

A PSO Optimized Fuzzy Control Scheme for Mobile Robot Path Tracking

Turki Y. Abdalla

Department of Computer Engineering
University of Basra
Basra, Iraq

Abdulkareem.A.A

Department of Electrical Engineering
University of Basra
Basra, Iraq

ABSTRACT

In this paper, a fuzzy controllers type Takagi_Sugeno is optimized by method of Particle Swarm Optimization (PSO). This algorithm automatically adjust the membership function of fuzzy controllers to control a trajectory of nonholonomic mobile robot that involves path trajectory using two optimized fuzzy controllers one for speed control and the other for azimuth control. The mobile robot is modelled in Simulink and PSO algorithm is implemented using MATLAB. Simulation results show good performance for the proposed control scheme. The results will be compared with PSO-PID controllers that control the same model of mobile robot.

Keywords

Mobil Robot, Fuzzy Control, Particle Swarm Optimization, Kinematic and Dynamic Model, Particle Swarm Fuzzy controller (PSFC), PSO-PID.

1. INTRODUCTION

The constrained motion robot control problem has received attention in the literature due to its complexity. This means that it is needed to evolve robot controllers that solve complicated problems and tackle complicated in the variable environments. Autonomous robots may act instead of human beings. The robots are able to accomplish many tasks in dangerous places where humans cannot enter, such as sites where harmful gases or high temperature are present a hard environment for humans. Cleaning robots and cargo delivery can work automatically and save costs by performing various routine tasks [1,2]. This means that it is needed to evolve robot controllers that solve complicated problems and tackle complicated in the variable environments. There are several types of controllers used to control mobile robot in this paper optimized fuzzy logic controllers optimized by PSO algorithm is used to control both speed and azimuth.

Recently, a new swarm intelligence technique, the particle swarm optimization (PSO), has been proposed [3,4] as an alternative to an evolutionary computation algorithms such as genetic algorithm (GA). The PSO development was based on study of the social behavior of animals such as bird flocking, fish schooling, and using the concept of swarm theory. Compared with GA, PSO has some attractive characteristics. It has memory, so knowledge of good solutions is retained by all particles; whereas in GA, previous knowledge of the problem is destroyed once the population changes. It has constructive cooperation between particles, particles in the swarm share information between each other. However, in GA, we have cross-over operator, the standard PSO is less random in nature than GA is.

Due to the simple concept, easy implementation and quick convergence, nowadays PSO has gained much attention and wide applications in different fields [5].

K. C. Ng and M. M. Trivedi showed a neural integrated Fuzzy controller which integrates the fuzzy logic representation of human

knowledge with the learning capability of neural networks; it is developed for nonlinear dynamic control problems [20], K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright and H. M. Tai presented results of their work in development of a genetic algorithm based path-planning algorithm for local obstacle avoidance (local feasible path) of a mobile robot in a given search space [21], S. E. Mahmoudi, A. A. Bitaghsir, B. Forouzandeh and AL. R. Marandi presented novel approach to mobile robot navigation combining path planning and auto-tuning motion control. Also, they discuss around the pursuit idea in robot motion control and show the modification with genetic algorithm to achieve a method for navigation of a two wheel mobile robot [22].

This paper presents fuzzy controllers optimized by PSO algorithm to control the trajectory of mobile robot and tries this control system for different trajectories and compared the results with the same mobile robot controlled by PSO-PID controllers for same trajectories. This paper has been organized as follows: in section 2 both kinematics and dynamic models of mobile robot are described. In section 3, the particle swarm optimization method is reviewed. Section 4 describes the structure of fuzzy logic controller. In section 5 describe how to implement PSFC. The simulation and the results are presented in section 6.

2. A Nonholonomic Mobile Robot Model

A mobile robot is located in a two dimensional Cartesian workspace, in which a global coordinate $\{X, 0, Y\}$ is defined. A local coordinate $\{X_c, C, Y_c\}$ is attached to the robot with the origin at point c , the middle points of two wheels which is the guide point of this mobile robot. A typical mobile model is shown in Figure 1, where b is the half distance between two wheels. There are several ways to set up a steering system for differential drive mobile robot. A robot must have a minimum of three wheels in order to work.

All the combinations require two motorized wheels and at least one swiveling wheel for balance [6]. Consider the mobile robot depicted in Figure 1 as front drive used in this paper. The platform moves by driving the two independent wheels as shown in the Figure 1. We assume that the speed at which this system moves is low and therefore the two driven wheels do not slip sideways.

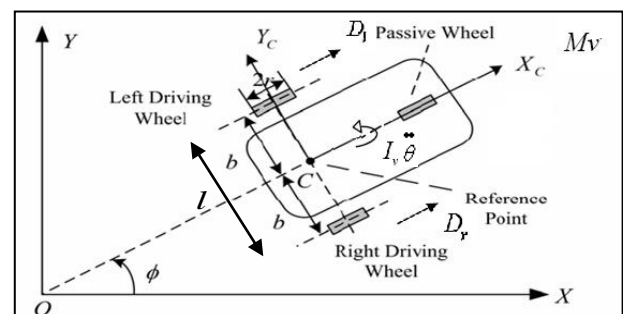


Figure 1: Model of mobile

Let's consider the kinematic model (study of the mathematics of motion without considering the forces that affect the motion), it deals with the geometric relationships that govern the system and deals with the relationship between control parameters and the behavior of a system in state space for an autonomous vehicle[7]. The position of the mobile robot in the plane is shown in Figure 1, the inertial-based frame (Oxy) is fixed in the plane of motion and the moving frame is attached to the mobile robot. The mobile robots are rigid cart equipped, with non-deformable conventional wheels, and it is moving on a non-deformable horizontal plane. During the motion: the contact between the wheel and the horizontal plane is reduced to a single point, the wheels are fixed, the plane of each wheel remains vertical, the wheel rotates about its horizontal axle and the orientation of the horizontal axle with respect to the cart can be fixed [8]. This means that the velocity of the contact point between each wheel and the horizontal plane is equal to zero. The rotation angle of the wheel about its horizontal axle is denoted by $\varphi(t)$ and the radius of the wheel by r . Hence, the position of the wheel is characterized by two constants: b and r and its motion by: $\varphi_r(t)$ – the rotation angle of the right wheel and $\varphi_l(t)$ – the rotation angle of the left wheel. The configuration of the mobile robot can be described by five generalized coordinates (q) such as [8, 9]:

$$q = (x_c, y_c, \theta, \varphi_r, \varphi_l) \quad (1)$$

where: x_c and y_c are the two coordinates of the origin P of the moving frame (the geometric center of the mobile robot), θ is the orientation angle of the mobile robot (of the moving frame). The vehicle velocity v can be found in Equation (2) [10]:

$$v = \frac{R(w_r + w_l)}{2} \quad (2)$$

where:

$$w_r = \frac{d\varphi_r}{dt} \quad (\text{Angular velocity of the right wheel})$$

$$w_l = \frac{d\varphi_l}{dt} \quad (\text{Angular velocity of the left wheel})$$

The position and the orientation of the mobile robot are determined by a set of differential previous equations in the following forms [9,10]:

$$\dot{x} = (R \cos\theta (w_r + w_l))/2 \quad (3)$$

$$\dot{y} = (R \sin\theta (w_r + w_l))/2 \quad (4)$$

$$\dot{\theta} = R(w_r - w_l)/2b \quad (5)$$

$$\text{Here, } \dot{x} = v \cos\theta, \dot{y} = v \sin\theta$$

Finally, the kinematics model of the vehicle velocity v and the orientation θ can be represented by the matrix as follows [8]:

$$\begin{bmatrix} \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ R/2b & -R/2b \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix} \quad (6)$$

A large number of researchers have used kinematic models to develop motion control strategy for mobile robots, their argument and

assumption that these models are valid if the robot has low speed, low acceleration and light load [9]. Dynamic modeling takes into account the forces acting on the vehicle. This model can be constructed using the no-slip condition [10] or allowing wheel slip [11]. In either case, the acceleration of the car is considered.

In dynamic modeling the vehicle's dynamic properties, such as mass, center of gravity, etc. enter into the equations. to drive this model, the non holonomic constraints of the system are utilized. Dynamic equation of wheeled mobile robot is described as[12]:

$$M(q)W + C(q, \dot{q})W + D\dot{W} = \tau \quad (7)$$

$$W = [w_r \ w_l]^T, \tau = [\tau_r \ \tau_l]^T \quad (8)$$

$$M(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{11} \end{bmatrix} \quad (9)$$

$$m = m_c + 2m_w \quad (10)$$

$$I = m_c a^2 + 2m_w b^2 + I_c + 2I_w \quad (11)$$

$$m_{11} = 0.25b^{-2}r^2(mb^2 + I) + I_w \quad (2)$$

$$m_{12} = 0.25b^{-2}r^2(mb^2 - I) \quad (13)$$

$$(14)$$

$$D \quad C(q, \dot{q}) = \begin{bmatrix} 0 & c\dot{\theta} \\ -c\dot{\theta} & 0 \end{bmatrix} \quad (15)$$

$$W = \dot{\theta} = d\theta/dt \quad (16)$$

Where:

$w_r \ w_l$: are angular velocities of right and left wheel respectively.

m_c : is the mass of body.

m_w : is the mass of the wheel with a motor.

I_c : the moment of inertia of the body about vertical axis through the center of mass.

I_w : is the moment of inertia of the wheel with a motor about the wheel diameter.

R : is the radius of the wheel.

a : is the distance between the robot's center of mass and the center of the wheel axle.

b : is the half distance between the two wheels.

d_{11}, d_{22} : are damping coefficients.

$q = (x, y, \theta)$: is the vector of generalized coordinates.

$\tau = [\tau_r \ \tau_l]^T$: is the vector of torque applied to the wheels of the robot.

$$[\dot{V} \ \dot{W}] = -M^{-1}(q)D[V \ W]^T + M^{-1}(q)(-C(q, \dot{q})[V \ W]^T + [\tau_r, \tau_l]^T)$$

$M(q)$; is 2*2 positive-definite inertia matrix.

These are equations that are used to build the model of robot.
From the above equation we can get,

((17))

3. OVERVIEW OF PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a technique used to explore the search space of a given problem to find the settings or parameters required to maximize a particular objective. This technique, first described by James Kennedy and Russell C. Eberhart in 1995 [14]. The basis behind the development of PSO is a hypothesis that social sharing of information among specious offers an evolutionary advantage. Each particle in PSO moves in the D-dimensional problem space with a velocity which is dynamically adjusted according to the moving experiences of its own and its neighbors. The location of the i th particle is denoted as $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$ where $x_{id} \in [ld, ud]$, $d \in [1, D]$, ld , ud , are the lower and upper bounds for d th dimension respectively. The best previous position of the i th particle is saved and denote by $P_i = (p_{i1}, \dots, p_{id}, \dots, p_{iD})$, which is also called *pbest*. The best particle among all the particles in the population is denoted by P_g which is also called *gbest*. The velocity for the i th particle is denoted by $V_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$, is limited by $V_{max} = (v_{max1}, \dots, v_{maxid}, \dots, v_{maxiD})$, which specified by the user.

The particle swarm optimization is based on changing the velocity and position of each particle toward its *pbest* and *gbest* locations according to the equations (1) and (2), respectively, at each time step:

$$v_{id}(t+1) = W_p v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \quad (18)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (19)$$

Where W_p is the inertia weight [15], c_1 and c_2 are acceleration constants [5], r_1 and r_2 are a random variables in range [0, 1]. In equation (1), the first part represents the inertia of pervious velocity; the second part is the “cognition” part, which represents the private thinking by itself; the third part is the “social” part, which represents the cooperation among the particles.

4. FUZZY LOGIC CONTROLLER (FLC)

FLC has found promising applications for a wide variety of industrial systems, which can guarantee various stability and performance criteria, not only for SISO nonlinear systems but also for MIMO nonlinear systems [16]. FLC, as one of the most useful approaches for utilizing expert knowledge, has had extensive research in the past decade. In the control systems, fuzzy logic is considered as an alternative for conventional control theory in the control of linear, nonlinear and complex nonlinear plants [17]. Basics of a fuzzy model are shown in Figure (2). The fuzzy model includes the following components or stages [18]:

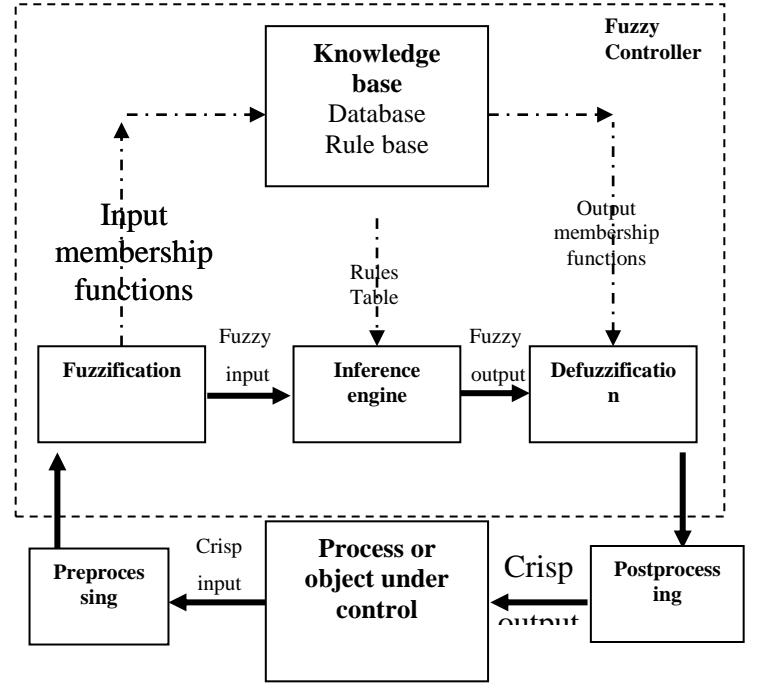


Figure 2: The fuzzy logic controller (basic structure).

Data preprocessing stage: It is the scaling process for the given physical values of the input to the fuzzy system. It is done by mapping it to proper normalized domains via scaling.

Fuzzification stage: It is the mapping of the crisp values of the preprocessed input of the model into a fuzzy sets represented by membership functions (MFs).

Rule-base stage: In the rule base the expert knowledge about how to control the system is represented in the form of (*if premise – then consequent*).

Inference engine stage: It is the computational method which calculates the degree to which each rule fires for a given fuzzified input pattern by considering the rule and label sets.

Defuzzification stage: It is the final component of fuzzy controllers. Defuzzification operates on the fuzzy output of the inference mechanism to provide the most certain controller output. In other words it converts fuzzy conclusions into crisp control actions. There are many approaches to achieve defuzzification process like center of gravity COG or sometimes referred to as center of area COA. If $B^*(\text{degree of membership})$ is defined in the universe X , the crisp output value is obtained for discrete values using:

$$z^* = \frac{\sum_{i=1}^q B^*(x_i) \cdot x_i}{\sum_{i=1}^q B^*(x_i)} \quad (20)$$

Where q is the number of quantization levels of universe X , x_i is the crisp value for quantization level i and $B^*(x_i)$ is its membership value in the inferred fuzzy set, B^* . [19].

Postprocessing stage: It gives the output of the fuzzy system based on the crisp signal obtained after defuzzification. This often means the scaling of the output.

5. PSFC IMPLEMENTATION

In this paper, an offline optimization process of the FLC was attempted by using PSO. PSO is given prior information in relation to the positions of rules and the shape of MFs for FLC.

A. Design Of The Membership function.

Constraints are introduced to the design of the FLC as follows:

1) All universes of discourses are normalized to lie between -6 and 6 with scaling factors external to the FLC used to give appropriate values to the variables.

2) It is assumed that the first and last membership functions have their apexes at -6 and 6 respectively. This can be justified by the fact that changing the external scaling would have similar effect to change these positions.

3) Triangular membership functions are used in middle of universes of discourses in input MF.

4) Trapezoidal membership functions are used in two edges of universes of discourses in input MF.

5) The number of fuzzy sets is constrained to be an odd integer greater than unity. Seven membership functions were used.

6) The base vertices of membership functions are coincident with the apex of the adjacent membership functions.

Using these constraints the design of the membership functions can be described using to determine the position of the apexes for all MFs. There are several methods for design membership functions, in this paper we'll use the method in which adjusts each parameters of MF in variable at alone and there parameters are co-operation with other two neighbors MFs as shown in Figure (3)

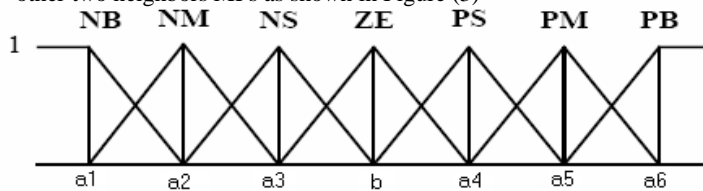


Figure 3: Six parameters for each membership functions.

Each controller will contain two input (error and change in error) and one output (speed or azimuth), each one has 7 membership function, so we have 18 parameter to be tuned for each controller.

B. Design Of Rule Base

As well as specifying the membership functions, the rule-base also needs to be designed. The FIS has two inputs (premises), error, $e(t)$, and error derivative, $\Delta e(t)$, and one output (consequent), control action, $u(t)$. Each of these three controller variables is used 7 membership functions (NB, NM, NS, Z, PS, PM, PB) with AND

connective, a rule-base consisting of 49 rules was therefore to be defined.

The following assumptions was initially mode:

1) The magnitude of the output control action is consistent with the magnitude of the input values. (i.e. in general, extreme input values (premise) result in extreme output values (consequent), mid-range input values in mid-range output values and small/zero input values in small/zero output values).

2) Similar combinations of input linguistic values lead to similar output values.

Using these assumptions the output space is partitioned into different regions corresponding to different output linguistic values as in Table (1). Each region has different membership functions (NB, NM, NS, Z, PS, PM or PB), where there are three regions (negative, zeroes and positive) as shown in Table (4), the four membership functions (NB, NM, NS or Z) will be tuned the negative region, while, the zeroes region will be tuned all membership functions (NB, NM, NS, Z, PS, PM or PB) and the positive region will tune four membership functions (Z, PS, PM or PB). The mobile robot using same Fuzzy Inference System (FIS) (same RB) for each two controller.

Table 1: Rules Table.

$e/\Delta e$	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PM	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

6. SIMULATION AND RESULTS

In order to build the mobile robot given by Equation (18) we use the following physical parameters given by [15]

Table 2: The physical parameters of mobile robot.

Parameter	Value	Unit
r	0.15	M
b	0.75	M
a	0.3	M
m_c	30	Kg
m_w	1	Kg
I_c	15.625	Kg. m^2
I_w	0.005	Kg. m^2
I_m	0.0025	Kg. m^2
d_{11}	10	—
d_{22}	10	—

C	0.135	—
---	-------	---

According to the trial, the following PSO parameters are used to verify the performance of each one of the PSFC controller parameters:

- Population size: 50 ;
- $W_p \text{ max}=0.9$, $W_p \text{ min}=0.4$;
- $c_1 = c_2 = 1.2$;
- Iteration :50 ;

The figures 4,5 and6 show the membership functions for controllers after optimization .

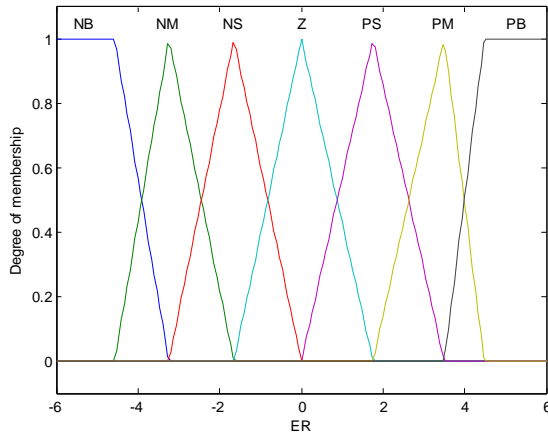


Figure 4: Optimized membership function for error.

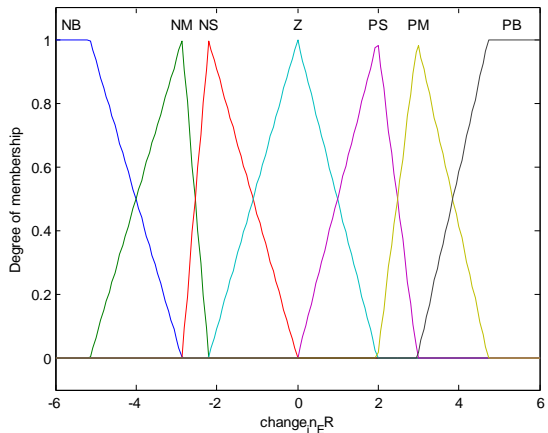


Figure 5: Optimized membership function for change in error.

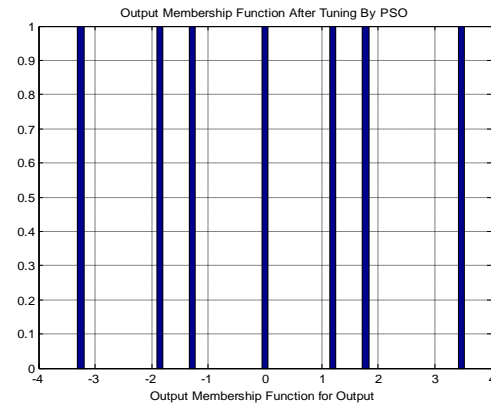


Figure 6: Optimized membership function for Output.

These membership functions shown above considered the optimal membership functions for fuzzy controllers and given lowest MSE. The system is tested for two different cases as follow:

- 1) The desired circular trajectory is given by a reference velocity v_d of 1 [meter/sec] and a reference azimuth θ_d given:

$$\theta_d = [(2 * \pi) / m \times f(t)] [\text{rad}]$$

Where $m(\text{slop})=0.1592$, $f(t) = t$, $0 \leq t \leq 8$

Figure 7 shows the velocity error, Figure 8 show the azimuth error, Figure 9 show the actual path for PSFC and PSO-PID and desired path for circular trajectory. The MSE for PSFC= $7.6 * 10^{-5}$.

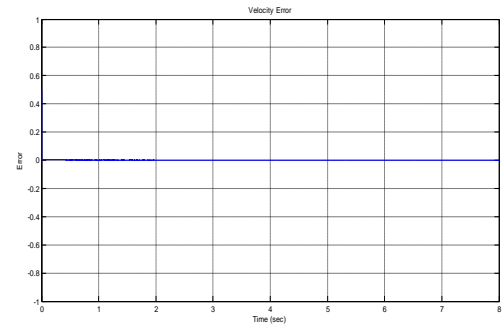


Figure 7: The error in velocity.

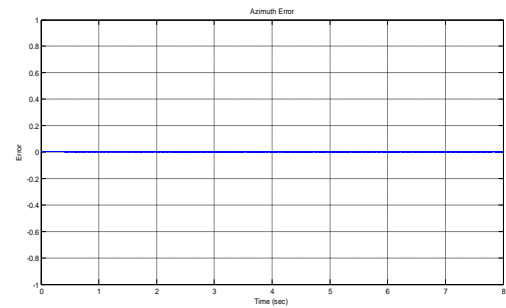


Figure 8: The error in azimuth

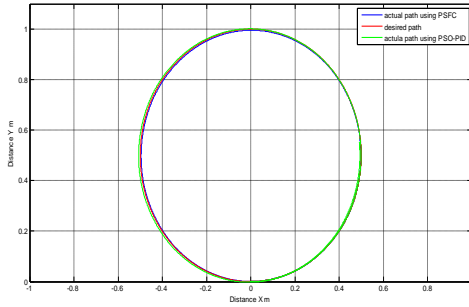


Figure 9: Circular trajectory compared PSFC and PSO-PID.

2) To follow a desired sine trajectory .

Figure 11 show the actual trajectory by using PSFC and PSO-PID controllers and desired trajectory, figure 12 show the error in azimuth and figure 13 show the error in velocity. $MSE=4.7*10^{-5}$. Table 2 the MSE for two cases and for PSFC controller and PSO-PID controller.

Table 3: MES for PSFC & PSO-PID controllers for two cases.

Trajectory	Circular	Sine
MSE for PSFC	$7.7*10^{-5}$	$4.7*10^{-5}$
MSE for PSO-PID	$1.2*10^{-4}$	$9.5*10^{-5}$

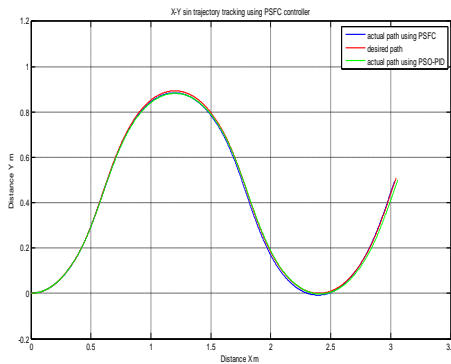


Figure 10: sine trajectory compared PSFC and PSO-PID.

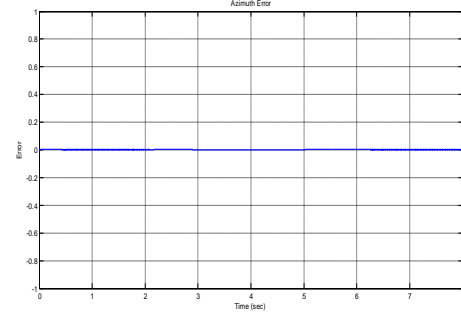


Figure 11: The Error in Azimuth.

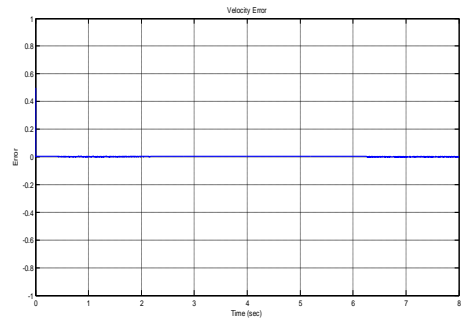


Figure 12: The Error in Velocity.

7. CONCLUSION

PSFC controller is built and implement in matlab / simulink software package and it is succeed to solve trajectory tracking problem. The Particle Swarm Optimization method is utilized to tune/optimize the parameters of fuzzy controller and it give a good results in short time compared with other optimization methods.

8. REFERENCES

- [1] Y. Shinoda, Y. Tan, J. Nakata and R. Beuran. 2007. Collaborative Motion Planning of Autonomous Robots. School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa Japan.
- [2] R. A. Felder. 1998. Mobile Robot Simulation of Clinical Laboratory Deliveries. M.Sc. Thesis, the University of Virginia, U.S.A.
- [3] Kennedy J, Eberhart RC. 1995 Particle swarm optimization. In *Proc. IEEE International Conference on Neural Networks*, WA Australia, p. 1942–8
- [4] Kennedy J, Eberhart RC, Shi Y, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001
- [5] Eberhart RC, Shi Y. 2001. Particle Swarm Optimization: developments, applications and resources. In *Proc. Congress on evolutionary computation*, Seoul, Korea, p. 81–6.

- [6] A. Haj-Ali and H. Ying. 2004. Structural Analysis of Fuzzy Controllers with Nonlinear Input Fuzzy Sets in Relation to Nonlinear PID Control with Variable Gains. Associate Editor Gary G. Yen under the direction of Editor Robert R. Bitmead, Wayne State University , Detroit , MI48202 ,USA .
- [7] D. R. Shircliff. 2002 Build A Remote-Controlled Robot. eBook, Copyright © by The McGraw-Hill Companies.
- [8] G. Mester. 2007 Obstacle Avoidance of Mobile Robots in Unknown Environments. SISY, International Symposium on Intelligent Systems and Informatics 24-25 Subotica, Serbia.
- [9] A. Albagul and Wahyudi. 2004. Dynamic Modelling and Adaptive Traction Control for Mobile Robots. 30th Annual Conference of the IEEE Industrial Electronics Society, November 2 - 6, Susan, Korea.
- [10] M. A. Abido. 2002 Optimal design of power-system stabilizers using particleswarm optimization. IEEE Trans. Energy Conversion, vol. 17, pp.406-413.
- [11] E.N.moret. 2003. Dynamic Modeling And Control Of a Car-like Robot. Master's thises, Virginia Polytechnic Institute and state University.
- [12] P.Kachoor and M.tomizuka. 1995.Vvehicle Ccontrol For Automated Hhighway Systems For Improve Lateral Maneuverability. in IEEE International conference on systems,Man, and Cybernetics , Vancouver ,B.C., Canada , vol. 1,pp.777-782.
- [13] Duc Do,K.,Zhong-Ping j.,Pan,J. 2004. A Global Output-feedback Controller For Simultaneous Tracking And Stabilization of Unicycle-Type Mmobile Robots. IEEE Trans. Automat Contr., V30,N3,pp. 589-594.
- [14] E J. Kennedy and R. Eberhart. 1994 Particle Swarm Optimization. In Proc.IEEE Int. Conf. Neural Networks , vol. IV, Perth, Australia, 1995,pp.1942–1948.
- [15] Y. Shi, R. Eberhart. 1998. A Modified Particle Swarm Optimizer. *Proc. IEEE Int. Conf. on Evolutionary Computation*, pp. 69-73.
- [16] J S. Tong, H. X. LI and G. Chen. 2003. Adaptive Fuzzy Decentralized Control for a Class of Large-Scale Nonlinear Systems. *IEEE Trans. on Cybern. Man Sys., Part B*.
- [17] H. Ying. 1998. Constructing Nonlinear Variable Gain Controllers via the Takagi – Sugeno Fuzzy Control. *IEEE Transactions on Fuzzy, Syst.*,vol.6,pp.226–234.
- [18] K. M. Zahrani. 2005. Fuzzy Takagi-Sugeno and LMS Baded Control Techniques. M.Sc thesis, Fahd University of Petroleum and Minerals.
- [19] . Hermadi. 2004. Genetic alorathm based test data generator. M.Sc. Thesis, King Fahd university of petroleum & minerals. Dhahran, Saudi Arabia.
- [20] K. C. Ng, M. M. Trivedi. 1998. A Neuro – Fuzzy Controller for Mobile Robot Navigation and Multirobot Convoying. *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, Vol. 28, No. 6, pp. 829-840, ISSN 1083-4419.
- [21] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright and H. M. Tai. 2003. Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm. *Proceedings of the IEEE Conference on Robotics and Automation*, Sacramento, California, 7-12, pp. 1398-1404.
- [22] S. E. Mahmoudi , A. A. Bitaghsir , B. Forouzandeh and A. R. Marandi. 2004. A New Genetic Method for Mobile Robot Navigation. *Electrical and Computer Engineering Dep. University of Tehran*, P.O Box : 14395/515, IRAN.