

PSO-based Optimum Design of PID Controller for Mobile Robot Trajectory Tracking

Turki Y. Abdalla
Department of Computer Engineering
University of Basrah
Basrah,Iraq

Abdulkareem. A. A
Department of Electrical Engineering
University of Basrah
Basrah,Iraq

ABSTRACT

This paper present a particles swarm optimization (PSO) method for determining the optimal proportional – integral derivative (PID) controller parameters, for the control of nonholonomic mobile robot that involves path tracking using two optimized PID 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.

Keywords

Mobile Robot, Particles Swarm Optimization, PID Controller, Kinematic and dynamic model, Trajectory tracking.

1. INTRODUCTION

Autonomous robots may act instead of human beings. The robots are able to accomplish many tasks in dangerous places where humans cannot enter, such 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 that used for control the mobile robot the simplest one is PID controller.

The PID controller has been used to control about 90% of industrial processes worldwide [3]. The main problem of that simple controller is the correct choice of the PID gains and the fact that by using fixed gains, the controller may not provide the required control performance, when there are variations in the plant parameters and operating conditions. Therefore, a tuning process must be performed to insure that the controller can deal with the variations in the plant [4]. To tune the PID controller, there are numbers of strategies, the most famous, which is frequently used in industrial applications, is the Ziegler-Nichols method [3], genetic algorithm GA, etc. Moreover, PSO was another method for tuning procedure. PSO first introduced by Kennedy and Eberhart is one of the modern heuristic algorithms, it has been motivated by the behavior of organisms, such as fish schooling and bird flocking [5]. Generally, PSO is characterized as a simple concept, easy to implement, and computationally efficient. Unlike the other heuristic techniques, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities [6]. In this paper, a novel PSO-based approach to optimally design a PID controller for a mobile robot trajectory tracking is proposed. 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 how PSO is used to design the PID controller optimally for mobile robot to control the velocity and

azimuth. The simulation and the results are presented in section 5.

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 robot 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 [7]. 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.

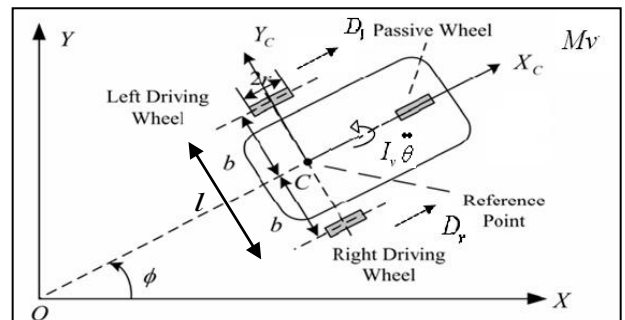


Figure 1: Model of mobile

Let us 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 [8]) for an autonomous vehicle. 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 [9]. 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 [9, 10]:

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

where: x_c and y_c are the two coordinates of the origin C 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) [3, 5, 6]:

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

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 equations in the following forms [7, 3, 6, 8]:

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

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

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

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 [9]:

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

A large number of researchers have used kinematic models to develop motion control strategy for mobile robots, their assumption that these models are valid if the robot has low speed, low acceleration and light load [10]. Dynamic modeling takes into account the forces acting on the vehicle. This model can be constructed using the no-slip condition [11] or allowing wheel slip [12]. In either case, the acceleration is considered.

In dynamic modeling the vehicle's dynamic properties, such as mass, center of gravity, etc. are entered into the equations. To drive this model, the nonholonomic constraints of the system are utilized. Dynamic equation of wheeled mobile robot is described as [13]:

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

$$w = [w_r \ w_l], \tau = [\tau_v \ \tau_w] \quad (8)$$

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

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

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

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

$$(12)$$

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

$$D = \begin{bmatrix} d_{11} & 0 \\ 0 & d_{22} \end{bmatrix} \quad (14)$$

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

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

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.

$\tau = [\tau_v, \tau_w]$: is the vector of torque applied to the wheels of the robot.

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

From the above equation we can get,

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

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]. PSO is one of the optimization techniques and a kind of evolutionary computation technique. The method has been found to be robust in solving problems featuring nonlinearity and non differentiability, multiple optima, and high dimensionality through adaptation, which is derived from the social-psychological theory [15]. The technique is derived from research on swarm such as fish schooling and bird flocking. According to the research results for a flock of birds, birds find food by flocking (not by each individual). The observation leads the assumption that every information is shared inside flocking. Moreover, according to observation of behavior of human groups, behavior of each individual (agent) is also based on behavior patterns authorized by the groups such as customs and other behavior patterns according to the experiences by each individual. The assumption is a basic concept of PSO [16]. In the PSO algorithm, instead of using

evolutionary operators such as mutation and crossover, to manipulate algorithms, for a d-variables optimization problem, a flock of particles are put into the d-dimensional search space with randomly chosen velocities and positions knowing their best values so far (Pbest) and the position in the d-dimensional space. The velocity of each particle, adjusted according to its own flying experience and the other particle's flying experience. For example, the i^{th} particle is represented as $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ in the d-dimensional space. The best previous position of the i^{th} particle is recorded and represented as:

$$Pbest_i = (Pbest_{i,1}, Pbest_{i,2}, \dots, Pbest_{i,d})$$

The index of best particle among all of the particles in the group is $gbest_d$. The velocity for particle i is represented as $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,d})$. The modified velocity and position of each particle can be calculated using the current velocity and the distance from $Pbest_{i,d}$ to $gbest_{i,d}$ as shown in the following formulas[15]:

$$v_{i,m}^{t+1} = w \cdot v_{i,m}^t + c_1 * Rand() * (pbest_{i,m} - x_{i,m}^{(t)}) + c_2 * rand() * (gbest_m -) \quad (18)$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)} \quad (19)$$

$$i = 1, 2, \dots, n ; \quad m = 1, 2, \dots, d$$

where

n Number of particles in the group.

d Dimension.

t Pointer of iteration (generations).

$v_{i,m}^{(t)}$ Velocity of particle i at iteration t
 $v_d^{min} \leq v_{i,d}^{(t)} \leq v_d^{max}$.

w Inertia weight factor.

c_1, c_2 Acceleration constant.

$Rand(), rand()$ Random number between 0 and 1.

$x_{i,d}^{(t)}$ Current position of particle i at iterations.

$Pbest_i$ Best previous position of the i th particle.

$gbest$ Best particle among all the particle in the Population.

4. IMPLEMENTATION OF PSO-PID CONTROLLER

4.1 Fitness Function Unit

In PID controller design methods, the most common performance criteria are integrated absolute error (IAE), the integrated of time weight square error (ITSE), integrated of squared error (ISE) and Mean Square Error (MSE) [17, 18]. These four integral performance criteria have their own advantages and disadvantages. For example, disadvantage of the IAE and ISE criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the ISE performance criterion weights all errors equally independent of time. Although the ITSE performance criterion can overcome the disadvantage of

the ISE criterion, the derivation processes of the analytical formula are complex and time-consuming [18]. The IAE, ISE, ITSE, and MSE performance criterion formulas are as follows:

a) *Integral of Absolute Magnitude of the Error (IAE)*

$$IAE_{Total} = (\int_0^\infty |e_\theta| dt) + (\int_0^\infty |e_v| dt) \quad (20)$$

b) *Integral of the Square of the Error (ISE)*

$$ISE_{Total} = (\int_0^\infty [e_v(t)]^2 dt) + (\int_0^\infty [e_\theta(t)]^2 dt) \quad (21)$$

c) *Integral of Time multiplied by Absolute Error (ITAE)*

$$ITAE_{total} = (\int_0^\infty t |e_\theta(t)| dt) + (\int_0^\infty t |e_v(t)| dt) \quad (22)$$

d) *Mean Square Error (MSE)*

$$MSE_{total} = \left(\frac{1}{n} \sum_{k=1}^n (e_\theta(k))^2 \right) + \left(\frac{1}{n} \sum_{k=1}^n (e_v(k))^2 \right) \quad (23)$$

n : represents number of samples, k : sample time.

In this paper a time domain criterion is used for evaluating the PID controller [15]. A set of good control parameters P, I and D can yield a good step response that will result in performance criteria minimization in time domain. In this paper we'll use Equation (23) as a fitness function.

3.2 Proposed PSO-PID Controller

In this paper the PSO algorithm is used to find the optimal parameters for two PID controllers for the control of velocity and azimuth of mobile robot. Figure 2 shows the block diagram of optimal PID controller for the mobile robot.

In the proposed PSO method each particle contains six members P_1, I_1 and D_1 (parameters of velocity controller), P_2, I_2 and D_2 (parameters of azimuth controller). The search space has six dimension and particles must 'fly' in six dimensional space. The flowchart of PSO-PID controller is shown in Figure 3.

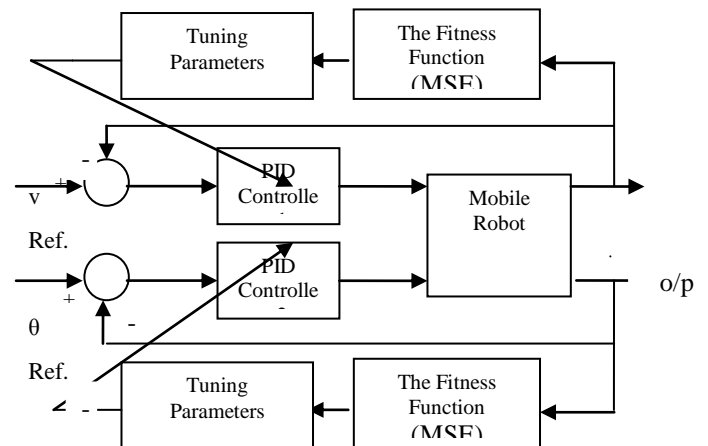


Figure 2: Optimal PID controller

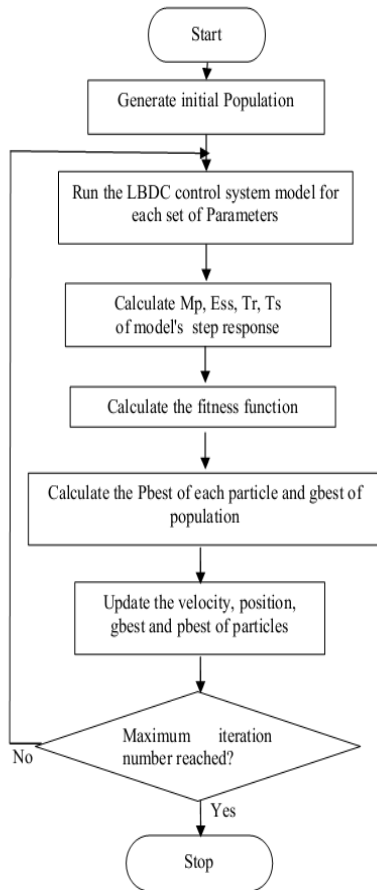


Figure 3: Flowchart of the PSO-PID Control System

Table 1: 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	–

5. SIMULATION AND RESULTS

In order to build the mobile robot given by Equation (17), the values of physical parameters shown in Table 1 are used [13].

By several experiment, the following PSO parameters are used to obtain the optimal performance of the controller:

$$W_{\max}=0.9, W_{\min}=0.4; c_1 = c_2 = 1.2;$$

By doing several experiment using different values for population size and number of iterations. The following values for them are considered to be acceptable.

Population size=30;

Number of iteration =40.

The result obtained in 40 iterations as shown in Table 2:

Table2:Parameters of PID controllers optimized by PSO

P_1	I_1	D_1	P_2	I_2	D_2
214.3	533.3	257	357.9	403.7	176

These values are considered as parameters of the two optimal PID controllers that give the lowest MSE. The system is tested for three different cases as follows:

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

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

where m (slop) = 0.1592, $f(t) = t, 0 \leq t \leq 7$

Figure 4 shows the velocity error, Figure 5 shows the azimuth error, Figure 6 shows the actual and desired path for circular trajectory. The MSE= 0.000095.

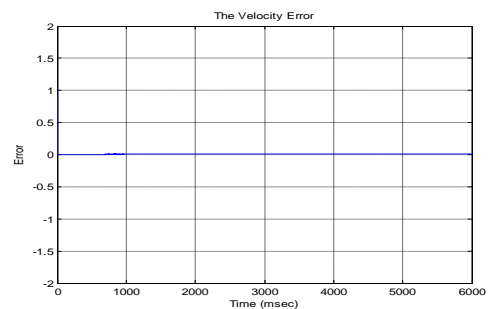


Figure 4: The error in velocity

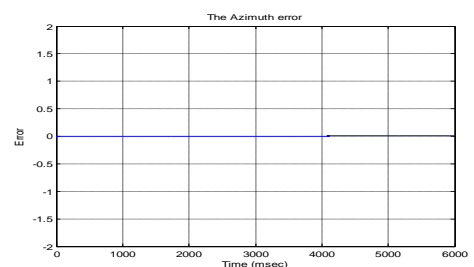


Figure 5: The error in azimuth

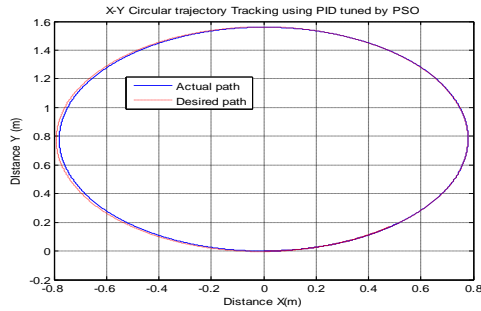


Figure 6: circular trajectory tracking using PSO-PID controller

2) To follow a desired square trajectory.

Figure 7 shows the desired and actual square trajectory Figure 8 shows the error in azimuth and Figure 9 shows the error in velocity . The MSE= 0.00055

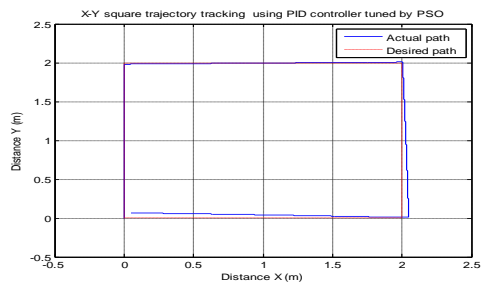


Figure 7: square trajectory tracking using PSO-PID controllers

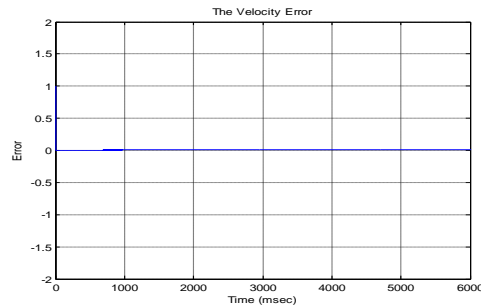


Figure 8: The error in velocity

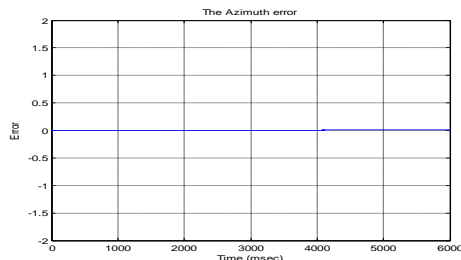


Figure 9: The error in azimuth

3) To follow a desired sine trajectory.

Figure 10 shows the desired and actual sine trajectory Figure 11 shows the error in azimuth and Figure 12 shows the error in velocity . The MSE= 0.000085. Table 3 shows the values of MSE in all cases.

Table 3: MSE for different trajectories

Trajectory	Circular	Square	Sine
MSE	0.000095	0.00055	0.000085

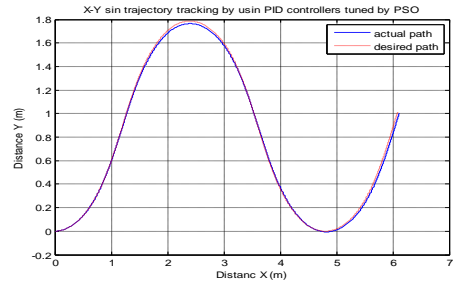


Figure 10: Sine trajectory tracking using PSO-PID controllers

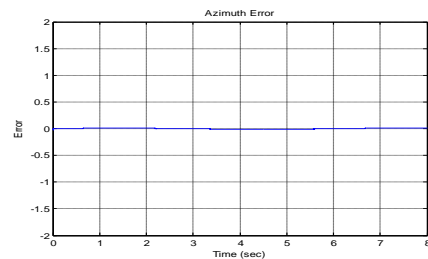


Figure 11 : Error in azimuth for sine trajectory tracing

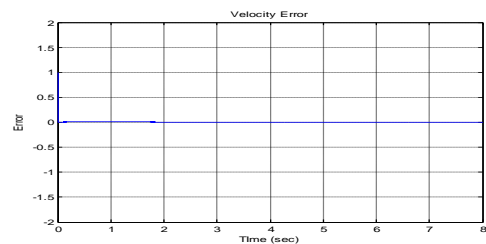


Figure 12: Error in velocity for sine trajectory tracking

6. CONCLUSION

PSO-PID controller is built and implemented in matlab / simulink software package and it is succeeded to solve the trajectory tracking problem in mobile robot . The Particle Swarm Optimization method is utilized to tune/optimize the parameters of PID controller and its gives us a good results in short time relatively with other optimization methods. Simulation results show good tracking performance with small Mean square error. The resulting good tracking capability and small MSE values show that the proposed method is more effective than other methods such as Ziegler-Nichols method and genetic algorithm approach.

7. REFERENCES

- [1] Y. Shinoda, Y. Tan, J. Nakata and R. Beuran , "Collaborative Motion Planning of Autonomous Robots", School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa Japan, 2007.
- [2] R. A. Felder , "Mobile Robot Simulation of Clinical Laboratory Deliveries", M.Sc. Thesis, the University of Virginia, U.S.A, 1998.

- [3] A. Haj-Ali and H. Ying , "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 ,2004, available on the following link: www.sciencedirect.com.
- [4] S. M. Gadoue, D. G. and J. W. Finch, "Tuning of PI Speed Controller in DTC of Induction Motor Based on Genetic Algorithms and Fuzzy Logic Schemes", *International Journal of Advanced Robotic Systems*, University of Newcastle upon Tyne, pp.187 - 194, ISSN 1229-3406, 2005.
- [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks* , vol. IV, Perth, Australia, 1995, pp. 1942–1948.
- [6] M. A. Abido, "Optimal design of power-system stabilizers using particleswarm optimization," *IEEE Trans. Energy Conversion*, vol. 17, pp.406-413, Sep. 2002.
- [7] D. R. Shircliff, "Build A Remote-Controlled Robot", eBook, Copyright © by The McGraw-Hill Companies, 2002.
- [8] .] M. I. Ribeiro and P. Lima, "Kinematics Models of Mobile Robots", Av. Rovisco Pais, 11049-001 Lisboa, April, 2002
- [9] G. Mester, "Obstacle Avoidance of Mobile Robots in Unknown Environments", SISY, International Symposium on Intelligent Systems and Informatics 24-25 Subotica, Serbia, August, 2007.
- [10] A. Albagul and Wahyudi, "Dynamic Modelling and Adaptive Traction Control for Mobile Robots", 30th Annual Conference of the IEEE Industrial Electronics Society, November 2 - 6, Susan, Korea 2004.
- [11] E.N.moret, "Dynamic modeling and control of a car-like robot" Master's thises, Virginia Polytechnic Institute and state University ,2003.
- [12] P.Kachoor and M.tomizuka, "vehicle control for automated highway systems for improve lateral maneuverability," in *IEEE International conference on systems,Man, and Cybernetics* , Vancouver ,B.C., Canada , vol. 1,pp.777-782, Oct. 1995.
- [13] Duc Do,K.,Zhong-Ping j.,Pan,J."A global output-feedback controller for simultaneous tracking and stabilization of unicycle-type mobile robots," *IEEE Trans. Automat Contr.*, V30,N3,pp. 589-594, 2004.
- [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc.IEEE Int. Conf. Neural Networks* , vol. IV, Perth, Australia, 1995,pp.1942–1948.
- [15] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. EnergyConversion*, vol. 19, pp. 384-391, June 2004.
- [16] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans. on Power Systems*, Vol. 15, No. 4, Nov. 2000, pp. 1232 – 1239.
- [17] R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejectionPID controllers using genetic algorithm," *IEEE Trans. Evol.Comput.*,vol. 5, pp. 78–82, Feb. 2001.
- [18] R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejectionPID controllers using genetic algorithm," *IEEE Trans. Evol. Comput.*,vol. 5, pp. 78–82, Feb. 2001.
- [19] Y. Mitsukura, T. Yamamoto, and M. Kaneda, "A design of self-tuningPID controllers using a genetic algorithm," in *Proc. Amer. Contr.Conf.*,San Diego, CA, June 1999, pp. 1361–1365.