Optimum synthesis of fuzzy logic controller for trajectory tracking by differential evolution

H. Nejat Pishkenari\(^a\), S.H. Mahboobi\(^b\), A. Alasty\(^a,\)*

\(^a\) Center of Excellence in Design, Robotics and Automation, School of Mechanical Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-9567, Iran
\(^b\) Institute for Nanoscience and Nanotechnology, Sharif University of Technology, Tehran, P.O. Box 14588-89694, Iran

Received 9 January 2010; revised 2 November 2010; accepted 19 January 2011

KEYWORDS

Trajectory tracking; Mobile robot; Differential evolution; Genetic algorithm; Fuzzy membership functions.

Abstract

Differential Evolution (DE) and Genetic Algorithms (GA) are population based search algorithms that come under the category of evolutionary optimization techniques. In the present study, these evolutionary methods have been utilized to conduct the optimum design of a fuzzy controller for mobile robot trajectory tracking. Comparison between their performances has also been conducted. In this paper, we will present a fuzzy controller to the problem of mobile robot path tracking for a CEDRA rescue robot. After designing the fuzzy tracking controller, the membership functions will be optimized by evolutionary algorithms in order to obtain more acceptable results.

© 2011 Sharif University of Technology. Production and hosting by Elsevier B.V. All rights reserved.

1. Introduction

The optimization of non-linear and sophisticated problems has been an interesting issue in science and technology [1]. In recent years, evolutionary algorithms have been applied to the solution of the above problems in many engineering applications. The best known algorithms in this category include Genetic Algorithms (GA), Evolutionary Programming (EP), Evolution Strategies (ES) and Genetic Programming (GP). There are some hybrid algorithms that blend features of these algorithms and are hard to classify, which have been referred to as Evolutionary Computation (EC) methods. In general, only the information related to the objective function is required by EC methods [2]. Differential Evolution (DE), developed by Price and Storn [3], is one of the best EC methods which provides one of the most efficient algorithms for solving the real-valued function. DE provides a very high convergence speed and has few control parameters in comparison to other major evolutionary algorithms [4].

In the present study, the Differential Evolution method has been utilized to conduct the optimum design of a fuzzy controller for mobile robot trajectory tracking. The problem of trajectory tracking for mobile robots has been an attractive issue in the robotic field in recent years. Implementation of classical control methods for this category of robot is usually complicated and time consuming, especially in the case of high mobility robots. Our purpose is to control a certain high mobility rover that is used for rescue operations.

In recent years, expert systems like fuzzy logic have been used to achieve the above mentioned task. In [5], Van Eck proposed a fuzzy controller for an autonomous boat without the nonlinear dynamics model of a vehicle. Sugeno and Murakami [6] has designed a fuzzy controller, based on the fuzzy modeling of human operator control actions, to navigate and park a car. Larkin [7] has proposed a fuzzy controller for aircraft flight control, where the fuzzy rules are generated by interrogating an experienced pilot and asking him a number of highly structured questions. Kurnaz et al. [8] have proposed a fuzzy logic based autonomous navigation controller for UAVs (Unmanned Aerial Vehicles). In [9], Wang and Mendel have solved the same problem by generating fuzzy rules using learning algorithms. Some other researchers have recently utilized fuzzy control approaches for mobile robot path tracking [10–14].

Design of a Fuzzy Logic System (FLS) includes the design of a rule base, input scale factors, output scale factors and, finally,
the design of the membership functions. A genetic learning process for the membership function design, coupled with a heuristic method for the rule base design, has been proposed in [15]. Also a fuzzy training process for input scale factors has also been proposed in [16]. Angelov and Buswell [17] have used genetic algorithms for automatic generation of a fuzzy rule-base. Cheong and Lai [18] have recently utilized differential evolution for designing a hierarchical fuzzy logic controller.

The subject of this paper is restricted to the tuning of membership functions. Researchers have used many different methods over the past decade to optimize fuzzy membership functions. These methods include genetic algorithms [19–24], neural networks [25], evolutionary programming [26], geometric methods [27], fuzzy equivalence relations [28], heuristic methods [29], gradient descent [30] and particle swarm optimization [31].

One method of removing the uncertainty associated with the selection of these variables is the use of evolutionary algorithms (e.g. GA and DE). In the present research, DE has been utilized for the optimization of a fuzzy control system, regarding its membership functions. As a case study, the path tracking for a mobile robot has been chosen and an optimized fuzzy controller will be set up for this aim. In addition, the optimization will be performed by GA to enable one to compare the efficacy of these approaches.

2. Kinematic model of the robot

A top view of a CEDRA rescue robot [32–34] (see Figure 1) is shown in Figure 2. Interaction between the robot and the path curve may be seen in Figure 3. This mobile robot consists of six wheels, the front and rear of which are steerable, and the remaining four are mounted beside the robot. All robot wheels are active and their speeds are adjusted according to the desired path. Robot kinematic equations are as follows:

\[
\begin{align*}
\dot{x}_c(t + \Delta t) & = x_c(t) + V(t) \cos(\theta(t)) \cos(\phi_c(t)) \Delta t, \\
\dot{y}_c(t + \Delta t) & = y_c(t) + V(t) \cos(\theta(t)) \sin(\phi_c(t)) \Delta t, \quad (1) \\
\phi_c(t + \Delta t) & = \phi(t) + \frac{2V(t)}{L} \sin(\theta(t)) \Delta t, \quad (2)
\end{align*}
\]

where \(V(t)\) is the speed of the front and rear wheels, and the other variables are shown in Figures 2 and 3.

3. Fuzzy logic controller design

The general configuration of the fuzzy controller, which is divided into four main parts, is shown in Figure 4. The first part is the fuzzifier which converts crisp values into fuzzy sets. Fuzzy sets enter the inference engine as inputs. Fuzzy decisions will be made upon the fuzzy rule base and the results will pass through a defuzzifier stage. Defuzzification is the inverse process of fuzzification.

Each fuzzy set consists of a number of membership functions to describe the heuristic variables in a mathematical manner. All membership functions used in input and output sets are in
the form of a trapezoid function, formulated as in Eq. (4) [35].
The trapezoid function can model different functions better
than the triangular function and, at a specific condition, when
is equal to \(c\), it is transformed to the triangular function.

\[
\mu_A(x; a, b, c, d) = \begin{cases}
0, & x \in (\infty, a) \\
x - a, & x \in (a, b) \\
b - a, & x \in [b, c) \\
d - x, & x \in [c, d) \\
0, & x \in [d, \infty)
\end{cases}
\] (4)

3.1. Fuzzy input sets

In a fuzzy logic controller, the input to the controller (cur-
vature, position error, and orientation error) is converted into
a series of fuzzy sets. The number and exact shape of these
fuzzy sets critically determine the performance of the con-
troller. These fuzzy sets describe qualitative situations in which
the output of the controller is qualitatively different.

In other words, whenever the desired behavior (e.g. change
from going straight to turning left or change from fast to
medium speed) of the controller changes in an input situation,
a fuzzy set is created to represent this case.

Curvature consists of three fuzzy sets: Left Curvature,
Straight and Right Curvature. The fuzzification of the positional
error includes five sets: NegHighDist (NHD) (Neg: Negative,
Dist: Distance), NegLowDist (NLD), ZeroDist (ZD), PosLowDist
(PLD) (Pos: Positive) and PosHighDist (PHD). It is desirable that
the robot be on the line (ZeroDist). Assume that the robot is
off the path, then the desired behavior is either turn right/left
towards the path, drive straight towards the path, and then turn
left/right to straighten out. Therefore, we require two extra sets
on each side of the path.

Similar reasoning leads to the design of the fuzzy sets for
the orientation error. A total of five sets have been used to
describe different cases: NegHighAngle (NHA), NegLowAngle
(NLA), ZeroAngle (ZA), PosLowAngle (PLA) and PosHighAngle
(PHA).

3.2. Fuzzy output sets

There are two outputs from the fuzzy controller to the robot:
(a) speed and (b) steering angle. There are four membership
functions for describing the speed heuristic variable: Zero, Slow,
Medium and Fast. The steering angle is determined using five
membership functions; SharpLeft (SL), LowLeft (LL), Straight
(ST), LowRight (LR) and SharpRight (SR). A crisp output value
is then computed from this fuzzy set. This step is called
defuzzification. In this research, we used the well-known
centroid defuzzification method which uses the center of the area
as the crisp output value.

3.3. Fuzzy rule base

Given these fuzzy input sets, a fuzzy controller uses a set of
fuzzy rules to specify the desired control behavior. After the
design of the fuzzy input and output sets, the design of the
fuzzy rules is straightforward. There are a total of 5\(^5\)3 =
75 possible different input configurations. For each of these
input configurations, a rule was specified to indicate the desired
speed and directional settings. Examples of fuzzy rules appear
in Table 1.

4. Evolutionary optimization: Genetic Algorithm (GA) and
Differential Evolution (DE)

Algorithms for function optimization are generally limited
to convex regular functions. However, many functions are
multi-model, discontinuous and non-differentiable. Genetic
Algorithms (GAs) are a class of stochastic search techniques,
loosely based on ideas from biological evolution, which have
been used successfully for a great variety of different problems
(e.g. [36–38]).

The GA searches for an optimal solution from a population
of candidate solutions according to an objective function, which
is used to establish the fitness of each candidate as a solution.
The governing process in the search is the application of
appropriate breeding operators to candidate solutions in a given
generation to form the candidates for the next generation.
These operators are designed to preserve the most successful
aspects of candidate fitness until the best possible solution is
attained.

At each generation, a new set of approximations is created
by the process of selecting individuals according to their level
of fitness in the problem domain, and breeding them together
using operators borrowed from natural genetics. This process
leads to the evolution of populations of individuals that are
better suited to their environment than the individuals that
they were created from, just as in natural adaptation [39].

Differential Evolution [3] is an improved version of Genetic
Algorithm [36–39] for faster optimization. The main difference
between DE and its antecedent, GA, is the use of real coding
instead of binary coding to represent problem parameters. We
can list many advantages for DE, including simple structure,
ease of use, low number of control parameters and speed. The
simple adaptive scheme used by DE ensures that these mutation
increments are automatically scaled to the correct magnitude.

DE uses non-uniform crossover in which the children inherit
the parameter values from the parent vectors in unequal propor-
tions. In reproduction, the child vector competes against one of its
parents in order to be selected.

DE utilizes a new scheme for generating trial parameter vec-
tors. In DE, new parameter vectors are generated by adding the
difference vector between two population members to a third
member with an adjustable weight factor. Next, a comparison
will be made between the newly generated vector and a prede-
termined vector, and the one with the lower objective function
value will survive [4].

5. Membership function optimization

The essential point of designing an FLC lies in the selection
of high-performance membership functions that represent
human expert interpretation of linguistic variables, because
different membership functions determine the extent to which
the rules affect the action and hence the performance [40].

The existing iterative approaches for choosing the member-
ship functions are basically a manual trial-and-error process
and lack learning capability or autonomy. Therefore, the more
efficient and systematic evolutionary optimization algorithm,
which acts on the survival-of-the-fittest, has been applied to
the FLC design for searching the poorly understood, irregu-
lar and complex membership function space with improved
performance.

From the point of view of an evolutionary search, member-
ship functions can be seen as functions, the parameters of which
are necessary to achieve optimization in general terms and are
Table 1: Fuzzy rule base.

| No. | Input | Output | No. | Input | Output |
|-----|-------|--------|-----|-------|--------|
|     | Curvature | Input | d | ∆φ | V | θ | Curvature | Input | d | ∆φ | V | θ |
| 1   | StraightLine | ZD | ZA | Fast | ST | 39 | LeftCircle | ZD | NLA | Slow | ST |
| 2   | StraightLine | ZD | PLA | Med | LL | 40 | LeftCircle | ZD | NLA | Slow | LR |
| 3   | StraightLine | ZD | PHA | Slow | SL | 41 | LeftCircle | PHD | NHA | Slow | LR |
| 4   | StraightLine | ZD | NLA | Slow | LR | 42 | LeftCircle | PHD | NLA | Slow | ST |
| 5   | StraightLine | ZD | NHA | Slow | SR | 43 | LeftCircle | PHD | ZA | Slow | LL |
| 6   | StraightLine | PHD | NHA | Slow | ST | 44 | LeftCircle | PHD | PLA | Zero | SL |
| 7   | StraightLine | PHD | NLA | Med | LL | 45 | LeftCircle | PHD | PHA | Slow | SL |
| 8   | StraightLine | PHD | PHA | Slow | SL | 46 | LeftCircle | PLD | NHA | Slow | LR |
| 9   | StraightLine | PHD | PLA | Slow | SL | 47 | LeftCircle | PLD | NLA | Slow | LL |
| 10  | StraightLine | PHD | ZA | Med | SL | 48 | LeftCircle | PLD | ZA | Slow | SL |
| 11  | StraightLine | PLD | NLA | Slow | LR | 49 | LeftCircle | PLD | PLA | Zero | SL |
| 12  | StraightLine | PLD | PHA | Slow | LL | 50 | LeftCircle | PLD | PHA | Slow | LR |
| 13  | StraightLine | PLD | ZA | Slow | LL | 51 | RightCircle | NHD | NHA | Slow | LL |
| 14  | StraightLine | PLD | PLA | Slow | LL | 52 | RightCircle | NHD | NLA | Slow | ST |
| 15  | StraightLine | PLD | PHA | Slow | SL | 53 | RightCircle | NHD | ZA | Slow | LR |
| 16  | StraightLine | NHD | ZA | Slow | SL | 54 | RightCircle | NHD | PLA | Slow | SR |
| 17  | StraightLine | NHD | PLA | Slow | SR | 55 | RightCircle | NHD | PHA | Slow | SR |
| 18  | StraightLine | NHD | NLA | Slow | LR | 56 | RightCircle | NLD | NHA | Slow | LL |
| 19  | StraightLine | NHD | NLA | Slow | LR | 57 | RightCircle | NLD | NLA | Slow | LR |
| 20  | StraightLine | NHD | NLA | Med | SL | 58 | RightCircle | NLD | ZA | Slow | SR |
| 21  | StraightLine | NLD | PHA | Slow | SL | 59 | RightCircle | NLD | PLA | Slow | SR |
| 22  | StraightLine | NLD | PHA | Slow | LR | 60 | RightCircle | NLD | PLA | Zero | SR |
| 23  | StraightLine | NLD | PLA | Slow | LR | 61 | RightCircle | ZD | ZA | Slow | LR |
| 24  | StraightLine | NLD | NLA | Slow | SR | 62 | RightCircle | ZD | PLA | Zero | SR |
| 25  | StraightLine | NLD | NHA | Slow | SR | 63 | RightCircle | ZD | PHA | Slow | SR |
| 26  | LeftCircle | NHD | NLA | Slow | SR | 64 | RightCircle | ZD | NLA | Slow | ST |
| 27  | LeftCircle | NHD | NLA | Slow | LR | 65 | RightCircle | ZD | NLA | Slow | LR |
| 28  | LeftCircle | NHD | ZA | Slow | LR | 66 | RightCircle | PHD | NHA | Slow | SL |
| 29  | LeftCircle | NHD | PLA | Slow | LL | 67 | RightCircle | PHD | NLA | Slow | LL |
| 30  | LeftCircle | NHD | PHA | Slow | SL | 68 | RightCircle | PHD | ZA | Slow | ST |
| 31  | LeftCircle | NLD | NLA | Slow | SR | 69 | RightCircle | PHD | PLA | Zero | LR |
| 32  | LeftCircle | NLD | NLA | Slow | LR | 70 | RightCircle | PHD | PLA | Zero | SR |
| 33  | LeftCircle | NLD | ZA | Slow | ST | 71 | RightCircle | PLD | NHA | Slow | SL |
| 34  | LeftCircle | NLD | PLA | Slow | LL | 72 | RightCircle | PLD | NLA | Slow | LL |
| 35  | LeftCircle | NLD | PHA | Zero | SL | 73 | RightCircle | PLD | ZA | Slow | ST |
| 36  | LeftCircle | ZD | ZA | Slow | LL | 74 | RightCircle | PLD | PLA | Slow | LR |
| 37  | LeftCircle | ZD | PLA | Zero | SL | 75 | RightCircle | PLD | PHA | Slow | SR |
| 38  | LeftCircle | ZD | PHA | Zero | SL | 76 | RightCircle | PLD | PHA | Slow | SR |

Figure 5: Sample of parametric membership function.

The objective function is defined as the summation of the distances of the robot from the desired path (Eq. (5)). The penalty function will be triggered when the constraints of the problem are violated (Eq. (6)). The first constraint is the limitation on the distance of the robot center from the path during tracking (Eq. (7)). The second constraint is the acceptable range for the design parameters of the membership functions (Eq. (8)).
The optimization process has been fulfilled using the two evolutionary algorithms mentioned earlier, GA and DE. We have developed a modified version of DE, which preserves the initial variable intervals in order to eliminate the unacceptable solution space. The algorithm parameters and their performance comparisons have been shown in Table 2. As can be seen, the number of population and generations is set to be equal in both methods, and the achieved error summation and mean have been compared. The convergence plot for optimization by DE is shown in Figure 6, which depicts how fast the convergence takes place for the utilized method.

The mentioned errors correspond to the tracking of an assumed test path with a highly complicated shape and sharp edges. Results have demonstrated that both algorithm performances are mostly the same, although the error summation

| Parameter name | GA  | DE  |
|----------------|-----|-----|
| $N_P$ (number of populations) | 100 | 100 |
| $N_{var}$ (number of variables) | 20  | 20  |
| Maximum generation | 200 | 200 |
| $F$ (scaling factor) | –   | 0.6 |
| $C_r$ (crossover probability) | –   | 0.7 |
| $G_{gap}$ (generation gap) | 0.9 | –   |
| $L_{ind}$ (length of individual vars.) | 10  | –   |
| Crossover rate | 0.9 | –   |
| Mutation rate | 0.07 | –   |
| Error summation | 19,879 | 19,377 |
| Number of sampling | 399 | 411 |
| Average error (mm) | 49.9 | 47.2 |
and mean achieved by DE is slightly lower than its competitor; GA. As can be seen, DE has gained an average error 5.4% lower than GA. Another advantage of DE is its lower number of control parameters, which makes it easier to use. The most time-consuming part of computations arises from calculation of the objective function, so by considering that we have equal amounts of $N_P$ and Maximum Generation, the total time of computations is approximately independent of the optimization algorithm.

Simulations are presented using a complicated path including several break points to show controller performance. At any instant, the position and orientation of the robot are assumed to be determined by an accelerometer and tilt sensor. The data sampling time of the controller is set to 0.5 s. Although the sampling time is relatively large, the tracking controller has responded in an acceptable way.

Figure 7 shows the response of an initial fuzzy logic controller. Figures 8 and 9 depict the performance of an optimized fuzzy logic controller, using GA and DE, respectively. As can be seen, deviation from the desired path has been extremely reduced after optimization independent of algorithm type. We can observe similar results of the two utilized techniques in practice also. Since we have designed the fuzzy logic controller for an elaborate path including several break points, it is expected to have a good response, but not optimal for other paths.

Because of the complexity and variety of curve sections, it is expected that the controller shows an acceptable behavior in case of any arbitrary trajectory. In order to illustrate this feature, the controller optimized by DE has been tested over a different path and its response is shown in Figure 10. Finally, the optimized forms of membership functions for input and output variables are depicted in Figures 11 and 12, respectively.

6. Conclusion

In this paper, a fuzzy logic controller has been developed for path tracking of a rescue robot. In order to tune the membership functions, (to achieve minimum deviation from the desired path) we have used two evolutionary methods; Genetic Algorithm and Differential Evolution.

In this trajectory tracking application, the fitness function evaluates the robot’s path, taking into account distance and orientation error from the desired path. In the present research, the control signals are robot velocity and steering angle, which are determined based on deviations between the desired and actual position and orientation. It should be mentioned
that the environment is completely known and the path is prescribed and, then, the optimized fuzzy controller is proposed for tracking this path. Here the fitness function evaluates the robot’s path, taking into account the distance and orientation error from the desired path. The optimum membership functions and weights of the rules differ from one path to another. Therefore, a very complicated desired path has been chosen in the optimization process to assure the robustness of the optimal fuzzy controller.

The improved performance of the controller relative to the original has been shown via tracking simulation over a complicated path. According to the results, the performance of the optimized controller is more acceptable than the initial controller. A comparison between the two mentioned algorithms has also been addressed and they were found to have similar results, except for a slightly lower error, and less control parameters for the DE method.

References

[1] Flosdus, C.A. Nonlinear and Mixed-Integer Optimization, Oxford University Press, New York, USA (1995).
[2] Dasgupta, D. and Michalewicz, Z. Evolutionary Algorithms in Engineering Applications, Springer, Germany (1997).
[3] Price, K. and Storn, R. “Differential evolution - A simple evolution strategy for fast optimization”, Dr. Dobb’s Journal, 22(4), pp. 18–24 (1997).
[4] Babu, B.V. and Munawar, S.A. “Differential evolution strategies for optimal design of shell-and-tube heat exchangers”, Chemical Engineering Science, 62, pp. 3720–3737 (2009).
[5] Vaneeck, T.W. “Fuzzy guidance controller for an autonomous boat”, IEEE Control Systems Magazine, 17(2), pp. 43–51 (1997).
[6] Sugeno, M. and Murakami, M. “An experimental study and fuzzy parking control using a model car”, In Industrial Applications of Fuzzy Control, pp. 125–128, North-Holland, Amsterdam (1985).
[7] Larkin, L.I. “A fuzzy logic controller for aircraft flight control”, In Industrial Applications of Fuzzy Control, pp. 87–107, North-Holland, Amsterdam (1985).
[8] Kurnaz, S., Cetin, O. and Kaynak, O. “A fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles”, Journal of Intelligent and Robotic Systems, 54, pp. 229–244 (2009).
[9] Wang, L. and Mendel, J. “Generating fuzzy rules by learning from examples”, IEEE Transactions on Systems, Man and Cybernetics, 22(6), pp. 1414–1427 (1992).
[10] Antonelli, G., Chiaverini, S. and Fusco, G. “A fuzzy-logic-based approach for mobile robot path tracking”, IEEE Transactions on Fuzzy Systems, 15(2), pp. 211–221 (2007).
[11] Hui, N.B., Mahendar, V. and Prathar, D.K. “Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches”, Fuzzy Sets and Systems, 157(16), pp. 2171–2204 (2006).
[12] Lee, T.H., Lam, H.K., Leung, F.H.F. and Tam, P.K.S. “A practical fuzzy logic controller for the path tracking of wheeled mobile robots”, IEEE Control Systems Magazine, 23(2), pp. 60–65 (2003).
[13] Hwang, C.L. and Lin, H.Y. “Trajectory tracking of robot using a fuzzy decentralized sliding-mode tracking control”, FUZZ-IEEE 2004, Budapest, Hungary (2004).
[14] Giap, N.H., Shin, J.H. and Kim, W.H. “Adaptive robust fuzzy control for path tracking of a wheeled mobile robot”, Artificial Life Robotics, 13, pp. 134–138 (2008).
[15] Cordon, O., Herrera, F., Magdalena, L. and Villar, P. “A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base”, Information Sciences, 136(1–4), pp. 85–107 (2001).
[16] Daugherity, W., Rathakrishnan, B. and Yen, J. “Performance evaluation of a self-tuning fuzzy controller”, IEEE International Conference on Fuzzy Systems, San Diego, CA, USA, pp. 389–397 (1992).
[17] Angelov, P.P. and Buswell, R.A. “Automatic generation of fuzzy rule-based models from data by genetic algorithms”, Information Sciences, 150, pp. 17–31 (2003).
[18] Cheong, F. and Lai, R. “Designing a hierarchical fuzzy logic controller using the differential evolution approach”, Applied Soft Computing, 7, pp. 481–491 (2007).
[19] Simon, D. and El-Sharief, H. “Fuzzy logic for digital phase-locked loop filter design”, IEEE Transactions on Fuzzy Systems, 3, pp. 211–218 (1995).
[20] Wong, C.C., Lin, Y.H., Lee, S.A. and Tsai, C.H. “GA based fuzzy system design in FPGA for an omnidirectional mobile robot”, Journal of Intelligent and Robotic Systems, 44, pp. 327–347 (2005).
[21] Pishkenari, H.N., Mahboobi, S.H. and Alasty, A. “Trajectory tracking of a mobile robot using fuzzy logic tuned by genetic algorithm”, International Journal of Engineering, Transactions A: Basics, 19(1), pp. 95–104 (2006).
[22] Doitsidis, L., Tsourveloudis, N.C. and Piperidis, S. “Evolution of fuzzy controllers for wall-following behavior in mobile robots”, Soft Computing, 10(10), pp. 881–899 (2006).
[23] Goddard, J., Parrales, R., Lopez, A. and Barro, S. “Evolutionary learning of a fuzzy controller for the control of wheeled mobile robots”, Proceedings of IEEE, 89(9), pp. 1318–1333 (2001).
[24] Angelov, P.P. and Buswell, R.A. “Automatic generation of fuzzy rule-based system database”, Journal of Intelligent and Robotic Systems, 56, pp. 469–484 (2009).
[25] Pishkenari et al. / Scientia Iranica, Transactions B: Mechanical Engineering 18 (2011) 261–267