Learning to Avoid Moving Obstacles Optimally for Mobile Robots Using a Genetic-Fuzzy Approach

Kalyanmoy Deb, Dilip Kumar Pratihar, and Amitabha Ghosh

Kanpur Genetic Algorithms Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur
Kanpur, Pin 208 016, India
E-mail: {deb, dkpra, amitabha}@iitk.ernet.in

Abstract. The task in a motion planning problem for a mobile robot is to find an obstacle-free path between a starting and a destination point, which will require the minimum possible time of travel. Although there exists many studies involving classical methods and using fuzzy logic controllers (FLCs), they are either computationally extensive or they do not attempt to find optimal controllers. The proposed genetic-fuzzy approach optimizes the travel time of a robot off-line by simultanously finding an optimal fuzzy rule base and optimal membership function distributions describing various values of condition and action variables of fuzzy rules. A mobile robot can then use this optimal FLC on-line to navigate in the presence of moving obstacles. The results of this study on a number of problems show that the proposed genetic-fuzzy approach can produce efficient rules and membership functions of an FLC for controlling the motion of a robot among moving obstacles.

1 Introduction

Building autonomous robots, which can plan its own motion during navigation through two-dimensional or three-dimensional terrains, has been one of the major areas of research in robotics [11, 6, 8]. Latombe [11] provides an extensive survey of different classical approaches of motion planning, particularly in the presence of stationary obstacles. Both graphical as well as analytical methods have been developed by several investigators to solve the mobile robot navigation problems among moving obstacles, known as dynamic motion planning problems. These methods include path velocity decomposition [6, 8], accessibility graph technique [7], incremental planning [12], probabilistic approach [17], potential field approach [15, 16, 1], and others. Moreover, different learning techniques have also been used by researchers to improve the performance of conventional controllers [4, 5].

Each of these methods has its own inherent limitations and is capable of solving only a particular type of problems. Canny and Reif [3] studied the computational complexity of some of these methods and showed that motion planning for a point robot in a two-dimensional plane with a bounded velocity is an

NP-hard problem, even when the moving obstacles are convex polygons moving at a constant linear velocity without rotation. Potential field method [15, 16, 1], in which a robot moves under the action of combined attractive and repulsive potentials created artificially, is the most widely used technique for solving dynamic motion planning problems. Since at every time step a new potential field must be created to find an obstacle-free direction, the method is local in nature and often has the chance of converging to a sub-optimal solution. Moreover, it is intuitive that many computation of such local travel directions using artificial potential field method may be computationally expensive.

To reduce the computational complexity, some heuristics have also been developed by several researchers. Fuzzy logic controllers (FLCs) have been used by several investigators in the recent past [2, 18, 14] to solve the dynamic motion planning problem. However, in all such studies, no effort is spent to find optimal FLCs (instead an FLC is designed based on a particular user-defined membership function and rules). With the availability of a versatile yet efficient optimization method (GA), optimal FLCs for dynamic motion planning problems can be developed, like they have been used in other applications of FLCs, such as the cart-pole balancing [10], cart centering [19], and others [9, 13].

In the present study, we concentrate on dynamic motion planning (DMP) problem, where the objective is to find an obstacle-free path between a starting point and a destination point, requiring the minimum possible time of travel. Since the DMP problem is unrealistic to solve on-line for every new scenario a robot faces, we convert the problem into a similar yet an approximate off-line optimization problem. The optimiation problem involves finding an optimal fuzzy rule base that the robot should use for navigation, when left in a number of author-defined scenarios of moving obstacles. Once the optimal rule base is obtained off-line, the robot can then use it on-line to navigate in other scenarios of moving obstacles.

In the remainder of this paper, we describe the genetic-fuzzy approach by drawing a simile of the motion planning problem with a natural learning process. The proposed approach incorporates some practical considerations, which, along with the use of a fuzzy logic controller, makes the overall approach easier to be used in practice. The efficacy of the proposed approach is demonstrated by solving a number of motion planning problems.

2 Proposed Genetic-Fuzzy Approach

We describe the genetic-fuzzy approach by drawing a connection between the motion planning problem with a natural learning process. The purpose of the DMP problem of a robot is to find an obstacle-free path which takes a robot from a point A to a point B with minimum time. There are essentially two parts of the problem:

1. Learn to find *any* obstacle-free path from point A to B, and
2. Learn to choose that obstacle-free path which takes the robot in a minimum possible time.

Both these problems are somewhat similar to the learning phases a child would go through while solving a similar obstacle-avoidance problem. If a child is kept in a similar (albeit hypothetical) situation (that is, a child has to go from one corner of a room to another by avoiding a few moving objects), the child learns to avoid an incoming obstacle by taking detour from its path. It is interesting that while taking the detour, it never calculates the precise angle of deviation form its path. This process of avoiding an object can be thought as if the child is using a rule of the following sort:

If an object is very near and is approaching, then turn right to the original path.

Because of the imprecise definition of the deviation in this problem, it seems natural to use a fuzzy logic technique in our study, instead of an exact representation of the deviation angle.

The second task of finding an optimal obstacle-free path arises from a simile of solving the same problem by an experienced versus an inexperienced child. An inexperienced child may take avoidance of *each* obstacle too seriously and deviate by a large angle each time it faces an obstacle. This way, this child may lead away from the target and take a long winding distance to reach the target. Whereas, an experienced child may deviate barely from each obstacle, thereby taking the quickest route to the target point. If we think about how the experienced child has learned this trick, the answer is through experience of solving many such similar problems in the past. Previous efforts helped the child find a set of good rules to do the task efficiently. This is precisely the task of an optimizer which needs to discover an optimal set of rules needed to avoid obstacles and to reach the target point in a minimum possible time. This is where the GA comes as a natural choice.

Thus, the use of fuzzy logic technique helps in quickly determining imprecise yet obstacle-free paths and the use of a GA helps in learning an optimal set of rules that a robot should use while navigating in presence of moving obstacles. This process is illustrated in Figure 1.

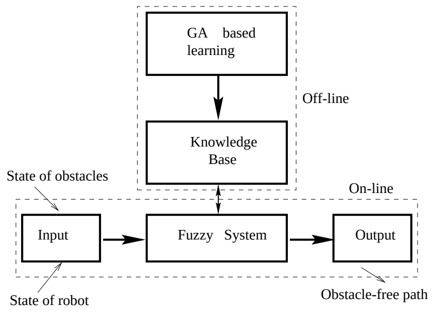

Fig. 1. Genetic-fuzzy approach

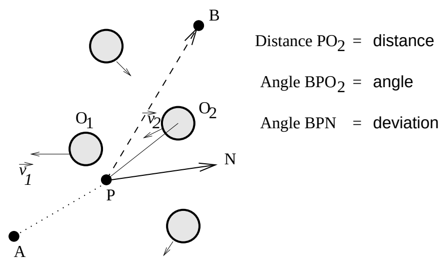

Fig. 2. A schematic showing condition and action variables

A GA is used to create the knowledge base (fuzzy rule base) of a robot off-line. For on-line application, the robot uses its optimal fuzzy rule base to find an obstacle-free path for a given input of parameters depicting the state of moving obstacles and the state of the robot.

2.1 Representation of a Solution

A solution to the DMP problem is represented by a set of rules which a robot will use to navigate from point A to point B (Figure 2). Each rule has three conditions: distance, angle, and relative velocity. The distance is the distance of the nearest obstacle forward from the robot. Four fuzzy values of distance is chosen: very near (VN), near (N), far (F), and very far (VF). The angle is the relative angle between the path joining the robot and the target point and the path to the nearest obstacle forward. The corresponding fuzzy values are left (L), ahead left (AL), ahead (A), ahead right (AR), and right (R). The relative velocity is the relative velocity vector of the nearest obstacle forward with respect to the robot. In our approach, we do not use this variable explicitly, instead follow a practical incremental procedure. Since, a robot can sense the position and velocity of each obstacle at any instant of time, the critical obstacle ahead of the robot can always be identified. In such a case (Figure 2), although an obstacle O_1 is nearer compared to another obstacle O_2, the relative velocity $\boldsymbol{v}_1$ of O_1 directs away from robot's path towards the target point B and the relative velocity $\boldsymbol{v}_2$ of O_2 directs towards the robot (Position P). Thus, the obstacle O_2 is assumed to be the critical obstacle forward.

The action variable is deviation of the robot from its path towards the target (Figure 2). This variable is considered to have five fuzzy values: L, AL, A, AR, and R. Triangular membership functions are considered for each membership function (Figure 3). Using this rule base, a typical rule will look like the following:

If distance is VN and angle is A, then deviation is AL.

With four choices for distance and five choices for angle, there could be a total of 4×5 or 20 valid rules possible. For each combination of condition variables, a suitable action value (author-defined) is associated, as shown in Table 1.

Table 1. All possible rules are shown

distance \ angle	L	AL	A	AR	R
VN	A	AR	AL	AL	A
N	A	A	AL	A	A
F	A	A	AR	A	A
VF	A	A	A	A	A

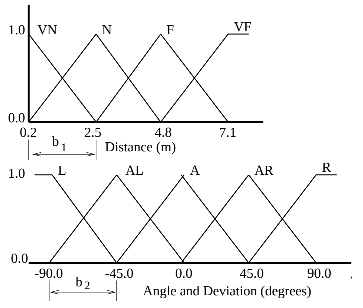

Fig. 3. Author-defined membership functions

The task of GAs is to find which rules (out of 20) should be present in the optimal rule base. We represent the presence of a rule by a 1 and the absence by a `0`. Thus, a complete solution will have a 20-bit length string of `1` and `0`. The value of i-th position along the string marks the presence or absence of the i-th rule in the rule base.

2.2 Evaluating a Solution

A rule base (represented by a 20-bit binary string) is evaluated by simulating a robot's performance on a number of scenarios (say S) and by keeping track of the travel time T in each scenario. Since a robot may not reach the destination using an arbitrary rule base, the robot is allowed a maximum travel time. In this case, a penalty proportional to a time needed to cover the Euclidean distance from the final position to the target point is added to the allowed maximum travel time. An average of travel times in all S scenarios is used as the *fitness* of the solution.

The robot's complete path is a collection of a number of small straight line paths traveled for a constant time ΔT in each step. To make the matter as practical as possible, we have assumed that the robot starts from zero velocity and accelerates during the first quarter of the time ΔT and then maintains a constant velocity for the next one-half of ΔT and decelerates to zero velocity during the remaining quarter of the total time ΔT. For constant acceleration and deceleration rates (say a), the total distance covered during the small time step ΔT is $3a\Delta T^2/16$. At the end of the constant velocity travel, the robot senses the position and velocity of each obstacle and decides whether to continue moving in the same direction or to deviate from its path. This is achieved by first determining the predicted position of each obstacle, as follows:

$$P_{\text{predicted}} = P_{\text{present}} + (P_{\text{present}} - P_{\text{previous}}). \qquad (1)$$

The predicted position is the linearly extrapolated position of an obstacle from its current position P_{present} along the path formed by joining the previous P_{previous} and present position. Thereafter, the nearest obstacle forward is determined based on $P_{\text{predicted}}$ values of all obstacles and fuzzy logic technique is applied to find the obstacle-free direction using the rule base dictated by the corresponding 20-bit string. If the robot has to change its path, its velocity is reduced to zero at the end of the time step; otherwise the robot does not decelerate and continues in the same direction with the same velocity $a\Delta T/4$. It is interesting to note that when the latter case happens (the robot does not change its course) in two consecutive time steps, there is a saving of $\Delta T/4$ second in travel time per such occasion. Overall time of travel (T) is then calculated by summing all intermediate time steps needed for the robot to reach its destination. This approach of robot navigation can be easily incorporated in a real-world scenario[1].

[1] In all the simulations here, we have chosen $\Delta T = 4$ sec and $a = 1$ m/s^2. These values make the velocity of the robot in the middle portion of each time step equal to 1 m/sec.

3 Results

We consider four different approaches:

Approach 1: Author-defined fuzzy-logic controller. A fixed set of 20 rules (Table 1) and author-defined membership functions (Figure 3) are used. No optimization method is used to find optimal rule base or to find the optimal membership function distributions.

Approach 2: Optimizing membership functions alone. Only the membership function distributions of condition and action variables are optimized. All 20 rules (Table 1) are used. The bases b_1 and b_2 (refer Figure 3) are coded in 10 bit substrings each. The parameters b_1 and b_2 are decoded in the ranges (1.0, 4.0) cm and (25.0, 60.0) degrees, respectively. Symmetry is maintained in constructing other membership function distributions. In all simulations here, the membership function distribution for deviation is kept the same as that in angle.

Approach 3: Optimizing rule base alone. Only the rule base is optimized in this approach. Author-defined membership functions (Figure 3) are used.

Approach 4: Optimizing membership functions and rule base simultaneously. Membership functions and the rule base are optimized. Here, a GA string is a 40-bit string with first 20 bits denoting the presence or absence of 20 possible rules, next 10 bits are used to represent the base b_1 and the final 10 bits are used to represent the base b_2.

In all runs of the proposed approach, we use binary tournament selection (with replacement), the single-point crossover operator with a probability p_c of 0.9 and the bit-wise mutation operator with a probability p_m of 0.02. A maximum number of generations equal to 100 is used. In every case, a population size of 60 is used. In all cases, $S = 10$ different author-defined scenarios are used to evaluate a solution.

We now apply all four approaches to eight-obstacle problems (in a grid of 25×20 m^2). The optimized travel distance and time for all approaches are presented in Table 2. The first three rows in the table show the performance of

Table 2. Travel distance D (in meter) and time T (in sec) obtained by four approaches

	Approach 1		Approach 2		Approach 3		Approach 4	
Scenario	D	T	D	T	D	T	D	T
1	27.203	28.901	26.077	27.769	26.154	27.872	26.154	27.872
2	26.957	28.943	25.966	27.622	26.026	26.546	26.026	26.546
3	29.848	36.798	28.623	35.164	26.660	34.547	27.139	35.000
4	33.465	43.365	26.396	27.907	26.243	27.512	26.243	27.512
5	32.836	41.781	27.129	33.000	26.543	32.390	27.041	33.000
6	33.464	43.363	28.001	31.335	27.164	31.000	27.164	31.000

all approaches on three scenarios that were used during the optimization process and the last three rows show their performance on new test (unseen) scenarios. The table shows that in all cases, Approaches 2, 3 and 4 have performed better than Approach 1 (no optimization).

Paths obtained using all four approaches for scenario 4 (unseen) are shown in Figure 4. It is clear that the paths obtained by Approaches 3 and 4 (travel

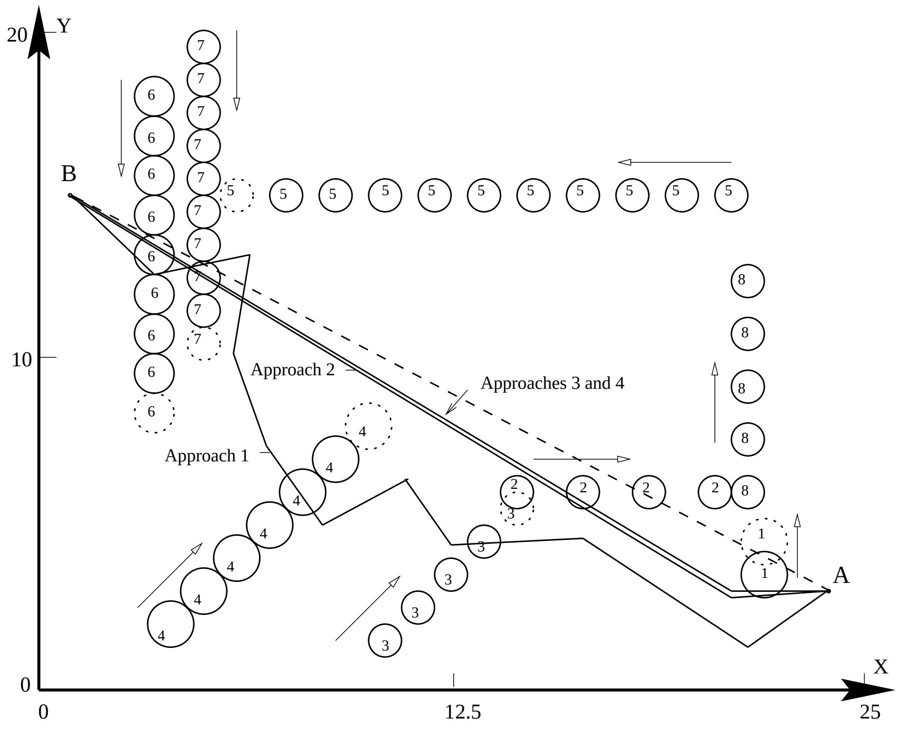

Fig. 4. Optimized paths found by all four approaches for the eight-obstacle problem are shown. The dashed circles mark the critical positions of obstacles found by the FLC.

time 27.512 sec) are shorter and quicker than that obtained by Approaches 1 (travel time 43.365 sec) and 2 (travel time 27.907 sec).

The optimized rule bases obtained using Approaches 3 and 4 are shown in Tables 3 and 4. The optimized membership functions obtained using Approaches 2 and 4 are shown in Figures 5 and 6, respectively. Here, Approach 4 (simultaneous optimization of rules and membership functions) has elongated the membership function distributions so that classification of relative angle is uniform in the range of $(-90, 90)$ degrees. In Approach 3, since membership functions are specified, the GA-optimized solution needed many rules specifying an appropriate action for each value of distance. In Approach 4, membership functions are not

Table 3. Optimized rule base (nine rules) obtained using Approach 3.

distance \ angle	L	AL	A	AR	R
VN					
N		A		A	
F	A	A			
VF	A	A	A	A	A

Table 4. Optimized rule base (five rules) obtained using Approach 4.

distance \ angle	L	AL	A	AR	R
VN		AR			
N	A	A			
F				A	
VF				A	

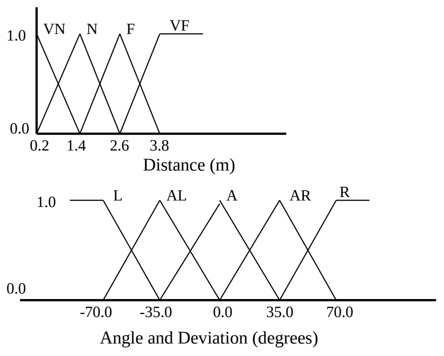

Fig. 5. The optimized membership function obtained using Approach 2.

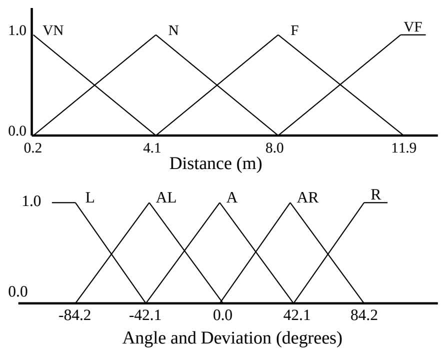

Fig. 6. The optimized membership function obtained using Approach 4.

fixed and the GA finds a solution which elongates the range of each distance value so that only about one rule is enough to classify each distance value.

It is also interesting to note that since all 20 rules are used in Approach 2 and since moving ahead (towards the target) is always optimal, GAs have adjusted the membership function for distance in way so as to have most cases appear as VF. Recall from Table 1 that for all rules with VF distance, action variable is always ahead.

Both Tables 3 and 4 show that there are more rules for L and AL angles than for R and AR angles. This is merely because only 10 scenarios are considered during the optimization process and it could have been that in most cases the critical obstacles come in the left of the robot, thereby causing more rules specifying L or AL to appear in the optimized rule base. By considering more scenarios during the optimization process, such bias can be avoided and equal number of rules specifying left and right considerations can be obtained.

From Table 2, it can be observed that Approach 3 (optimization of rule base only) has resulted in a much quicker path than Approach 2 (optimization of membership function only). This is because finding a good set of rules is more important for the robot than finding a good set of membership functions. Thus, it may be argued that the optimization of rule base is a rough-tuning process and the optimization of the membership function distributions is a fine-tuning process. In most scenarios, the optimized solutions are already obtained during the optimization of rule-base only and optimization of membership function has a marginal effect to improve the solution any further.

Although the performance of Approaches 3 and 4 are more-or-less similar, we would like to highlight that Approach 4 is more flexible and a more practical approach. Since the membership functions used in Approach 3 are well-chosen by the authors, the performance of Approach 3 is good. However, for more complicated problems, we recommend using Approach 4, since it optimizes both the rule base and membership functions needed in a problem.

4 Conclusions

In this study, learning capability of a genetic-fuzzy approach has been demonstrated by finding optimal/near-optimal FLCs for solving motion planning problem of a mobile robot. In the genetic-fuzzy approach, obstacle-free paths have been found locally by using fuzzy logic technique, where optimal membership functions for condition and action variables and an optimal rule base have been found using genetic algorithms. Based on this basic approach, three different approaches have been developed and compared with an author-defined (non-optimized) fuzzy-logic controller (FLC).

The genetic-fuzzy approach developed here is also practical to be used in a real-world situation. One of the major advantages of the proposed method is that the optimization is performed off-line and an optimal rule base is obtained before-hand. Robots can then use this optimal rule base to navigate in presence of unseen scenarios in an optimal or a near-optimal manner. This paper shows how such a rule base can be achieved.

This study can be extended in a number of ways. Since the optimized travel time depends on the chosen incremental time step ΔT, this parameter can also be kept as an action variable. This way, a robot can make a longer leap in a particular obstacle-free direction or make shorter leap if there are a crowd of obstacles in the course of path. In this connection, controlling speed of the robot to allow passing of moving obsctacles may also be considered.

In this study, we have used an author-defined set of 20 rules (Table 1), all of which may not have the optimal combination of condition and action variables. GAs can be used to eliminate this bias by using a different representation scheme, as follows:

`201500130...4`

The above string has 20 positions (representing each combination of action and condition variables) and each position can take one of six values: `0` for absence of the rule, `1` for first option of action variables (say, `L`), `2` for the second option, and so on. This way every solution represented by a 20-position vector represents a valid rule base. Nevertheless, the results of this study show that the proposed GA-fuzzy approach is efficient and a natural choice to the robot navigation problem, which should get more attention in applications of robotics in the coming years.

References

1. Barraquand J., Langlois B. and Latombe J.C., Numerical potential field techniques for robot path planning, *IEEE Trans. Syst., Man and Cybern.*, 22, 224-241, 1992.
2. Beaufrere. B. and Zeghloul, S. "A mobile robot navigation method using a fuzzy logic approach", *Robotica* 13, 437-448 (1995).
3. Canny J. and Reif J., New lower bound techniques for robot motion planning problems, *Proc. 27-th IEEE Symp. on Foundations of Computer Science*, Los Angeles, CA, 49-60, 1987.
4. Donnart J.Y. and Meyer J.A., Learning reactive and planning rules in a motivationally autonomous animat, *IEEE Trans. on Systems, Man and Cybernetics -Part B: Cybernetics*, 26(3), 381-395, 1996
5. Floreano D. and Mondada F., Evolution of Homing Navigation in a Real Mobile Robot, *IEEE Trans. on Systems, Man and Cybernetics -Part B: Cybernetics*, 26(3), 396-407, 1996.
6. Fujimura K. and Samet H., A hierarchical strategy for path planning among moving obstacles, *IEEE Trans. on Robotics and Automation*, 5(1), 61-69, 1989.
7. Fujimura K. and Samet H., Accessibility : a new approach to path planning among moving obstacles, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Ann Arbor, MI, 803-807, 1988.
8. Griswold N.C. and Eem J., Control for mobile robots in the presence of moving objects, *IEEE Trans. on Robotics and Automation*, 6(2), 263-268, 1990.
9. Herrera F., Herrera-Viedma E., Lozano M. and Verdegay J.L., Fuzzy tools to improve genetic algorithms, *Proc. of the Second European Congress on Intelligent Techniques and Soft Computing*, 1532-1539, 1994.
10. Karr C., Design of an adaptive fuzzy logic controller using a genetic algorithm, *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, San Mateo CA, 450-457, 1991.
11. Latombe J.C., *Robot Motion Planning*, Kluwer Academic Publishing, Norwell, MA, 1991.
12. Lamadrid J.G., Avoidance of obstacles with unknown trajectories: Locally optimal paths and periodic sensor readings, *The Int. Jl. of Robotics Research*, 496-507, 1994.
13. Lee M. and Takagi H., Integrating design stages of fuzzy systems using genetic algorithms, *Proc. of the Second IEEE Int. Conf. on Fuzzy Systems*, 612-617, 1993
14. Martinez A. et al, Fuzzy logic based collision avoidance for a mobile robot, *Robotica*, 12, 521-527, 1994.
15. Okutomi M. and Mori M., Decision of robot movement by means of a potential field, *Advanced Robotics*, 1(2), 131-141, 1986.
16. Newman W.S. and Hogan N., High speed robot control and obstacle avoidance using dynamic potential functions, *Proc. IEEE Int. Conf. on Robotics and Automation*, 14-24, 1987.
17. Sharma R., A probabilistic framework for dynamic motion planning in partially known environments, *Proc. of IEEE Int. Conf. on Robotics and Automation*, Nice, France, 2459-2464, 1992.
18. Takeuchi T., Nagai Y. and Enomoto Y., Fuzzy Control of a Mobile Robot for Obstacle Avoidance, *Information Sciences*, 45(2), 231-248, 1988.
19. Thrift P., Fuzzy logic synthesis with genetic algorithms, *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, San Mateo CA, 509-513, 1991.