# Mohamed J. Mohamed

Department of Control and Systems Engineering, University of Technology, Baghdad-Iraq. <u>60098@uotechnology.edu.iq</u>

## Mohammed K. Hamza

Department of Control and Systems Engineering, University of Technology, Baghdad-Iraq. <u>60115@uotechnology.edu.iq</u>

Received on: 09/12 /2018 Accepted on: 23/05/2019 Published online: 25/12/2019

# Design PID Neural Network Controller for Trajectory Tracking of Differential Drive Mobile Robot Based on PSO

**Abstract** – This paper introduces a nonlinear (Proportional-Integral-Derivative Neural Network) (PID NN) controller for a differential wheeled mobile robot trajectory tracking problem. This neural controller is built based on the principles of neural network (NN) and the equation of conventional structure of PID controller and is applied on kinematic model of the mobile robot. The particle swarm optimization algorithm (PSO) is utilized to find the best values of three PID NN parameters and connection weights that minimize the error between the reference path and the actual path. The results illustrate that the PID NN controller has a satisfied ability to make the mobile robot tracking any path with good performance, high accuracy and acceptable robustness.

Keywords - Two Wheel Mobile Robot, PID Neural Network Controller, Particle Swarm Optimization.

**How to cite this article**: M.J. Mohamed and M.K. Hamza "Design PID Neural Network Controller for Trajectory Tracking of Differential Drive Mobile Robot Based on PSO," *Engineering and Technology Journal*, Vol. 37, Part A, No. 12, pp. 574-583, 2019.

# 1. Introduction

There are many applications need to use mobile robots in order to facilitate doing some jobs. One of these applications is using the mobile robots to move materials between workstations. They can also be used in many fields such as medical environment, industrial, security and even domestic machines. Since the mobile robots are growingly utilized in a broad scope of applications, the researches in this field attracted so much attention in the recent years [1,2] Therefore, many types of controllers were proposed to tackle the trajectory tracking problem the nonholonomic constraints when were considered. There are many techniques proposed to tackle the problems of path tracking and to guarantee that the mobile robots move on the predefined path without any slipping. The classical controller has utilized the techniques of linear and nonlinear feedback control to result closed loop system for the kinematic model of mobile robot [3,4]. The Artificial Intelligent Controllers (AIC) like fuzzy controllers or neural network controllers are greatly used to solve this problem [5,6]. In addition, there are other methods proposed for this problem such as, the proposed adaptive controllers based on dynamics of mobile robots [7], also the method of back stepping control technique was proposed in [8,9] but, the designing method and the resultant controller structure are very complex. The sliding mode technique was introduced in [10], in this method, the corporation between the kinematic

and dynamic controllers is applied to track the required path. One of the disadvantages of traditional neural network is time consuming in learning because of the random initialization weights of the network connections. This property leads the algorithm to unable achieve fast response of the control system and does not give a stationary performance. In [11] a new control algorithm using a PID NN controller is introduced, this algorithm is a type of Multi-Layer Perceptron (MLP) but the hidden layers of the network are different from traditional layers. This algorithm is composed of three elements, proportional neuron, integral neuron, and derivative neuron. Therefore, PID NN exploits the advantages of PID, the neural network convergence and short learning time [12] [13]. There are two categories of PIDNN, the singleoutput (SPIDNN) and the multi-output (MPIDNN). The two types of the networks can be learned off-line or on-line to optimize the values of the connection weights between the three kinds of network layers. In this paper, PSO algorithm was utilized to find the optimum values of connection weights between neural network layers. The rest of this paper is arranged as follows: in section two, the kinematic model equations of the differential wheeled nonholonomic mobile robot are derived; in section three, the proposed structure of the PID neural network controller is depicted as well as the main steps of applying the proposed PSO are explained, in section four the simulations and the

obtained results are illustrated, in section five the main points of conclusions are introduced.

#### 2. Kinematical Model of Mobile Robot

The main property of this kind of mobile robot is; the wheel rotations are limited to only one axis. So, the navigation of mobile robot is controlled by changing the left or right-wheel speed on either side of the mobile robot. Figure 1 depicts the kinematic scheme of a differential wheel mobile robot, where  $V_I$  is the robot's linear velocity, r represents the radius for each wheel, L represents the distance between the two wheels,  $V_R$  symbolizes the right-wheel angular velocity, xand y are the two coordinates of the mobile robot position, and the angle of mobile robot orientation is denoted by  $\theta$  [14,15].



Figure 1: Model of nonholonomic mobile robot

Depending on the principles of motion to derive a kinematic model of rigid body, the mobile robot can be characterized by the two Eqs. (1) and (2), where  $V_w$  is the mobile robot angular velocity ,  $V_I$  is the mobile robot linear velocity [14,15].

$$V_{w(t)} = \frac{d\theta(t)}{dt} \tag{1}$$

The definition of a mobile robot total linear velocity can be as; a multiplier value of an angular velocity and an instantaneous bending radius of a mobile robot trajectory R.  $V_I = V_{w(t)}R(t)$ 

Therefore, the left-wheel velocity and right-wheel velocity can be represented by Eqs (3) and (4) respectively.

$$V_{L(t)} = \left(R(t) + \frac{L}{2}\right)V_{w}$$
(3)
$$V_{R(t)} = \left(R(t) - \frac{L}{2}\right)V_{w}$$
(4)

From the solution of Eq. (3) and Eq. (4), an instantaneous bending radius of the mobile robot trajectory compared to the midpoint axis can be computed as shown in Eq. (5).

$$R(t) = \frac{L(V_L(t) + V_R(t))}{2(VL(t) - V_R(t))}$$
(5)

Now, by rewriting Eq. (2). The resulting Eq. (6) represents the mobile robot angular velocity [14,15]:

$$V_w(t) = \frac{V_L(t) - V_R(t)}{L}$$
(6)

At last, Eq. (7) represents the mobile robot linear velocity.

$$V_{I}(t) = \frac{V_{L}(t) + V_{R}(t)}{2}$$
(7)

The kinematic equations of mobile robot can be defined as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_l \\ V_w \end{bmatrix}$$
(8)

$$\dot{x}(t) = V_l \cos \theta(t) \tag{9}$$

$$\dot{y}(t) = V_I \sin\theta(t) \tag{10}$$

 $\dot{\theta}(t) = V_w(t) \tag{11}$ 

By performing integration to both sides of the Eqs (9), (10) and (11):

$$x(t) = x_o + \int_0^t V_I(\tau) \cos \theta(\tau) d\tau$$
(12)

$$y(t) = y_0 + \int_0^t V_I(\tau) \sin \theta(\tau) d\tau$$
(13)

$$\theta(t) = \theta_o + \int_0^t V_w(\tau) d\tau$$
(14)

Where,  $(x_o, y_o, \theta_o)$  is the initial pose of position and orientation of mobile robot. The kinematics model of a two wheeled mobile robot is described by Eqs. (12), (13) and (14). The position and orientation of the kinematic model of mobile robot represent the control variables (x, y,and  $\theta$ ) while the angular velocities of left wheel and right wheel ( $V_L$ ,  $V_R$ ) of the mobile robot represent the control variables of the system.

## 3. The Proposed PID NN Controller

In general, the control system of the mobile robot composed of three major portions; the kinematic model of the differential wheel mobile robot, the suggested neural network controller PID NN, and the proposed PSO algorithm for adjusting the parameters of the controller. Figure 2 depicts the control system block diagram and the three parts of the controller.



Figure 2: The proposed structure of the PID NN control system

## I. The Design of PIDNN Controller

The proposed PID NN controller is represented as a forward multi-layer neural network. The input/output functions of hidden layer neurons are selected according to PID control law, where these neuron functions are made to represent the same role of proportional, integral and derivative actions. The PID control law can be defined as;

$$u = kp \left[ e + \frac{1}{\tau_i} \int_{to}^t e dt + \frac{1}{\tau_i} e' \right]$$
(15)  

$$u = kpe + k_i \int_{to}^t e(\tau) d\tau + kde'$$
(16)

Where, e is the error between the system input and desired output.

The accumulator error or the integral function of the error is employed to realize better tracking. e' is the error derivative, or the change rate of error which assists to decrease the overshoot of the output.

The suggested PID NN controller consists of an input layer, hidden layer, and the output layer (3nX9nX2n). Figure 3 shows this neural network. The first layer or input layer has three inputs;  $e_x, e_y$ , and  $e_\theta$ . These inputs represent the distance error between the required or reference trajectory and the actual trajectory of the mobile robot. The hidden layer owns nine neurons and the inputs to these nine neurons represent the gains of three PID controllers, because we have three terms must be controlled (x, y, z), where each neuron represents either P or I or D function.

 $V_{ji}$  symbolize the weights connections between

input layer and hidden layer, while  $W_{kj}$  symbolizing the weights connections between the hidden layer and output layer.



**Figure 3: The PID NN Neural Network Controller** The input/output functions of PID NN at any time of sampling subject to the following specifications:

1-All input layer neurons are described by the following input/output function.

$$i/pi_i(k) = e_{\sigma i}(k) = o/pi_i(k)$$
  

$$i = 1.2.3 , \sigma = x, y, \theta$$
(17)  
Where:

 $i/p_{i_i}(k)$  the input value

 $o/p_{i_i}(k)$  the output value of input layer neuron.

i, is the part number of sub network.

2- The hidden layer neuron input function is presented as:

 $i/ph_{\sigma j}k = vh_{\sigma ij} \cdot o/pi_i(k)$ (18)  $\sigma = (x, y \text{ and } \theta)$ 

j = (1, 2, 3 for each error).

Where  $i/ph_{\sigma j}$ , are the hidden layer input values and  $vh_{\sigma ij}$  are the weights of connections between the input layer and hidden layer of neural network.

The outputs of the hidden neurons, which participate to create proportion action is:

$$o/ph_{\sigma 1}(k) = H(i/ph_{\sigma 1}(k))$$
(19)

The outputs of the hidden neurons, which participate to create integral action is:

 $o/ph_{\sigma_2}(k) = H(i/ph_{\sigma_2}(k) + o/ph_{\sigma_2}(k-1))$  (20) The outputs of the hidden neurons, which participate to create differential action is:

$$o/ph_{\sigma_3}(k) = H(i/p_{\sigma_3}(k) - i/ph_{\sigma_3}(k-1))$$
 (21)  
Where;

 $H(\text{net}) = \frac{2}{1+e^{-\text{net}}} - 1$  (Sigmoid function). (22) 3-The neuron function of the output layer is a summation function which sum all its inputs (outputs of hidden layer neurons) and give its output. This function is illustrated as follows.

$$o/po_k(k) = \sum_{k=1}^{2} \sum_{j=1}^{3} wh_{\sigma j k}(k), o/ph_{\sigma j}(k)$$
  
for  $k = 1, 2$  (23)

The two outputs o/p (k) of two output layer neurons represent the control actions to the mobile robot  $V_R$  and  $V_L$ . Where,  $W_{jk}$  represent the weight connections between the hidden layers and the output layers. The real numbers of the output neurons are k =1,2. The characteristics of PID control of PIDNN controller are reflected by the connections between the input layer and hidden layer as well as the functions of the hidden layer neurons. The optimal values of weighted coefficients  $Vh_{ij}$  and  $wh_{jk}$  are found by the PSO algorithm.

#### II. Particle Swarm Optimization (PSO).

The PSO is considered as another kind of evolutionary computation technique which was proposed by Kennedy and Eberhart [16]. This algorithm is based on simplified social behavior of animal such as bird flocking and fish schooling. In PSO, each solution is represented as a flying bird in the searching space which is called particle. The algorithm of PSO simulates the social behavior of each particle in the swarm. At each generation or iteration, the algorithm adapting the trajectory of each particle in the swarm based on the best discovered trajectory for this particle and toward the trajectory of global best particle in the entire swarm. Many optimization problems are solved successfully by the PSO algorithm. In comparison, between PSO and genetic algorithm, the PSO has advantages which are easier than GA in the implementation and fewer adjusted parameters to control the algorithm. In this paper, in order to discover the optimal parameters of the PIDNN controller, PSO algorithm was suggested to find them in order to precisely control the velocities of right wheel and left wheel  $(V_R, V_L)$ .

1. Procedure of PSO

The mathematical description of PSO is as follow;

Suppose the search space has D dimensions and the number of particles is n.

The vector  $x_{j=}(x_{j1}, x_{j2}, \dots, x_{jd})$  performs the vector of position of *i-th* particle in the population.

The  $p_j = (j_1.j_2.j_{3d})$  is the best position for particle *j* searched by now, and the best position of whole particles in the swarm is symbolized as:

$$p_g = (p_{g1}, p_{g2}, p_{g3})$$

The vector  $\vec{v_j} = (v_{j1}, v_{j2}, \dots, v_{jd})$  is the rate of change for the *i-th* particle position. The position

of each particle is updated depending on the following formulates.

$$v_{jd}(k + 1) = w * v_{jd}(k) + c_1 * rand (p_{jd}(k) - x_{jd}(k) + c_2 * rand * (p_{jd}(k) - x_{jd}(k)) (24)x_{jd}(k + 1) = x_{jd}(k) + v_{jd}(k) (25)$$

Where, c1 and c2 are acceleration coefficients that have positive constant values.

The rand () is a function of random number or a function which generates a random number in range (0 to 1). The symbol w is the weight of inertia (where, w < 1).

#### 2. Tuning PID NN Parameters by PSO.

The following steps illustrate in details the implementation of PSO algorithm to discover the optimum values of the PIDNN controller parameters.

**Step 1**: Determine the dimension space of the problem, we have 27wieghts.

**Step 2**: Initialize group of particles as an array with random velocities and random positions

**Step 3**: Using the fitness function Eq. (26) to evaluate the fitness of each particle.

$$fit = \frac{1}{k} \int_{n=1}^{k} (e_x^2(n) + e_y^2(n) + e_\theta^2(n)) \quad (26)$$

**Step 4**: Choose the best individual position saw so far for each particle in the group.

**Step 5**: Decide the superior universal position encountered so far by all the particles in the group.

**Step 6**: Using Eq. (24) to update the velocities of all particles.

**Step7**: Using Eq. (25) to update positions of all particles.

**Step 8**: If the maximum number of iterations is reached go to step 9 otherwise increase the counter of iteration by one and return to step 3.

**Step 9:** The superior universal position encountered so far is considered as the solution for the PIDNN controller parameters.

#### Step 10: End.

From Figure 2 which shows the block diagram of a differential wheel mobile robot, the error is determined from the difference between the desired path  $q_r = (x_r \ y_r \ \theta_r)^T$  and the actual path  $q_m = (x_m \ y_m \ \theta_m)^T$  as depicted below:

$$\begin{bmatrix} e_{mx} \\ e_{my} \\ e_{m\theta} \end{bmatrix} = [R] \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \\ \theta_m \end{bmatrix}$$
(27)

where;

Rotation Matrix: R =  $\begin{bmatrix} \cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1 \end{bmatrix}$ (28)

The resulting errors are fed to the rotation matrix which are utilized to convert the frame of coordinate i.e from the global frame to the local frame of the mobile robot. The rotation matrix output is passed into PID NN controllers. The error of each control variable is passed to the PID NN controller. Therefore, there are three PID NN controller, each one consists of three elements (P, I and D). The first PID NN controller for x axis and the second PID NN controller for y axis while the third PID NN controller for angle  $\theta$ . The proposed PSO algorithm is applied to tune off-line the parameters of the PID NN controllers. The outputs of the PID NN controllers produce the right-wheel and left- wheel velocities  $(V_R, V_L)$ of the mobile robot which are used to apply at the kinematic model of mobile robot. Each particle represents a certain path which is needed evaluate. The actual to output  $q_m =$  $(x_m \quad y_m \quad \theta_m)^T$  for each particle is calculated and compared with the desired output path in order to evaluate the fitness of this particle. This operation is repeated every generation to adjust the particle's position until to reach the minimum tracking error between the desired and actual path.

#### 4. Analysis and Results

The proposed control system of mobile robot shown in Figure 2 was simulated using MATLAB 2013. The differential wheel mobile robot kinematic model was employed and the values of its physical parameters were chosen as: (model is taken from [9]: M=0.65kg, L=0.105 m, r =0.033 m and the sampling time is 0.1 second). The proposed PSO algorithm is applied to find the optimal values of PID NN controller's weights. The control parameters of PSO algorithm are chosen as: c1=2, c2=2, rand between (0-1), swarm population size 30 particles, the number of maximum iterations 100. In order to verify the ability of the PID NN controller to force the mobile robot tracking any path, two paths were taken as test in order to check.

**Trajectory path (1):** The first path has like infinity symbol and the following equations described this type of path.

$$xr(n) = 0.75 + 0.75 * sin\left(2 * pi * \frac{t}{100}\right)$$
$$yr(n) = sin(4 * pi * t/100)$$

 $\theta_r(n)$ 

$$= atan2\left(\left(\frac{yr(n) - yr(n-1)}{t + eps}\right), \left(\frac{xr(n) - xr(n-1)}{t + eps}\right)\right)$$

Where: t = 0.01 sec is sampling time and the observation time is 100 sec. yr(n), yr(n-1), xr(n), xr(n-1) and  $\theta r(n)$ ,  $\theta r(n-1)$  are the current and past position and orientation. The initial position and orientation of this path is (0.75, 0, 0). Figures 4-7 show the actual and the reference paths and responses for x, y and  $\theta$ . The mobile robot left-wheel and right-wheel velocities  $V_L$ ,  $V_R$  are shown in Figures 8 and 9 respectively. Finally, Figure 10 depicts the MSE between actual infinity path and reference path during tracking.



Figure 4: The actual and reference infinity paths of mobile robot



Figure 5: The *x*-variable tracking reference infinity path



Figure 6: The *y*-variable response during tracking reference infinity path



Figure 7: The  $\theta$ -variable response during tracking reference infinity path



Figure 8: The velocity response of left-whee during tracking reference infinity path.



Figure 9: The velocity response of right wheel during tracking reference infinity path



Figure 10: The MSE between reference and actual infinity paths during tracking.

**Trajectory path (2):** The second path is like star symbol and this path is described by the following equations.





Figure 11: The mobile robot reference and actual star path



Figure 12: The *x*-variable response during tracking reference star path.

The sampling time t = 0.01 and the observation time is 60 sec. The path initial position is taken as (0, 0, and 4.12). Figures 11-14 show the actual and desired star path and responses for  $x.y, \theta$ . The mobile robot left-wheel and right-wheel velocities  $V_L, V_R$  are shown in Figures 15 and 16 respectively. Eventually, Figure 17 depicts the MSE between reference and actual infinity paths during tracking.



Figure 13: The y-variable response during tracking reference star path.



Figure 14: The  $\theta$  variable response during tracking reference star path



Figure 15: The left wheel velocity response during tracking reference star path



Figure 16: The right wheel velocity response during tracking reference star path



Figure 17: The MSE between the actual and reference star paths

The obtained results show clearly that, the suggested PID NN controller is effective to make the mobile robot tracks the required trajectory with a very small tracking error. The MSE values for all variables x, y and  $\theta$  for the two paths mentioned above are shown in Table 1.



Figure 18: The mobile robot reference and actual star paths



Figure 19: The x-variable response



Figure 20: The y-variable response



Figure 21: The  $\theta$ -variable response



Figure 22: The velocity response of left wheel



#### Figure 23: The velocity response of right wheel

Figure 3 depicts that there are 27 weights in the structure of the proposed PID NN controller. The best values of these weights were determined by the PSO algorithm are illustrated in Table 2 and Table 3.

In fact, the proposed PID NN controller structure is basically a special structure. The PID NN controller is equivalent to the classical PID controller when suitable connective weights are chosen. These weights represent PID controller gains proportional, integral, and derivative as:  $k_{Pij}$ .  $k_{Iij}$  and  $K_{Dij}$ .

In Table 2 and Table 3 the optimal values of weights for the three PID NN controllers are shown. These values of weights, specify the lowest possible mean square error (MSE) between the reference variables and actual variables for position and orientation. Finally, from the obtained results, we have noticed clearly that the proposed PID NN intelligent controller proves its ability to drive the differential wheeled mobile robot kinematic model well for any path. Moreover, in order to check the robustness of the PID NN controller, another initial posture (0.2, 0.1, -3.9). (0, 0, -4.1) was applied to test the behavior of mobile robot. The results are shown in Figures 18 - 23.

The resultant Figures 18 - 23 show that the proposed PID NN controller results a good performance and fast response to track the desired trajectory when the disturbance applied for a short time or the deviation of the actual trajectory from the reference trajectory is small. This property confirms that the method is feasible and applicable.

Table 1: The MSE values for all variables of the two reference paths

Type of Trajectory	MSE Value of <i>x</i> -axis	MSE Value of y-axis	MSE Value of θ-axis	MSE Value of path
Path (1)	4.2417e-07	4.8446e-09	4.9408e-07	9.2108e-07
Path (2)	9.7151e-05	1.0155e-05	4.1842e-05	0.0001491

Table 2:	The values	of weights	between inj	put and hidd	en layer

Weight's Label	Weight's Value
$V_{x1}$	-4.524
$V_{x2}$	0.69249
$V_{x4}$	-2.76
$V_{y1}$	4.0446
$V_{y2}$	-0.266
$V_{y4}$	5.5977
$V_{\theta 1}$	0.82878
$V_{\theta 2}$	-0.902
$V_{ heta 4}$	1.0471

Table 3: The values of weights between hidden layer and output layer

Weight's	Weight's	Weight's	Weight's
Label	Value	Label	Value
$w_{x11}$	-1.9825	$W_{y21}$	0.14448
$W_{x12}$	0.14427	$W_{y22}$	-2.5786
$W_{x14}$	-1.4454	W <sub>y24</sub>	4.4589
$W_{x21}$	-0.76918	$W_{\theta 11}$	1.8011
$W_{x22}$	2.8557	$w_{\theta 12}$	1.6464
$W_{\chi 24}$	-1.2544	$W_{\theta 12}$	-0.97992
$W_{y11}$	0.40985	$w_{\theta 21}$	4.4647
$W_{v12}$	1.7484	$W_{\theta 22}$	-1.441
<i>w</i> <sub>y14</sub>	4.4662	$W_{\theta 24}$	1.0617

Type of	MSE Value of	MSE Value of	MSE Value of	MSE Value of
Trajectory	<i>x</i> - axis	<i>y</i> -axis	<i>Ө</i> -axis	path
Star Path test (2)	9.7776e-05	0.00021269	0.00067448	0.00098494

# 5. Conclusion

This paper introduced the proposed PID NN controller to apply on the kinematic model of differential wheel mobile robot. The role of PSO algorithm in the design was to find the best controller parameters that minimize the error between the reference trajectory and the actual trajectory. In fact, the proposed PID NN controller is a dynamic multilayer neural network. This controller consists of three PID sub controllers. One PID sub controller for each control variable x, y, and  $\Theta$ . Each PID sub controller has three neurons P- neuron, I-neuron and D-neuron. Therefore, the PID NN controller has nine neurons in its hidden layer. Table 4 shows the values of MSE for all control variables when the initial posture is changed, although the controller parameters were not adapted. Figure 18 shows that the controller can quickly force the mobile robot to join the reference path with small values of tracking errors.

The above results display that the PID NN controller has a good ability to make the mobile robot tracking any path with good performance, high accuracy and acceptable robustness.

## References

[1] K.Z. Karam and A. Albagul, "Dynamic Modelling and Control Issues for a Three Wheeled Vehicle," Proc. 5<sup>th</sup> Int. Con. on Control Automation, Robotics and Vision, Singapore, 1998.

[2] G. Walsh, D. Tilbury, S. sastry, et al, "Stabilization of Trajectories for Systems with Nonholonomic Constraints," IEEE Tran. Automatic Control, Vol. 39, No. 1, pp. 216-222, 1994.

[3] B.D. Andrea-Novel, G. Camnion and G. Bansit, "Control of Nonholonomic Wheeled Mobile Robots by State Feedback Linearization," *Robotics Research*, Vol. 14, No. 6, pp.543-559, 1995. [4] N. Sakar, X. YLin and V. Kumar, "Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile," Robotics *Research*, Vol.13, No. 1, pp.55-69, 1994.

[5] M.J. Mohamed and M.Y. Abbas, "Design a Fuzzy PID Controller for Trajectory Tracking of Mobile Robot," Engineering and Technology Journal, Baghdad, Iraq. Vol 36, Part A, No. 1, pp.100-110, 2018.

[6] T. Das and N. Kar, "Design and Implementation of an Adaptive Fuzzy Logic-based Controller for Wheeled Mobile Robots," IEEE Transactions on Control System Technology, 14, 3, 501-510, 2006.

[7] F.N. Martins, W.C. Celesta, R. Carelli, M. Sarcinelli–Filho, and T. F. Bastos–Filho, "An Adaptive Dynamic Controller for Autonomous Mobile Robot Trajectory Tracking," Control Engineering Practice. Vol. 16, 2008.

[8] J.Z. Ping and H. Nijmeijer, "Tracking Control of Mobile Robot: A case Study in Back Stepping," *Automatic*, Vol. 33, No.7, pp.1393-1399, 1997.

[9] W.W. Guo, C.H. Tang, and W.Y. Juan, "Global Trajectory Tracking Control of Mobile Robots," *Acta automatic Sinica*, Vol.27, No.3, pp. 326-331, 2001.

[10] N.A. Martins, D.W. Bertol, E.R. DePieri, E.B. Castelan and M. M. Dias, "Neural Dynamic Control of a Nonholonomic Mobile Robot Incorporating the Actuator Dynamics," CIMCA 2008, IAWTIC 2008, and ISA2008, IEEE Computer Society, 2008.

[11] H. Shu, "Analysis of PID Neural Network Multivariable Control Systems," Acta Automatica Sinica, 25(1), 105–110, 1999.

[12] H. Shu, Y. Pi, "PID Neural Networks for Timedelay systems [J]," Computers and Chemical Engineering, 24, 2, 859 – 886, 2000.

[13] H. Shu. PID "Neural Networks Algorithm for Control Application [M]," Beijing: Publishing House of National Defense Industry, 2006.

[14] A.S. Al-Araji "Design of a Cognitive Neural Predictive Controller for Mobile Robot," PhD Thesis Electronic and Computer Engineering, School of Engineering and Design, Brunel University, United Kingdom, 2012.

[15] K.E. Dagher and A.S. Al- Araji, "Design of a Nonlinear PID Neural Trajectory Tracking Controller for Mobile Robot Based on Optimization Algorithm," Eng. & Tech. Journal, Vol.32, Part (A), No.4, 2014.

[16] C-Y. Tsai and K-T. Song, "Visual Tracking Control of a Wheeled Mobile Robot with System Model and Velocity Quantization Robustness," IEEE Transactions on Control Systems Technology, Vol. 17, No. 3, pp. 520-527, 2009.