Optimization of Fuzzy Control Systems for Mobile Robots Based on PSO

David de la O, Oscar Castillo and Abraham Meléndez

Abstract This paper describes the optimization of a navigation controller system for a mobile autonomous robot using the PSO algorithm to adjust the parameters of each fuzzy controller, the navigation system is composed of 2 main controllers, a tracking controller and a reactive controller, plus an integrator block control that combines both fuzzy inference systems (FIS). The integrator block is called Weighted Fuzzy Inference System (WFIS) and assigns weights to the responses in each block of behavior in order to combine them into a single response. A comparison with the results obtained with genetic algorithms is also performed.

1 Introduction

The mobile robots must be able to operate in a real environment, and navigate in an autonomous manner. In this case an intelligent control strategy that can handle the uncertainty can be implemented by the working environment, while comparing with the performance in real time of a relatively low computational load.

One of the applications of fuzzy logic is the design of fuzzy control systems. The success of this control lies in the correct selection of the parameters of fuzzy controller; it is here where the Particle Swarm Optimization (PSO) metaheuristic will be applied, which is one of the most used for optimization with real

D. de la O · O. Castillo (✉) · A. Meléndez
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: ocastillo@tectijuana.mx

D. de la O
e-mail: ddsh@live.com

A. Meléndez
e-mail: abraham.ms@gmail.com

O. Castillo et al. (eds.), *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, Studies in Computational Intelligence 547,
DOI: 10.1007/978-3-319-05170-3_14,

parameters. PSO, because of their ease of implementation, converges faster than the Evolutionary Algorithms (EA) has better performance [9–30].

The present research work deals with the problem of mobile robot autonomy, for which a fuzzy control system is developed using fuzzy logic and the PSO algorithm [31–52].

This paper is organized into four sections as follows: Sect. 2: the points are developed and progress of the research work is shown. Sections 3 and 4: In part disclose the theory underlying the present work, in which issues such as fuzzy logic, PSO algorithm and a bit on the operation of autonomous mobile robot are discussed. Section 5 shows the results of the simulations are presented.

2 Mobile Robots

The particular mobile robot considered in this work is based on the description of the Simulation toolbox for mobile robots [1], which assumes a wheeled mobile robot consisting of one conventional, steered, unactuated and not-sensed wheel, and two conventional, actuated, and sensed wheels (conventional wheel chair model). This type of chassis provides two DOF (degrees of freedom) locomotion by two actuated conventional non-steered wheels and one unactuated steered wheel. The Robot has two degrees of freedom (DOFs): y-translation and either x-translation or z-rotation [1]. Figure 1 shows the robot's configuration, it has 2 independent motors located on each side of the robot and one castor wheel for support located at the front of the robot.

The kinematic equations of the mobile robot are as follows:

Equation 1 shows the sensed forward velocity solution [2]

$$\begin{pmatrix} V_{Bx} \\ V_{By} \\ \omega_{Bz} \end{pmatrix} = \frac{R}{2l_a} \begin{bmatrix} -l_b & l_b \\ -l_a & -l_a \\ -1 & -1 \end{bmatrix} \begin{pmatrix} \omega_{W1} \\ \omega_{W2} \end{pmatrix} \tag{1}$$

Equation 2 shows the Actuated Inverse Velocity Solution [3]

$$\begin{pmatrix} \omega_{W1} \\ \omega_{W2} \end{pmatrix} = \frac{1}{R(l_b^2+1)} \begin{bmatrix} -l_a l_b & -l_b^2-1 & -l_a \\ l_a l_b & -l_b^2-1 & l_a \end{bmatrix} \begin{pmatrix} V_{Bx} \\ V_{By} \\ \omega_{Bz} \end{pmatrix} \tag{2}$$

Under the Metric system are define as:

V_{Bx}, V_{By} Translational velocities $\left[\frac{m}{s}\right]$,
ω_{Bz} Robot z-rotational velocity $\left[\frac{rad}{s}\right]$,
ω_{W1}, ω_{W1} Wheel rotational velocities $\left[\frac{rad}{s}\right]$,
R Actuated wheel radius [m],
l_a,l_b Distances of wheels from robot's axes [m].

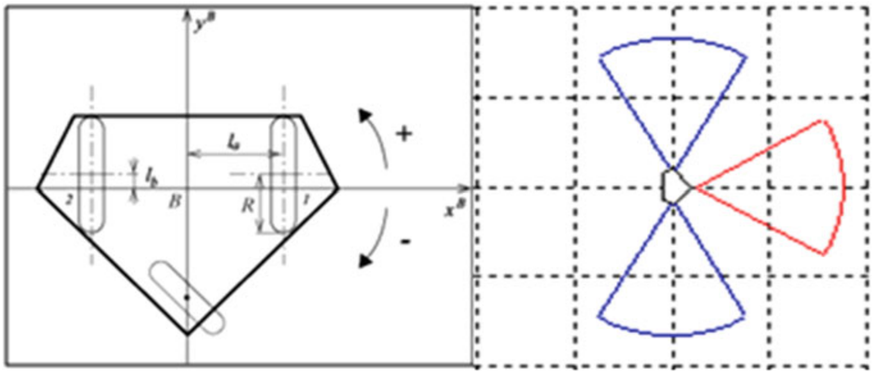


Fig. 1 Kinematic coordinate system assignments [2]

3 Navigation Control System

The proposed control system consists of three main fuzzy blocks: two controllers are based on the behavior of the robot (tracking and reactive) and one is in charge of combining the responses of the other two drivers, called block integration.

Each controller has a specific behavior, the problem is that they seem to be in conflict with each other when an unexpected obstacle appears, because if the route is planned when obstacles are present, the route can be designed to avoid them but when there are obstacles we do not realize that the two behaviors are in contradiction, one is designed to prevent the collision object and the other to keep the robot on the path.

The most common solution is to simply change among drivers as needed. However, this method is not very effective because of the lack of knowledge of the two controllers on to one another, the reagent remains the robot free from a collision, but may redirect the robot further from your destination to a point at which the tracking controller can no longer find their way back to the reference, or the tracking controller can directly guide the robot to an obstacle if the control reagent provides no actuation time. The proposed reference for navigation control always has both active controls and responses combined to create the movement of the robot. The integration is performed with another block called diffuse WFIS [2] (Weight-Fuzzy Inference System) so that this controller assigning weights is made responsive to each of the response values of the drivers.

The inputs are gathered from the information we can collect from the robot (sensors) or the environment by other means (cameras) and from this we need to create the knowledge rule base to give higher activation values to the response. If we want to take the lead on the robot movement one example of the rule is the following (if Front_Sensor_Distance is Close Then TranckingWeight is Medium and ReactiveWeight is Medium), both of our controls provide the right and left

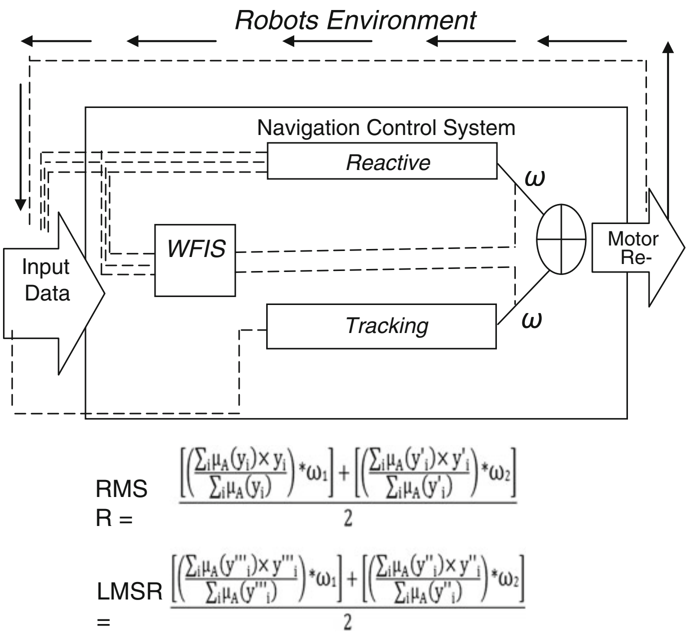


Fig. 2 Navigation control system [2]

motor speed and we combine each one with the weight given by the WFIS block. Figure 2 shows the proposed navigation control [2].

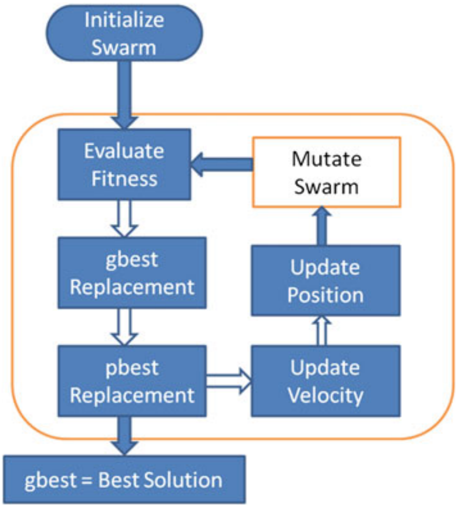
4 PSO Algorithms

The Particle Swarm Optimization Algorithm (PSO) was applied to each of the design problems in order to find the best fuzzy controller reactive and tracking. The purpose of using a behavior-based method is to find the best controllers of each type and this can be achieved by PSO as it searches along the search space of the solution, which combines the knowledge of the best controllers (particles), and we can handle the exploration and exploitation throughout the iterations. The main task of the algorithm is to convert the particle into a FIS and then evaluate each particle to determine its performance. Figure 3 shows the flowchart of the PSO.

4.1 Particle Encoding

The particle consists of 60 real-valued vectors, representing the parameters for the triangular membership function; we use five membership functions for each variable. This encoding is shown Fig. 4.

Fig. 3 Flowchart of the PSO



4.2 Reactive Controller

The role of reactive control is to apply the same capacity when we are driving, i.e. to react to unexpected situations, traffic jams, traffic lights, etc., but in a more basic concept. The objective is to guide the robot through the maze avoiding any collision. It is our goal to optimize the robot to find the exit of the maze, we used a maze to optimize the reactive control because of the feature that conditions the simulation, i.e. it is a closed space in which the robot cannot move easily and each wall is considered an obstacle to the robot to avoid while moving. The FIS is Mamdani type; each consisting of 3 entries, namely the distances obtained by the sensors of the robots described in Sect. 2, and 2 outputs that control the speed of the servo motors on the robot, all this information is encoded in each particle.

4.3 Tracking Controller

The tracking controller has the responsibility to keep the robot on the right track, given a reference; the robot will move on the reference and keep it on the road, allowing moving from point A to B without obstacles present in the environment.

The controller will work keeping the error (Δep, $\Delta\theta$) in the minimum values, Fig. 5, these minimum values are the relative position error and the relative error of the orientation of the front of the robot, the Mamdani fuzzy system and its 2 inputs are (Δep, $\Delta\theta$) and two outputs which control the speed of each servomotor of the robot (Fig. 6).

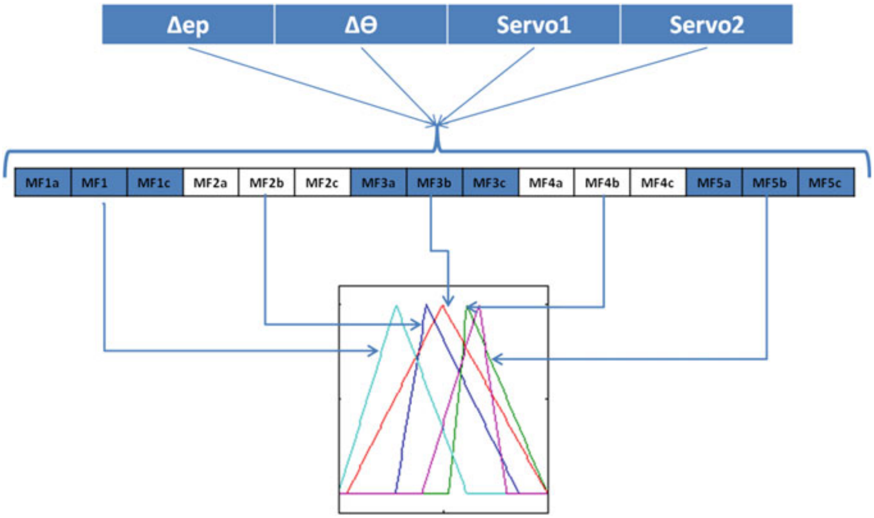


Fig. 4 Particle encoding

Controller performance is measured using the equation of the mean square error between the reference and the robot trajectory. We perform each test three times and take the average, above, below and random reference.

In Fig. 7 we can see the overall flowchart of the PSO. At the point of evaluation of the particle, we measure the effectiveness of the tracking controller FIS (Inference System Fuzzy) in our toolbox of evidence, which will be in a closed circuit with a given reference by a straight line [2, 4–7] environment.

4.4 WFIS Controller

The function of the WFIS control is to correctly combine the two behaviors of tracking and reaction and obtain a new global behavior that resembles the same ability that we apply when we are driving, that is to react to unexpected objects, but in a more basic concept and ability, to the problem that is the navigation of the robot. A forward moving behavior response out of the global control is desired. The objective is to guide the robot through the reference avoiding any collision with any obstacle present. It's not our objective to optimize the robot to find the maze exit. We use a closed space where the robot cannot easily wonder off and each wall is considered an obstacle to the robot that it must avoid while it moves around. The FISs are Mamdani fuzzy systems [8], each consisting of three inputs, which are the distances obtained by the robots sensors described on Sect. 2, and two outputs that are the weights that will be used to integrate the responses of the other two controllers. All this information is encoded into each particle.

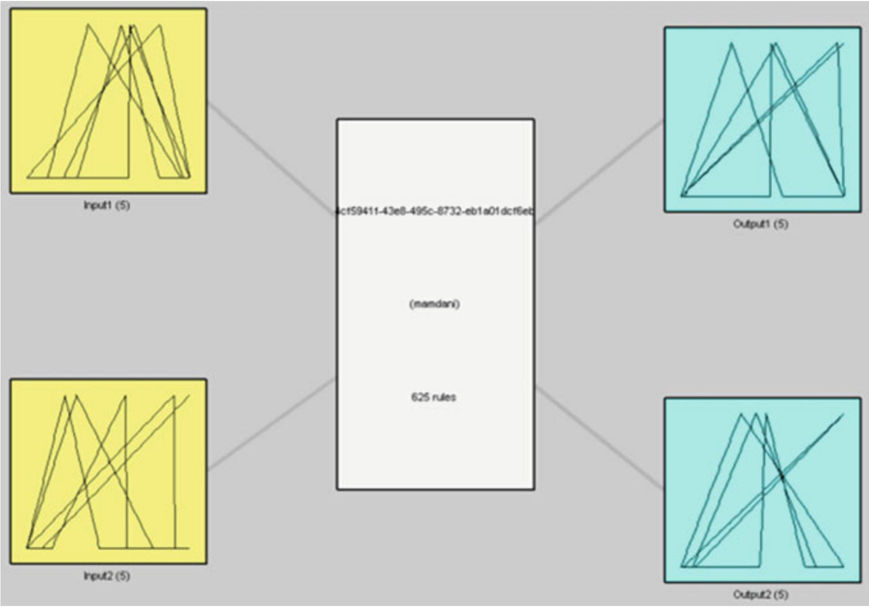


Fig. 5 Fuzzy controller inputs ep, eθ

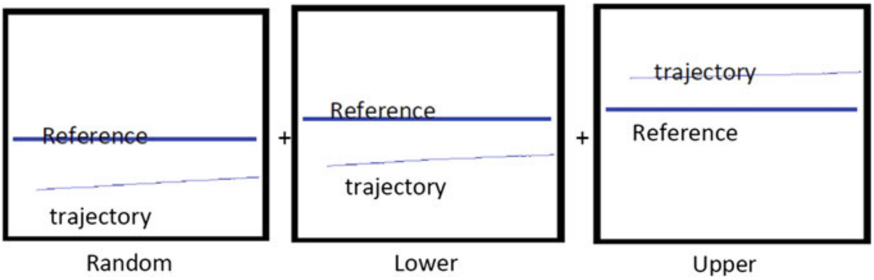


Fig. 6 Calculation of controller performance

4.5 Objective Function

The PSO will be generating particles that will need to be evaluated and assigned a crisp value that will represent the controller performance on each of the criteria that we want to improve. For this, we need to provide the PSO with a good evaluation scheme that will penalize unwanted behaviors and reward with higher fitness values those individuals that provide the performance we are looking for in our controller; if we fail to provide a proper evaluation method we can guide the population to suboptimal solutions or no solution at all [2, 4–7].

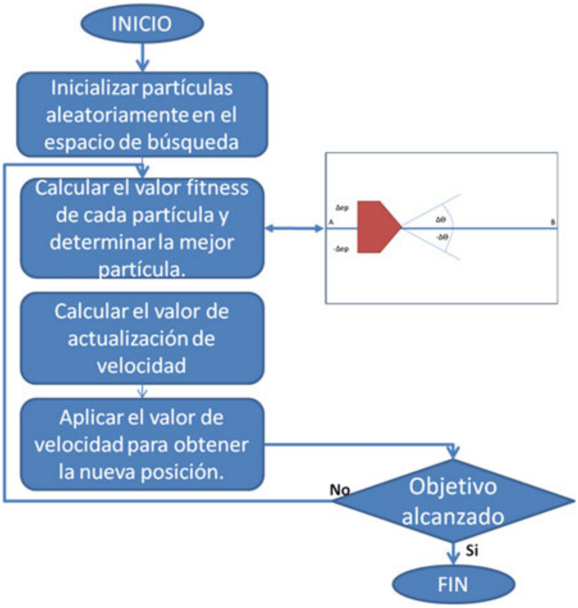


Fig. 7 Flowchart of the PSO

4.5.1 Reactive Controller Objective Function

The criteria used to measure the Reactive controller performance are the following

- Covered Distance
- Time used to cover the distance
- Battery life.

A Fitness FIS will provide the desired fitness value, adding very basic rules that reward the controller that provided the longer trajectories and smaller times and higher battery life. This seems like a good strategy that will guide the control population into evolving and provide the optimal control, but this strategy on its own is not capable of doing just that: it needs to have a supervisor on the robots trajectory to make sure it is a forward moving trajectory and that they does not contain any looping parts. For this, a Neural Network (NN), is used to detect cycle trajectories that do not have the desired forward moving behavior by giving low activation value and higher activation values for the ones that are cycle free. The NN has two inputs and one output, and 2 hidden layers, see Fig. 8.

The evaluation method for the reactive controller has integrated both parts of the FIS and the NN where the fitness value for each individual is calculated with Eq. 3. Based on the response of the NN the peak activation value is set to 0.35, this meaning that any activation lower than 0.35 will penalize the fitness given by the FIS [2, 4–7].

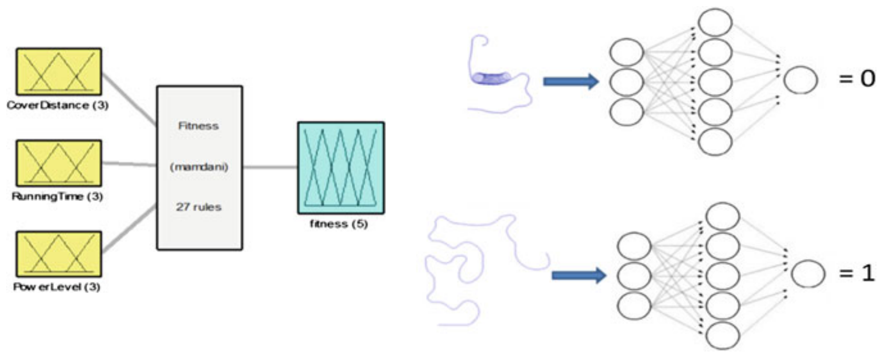


Fig. 8 Fitness function for the reactive controller

Equation 3 expresses how to calculate the fitness value of each individual

$$f(i) = \begin{cases} fv \, * \, nnv & nnv < 0.35 \\ fv & nnv \geq 0.35 \end{cases} \tag{3}$$

where:

fi Fitness value of the i-th individual,
fv Crisp value out of the fitness FIS,
nnv Looping trajectory activation value.

4.5.2 Tracking Controller Objective Function

The Tracking controller performance is measured with the RMSE between the reference and the robots trajectory; we apply the test three times and take the average on each of the three tests. The robot and the reference vertical position is random, but it's ensured that on one test the robots' vertical position is above the reference and on another test is below it. We do this to ensure the controller works properly for any case the robot may need it when its above or below (Fig. 9) [2, 4–7].

4.5.3 WFIS Controller Objective Function

The WFIS controller performance is measured by the RMSE between the reference and the robot's trajectory. We apply the test three times and take the average. On each of the three tests the robot's and the reference vertical position are random, but we make sure that on one test the robot's vertical position is above the reference and on another test is below it. We do this to ensure the controller works properly for any case the robot may need may need to deal with (Fig. 10) [2, 4–7].

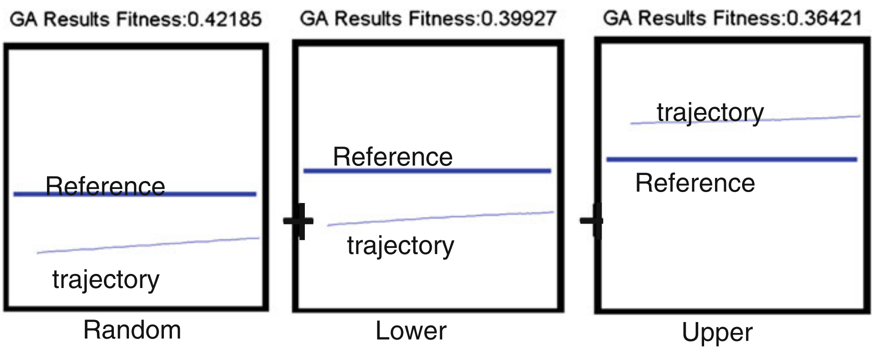


Fig. 9 Fitness function for the tracking controller

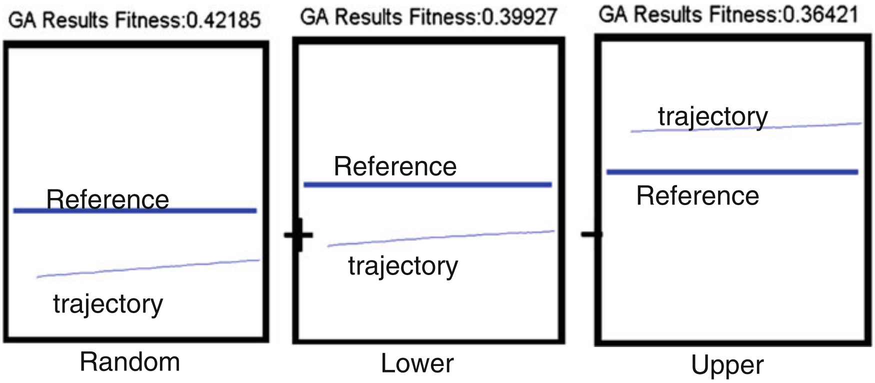


Fig. 10 Fitness functions for the WFIS controller

5 Simulation Results

This section presents the results of experiments carried out for the robot control system, with individual optimization of each behavior (tracking and reactive), necessary to obtain shown WFIS controller results.

The results are divided in 3 main principals:

- Tracking Controller
- Reactive Controller
- WFIS Controller

The tools that were used to conduct the experiments are Matlab and simulation tool Simrobot.

Table 1 Summary of tracking results

Particle	Iteration	C1	C2	Inertia weight	Constraint factor
20	500	1.4962	1.4962	LD	D
	Fitness		**Fitness**		**Fitness**
1	0.3206	11	0.3266	21	0.3293
2	0.3256	12	0.3148	22	0.3229
3	0.3014	13	0.3266	23	0.3117
4	0.3219	14	0.3255	24	0.3102
5	0.3266	15	0.2944	25	0.3117
6	0.3229	16	**0.3313**	26	0.3233
7	0.3164	17	0.2557	27	0.3136
8	0.3271	18	0.2806	28	0.3233
9	0.2919	19	0.2926	29	0.3313
10	0.2796	20	0.3154		
Average					0.312924138
Best					0.2557
Poor					0.3313
Std dev					0.000334708

5.1 Tracking Controller

Table 1 shows the configuration of the PSO and the results, in which we have the fitness value obtained in each experiment. It also shows the mean, variance, as well as the best and worst value obtained. In optimizing the weight of the inertia controller a constraint factor was used, not at the same time, but when the weight iteration inertia torque used as odd iteration a constraint factor is used.

Figure 11 shows the best path simulation during cycle PSO obtained for the tracking controller. The reference is defined by the red line and the smallest dot line is the trajectory of the robot, we can also see the graph of the FIS where input1 and input2 indicate the error on the position and orientation respectively, output1 and output2 speed to be applied to each actuator.

5.2 Reactive Controller

In this section, we show test of reactive controller, which includes the creation of a PSO algorithm to optimize the controller. The fitness of each controller is determined by their performance in the simulation tool. The robot should react in a closed environment (maze), avoiding obstacles present (walls) and must perform movements and avoid repeated.

Table 2 shows the configuration of PSO and displays the results, where we have the fitness value obtained in each experiment. It also shows the mean, variance, the best and worst value obtained. In this controller optimization constraint factor was used.

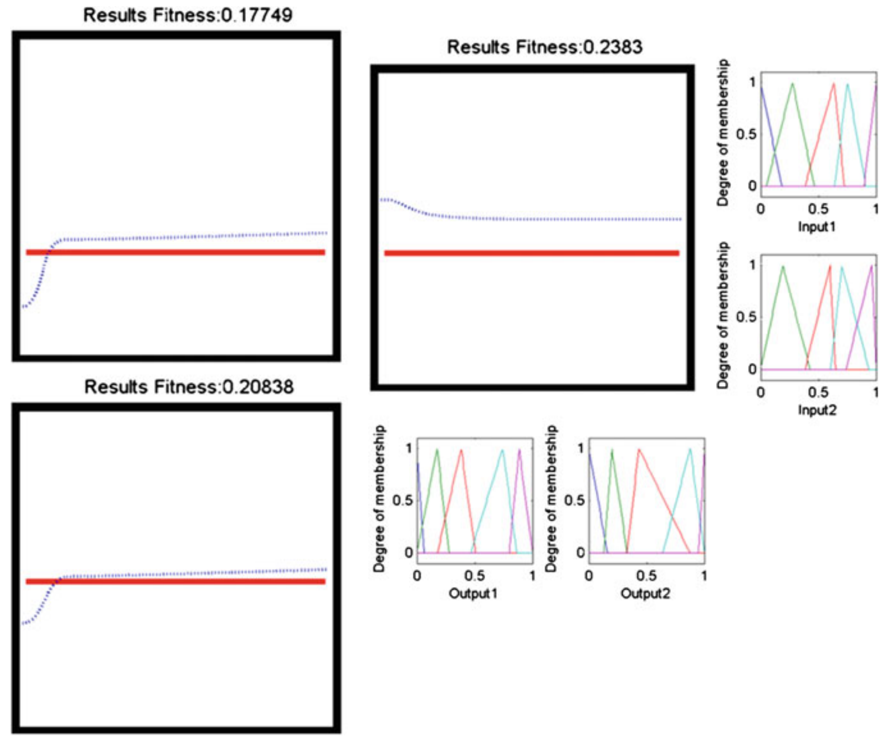


Fig. 11 Tracking controller results

Table 2 Summary of reactive controls results

Particle	Iteration	C1	C2	Inertia weight	Constraint factor
20	500	1.4962	1.4962	–	D
		Rank	Fitness		
		1	0.36249		
		2	0.36079		
		3	0.36219		
		4	0.35533		
		5	**0.35518**		
		6	0.39059		
		7	0.39018		
		8	0.37154		
		9	**0.41107**		
		10	0.39917		
Average					0.375853
Best					0.41107
Poor					0.35518
Std dev					0.0201816

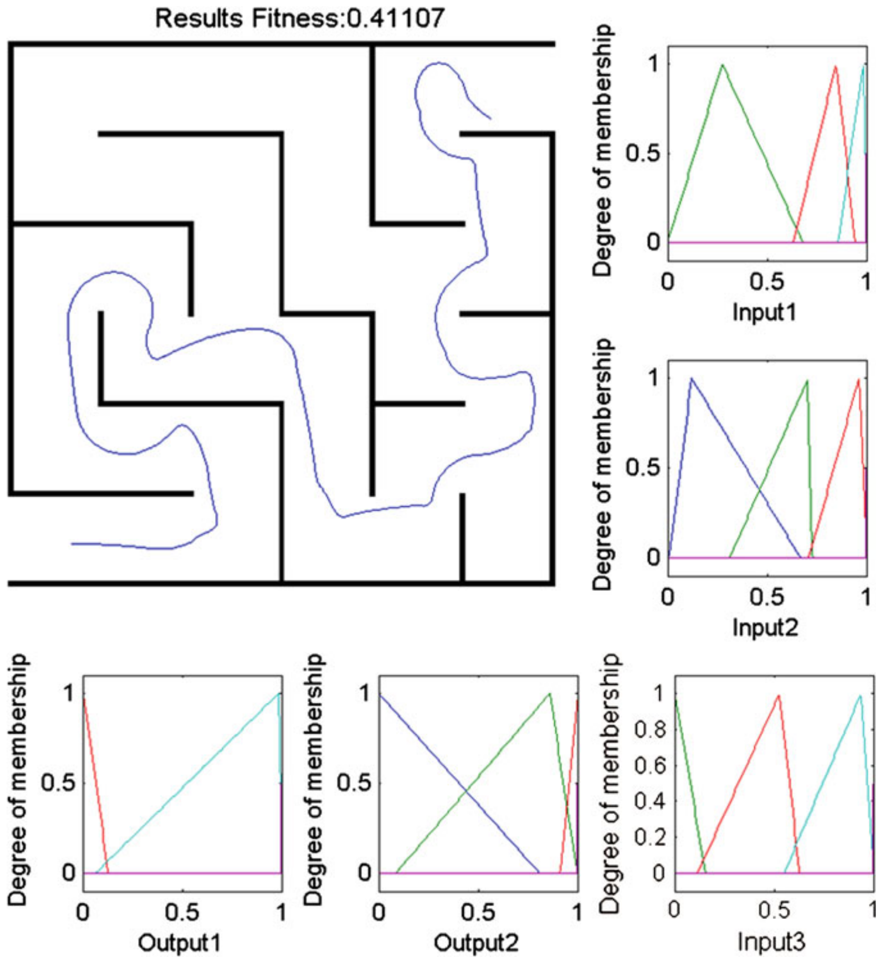


Fig. 12 Reactive controller results

Figure 12 shows the best robot path during the simulation of the PSO cycle obtained for the reactive controller. We can also see the graph of the FIS, where input1, input2 and input3 refer to the reading of the sensors robot orientation (straight, left and front), respectively, output1 and output2 speed to be applied to each servomotor.

5.3 WFIS Controller

In this section, we discuss the WFIS test driver. The best of each type of reagent and tracking controller are used as blocks in the behavioral integration system, which include the creation of a PSO algorithm to optimize the controller.

Table 3 Summary of WFIS results

Particle	Iteration	C1	C2	Inertia weight	Constraint factor
20	500	1.4962	1.4962	–	D
		Rank	Fitness		
		1	0.3497		
		2	0.3497		
		3	0.3497		
		4	0.3497		
		5	0.3497		
		6	0.3497		
		7	0.3497		
Average					0.3497
Best					0.3497
Poor					0.3497
Std dev					5.99589E−17

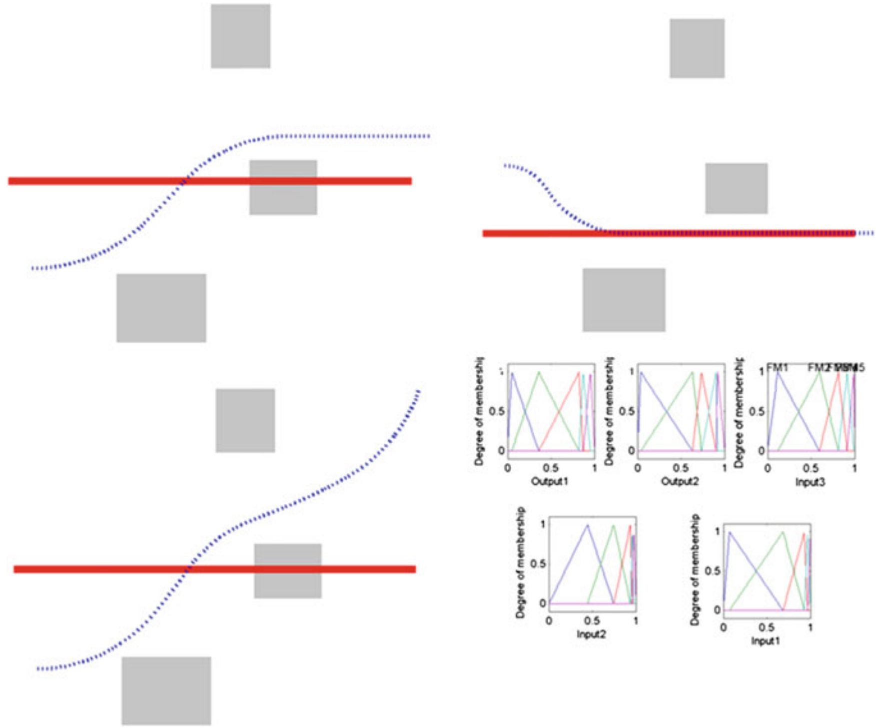


Fig. 13 WFIS controller results

The fitness of each controller is determined by its performance in the simulation tool, in which the robot must start from a random starting point and move forward on the reference line to avoid any collision. This test is performed 3 times forcing each robot controller to start at least once above and below the reference line.

Table 3 shows the configuration of PSO and results, in which we have the fitness value obtained in each experiment. It also shows the mean, variance, the best and worst value obtained. In this controller optimization a constraint factor was used.

Figure 13 shows the best robot path during the simulation of the PSO cycle obtained for the WFIS controller. We can also see the graph of the FIS, where input1, input2 and input3 refer to the reading of the sensors robot orientation (straight, left and front), respectively. Also output1 and output2 fuzzy weights will be applied to the controllers.

6 Conclusions

In this paper, the improved PSO algorithm used to tune the parameters of the fuzzy controller for the Reactive and Tracking Controllers, and we are currently working on the WFIS Optimization.

The fuzzy controllers both provide good results as they are able to guide the robot through the maze without hitting any wall and keep the robot on track. In comparison with the GA, only one of the two controllers in the tests that performed with the PSO proved to be better than the GA. Also, in the PSO less iterations have been performed, therefore consuming less computational resources.

Acknowledgments We would like to express our gratitude to CONACYT, and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Measurement and Instrumentation, Faculty of Electrical Engineering and Computer Science, Brno University of Technology, Czech Republic Department of Control: Autonomous Mobile Robotics Toolboxfor Matlab 5 (2001). Online. http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot%20en.html
2. Melendez, A., Castillo, O.: Hierarchical genetic optimization of the fuzzy integrator for navigation of a mobile robot. In: Melin, P., Castillo, O. (eds.) Soft Computing Applications in Optimization, Control, and Recognition. Volume 294 of Studies in Fuzziness and Soft Computing, pp. 77–96. Springer, Berlin (2013)
3. Astudillo, L., Melin, P., Castillo, O.: Nature optimization applied to design a type-2 fuzzy controller for an autonomous mobile robot. In: 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 212, 217, 5–9 Nov 2012. doi: 10.1109/NaBIC.2012.6402264

4. Melendez, A., Castillo, O.: Optimization of type-2 fuzzy reactive controllers for an autonomous mobile robot. In: 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 207–211 (2012)
5. Melendez, A., Castillo, O.: Evolutionary optimization of the fuzzy integrator in a navigation system for a mobile robot. In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) Recent Advances on Hybrid Intelligent Systems, volume 451 of Studies in Computational Intelligence, pp. 21–31. Springer, Berlin (2013)
6. Melendez, A., Castillo, O., Soria, J.: Reactive control of a mobile robot in a distributed environment using fuzzy logic. In: Annual Meeting of the North American on Fuzzy Information Processing Society, 2008. NAFIPS 2008, pp. 1–5 (2008)
7. Melendez, A., Castillo, O., Garza, A., Soria, J.: Reactive and tracking control of a mobile robot in a distributed environment using fuzzy logic. In: FUZZ-IEEE, pp. 1–5 (2010)
8. Astudillo, L., Castillo, O., Aguilar, L.: Intelligent control of an autonomous mobile robot using type-2 fuzzy logic. In: IC-AI 2006, pp. 565–570
9. Adika, C.O., Wang, L.: Short term energy consumption prediction using bio-inspired fuzzy systems. In: North American Power Symposium (NAPS), 2012, pp. 1, 6, 9–11 Sept 2012
10. Amin, S., Adriansyah, A.: Particle swarm fuzzy controller for behavior-based mobile robot. In: 9th International Conference on Control, Automation, Robotics and Vision, 2006. ICARCV '06, pp. 1, 6, 5–8 Dec 2006. doi: 10.1109/ICARCV.2006.345293
11. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley, New York, 2006
12. Astudillo, L., Castillo, O., Aguilar, L., Martínez, R.: Hybrid Control for an Autonomous Wheeled Mobile Robot under Perturbed Torques. IFSA (1), 594–603 (2007)
13. Cardenas, S., Garibaldi, J., Aguilar, L., Castillo, O.: Intelligent Planning and Control of Robots using Genetic Algorithms and Fuzzy Logic. In: IC-AI 2005, pp. 412–418
14. Castillo, O., Martinez, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf. Sci. **192**, 19–38 (2012)
15. Cervantes, L., Castillo, O.: Design of a fuzzy system for the longitudinal control of an F-14 airplane. In: Soft Computing for Intelligent Control and Mobile Robotics, pp. 213–224 (2011)
16. Tsai, C.C., Tsai, K.I., Su, C.T.: Cascaded fuzzy-PID control using PSO-EP algorithm for air source heat pumps. In: 2012 International Conference on Fuzzy Theory and it's Applications (iFUZZY), pp. 163, 168, 16–18 Nov 2012
17. De Santis, E., Rizzi, A., Sadeghiany, A., Mascioli, F.M.F.: Genetic optimization of a fuzzy control system for energy flow management in micro-grids. In: IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint, pp. 418, 423, 24–28 June 2013. doi: 10.1109/IFSA-NAFIPS.2013.6608437
18. Wang, D., Wang, G., Hu, R.: Parameters optimization of fuzzy controller based on PSO. In: 3rd International Conference on Intelligent System and Knowledge Engineering, 2008. ISKE 2008, vol. 1, pp. 599, 603, 17–19 Nov 2008
19. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation, 2000, vol. 1, pp. 84, 88 (2000). doi: 10.1109/CEC.2000.870279
20. Esmin, A.A.A., Aoki, A.R., Lambert-Torres, G.: Particle swarm optimization for fuzzy membership functions optimization. In: 2002 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, 6 pp., 6–9 Oct 2002
21. Fierro, R., Castillo, O.: Design of fuzzy control systems with different PSO variants. In: Recent Advances on Hybrid Intelligent Systems, pp. 81–88 (2013)
22. Fang, G., Kwok, N.M., Ha, Q.: Automatic fuzzy membership function tuning using the particle swarm optimization. In: Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008. PACIIA '08, vol. 2, pp. 324, 328, 19–20 Dec 2008
23. Hassen, T., Ahmed, M., Mohamed, E.: Pso-belbic scheme for two-coupled distillation column process. J. Adv. Res. **2**(1), 73–83 (2011)

24. Chen, J., Xu, L.: Road-junction traffic signal timing optimization by an adaptive particle swarm algorithm. In: 9th International Conference on Control, Automation, Robotics and Vision, 2006. ICARCV '06, pp. 1, 7, 5–8 Dec 2006
25. Kamejima, T., Phimmasone, V., Kondo, Y., Miyatake, M.: The optimization of control parameters of PSO based MPPT for photovoltaics. In: 2011 IEEE Ninth International Conference on Power Electronics and Drive Systems (PEDS), pp. 881, 883, 5–8 Dec 2011
26. Astudillo, L., Melin, P., Castillo, O.: Optimization of a fuzzy tracking controller for an autonomous mobile robot under perturbed torques by means of a chemical optimization paradigm. In: Recent Advances on Hybrid Intelligent Systems, pp. 3–20 (2013)
27. Wang, L., Kang, Q., Qiao, F., Wu, Q.: Fuzzy logic based multi-optimum programming in particle swarm optimization. In: Proceedings. 2005 IEEE Networking, Sensing and Control, 2005, pp. 473, 477, 19–22 March 2005
28. Mahendiran, T.V., Thanushkodi, K., Thangam, P., Gunapriya, B.: Speed control of three phase switched reluctance motor using particle swarm optimization. In: 2012 International Conference on Advances in Engineering, Science and Management (ICAESM), pp. 315, 319, 30–31 March 2012
29. Martínez, R., Castillo, O., Soria, J.: Particle swarm optimization applied to the design of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. In: Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition, pp. 247–262 (2009)
30. Martínez, R., Castillo, O., Aguilar, L.: Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Inf. Sci. **179**(13), 2158–2174 (2009)
31. Martinez, R., Castillo, O., Aguilar, L., Baruch, I.: Bio-inspired optimization of fuzzy logic controllers for autonomous mobile robots. In: 2012 Annual Meeting of the North American on Fuzzy Information Processing Society (NAFIPS), pp. 1–6 (2012)
32. Martínez, R., Castillo, O., Aguilar, L., Melin, P.: Fuzzy logic controllers optimization using genetic algorithms and particle swarm optimization. MICAI **2**, 475–486 (2010)
33. Melin, P., Astudillo, L., Castillo, O., Valdez, F., Garcia, M.: Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm. Expert Syst. Appl. **40**(8), 3185–3195 (2013)
34. García, M.A.P., Montiel, O., Castillo, O., Sepúlveda, R.: Optimal path planning for autonomous mobile robot navigation using ant colony optimization and a fuzzy cost function evaluation. In: Analysis and Design of Intelligent Systems using Soft Computing Techniques, pp. 790–798 (2007)
35. Milla, F., Sáez, D., Cortés, C.E., Cipriano, A.: Bus-stop control strategies based on fuzzy rules for the operation of a public transport system. In: IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 3, pp. 1394, 1403, Sept 2012
36. Yang, M., Wang, X.: Fuzzy PID controller using adaptive weighted PSO for permanent magnet synchronous motor drives. In: Second International Conference on Intelligent Computation Technology and Automation, 2009. ICICTA '09, vol. 2, pp. 736, 739, 10–11 Oct 2009
37. Montiel, O., Camacho, J., Sepúlveda, R., Castillo, O.: Fuzzy system to control the movement of a wheeled mobile robot. In: Soft Computing for Intelligent Control and Mobile Robotics, pp. 445–463 (2011)
38. Porta, M., Montiel, O., Castillo, O., Sepúlveda, R., Melin, P.: Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. Appl. Soft Comput. **9**(3), 1102–1110 (2009)
39. Martínez, R., Castillo, O., Aguilar, L.: Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. In: Soft Computing for Hybrid Intelligent Systems, pp. 3–18 (2008)
40. Rajeswari, K., Lakshmi, P.: PSO optimized fuzzy logic controller for active suspension system. In: 2010 International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), pp. 278, 283, 16–17 Oct 2010

41. Vaneshani, S., Jazayeri-Rad, H.: Optimized fuzzy control by particle swarm optimization technique for control of CSTR **5**(11), 464, 470 (2011)
42. Aguas-Marmolejo, S.J., Castillo, O.: Optimization of membership functions for type-1 and type 2 fuzzy controllers of an autonomous mobile robot using PSO. In: Recent Advances on Hybrid Intelligent Systems, pp. 97–104 (2013)
43. Selvakumaran, S., Parthasarathy, S., Karthigaivel, R., Rajasekaran, V.: Optimal decentralized load frequency control in a parallel ac-dc interconnected power system through fHVDCg link using fPSOg algorithm. Energy Procedia **14**(0), 1849, 1854 (2012). In: 2011 2nd International Conference on Advances in Energy Engineering (ICAEE)
44. Singh, R., Hanumandlu, M., Khatoon, S., Ibraheem, I.: An adaptive particle swarm optimization based fuzzy logic controller for line of sight stabilization tracking and pointing application. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 1259, 1264, 11–14 Dec 2011
45. Talbi, N.; Belarbi, K.: Fuzzy rule base optimization of fuzzy controller using hybrid tabu search and particle swarm optimization learning algorithm. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 1139, 1143, 11–14 Dec 2011
46. Valdez, F., Melin, P., Castillo, O.: Fuzzy control of parameters to dynamically adapt the PSO and GA Algorithms. In: FUZZ-IEEE 2010, pp. 1–8
47. Vázquez, J., Valdez, F., Melin, P.: Comparative study of particle swarm optimization variants in complex mathematics functions. In: Recent Advances on Hybrid Intelligent Systems, pp. 223–235 (2013)
48. Venayagamoorthy, G., Doctor, S.: Navigation of mobile sensors using PSO and embedded PSO in a fuzzy logic controller. In: Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE, vol. 2, pp. 1200, 1206, 3–7 Oct 2004
49. Wong, S., Hamouda, A.: Optimization of fuzzy rules design using genetic algorithm. Adv. Eng. Softw. **31**(4), 251–262 (2000). ISSN 0965-9978, http://dx.doi.org/10.1016/S0965-9978(99)00054-X
50. Yen, J, Langari R.: Fuzzy Logic: Intelligence, Control, and Information. Prentice Hall, Englewood Cliffs (1999)
51. Liu, Y., Zhu, X., Zhang, J., Wang, S.: Application of particle swarm optimization algorithm for weighted fuzzy rule-based system. In: 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004, vol. 3, pp. 2188, 2191, 2–6 Nov 2004
52. Zafer, B., Oğuzhan, K.: A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control. Expert Syst. Appl. **38**(1), 1017–1031 (2011). ISSN 0957-4174, http://dx.doi.org/10.1016/j.eswa.2010.07.131