle avoiding behavior

INPUT			OUTPUT	
left front	front	right front	translational	rotational
close	close	close	low	right/left
close	close	medium	medium	right
close	close	far	medium	right
close	medium	close	medium	forward
close	medium	medium	medium	right
close	medium	far	medium	right
close	far	close	high	forward
close	far	medium	high	right
close	far	far	high	right
medium	close	close	medium	left
medium	close	medium	low	right/left
medium	close	far	medium	right
medium	medium	close	medium	left
medium	medium	medium	medium	forward
medium	medium	far	medium	forward
medium	far	close	high	left
medium	far	medium	high	forward
medium	far	far	high	forward
far	close	close	medium	left
far	close	medium	medium	left
far	close	far	low	right/left
far	medium	close	medium	left
far	medium	medium	medium	forward
far	medium	far	medium	forward
far	far	close	high	left
far	far	medium	high	forward
far	far	far	high	forward

An office-room scenario has been used for testing all behaviors and behavior combination of mobile robot. Its overall dimensions are 10 by 10 meters, with corridor of 2 meters of width, and doorways of 1 meter. Fig. 5 shows simulation of mobile robot movement from different start position to reach a goal without any collision with wall or obstacles. First, the robot moves to go to goal position. Then, while it detects any obstacle, the robot starts to avoid that. After that, before avoiding obstacles is finished, the robot starts to adjust again its orientation to goal position. Also, following a wall is done when detecting. Finally, the robot goes to goal position when there are no obstacles and wall the field. From the figures, there is smooth transition between behaviors.

Fig. 6 shows behavior coordination as previous figure, but illustrated in different point of view, i.e. degree of behavior activation. The figure, based on Fig. 6(a), explains that individual behavior activated according to their context with some degree of activation. This degree shows the behavior is done by the robot in some gradation to make sure continuously movement of robot, Also, there is shows transition between two behaviors. It means, the robot can do more than one behavior in any particularly time.

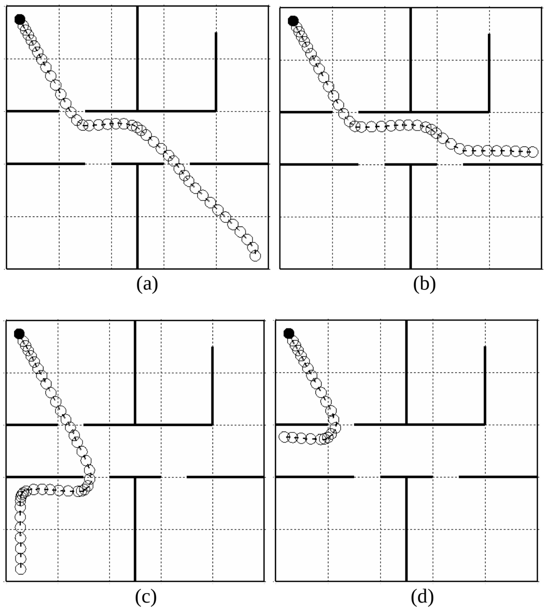

Fig. 5. Mobile robot movement in office-room from different start position

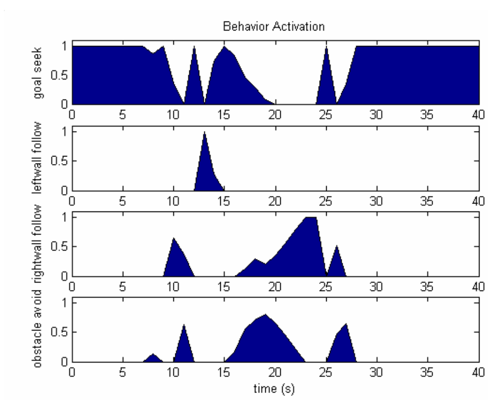

Fig. 6. Behaviors Activation of Fig. 5(a)

6. Conclusion

This paper has presented a Fuzzy Logic Controller (FLC) where the fuzzy knowledge based, i.e. membership functions and rule bases, are tuned by Genetic Algorithm (GAs), known as Genetic Fuzzy System (GFS), to generate individual command action. The model is designed in order to overcome behavior-based problems. So as to produce final control action, the behaviors activated are fused by Context-Dependent Blending (CDB) based on meta fuzzy rules. From the experiments, the proposed schema offers hopeful advantages, such as tuning fuzzy knowledge bases automatically. The best fitness knowledge base is obtained by learning the RB in advance and then tuning the MF after. Beside that, the robot has improved its performance, for instance it can generate robust control for individual behavior, able to produce the behavior in a degree of activation, changeover between two behaviors and move smoothly.

References

[1] Arkin, R.C., Behavior-based Robotics, The MIT Press, 1998, London, England.

[2] Brooks, R. A., “Robust Layered Control System for a Mobile Robot”, IEEE J. of Robotics and Automation, Vol. 2, No.1, (1986), pp. 14-23.

[3] Mataric, M. J., “Behavior-based Control: Examples from Navigation, Learning, and Group”, Journal on Experimental and Theoretical Artificial Intelligence, Special Issue on Software Architecture for Physical Agents, Vol. 9, No. 2-3, Hexmor, Horswill, and Kortenkamp, eds., (1997), pp. 1-15.

[4] Saffiotti, A., “The uses of fuzzy logic in autonomous robot navigation”, Soft Computing Vol. 1, Springer-Verlag, (1997), pp. 180-197.

[5] Pirjanian, P., “Behavior Coordination Mechanisms: State-of-the-art”, Tech-report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, October, (1999), pp. 1-49.

[6] Hoffmann, F., “An overview on soft computing in Behavior Based Robotics”, Proc. on Int’l Fuzzy System Association World Congress IFSA 2003, (2003), pp. 544-551.

[7] Saffiotti, A., “Fuzzy Logic in autonomous robotics: behaviour co-ordination”, Proceeding of the 1996 IEEE International Conference on Robotics and Automation, (1996), pp. 951-956.

[8] Saffiotti, A., Ruspini, E. H., and Konilige, K., “Using Fuzzy Logic for Mobile Robot Control”, Practical Applications of Fuzzy Technologies H-J. Zimmermann, editor, Kluwer Academic Publisher, (1999), pp. 185-206.

[9] Cordon, O., Gomide, F., Herrera, F., Hoffman, F., and Magdalena, L., “Ten years of genetic fuzzy systems: current framework and new trends”, J. Fuzzy Sets and Systems 141, (2004), pp. 5-31.

[10] Goldberg, D. E., Genetic Algorithm in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, (1989).

[11] Davis, L., Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[12] Cordon, O., Herrera, F., Hoffman, F., Magdalena, L., Genetic Fuzzy Systems – Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, Singapore, 2001.

[13] Hagras, H., Callaghan, V., Coley, M., and Carr-West, M., “A Behavior based hierarchical fuzzy control architecture for agricultural autonomous mobile robots”, Proc. of the Int. Conf. on Computational Intelligence for Modelling, Control and Automation, Vienna- Austria, (1999), pp. 166-171.