A Comparative Study of Various Intelligent Algorithms Based Nonlinear PID Neural Trajectory Tracking Controller for the Differential Wheeled Mobile Robot Model

Ahmed Sabah Al-Araji
Assistance Professor
Control and Systems Engineering Dept - University of Technology
E-mail:ahmedsas2040@yahoo.com

ABSTRACT

This paper presents a comparative study of two learning algorithms for the nonlinear PID neural trajectory tracking controller for mobile robot in order to follow a pre-defined path. As simple and fast tuning technique, genetic and particle swarm optimization algorithms are used to tune the nonlinear PID neural controller's parameters to find the best velocities control actions of the right wheel and left wheel for the real mobile robot. Polywog wavelet activation function is used in the structure of the nonlinear PID neural controller. Simulation results (Matlab) and experimental work (LabVIEW) show that the proposed nonlinear PID controller with PSO learning algorithm is more effective and robust than genetic learning algorithm; this is demonstrated by the minimized tracking error and obtained smoothness of the velocity control signal, especially when external disturbances are applied.

Key words: genetic algorithm, particle swarm optimization, nonlinear PID controller, NI mobile robots, trajectory tracking.

دراسة مقارنة لخوارزميات ذكية متنوعة أساسه مسيطر تتابع مسار عصبي لأخطي تناسبي تكاملي تفاضلي لنموذج التحرك التفاضلي لإنسان آلي متنقل.

أحمد صباح عبد الأمير الأعرجي
أستاذ مساعد دكتور
قسم هندسة السيطرة والنظم - الجامعة التكنولوجية

الخلاصة

يقدم هذا البحث, دراسة مقارنة لخوارزميتين لتعليم مسيطر تناسبي تكاملي تفاضلي عصبي لاخطي تتابعي لمسار عجلة الإنسان آلي متحرك لكي يتبع مسار مستمر معرف مسبقا.
كتقنيات سهله وسريعة التنغيم, لقد تم استخدام الخوارزمية الوراثية وخوارزمية حشد الجسيمات الامثلية لتنغيم عناصر المسيطر اللاخطي التناسبي التكاملي التفاضلي لإيجاد أفضل أشارة سرعة لعجلة للإنسان الآلي المتحرك الحقيقي. لقد تم استخدام الدالة الفعالة موجة البوليونك في هيكلية المسيطر العصبي.
من خلال نتائج المحاكاة والعمل التجريبي, أثبتت أن المسيطر التناسبي التكاملي التفاضلي العصبي اللاخطي المقترح تنغيمه بواسطة خوارزمية حشد الجسيمات الامثلية أكثر فعالية و متانة مقارنة بالمسيطرة الذي تم تنغيمه بواسطة الخوارزمية الوراثية, وهذا واضح من خلال تقليل الخطأ ألتتابعي لمسار الإنسان الآلي المتحرك مع توليد أشارة سرعة ناعمة, برغم من وجود التأثير الاضطرابي الخارجي.

الكلمات الرئيسية:
الخوارزمية الوراثية, حشد الجسيمات الأمثلية, المسيطر التناسبي التكاملي التفاضلي اللاخطي, الإنسان الآلي المتحرك, تتابع مسار.

1. INTRODUCTION

In the last decade, there has been an increasing amount of research on the subject of wheel-based mobile robots which have attracted considerable attention in various industrial and service applications. For example, room cleaning, lawn mowers, factory automation, transportation, nuclear-waste cleaning, **Wai,** and **Liu, 2009**.

These applications require mobile robots to have the ability to track specified path stably; therefore, several studies have been published for solving the mobile robot path tracking control problems which can be classified into three categories: The first category is the position estimation control approach for navigation problems of the mobile robot on interactive motion planning in dynamics environments and obstacle motion estimation, **Chang, et al., 2009**. The second category for navigation problems of the mobile robot is path planning and execution. The path planning is generated based on a prior map of the environment while the executed path is planned using certain optimization algorithms based on a minimal time, minimal distance or minimal energy performance index. Many methods have been developed for avoiding both static and moving obstacles as presented in, **Sahin,** and **Zergeroglu, 2007**. The third category for the navigation problems of mobile robot is the design and implementation of the driving control that the mobile robot must track to follow a desired path accurately and minimize the tracking error.

Tracking errors of mobile robot causes collisions with obstacles due to deviations from the planned path and also causes the robot to fail in accomplishing the mission successfully. It also causes an increase of the traveling time, as well as the travel distance, due to the additional adjustments needed to satisfy the driving sates. The major reasons for tracking error for mobile robot are the small rotation radius or not constant on the path such as the complex curvature or randomly curvature, **Takanori, et al., 2000**.

In, **Mnif,** and **Touati, 2005.** artificial intelligent controllers were carried out using neural networks or fuzzy inference in order to control the trajectory tracking for the mobile robot while the traditional control methods for path tracking have been used linear or non-linear feedback controller. There are other techniques for trajectory tracking controllers such as predictive control technique. Predictive approaches to path tracking seem to be very promising because the reference trajectory is known beforehand. Model predictive trajectory tracking control was applied to a mobile robot where linearized tracking error dynamics was used to predict future system behaviour and a control law was derived from a quadratic cost function penalizing the system tracking error and the control effort, **Klancar,** and **Skrjanc, 2007.**

In addition, an adaptive trajectory-tracking controller based on the robot dynamics was proposed in, **Park, et al. 2010**. Intelligent control architecture for two autonomously driven wheeled robot was developed in, **Su, et al. 2010**, that consists of the fuzzy inference as main controller and the neural network is an auxiliary part.

The fundamental essence of the contribution of this work can be understood considering the following points.

- The analytically derived control law which has significantly high computational accuracy to obtain the best control action and lead to minimum tracking error of the mobile robot based on genetic algorithm and particle swarm optimization.
- Investigation of the controller robustness and adaptation performance through adding boundary unknown disturbances.
- Verification of the proposed controller capability of tracking continuous trajectory is done by an experimental work using NI mobile robot model.

Simulation results and experimental work show that the proposed controller with PSO learning algorithm is more robust and effective than controller with genetic learning algorithm in terms of

minimum tracking error and in generating best velocity control action despite of the presence of bounded external disturbances.
The remainder of the paper is organized as follows: Section two is a description of the kinematics model of the differential wheeled mobile robot model and check controllability for the mobile robot. In section three, the proposed nonlinear PID neural controller is derived based on genetic algorithm and particle swarm optimization. The simulation results and experimental work of the proposed controller are presented in section four and the conclusions are drawn in section five.

2. DIFFERENTIAL WHEELED MOBILE ROBOT PLATFORM

The schematic of the differential drive mobile robot, shown in **Fig. 1**, consists of a cart with two driving wheels mounted on the same axis and one castor wheel as an omni-directional in the front of cart in order to carry the mechanical structure and keep the platform of the mobile robot more stable, **Al-Araji, et al, 2013**. Two independent analogous DC motors are the actuators of left and right wheels for motion and orientation. The two wheels have the same radius denoted by r, and L is the distance between the two wheels. The center of mass of the mobile robot is located at point c, center of axis of wheels.
The pose of mobile robot in the global coordinate frame $[O,X,Y]$ and the pose vector in the surface is defined as:

$$q = (x, y, \theta)^T \tag{1}$$

where $q(t) \in \Re^{3\times1}$,
x and y are coordinates of point c and θ is the robotic orientation angle measured with respect to the X-axis. These three generalized coordinates can describe the configuration of the mobile robot.
It is assumed that the mobile robot wheels are ideally installed in such a way that they have ideal rolling without skidding, **Yang, et al. 2004**, as shown in Eq. (2):

$$-\dot{x}(t)\sin\theta(t) + \dot{y}(t)\cos\theta(t) = 0 \tag{2}$$

Therefore, the kinematics equations in the world frame can be represented as follows **Han, et al. 2008**:

$$\dot{x}(t) = V_l(t)\cos\theta(t) \tag{3}$$

$$\dot{y}(t) = V_l(t)\sin\theta(t) \tag{4}$$

$$\dot{\theta}(t) = V_w(t) \tag{5}$$

where V_l and V_w, are the linear and angular velocities respectively.
In the computer simulation, the currently form of the pose equations are as follows:

$$x(k) = 0.5[V_R(k) + V_L(k)]\cos\theta(k)\Delta t + x(k-1) \tag{6}$$

$$y(k) = 0.5[V_R(k) + V_L(k)]\sin\theta(k)\Delta t + y(k-1) \tag{7}$$

$$\theta(k) = \frac{1}{L}[V_L(k) - V_R(k)]\Delta t + \theta(k-1) \tag{8}$$

where $x(k), y(k), \theta(k)$ are the components of the pose at the k step of the movement and Δt is the sampling period between two sampling times.

Using Jacobi-Lie-Bracket to check controllability of the nonlinear MIMO kinematic mobile robot system in Eqs. (3, 4, 5)**,** the accessibility rank condition is globally satisfied controllability. The mobile robot kinematics can be described by the left and right velocities as follows:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0.5\cos\theta(t) & 0.5\cos\theta(t) \\ 0.5\sin\theta(t) & 0.5\sin\theta(t) \\ 1/L & -1/L \end{bmatrix} \begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix} \tag{9}$$

$$[\dot{q}] = [f]V_R(t) + [g]V_L(t) \tag{10}$$

and f and g can be defined as two vectors with components:

$$f = \begin{bmatrix} 0.5\cos\theta(t) \\ 0.5\sin\theta(t) \\ 1/L \end{bmatrix} \text{ and } g = \begin{bmatrix} 0.5\cos\theta(t) \\ 0.5\sin\theta(t) \\ -1/L \end{bmatrix} \tag{11}$$

$$[f,g] = \begin{bmatrix} [f,g]^1 \\ [f,g]^2 \\ [f,g]^3 \end{bmatrix} = \begin{bmatrix} \frac{-1}{L}\sin\theta(t) \\ \frac{1}{L}\cos\theta(t) \\ 0 \end{bmatrix} \tag{12}$$

$$rank\{f,g,[f,g]\} = rank\begin{bmatrix} 0.5\cos\theta(t) & 0.5\cos\theta(t) & \frac{-1}{L}\sin\theta(t) \\ 0.5\sin\theta(t) & 0.5\sin\theta(t) & \frac{1}{L}\cos\theta(t) \\ 1/L & -1/L & 0 \end{bmatrix} \tag{13}$$

The determent of the matrix in Eq. (13) is equal to $(1/L^2) \neq 0$, then the full rank of matrix is equal to 3, therefore, the system in Eqs. (3, 4 and 5) is controllable.

3. CONTROL METHODOLOGY

The proposed structure of the nonlinear PID neural controller can be given in the form of block diagram, as shown in **Fig. 2**. The approach to control the wheeled mobile robot depends on the available information of the unknown nonlinear system and can be known by the input-output data and the control objectives. The genetic algorithm and particle swarm optimization will generate the optimal parameters for the nonlinear PID neural controller in order to obtain best velocity control signal that will minimize the tracking error of the mobile robot in the presence of external disturbance.

The feedback PID neural controller is very important because it is necessary to stabilize the tracking error of the system when the output of the mobile robot is drifted from the desired point. The nonlinear PID neural controller for MIMO mobile robot system is shown in **Fig. 3**. The proposed nonlinear PID neural controller has the characteristics of control agility, strong

adaptability, good dynamic characteristic and robustness because it is based on a conventional PID controller that consists of three terms: proportional, integral and derivative where the standard form of a PID controller is given in the s-domain as Eq. (14), **Zhong, 2006** and **Omatu, 1995**.

$$Gc(s) = P + I + D = K_p + \frac{K_i}{s} + K_d s \tag{14}$$

where K_p, K_i and K_d are called proportional gain, integral gain and derivative gain, respectively.
The proposed nonlinear PID neural controller scheme is based on the discrete-time PID as Eq. (15).

$$u_{1,2}(k) = u_{1,2}(k-1) + Kp_\gamma[e_\gamma(k) - e_\gamma(k-1)] + Ki_\gamma e_\gamma(k) + Kd_\gamma[e_\gamma(k) - 2e_\gamma(k-1) + e_\gamma(k-2)] \tag{15}$$

where $\gamma = x, y, \theta.$
Therefore, the tuning PID input vector consists of $e_\gamma(k)$, $e_\gamma(k-1), e_\gamma(k-2)$ and $u_{1,2}(k-1)$, where $e_\gamma(k)$ and $u_{1,2}(k)$ denote the input error signals and the PID output signal respectively.

The proposed control law of the feedback right and left velocity (u_1 and u_2) respectively can be proposed as follows:

$$u_1(k) = u_1(k-1) + o_x + o_y \tag{16}$$

$$u_2(k) = u_2(k-1) + o_\theta + o_y \tag{17}$$

o_x, o_y and o_θ are the outputs of the neural networks that can be obtained from non-linear Polywog wavelet activation functions, **Righeto, 2004,** as shown in **Fig. 4** and has nonlinear relationship as presented in the following function:

$$o_\gamma = (3(net_\gamma)^2 - (net_\gamma)^4)e^{-0.5(net_\gamma)^2} \tag{18}$$

net_γ is calculated from this equation:

$$net_\gamma(k) = Kp_\gamma[e_\gamma(k) - e_\gamma(k-1)] + Ki_\gamma e_\gamma(k) + Kd_\gamma[e_\gamma(k) - 2e_\gamma(k-1) + e_\gamma(k-2)] \tag{19}$$

The control parameters Kp_γ, Ki_γ and Kd_γ of the nonlinear PID neural controller are adjusted using genetic algorithm and particle swarm optimization.

3.1 Learning Genetic Algorithm (GA)

Genetic algorithm is an intelligent optimization technique that relies on the parallelism found in nature; in particular its searching procedures are based on the mechanics of natural selection and genetics, **Ali, et al, 2011**.
GAs includes three major operators: selection, crossover, and mutation, in addition to four control parameters: population size, selection pressure, mutation rate and crossover. Genetic algorithms start parallel searching from independent points of searching space in which the

solution knowledge is poor or not available. The solution depends on interaction of the surroundings and genetic operators. For that reason, obtaining the suboptimal solutions of genetic algorithm is a small probability, **Muhammet, 2010.**

In this paper, the first learning algorithm for determining the nonlinear PID neural controller parameters is genetic algorithm and the steps of the tuning these parameters can be described as follows:

- **Step1** GA makes the problem parameters into chromosomes then simulates evolutionary operation.
- **Step2** To determine the size of the population and to initialize. Set the population size 30, the evolution generation 100.
- **Step3** Decode the individual of the population into optimal parameters for calculating the value of fitness function.
- **Step4** Do population selection, crossover and mutation operations to produce the next generation population. The crossover probability is 0.75, mutation probability is 0.01.
- **Step5** Repeat steps 3 and 4 until the fitness function condition is achieved or the maximum number of generations is reached.

Mean square error (MSE) function for multi-input multi-output (MIMO) mobile robot system is chosen as a criterion for estimating the model performance and an objective function to be minimized with the GA as Eq. (20):

$$MSE = \frac{1}{pop}\sum_{j=1}^{pop}((x_r(k+1)^j - x(k+1)^j)^2 + (y_r(k+1)^j - y(k+1)^j)^2 + (\theta_r(k+1)^j - \theta(k+1)^j)^2) \quad (20)$$

Since the GA maximizes its fitness function, it is necessary therefore to map the objective function (MSE) to a fitness function. It is used objective –to- fitness transformation is of the form, **Al-Araji, 2005**.

$$fitness = \frac{1}{objectivefunction + \mu} \quad (21)$$

where μ is a constant chosen to avoid division by zero.

3-2 Learning Particle Swarm Optimization Algorithm

Particle Swarm optimization (PSO) is a kind of algorithm to search for the best solution by simulating the movement and flocking of birds. PSO algorithms use a population of individual (called particles) "flies" over the solution space in search for the optimal solution.

Each particle has its own position and velocity to move around the search space. The particles are evaluated using a fitness function to see how close they are to the optimal solution, **Derrac, et al, 2005**. The previous best value is called as *pbest*. Thus, *pbest* is related only to a particular particle. It also has another value called *gbest*, which is the best value of all the particles *pbest* in the swarm.

The nonlinear PID neural controller with nine weights parameters and the matrix is rewritten as an array to form a particle. Particles are then initialized randomly and updated afterwards according to Eqs. (22, 23, 24, 25, 26 and 27) in order to tune the PID parameters:

$$\Delta Kp_{\gamma,m}^{k+1} = \Omega(\Delta Kp_{\gamma,m}^{k}) + c_1 r_1 (pbest_{\gamma,m}^{k} - Kp_{\gamma,m}^{k}) + c_2 r_2 (gbest^{k} - Kp_{\gamma,m}^{k}) \quad (22)$$

$$Kp_{\gamma,m}^{k+1} = Kp_{\gamma,m}^{k} + \Delta Kp_{\gamma,m}^{k+1} \quad (23)$$

$$\Delta Ki_{\gamma,m}^{k+1} = \Omega(\Delta Ki_{\gamma,m}^{k}) + c_1 r_1 (pbest_{\gamma,m}^{k} - Ki_{\gamma,m}^{k}) + c_2 r_2 (gbest^{k} - Ki_{\gamma,m}^{k}) \quad (24)$$

$$Ki_{\gamma,m}^{k+1} = Ki_{\gamma,m}^{k} + \Delta Ki_{\gamma,m}^{k+1} \quad (25)$$

$$\Delta Kd_{\gamma,m}^{k+1} = \Omega(\Delta Kd_{\gamma,m}^{k}) + c_1 r_1 (pbest_{\gamma,m}^{k} - Kd_{\gamma,m}^{k}) + c_2 r_2 (gbest^{k} - Kd_{\gamma,m}^{k}) \quad (26)$$

$$Kd_{\gamma,m}^{k+1} = Kd_{\gamma,m}^{k} + \Delta Kd_{\gamma,m}^{k+1} \quad (27)$$

$m = 1,2,3,.....pop$

where

pop is number of particles,

$K_{\gamma,m}^{k}$ is the weight of particle *m* at *k* iteration,

c_1 and c_2 are the acceleration constants with positive values equal to 1.47,

r_1 and r_2 are random numbers between 0 and 1,

$pbest_{\gamma,m}$ is best previous weight of m^{th} particle.

$gbest$ is best particle among all the particle in the population.

Ω is the inertia weight factor and it is equal to 0.75,

The number of dimension in particle swarm optimization is equal to nine because the proposed nonlinear PID controller has nine parameters. The mean square error function is chosen as a criterion for estimating the model performance as given in Eq. (20).

In this paper, the second learning algorithm for determining the nonlinear PID neural controller parameters is PSO algorithm and the steps of the tuning these parameters can be described as follows:

- **Step1** Initial searching points $Kp_{\gamma}^{0}, Ki_{\gamma}^{0}, Kd_{\gamma}^{0}, \Delta Kp_{\gamma}^{0}, \Delta Ki_{\gamma}^{0}$ and ΔKd_{γ}^{0} of each particle are usually generated randomly within the allowable range. Note that the dimension of search space consists of all the parameters used in the nonlinear PID neural controller as shown in **Fig. 2**. The current searching point is set to *pbest* for each particle. The best-evaluated value of *pbest* is set to *gbest* and the particle number with the best value is stored.
- **Step2** The objective function value is calculated for each particle by using Eq. (20). If the value is better than the current *pbest* of the particle, the *pbest* value is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, *gbest* is replaced by the best value and the particle number with the best value is stored.
- **Step3** The current searching point of each particle is update by using Eqs. (22, 23, 24, 25, 26 and 27).
- **Step4** If the current iteration number reaches the predetermined maximum iteration number, then exit. Otherwise, return to step 2.

The fundamental essences for applying the proposed control algorithms are minimizing tracking error and obtaining smoothness of the velocity control signal through speed up the process of getting the optimal parameters of the nonlinear PID controller.

4. SIMULATION RESULTS

The kinematic model of the differential wheeled mobile robot described in section 2 is used and the proposed controller is verified by means of computer simulation using MATLAB package. The simulation is carried out by tracking a desired position (x, y) and orientation angle (θ) with continuous trajectory in the tracking control of the NI mobile robot. The parameter values of the NI robot model are taken from, **Al-Shibaany, 2012**: L=0.36 m, r=0.05 m and sampling time is equal to 0.5 second.

The proposed nonlinear PID neural controller scheme as in **Fig. 2** is applied to the mobile robot model and it uses the proposed learning algorithms steps of genetic algorithm and particle swarm optimization for tuning the nonlinear PID controller's parameters. The fist stage of operation is to set the following parameters of the GA and PSO:

Population size is equal to 30 and number of iteration is equal to 100 for GA.

Population of particle is equal to 30 and number of iteration is equal to 100 for PSO.

Number of weight in each chromosome and particle is 9 because there are nine parameters of the nonlinear PID controller.

The desired path which has explicitly continuous gradient with rotation radius changes can be described by the following equations:

$$x_r(t) = -0.5 \times \sin(\frac{2\pi t}{30}) \tag{28}$$

$$y_r(t) = 0.5 \times \sin(\frac{2\pi t}{20}) \tag{29}$$

$$\theta_r(t) = 2\tan^{-1}(\frac{\Delta y_r(t)}{\sqrt{(\Delta x_r(t))^2 - (\Delta y_r(t))^2} + \Delta x_r(t)}) \tag{30}$$

For simulation purposes, the desired path is chosen as described in Eq. (28) and Eq. (29) while the desired orientation angle is taken as expressed in Eq. (30).

The mobile robot model starts from the initial posture $q(0) = [-0.1,0,0]$ as its initial conditions. A disturbance term $\overline{\pi d} = [0.01\sin(2t) \quad 0.01\sin(2t)]^T$ **Al-Araji, et al, 2013**, is added to the mobile robot system as unmodelled kinematics disturbances in order to prove the adaptation and robustness ability of the proposed controller. The mobile robot trajectory tracking obtained by the proposed nonlinear PID neural controller is shown in **Fig. 5**.

The adaptive learning and robustness of nonlinear PID neural controller based on PSO show excellent position and orientation tracking performance and small effect of the disturbances than learning GA because the PSO algorithm has capability to obtain smooth values of the nonlinear PID controller's parameters with smoothness convergence behaviour in the parameters values that depend on previous values without using mutation and crossover processes for GA technique.

Table 1. shows the tuning values of the nonlinear PID neural controller parameters (k_x, k_y, k_θ) which are based on genetic and particle swarm optimization learning algorithms.

The effectiveness and robustness of the proposed nonlinear PID neural control algorithm based on PSO is clear by showing the convergence of the pose trajectory and orientation errors for the robot model motion, as shown in **Fig. 6**.

MSE of the pose trajectory and orientation errors for the robot model motion at 100 iterations is shown in **Fig. 7**.

The simulation results demonstrated the effectiveness of the proposed controller based on PSO technique by showing its ability to generate small smooth values of the control input velocities for right and left wheels without sharp spikes.

The actions described in **Fig. 8** show that smaller power is required to drive the DC motors of the mobile robot model.

The mean linear velocity of the NI mobile robot is equal to 0.0897 m/sec, and the maximum peak of the angular velocity is equal to ± 0.42 rad/sec. Both are shown in **Fig. 9**.

In order to validate the applicability of the proposed nonlinear neural PID control methodology, experiments have executed by using mobile robot from National Instrument Company type differential wheeled mobile robot shown in **Fig. 10**. The wheeled mobile robot is equipped with Lab VIEW package guided.

In the experiments, PSO technique control methodology of the simulation has been applied on real NI mobile robot because the controller was robust and effective in terms of minimum tracking error and in generating best velocity control action despite of the presence of bounded external disturbances in comparison with the GA simulation results. Therefore, these control data has transmitted to the NI mobile robot model, which admits right wheel velocity and left wheel velocity as input reference signals by using wireless communication after has been converted the data format from MATLAB file of simulations to LabVIEW package version 2010 format as a lookup table in the NI mobile robot.

Velocities commands sent by the computer are coded messages which are recognized by microcontroller. Based on received characters, the microcontroller creates control actions for servo motors. The output voltages of the two encoder sensors are converted to coded messages by microcontroller and sent to the personal computer in order to calculate the tracking error of the mobile robot during motion. Wireless communication technique has been used for transmitting the data between the NI mobile robot and main computer.

The velocities of the simulation results for right and left wheels have been downloaded to the memory of the NI mobile robot as commands which have smooth values without sharp spikes, as shown in **Fig. 8**.

The initial pose for the NI mobile robot starts at position (-0.1 and 0) meter and orientation 0 radian and should follow desired continuous trajectory, as show in **Fig. 11**. The desired trajectory starts at position (0, 0, 0).

After 61 seconds, the real mobile robot has followed and finished the tracking of the desired path, as shown in **Fig. 12a** with small drifted from the desired trajectory and the distance of the trajectory did not exceed 5.47m. **Fig. 12b** shows the actual orientation of the mobile robot with small error. As shown in **Fig. 12c** the tracking error in x-coordinate and y-coordinate were reasonably accurate because the mobile robot was trying to correct the pose and orientation errors.

The mean-square error for each component of the state error vector $[ex, ey, e\theta]$ for simulation results and experimental work for two control algorithms are calculated, as shown in **Table 2**.

Table 2. shows the effectiveness and robustness of proposed controller based on PSO in terms of minimum tracking error and excellent position and orientation tracking performance than genetic learning algorithm because the PSO learning algorithm has capability to obtain best values of the parameters of the proposed controller with smoothness convergence behaviour in these parameters that depend on previous values without using mutation and crossover processes for GA technique.

The percentage of the mean square error between simulation results and experimental work can be shown in **Table 3**.

Table 3. shows the difference between simulation results and experimental work which caused by the residual errors in the experimental results due to the inherent friction present in the real system, especially during tracking the continuous gradient path and modelling errors, due to the difficulty of estimating or measuring the geometric, kinematics or inertial parameters, or from incomplete knowledge of the system components.

5. CONCLUSIONS

The nonlinear PID neural trajectory tracking control methodology with two learning algorithms for the differential drive mobile robot model have been presented in this paper. It has been designed and tested using Matlab package and carried out on real NI mobile robot using LabVIEW package.

The simulation results and experimental work show evidently the capability of adaptation and robustness of the proposed nonlinear PID neural controller that based on PSO learning algorithm which has excellent position tracking performance and it has the capability of generating smooth and suitable velocity commands than the controller based on genetic learning algorithm because the PSO algorithm has capability to obtain best values of the nonlinear PID controller's parameters as well as smooth parameters convergences behaviour which depends on previous values and this with few parameters adjustment unlike GA which depends on more evolutionary operators such as mutation and crossover.

REFERENCES

Al-Araji, A., Abbod, M. and Al-Raweshidy, H., 2013, *Applying Posture Identifier in Designing an Adaptive Nonlinear Predictive Controller for Nonholonomic Mobile Robot*. Neurocomputing, Vol. 99, pp. 543-554.

Al-Araji, A., 2005, *Genetic Algorithm and Elman Neural Network Used for Tuning the Parameters of the PID Neural Controller Based Model Reference.* Iraqi journal of computers, communication, control and systems engineering by University of Technology Vol. 5, No.1, pp. 111-126.

Ali, O. M., Koh, S. P., Chong, K. H., Hamoodi, A. S., 2011, *Design a PID Controller of BLDC Motor by Using Hybrid Genetic-Immune*, Modern Applied Science .Vol. 5, No. 1, pp. 56-63.

Al-Shibaany, Z., Hedley, J. and Bicker, R., 2012, *Design of an Adaptive Neural Kinematic Controller for a National Instrument Mobile Robot System*, IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, pp. 623-628.

Chang, Y-C, Lwin, Y.Y. and Yamamoto, Y., 2009, *Sensor-Based Trajectory Planning Strategy for Nonholonomic Mobile Robot with Laser Range Sensors*, IEEE International Symposium on Industrial Electronics, Seoul Olympic Parktel, Seoul, Korea. pp. 1755-1760.

Derrac, J., Garc, S., Molina, D. and Herrera, F., 2011, *A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms*, Journal of Swarm and Evolutionary Computation, Vol.1, pp. 3–18.

Han, S., Choi, B. and Lee, J., 2008, *A Precise Curved Motion Planning for a Differential Driving Mobile Robot*. Mechatronics, Vol. 18, pp. 486-494.

Klancar, G. and Skrjanc, I., 2007, *Tracking Error Model-Based Predictive Control for Mobile Robots in Real Time*, Robotics and Autonomous Systems, Vol. 55, pp. 460-469.

Mnif, F. and Touati, F., 2005, *An Adaptive Control Scheme for Nonholonomic Mobile Robot with Parametric Uncertainty*, International Journal of Advanced Robotic Systems, Vol. 2, No. 1, pp. 59-63.

Muhammet, U., 2010, *Trajectory Tracking Performance Comparison Between Genetic Algorithm and Ant Colony Optimization for PID Controller Tuning on Pressure Process*, Wiley Periodicals Inc.

Omatu, S., Khalid, M. and Yusof, R., 1995, *Neuro-Control and its Applications*. London: Springer-Velag.

Park, B. S., Yoo, S. J., Park, J. B. and Choi, Y. H., 2010, *A Simple Adaptive Control Approach for Trajectory Tracking of Electrically Driven Nonholonomic Mobile Robots*, IEEE Transactions on Control Systems Technology, Vol. 18, No. 5, pp. 1199-1206.

Righeto, E., Grassi, L-H M. and Pereira, J. A., 2004, *Nonlinear Plant Identification by Wavelet*, Proceeding of ABCM Symposium Series in Mechatronics, Vol. 1, pp. 392-398.

Sahin, T. and Zergeroglu, E., 2007, *A Computationally Efficient Path Planner for a Collection of Wheeled Mobile Robots with Limited Sensing Zones*, IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 1074-1079.

Su, K-H., Chen, Y-Y. and Su, S-F., 2010, *Design of Neural-Fuzzy-Based Controller for Two Autonomously Driven Wheeled Robot*, Neurocompting. Vol. 73, pp. 2478-2488.

Takanori, F., Kiroshi, N. and Norihiko, A., 2000, *Adaptive Tracking Control of a Nonholonomic Mobile Robot*, IEEE Trans Robotic Autonomous, Vol. 16, No. 2, pp. 609-615.

Wai, R-J and Liu, C-M, 2009, *Design of Dynamic Petri Recurrent Fuzzy Neural Network and its Application to Path-Tracking Control of Nonholonomic Mobile Robot*, IEEE Transactions on Industrial Electronics. Vol. 56, No. 7, pp. 2667-2683.

Yang, S. X., Yang, H., Max, Q. and Meng, H., 2004, *Neural Dynamics Based Full-State Tracking Control of a Mobile Robot*, IEEE International Conference on Robotics & Automation, New Orleans, LA, pp. 4614-4619.

Zhong, Q., 2006, *Robust Control of Time-delay Systems*, Springer – Verlag London Limited.

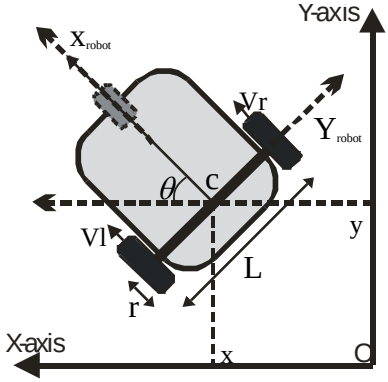
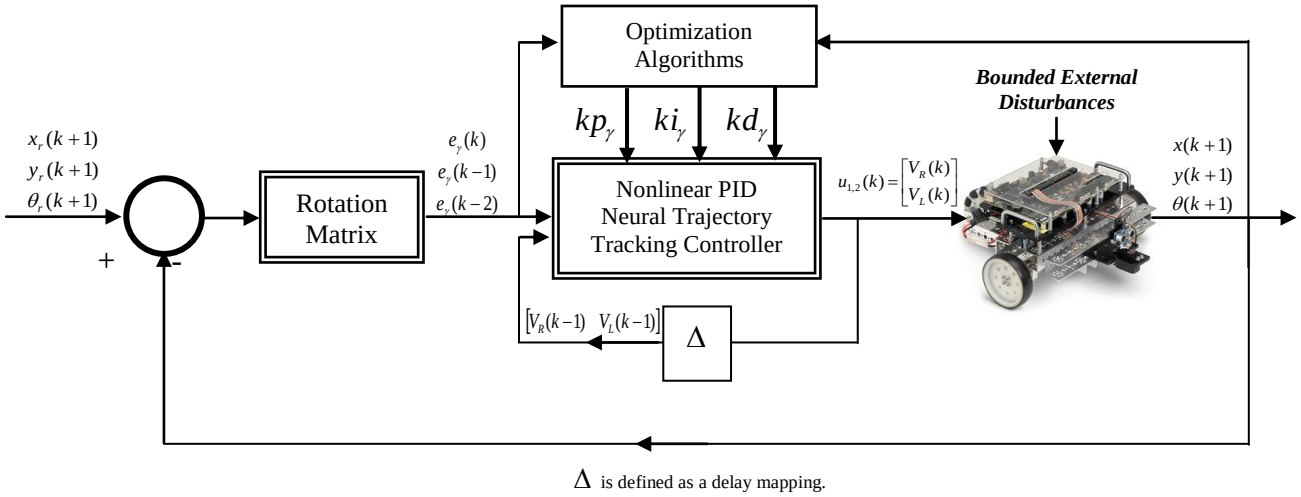


Figure 1. Mobile robot platform.



Δ is defined as a delay mapping.

Figure 2. Nonlinear PID Neural trajectory tracking controller structure for mobile robot.

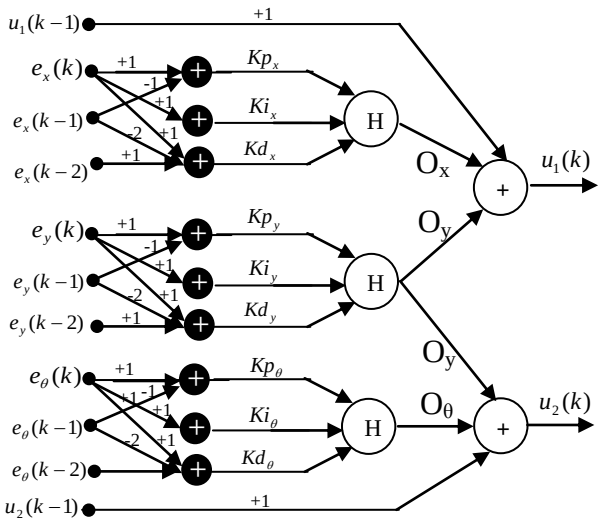


Figure 3. The nonlinear PID neural feedback controller structure.

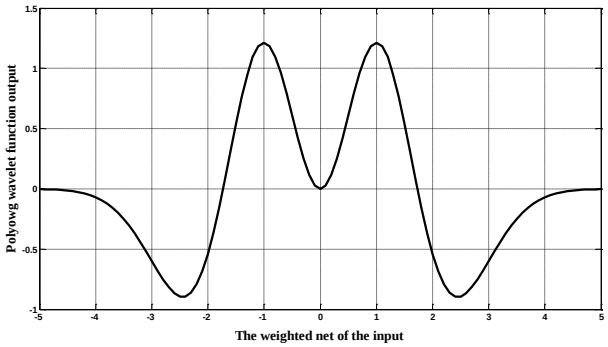


Figure 4. Polywog wavelet function.

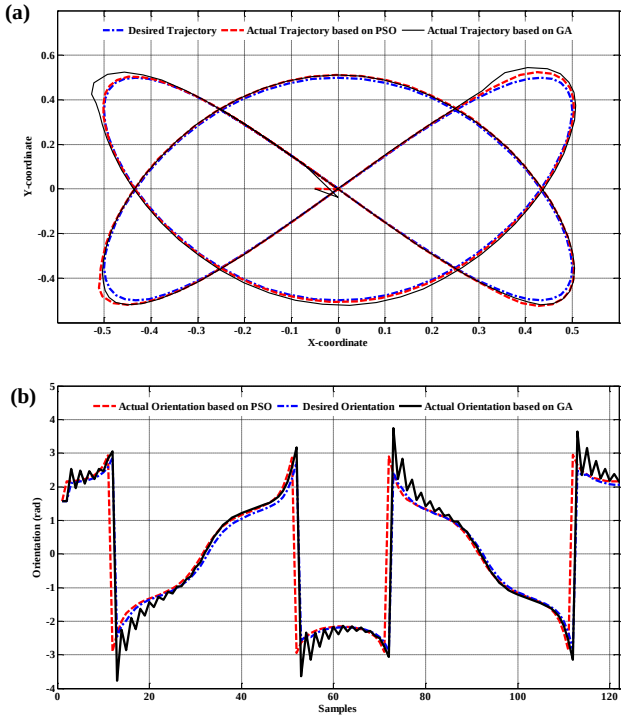


Figure 5. Simulation results (a) desired trajectory and actual mobile robot trajectory; (b) desired orientation and actual mobile robot orientation.

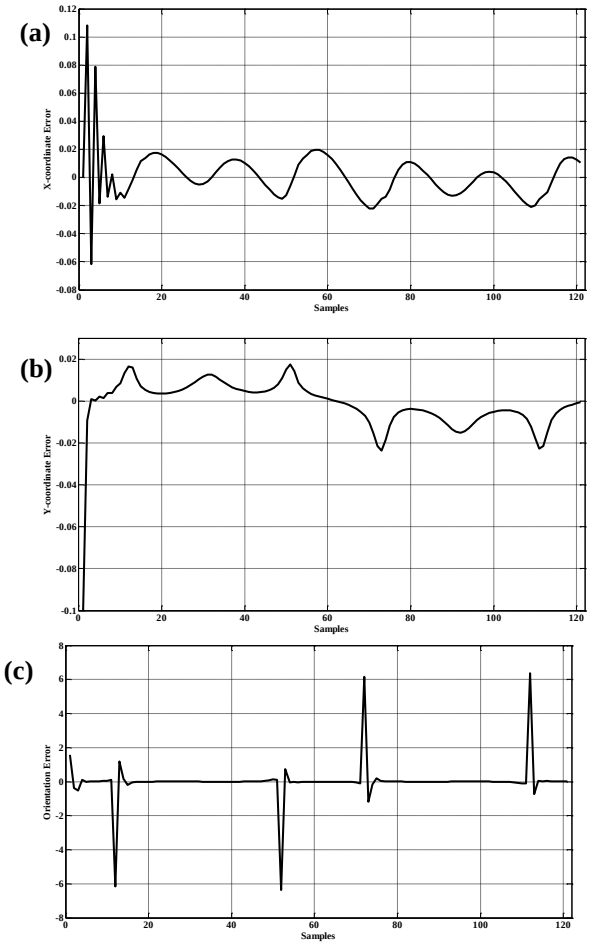


Figure 6. Position tracking error (a) in X- coordinate; (b) in Y-coordinate; (c) Orientation tracking error.

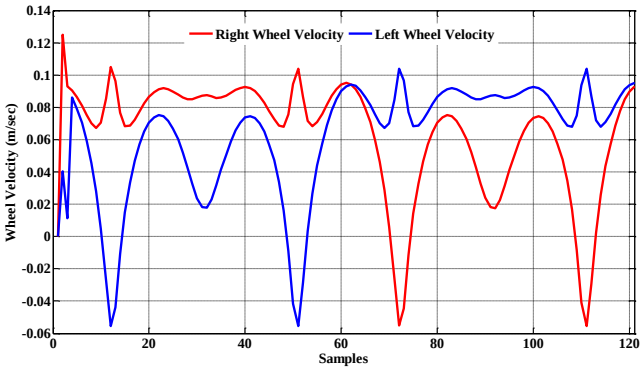


Figure 7. The right and left wheel action velocity.

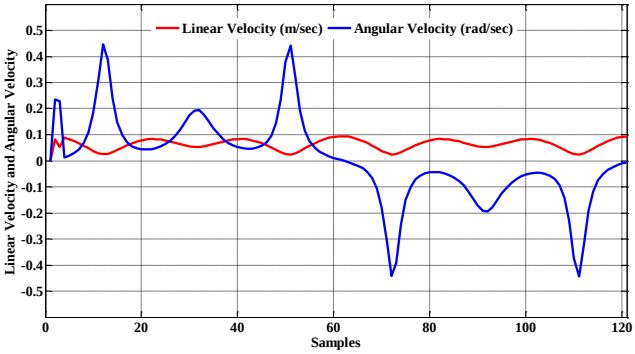


Figure 8. The linear and angular velocity.

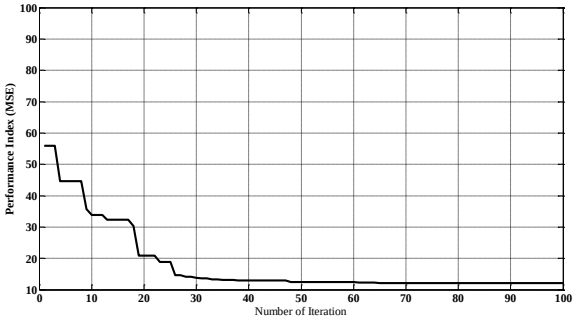


Figure 9. The performance index (MSE).

Figure10. NI mobile robot for the experiments.

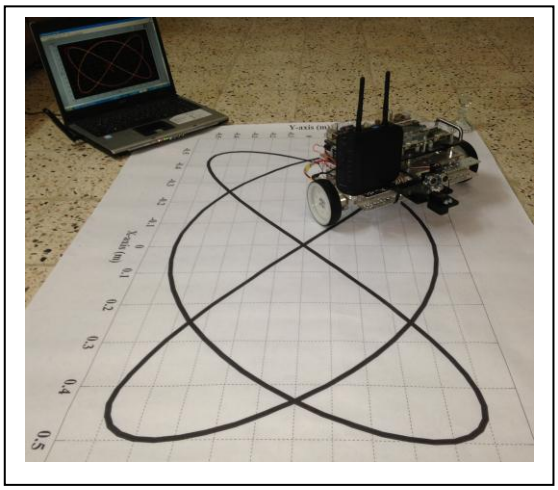


Figure 11. Real set-up experiment of NI mobile robot for continuous

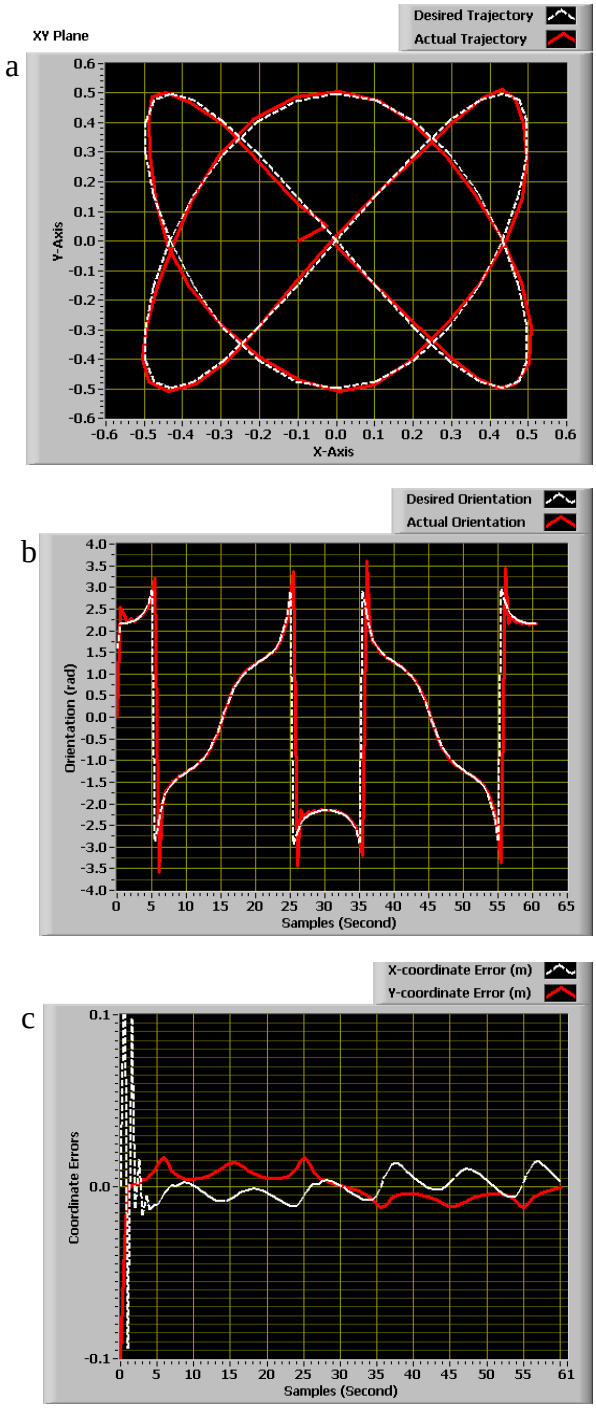


Figure 12. Practical results (a) desired trajectory and actual mobile robot trajectory; (b) desired orientation and actual mobile robot orientation; (c) X-coordinate and Y-coordinate errors.

Table 1. The nonlinear PID neural controller parameters.

Tech	kp_x	ki_x	kd_x	kp_y	ki_y	kd_y	kp_θ	ki_θ	kd_θ
PSO	0.75	0.72	0.131	-0.307	-0.36	1.039	-0.08	2.78	0.014
GA	0.007	0.42	0.460	0.316	0.407	0.554	0.242	0.454	0.5261

Table 2. The MSE for simulation results and experimental work.

PID Control Algorithm	GA			PSO		
MSE	(ex)	(ey)	(eθ)	(ex)	(ey)	(eθ)
Simulation Results	0.0021	0.0059	1.46	0.0013	0.0038	1.18
Experimental work	0.0032	0.0074	1.77	0.0019	0.0047	1.23

Table 3. The percentage of MSE between simulation results and experimental work.

PID Control Methodology	GA	PSO
(MSE of X-coordinate) 100%	34.3%	31.5%
(MSE of Y-coordinate) 100%	20.3%	19.1%
(MSE of Orientation) 100%	17.5%	4%