https://doi.org/10.33261/jaaru.2019.26.1.008 *Association of Arab Universities Journal of Engineering Sciences (2019) 26(1): 55–66*

Association of Arab Universities Journal of Engineering Sciences
مجلة اتحاد الجامعات العربية للدراسات والبحوث الهندسية

Motion Control of Non-Holonomic Wheeled Mobile Robot Based on Particle Swarm Optimization Method (PSO)

Mauwafak A. Tawfik

Department of Mechanical Engineering, University of Technology, Baghdad, Iraq, 20040@uotechnology.edu.iq

Published online: 31 March 2019

Abstract— Using (PID) controller to control the trajectory motion of non-holonomic wheeled mobile robot may not be efficient especially for non-linear systems. Hence this work introduces a combination of back stepping method with the (PID) controller to obtain an efficient controller for (WMR) to deal with the non-linear systems. Different common trajectories such as infinity, circle and straight line were applied to be tracked by (WMR) to examine the control system. The results of the simulation tests of the designated trajectories with the desired trajectories were achieved through the implementation of the mean square error for x, y and the orientation. Practical swam optimization method was used to find the control gain to investigate an optimized minimum error percentage. The results of simulation show a good tracking performance with the desired trajectories.

Keywords— PID controller, Back-stepping technique, mobile robot, Intelligent optimization**.**

1. Introduction

Many investigators have paid their attention to study the problems of controlling the non- holonomic system and as a result of that a number of control algorithms have been achieved to sort out the path-tracking control problems as neural network controller [1], the kinematic back-stepping method controller [2],[3]. Lyapunov criterion is implemented to prove the stability of the proposed control law since this criterion is active in specifying the cited stability [4]. One of the practical studied cases with great importance is the kinematic model of the wheeled mobile robot like a unicycle and the differentially mobile robots [5]. More difficult problems of the dynamic model stability for different types of mobile robots have been studied by the researchers [6], [7]; kinematic model plays a great part in specifying the limitations of controlling the mobile robot as presented in [8],[9],[10]. The velocity command is often transmitted through high level hardware which in turn provides the current control objective. It is well known that the problem of controlling the mobile robot has been tackled by point stabilization or by tracking control [11], [12]. Both cited problems are also studied through some approaches simultaneously [13]. The tracking control approach is considered somehow more suitable because of the non-holonomic limitations and other targets were sorted out by the path planning procedure [14], [15], [16]. This approach can be extended easily to sophisticated case such as controlling the mobile robot platoon [17]. Many control algorithm proposals were adopted through studying the path-tracking framework, such as PID [18], Lyapunov nonlinear controllers [19], adaptive and model-based predictive controllers, fuzzy neural network and fuzzy [20], [21], [22], [23], [24]. Fuzzy controllers are suitable to be used for high level control and it may also be used sometimes for chips or other industrial hardware. It is of importance to obtain a (kinematic) control law that is able to give an efficient control signal. Otherwise the dynamic model would not give the required controlling process. A discontinuity in orientation error may lead to a discontinuity in the angular-velocity control orders since the classical kinematic model does not taken in to account the mapped interval at (-π,π). This problem is sorted out in the present work in spite of its difficulty. Also, a proposed control law is adopted to achieve the globule asymptotic convergence to predesign the path under mild conditions. This law is compared with other common control laws.

2. Modeling of the Non-Holonomic Wheeled Mobile Robot

A mobile robot system having an n-dimensional configuration space (ρ) with generalized coordinates (q1. . . qn) and subjected to m constraints is described by [26]:

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (1)$$

where $M(q) \in R^{n\times n}$ is a symmetric positive definite inertia matrix $,V_m(q,\dot{q}) \in R^{n\times n}$ is the centeriptal and corilis matrix, $F(\dot{q}) \in R^{n\times 1}$ denotes the surface friction , $G(q) \in R^{n\times 1}$ is the gravitational vector, τ_d denotes bounded unknown disturbances including unstructured *unmolded* dynamics, $B(q) \in R^{n\times r}$ is the input transformation matrix, $\tau \in R^{n\times 1}$ is the input vector, $A(q) \in R^{m\times n}$ is the matrix associated with the constraints, and $\lambda \in R^{m\times 1}$ is the vector of constraint forces.
In this work, all kinematic equality constrains are considered to be time independence. Hence [2];

$$A(q)\dot{q} = 0 \tag{2}$$

Let S(q) be a full rank matrix (*n* x *m*) which is able to spanning the null space of A(q), then:

$$S^T(q)A^T(q) = 0 \tag{3}$$

Per (2) and (3), it is possible to find an auxiliary vector time function $v(t) \in R^{n-m}$ such that, for all *t*

$$\dot{q} = S(q)V(t) \tag{4}$$

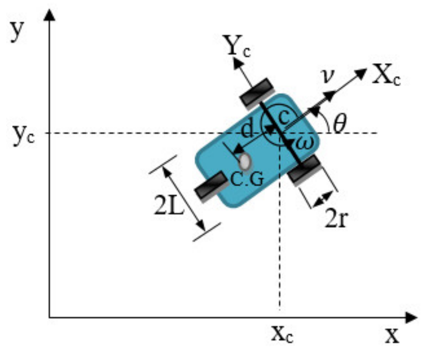

Figure 1: A non-holonomic mobile platform

Fig. 1 shows a common mobile robot of a non-holonomic mechanical system. It is an accomplished of a vehicle with two mounted wheels on the same axis and another front free wheel. Independent actuators (D.C motors) are implemented to achieve the torques of the required motion and orientation. The position of the robot is specified by a Cartesian frame {O, X, Y} which is by the vector $q = [x_c\ y_c\ \theta]^T$ where x_c,y_c are the coordinates of vehicle mass center and θ is the orientation with respect to the inertial basis.

The constraint of non-holonomic is considered that the mobile base is always under pure rolling condition. i.e. no slipping. Hence the robot only moves in the direction normal to the axis of the driving wheels. It is of importance that a frame reference x_c & y_c are implemented in order to specify momently the new position of (WMR).

The motion of "C" in terms of its linear and angular velocity can be specified as:

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

and

$$V = [\nu\ \ \omega]^T \tag{6}$$

The equations of motion for the MR may be found by using Lagrange formula. In this case G(q) = 0, since the base mobile trajectory is confined to move in a horizontal plane. The potential energy remains constant because there is no change in the system vertical position. The kinetic energy K_E is given by:

$$k_E^i = \frac{1}{2}m_i v_i v_i + \frac{1}{2}w_i^T I_i w_i, K_E = \sum_{i=1}^{n_i} k_E^i = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{7}$$

The mobile base dynamical equations [parameters of equation 1] can be written as [26]:

$$M(q) = \begin{bmatrix} m & 0 & mdsin\theta \\ 0 & m & -mdcos\theta \\ mdsin\theta & -mdcos\theta & I \end{bmatrix} \tag{8}$$

$$B(q) = \frac{1}{r}\begin{bmatrix} cos\theta & cos\theta \\ sin\theta & sin\theta \\ L & -L \end{bmatrix} \tag{9}$$

$$V_m(q,\dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\theta}cos\theta \\ 0 & 0 & md\dot{\theta}sin\theta \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$A^T(q) = \begin{bmatrix} sin\theta \\ -cos\theta \\ 0 \end{bmatrix} \tag{11}$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{12}$$

G(q) = 0,

$$I = I_c + md^2 \tag{13}$$

2.1 *Structural Properties of a Mobile Platform*

For the sake of simplicity and control regards, the system is specified properly as follows;

Multiplying equation (1) by S^T and substituting for $\ddot{q}$ from the differentiation of equation (4), $\mathbf{A}^{T}(\mathbf{q})$ can be eliminated and the equation yields:

$$S^TMS\dot{V} + S^T(M\dot{S} + V_mS)V + S^T\tau_d = S^TB\tau \tag{14}$$

3. Control Algorithm

In order to achieve and control the robot task reference position, velocity, sensory information and actuator commands, are specified and designed. For mobile robot, the controller design problem can be specified when the reference position $q_r(t)$,velocity $\dot{q}_r$ (t) and a control law for the actuator torques, which drive the mobile robot are known. As a result of that the mobile robot velocity is tracking precise velocity control input and reference position.

The components of the velocity and position for the robot are stated as:

$$\dot{q_r} = [\dot{x_r}\dot{y_r}\dot{\theta_r}]^T \quad (15)$$

$$\dot{x_r} = \nu_r \cos\theta_r \quad (16)$$

$$\dot{y_r} = \nu_r \sin\theta_r \quad (17)$$

$$\dot{\theta_r} = \omega_r \quad (18)$$

where $\nu_r > 0$ (linear velocity) at any time t and ω_r can be estimated from the following equations [20]:

$$\nu_r = \sqrt{(\dot{x_r})^2 + (\dot{y_r})^2} \quad (19)$$

$$\omega_r = \frac{\ddot{y_r}\dot{x_r} - \ddot{x_r}\dot{y_r}}{(\dot{x_r})^2 + (\dot{y_r})^2} \quad (20)$$

Then the error between the reference and the actual tracking position in the robot local frame is specified as:

$$\mathbf{e}(t) = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} = T_e \mathbf{e_p}(t) \quad (21)$$

A smooth velocity input v_c can be selected and in this paper back stepping model from [14] is chosen.

$$V_c = f_c(e, \nu_r, K_1) = \begin{bmatrix} \nu_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 \nu_r e_2 + k_3 \nu_r \sin e_3 \end{bmatrix} = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \quad (22)$$

where, $K_1 = \{k_1, k_2, k_3\}$ are design parameters and assumed that: $k_1, k_2, k_3 > 0$.

The feedback control input of the assumed nonlinear acceleration is:

$$u_1 = \dot{V_c} - K_2(V - V_c) \quad (23)$$

where, K_2 is a positive definite diagonal matrix given by:

$$K_2 = [k_4 \;\; 0; 0 \;\; k_4] \quad (24)$$

After taken time derivative of equation (21), the configuration error for the mobile robot can be express as follows:

$$\dot{\mathbf{e}}(t) = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - \nu + \nu_r \text{cos} e_3 \\ -\omega e + \nu_r \text{sin} e_3 \\ \omega_r - \omega \end{bmatrix} \quad (25)$$

Let $\nu_c = \nu$ and $\omega_c = \omega$, then the Equation (25) can be rewritten as:

$$\dot{\mathbf{e}}(t) = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega_r e_2 + k_2 e_2{}^2 + k_3 \nu_r e_2 \text{sin} e_3 - k_1 e_1 \\ -\omega_r e_1 + k_2 \nu_r e_1 e_2 - k_3 \nu_r e_1 \text{sin} e_3 + \nu_r \text{sin} e_3 \\ \omega_r - \omega \end{bmatrix} \quad (26)$$

Now Lyapunov criterion is implemented to prove the proposed control law and to state the controller stability. This is as follows:

The error of an auxiliary velocity can be defined as follows:

$$e_c = V - V_c \quad (27)$$

Or

$$\mathbf{e}_c = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} = \begin{bmatrix} \nu - v_c \\ \omega - \omega_c \end{bmatrix} \quad (28)$$

Then, the inter error vector can be express as:

$$\mathbf{e}_t = [\mathbf{e}^T \mathbf{e}_c^T]^T \quad (29)$$

The time derivative of auxiliary error vector in Equation (30) is equal to:

$$\dot{\mathbf{e}}_c = -\mathbf{K}_4 \mathbf{e}_c \quad (30)$$

Now, considered the following Lyapunove function candidate:

$$V^* = \frac{k_1}{2}(e_1^2 + e_2^2 + e_4^2 + e_5^2) + \frac{1}{k_2}(1 - \cos e_3) \quad (31)$$

The time derivative of Equation (31) becomes:

$$\dot{V}^* = k_1(e_1\dot{e_1} + e_2\dot{e_2} + \dot{e_4}e_4 + \dot{e_5}e_5) + \frac{\dot{e_3}}{k_2}\sin e_3 \quad (32)$$

Then

$$\dot{V}^* = k_1(\omega_d e_2 + k_2 v_d e_2^2 + k_3 v_d \sin e_3 - k_1 e_1)e_1$$

$$+k_1(-\omega_d e_1 - k_2 v_d e_1 e_2 - k_3 v_d \sin e_3 + v_d \sin e_3)e_2$$

$$-k_1 k_4 e_4^2 - k_1 k_4 e_5^2 + \frac{\sin e_3}{k_2}(-k_2 v_d e_2 - k_3 v_d \text{sin} e_3) \quad (33)$$

$$\dot{V}^* = -k_1^2 e_1^2 - k_1 k_4 e_4^2 - k_1 k_4 e_5^2 - \frac{k_3}{k_2} v_d \sin^2 e_3 \quad (34)$$

From Equations (34) and (21), one can find;

$V^* \geq 0$, if $e_t = 0$ then $V^* = 0$ and $\dot{V}^* = 0$

if $e_t \neq 0$ then $V^* > 0$ *and* $\dot{V}^* < 0$

Then V^* becomes a Lyapunov function, therefor the equilibrium point $e_t = 0$ is asymptotically stable. The controller gains (k_1, k_2, k_3, and k_4) are determined by using optimization method (Particle Swarm Optimization (PSO)).

The second part of the proposed controller is PID controller (proportional, Derivative, and Integral). The standard form of the PID controller is given in time domain as in equation (35).

$$u(t) = (K_p e(t) + K_d e(t) + K_i e(t)) \quad (35)$$

where $e(t)$ is the input to the controller.

The controller parameters $K_p + K_d, and\ K_i$ are the proportional, derivative, and integral gains respectively. These gains had been tuned using optimization methods (PSO). The control input u_1 in equation (23) is used to determine the torque control (τ). In this work and to reduce the trajectory tracking error the control input u_1 is used as an input to PID controller and the output of the PID controller is used to determine the torque control (τ) for the wheeled mobile robot, as follows;

$$\text{Let } \bar{e} = u_1 = \dot{V}_c - K_2(V - V_c) \tag{36}$$

Define an auxiliary control input u_2 where:

$$\dot{V} = u_2 \tag{37}$$

Neglecting the disturbance torque ($\boldsymbol{\tau_r} = 0$) and subsisted equation (37) in equation (14), then the torque equation becomes:

$$\tau = [S^T B]^{-1} [S^T M S u_2 + S^T (M\dot{S} + V_m S)] \tag{38}$$

Choosing the control input $\mathbf{u_2}$ as the output of the PID controller, equation (38) becomes as follows;

$$\tau = [S^T B]^{-1} [S^T M S (K_p \bar{e} + K_d \bar{e} + K_i \bar{e}) + S^T (M\dot{S} + V_m S)] \tag{39}$$

Fig. 2 shows the schematic diagram of the proposed controller of the Simulink model (FOPID). The evolutionary algorithm is adopted to modify the PID parameter which is optimized by using optiy program.

K_p, K_d, K_i, a and b parameters must be considered (equation (36)). In order to reduce the optimization time, the variable vector is chosen with five dimensions. The ranges of FOPID parameters are specified.

4. Particle Swarm Optimization (PSO)

Particle swarm optimization is used to solve optimization problems which are considered as computationally hard. A robust optimization technique is implemented accurately to sort out many optimization problems. The adopted algorithm applies multiple of flying particles in specified space in order to optimize its global location. The particles communicate with each other by searching an optimize direction. Each particle is updating its location relaying on three aspects. These aspects are determining its velocity relying on its best previous velocity, location and neighborhood position. Fig. 3 shows the flowchart of PSO algorithm. The mean concept of PSO is to accelerate each particle in the best position direction (pbpd) and the obtained global best position (gbp) by any particle is accelerated by a random weight at each time step. Equations (40) and (41) demonstrate that:

$$v_{t+1} = w * v_t + c_1 * rand(0,1) * (pbpd - x_t) + c_2 * rand(0,1) * (gbp - x_t) \tag{40}$$

$$x_{t+1} = x_t + v_{t+1} \tag{41}$$

where:

gbp=Global Best Position.

$pbpd$= Self Best Position.

$c1$ and $c2$ =Acceleration Coefficients.

w = Inertial Weight (the value is taken unity).

Once the new xt is computed by the particle a new position is located. When fitness (xt) is better than fitness (pbpd) then pbpd= xt and fitness (pbpd) = fitness (xt). Finally, the iteration will converge to the fitness (*gbp*) = the better fitness (*pbpd*) [17, 25].

The PSO algorithm method is used as M file to be connected to the Simulink model in which PID controller parameters are calculated and fed to the GUI of the controller. The initial parameters of optimization are number of particles 50, number of dimensions 7, maximum iteration 50, C_1=1, C_2=1, with the objective function ITAE.

The initial values of the parameters K_p ,Ki, K_d, k_1, k_2, k_3, and k_4 of the proposed (back stepping –PID) controller will be generated in PSO program and submitted in simulation diagram **Fig. 2**. The simulation will be run automatically and computing the objective function ITAE which is fed to PSO program to parameters are calculated and fed to the GUI of the controller. The initial parameters of optimization are number of particles 50, number of dimensions 7, maximum iteration 50, C_1=1, C_2=1, with the objective function ITAE.

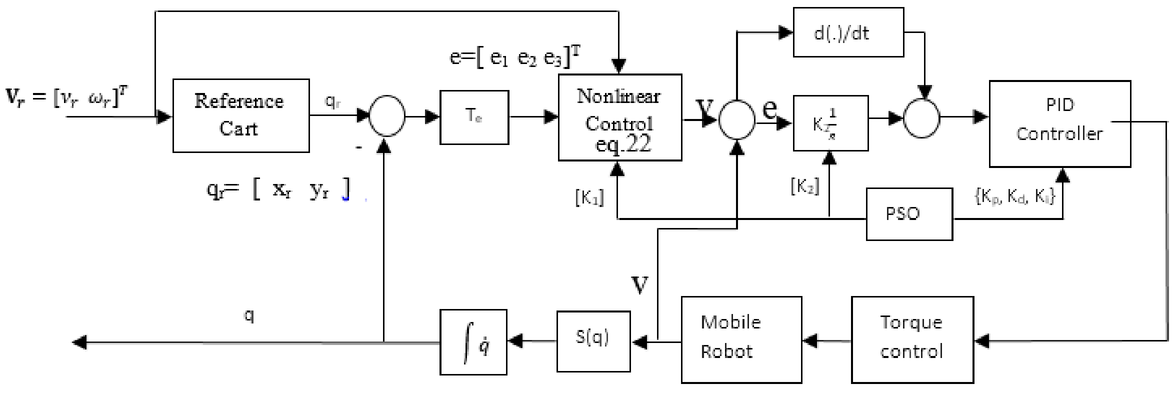

Figure 2: The proposed structure of back stepping – PID trajectory tracking controller.

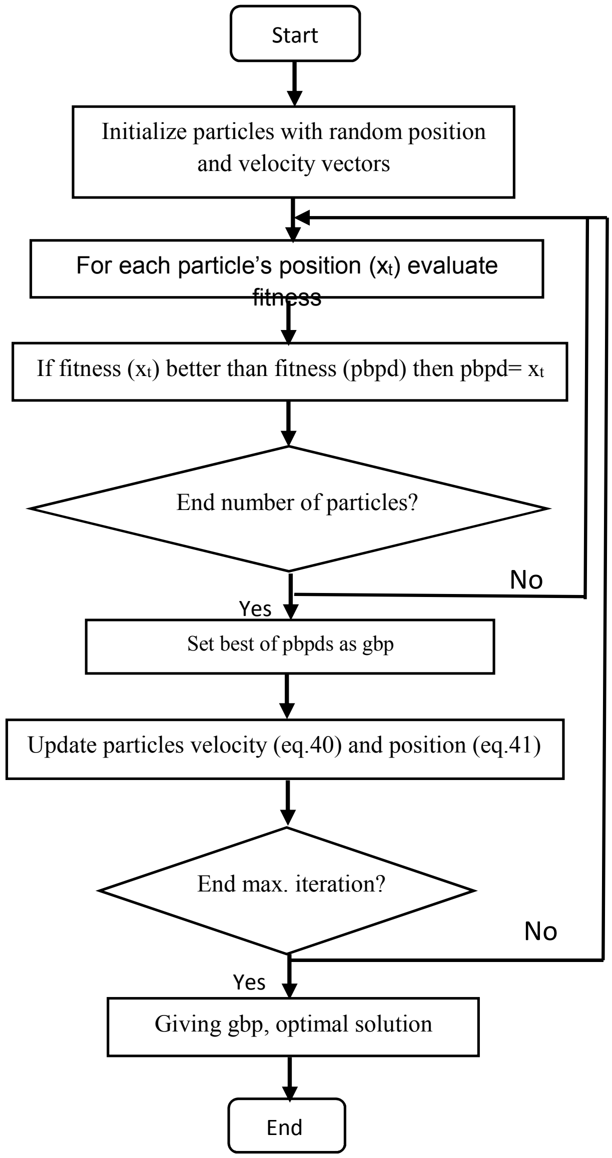

Figure 3: Flow chart of PSO algorithm

The initial values of the parameters Kp ,Ki, Kd, k1, k2, k3, and k4 of the proposed (back stepping –PID) controller will be generated in PSO program and submitted in simulation diagram Fig. 2. The simulation will be run automatically and computing the objective function ITAE which is fed to PSO program to improve its parameter value and so on.

At the end of iteration, the parameters Kp, Ki, Kd, k1, k2, k3, and k4 have been obtained directly according to the minimum value of objective function Mean Square Error (MSE). The obtained results are shown in Table 1, while Fig. 4 shows how MSE is changing with the number of iteration.

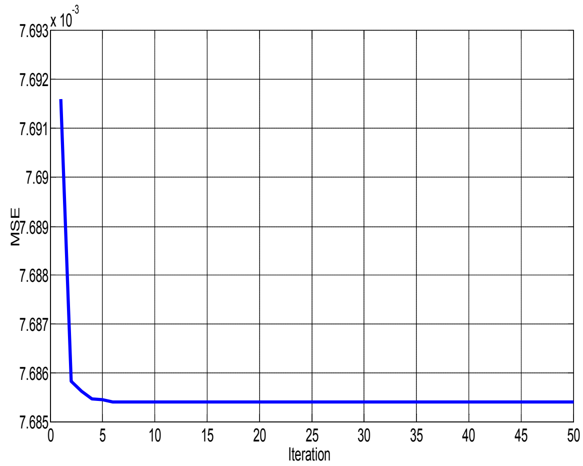

Figure 4: Mean square error versus number of iteration

Table 1: The parameters of Back stepping-PID controller obtained by PSO

parameters	K_p	Ki	K_d	k_1	k_2	k_3	k_4
PSO algorithm value	1.530	0.739	0.0081	1.317	17.003	16.587	8.286

5. Simulation Results

MATLAB/SIMULINk is implemented for the purpose of the designed controller verification. The kinematics and dynamic model of the non-holonomic MR described in Sections 2 and 3 are used. The simulation is achieved by tracking a local position $(\boldsymbol{x_r}, \boldsymbol{y_r})$ and orientation angle $(\boldsymbol{\theta_r})$ for three different types of trajectory. These trajectories were circular, infinity, and linear trajectory. To verify the stability and efficiency of the proposed controller each trajectory is performed with two different initial conditions. The parameter values of the robot model are taken from [19], [20] and presented in Table 2. The designed controller is used based on the structure shown in Fig. 2. The robot trajectory tracking obtained by the proposed Back stepping-PID controller is illustrated in Figs. (5 - 8). The sampling period was taken to be T0= 0.1s.

Table 2: Parameter values of robot model

m (Kg)	0.65
I (Kg.cm2)	0.36
L (m)	0.105
r (m)	0.033
d (m)	0.03

5.1 *Circular Trajectory*

The circular trajectory was performed by using: linear velocity $v_r = 0.1\ m/s$ and angular velocity $\omega_r = 0.1\ rad/s$. The desired circular trajectory can be described as follows:

$$\theta_r(t) = \frac{\pi}{2} + \frac{t}{10} \quad (42)$$

$$x_r(t) = 1 + \cos(\frac{t}{10}) \quad (43)$$

$$y_r(t) = \sin\left(\frac{t}{10}\right) \quad (44)$$

5.1.1 Case study -1

The desired trajectory has initial position of $q_r(0) = [2,0,\frac{\pi}{2}]^T$. However, the actual initial location of the robot is $q(0) = [\,2.5,0,\pi]^T$. The simulation of the circular trajectory tracking and the curves of the posture error are illustrated in Figs. (5 - 28) respectively. It is quite obvious that the proposed controller achieved its function properly good.

MSE=
Xr= (0.00194786921055269),
Yr= (0.000146676084870509),
Θ_r = (0.0240028846935643)

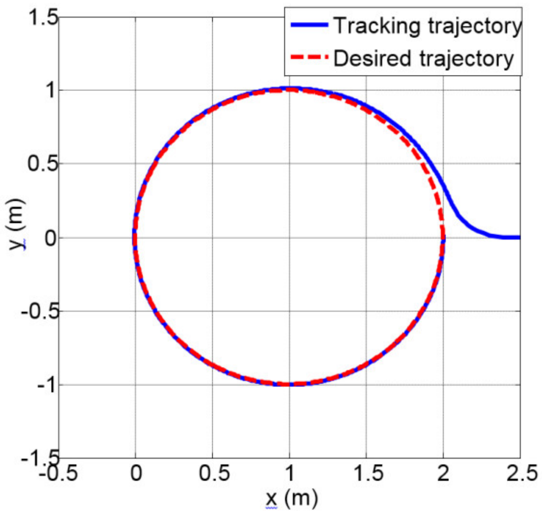

Figure 5: Circular trajectory tracking

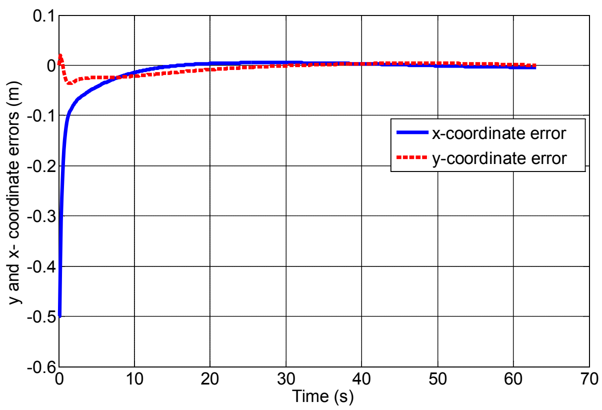

Figure 6: x-y Coordinate trajectory tracking error

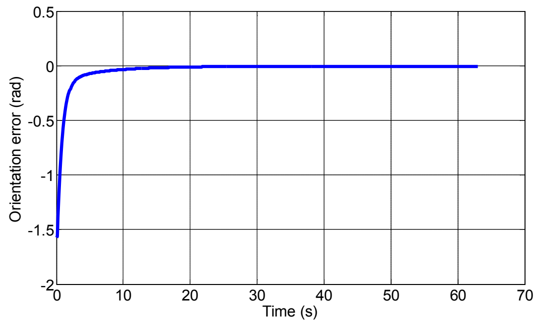

Figure 7: Orientation trajectory tracking error

5.1.2 Case study-2

The initial location of the desired trajectory is $q_r(0) = [2,0,\frac{\pi}{2}]^T$ and the actual initial posture of the robot is $q(0) = [\,1.5,0,0]^T$, circular trajectory tracking simulation and posture error curves are shown in **Figs. 9** to **12** respectively. It is quite obvious that the proposed controller achieved its function properly good.

MSE=
X_r= (0.000216878234982221),
Y_r= (0.000308521129642213),
Θ_r= (0.0246292867874308)

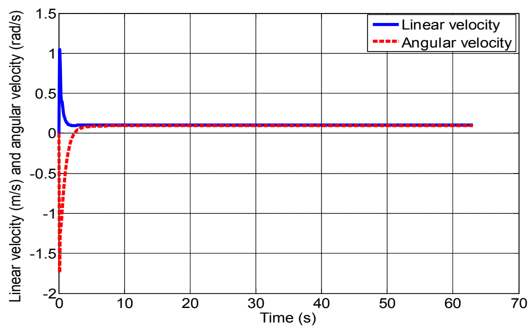

Figure 8: Linear and angular velocity of mobile robot

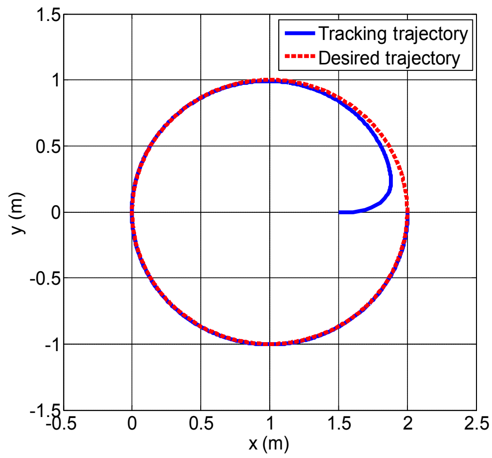

Figure 9: Circular trajectory tracking

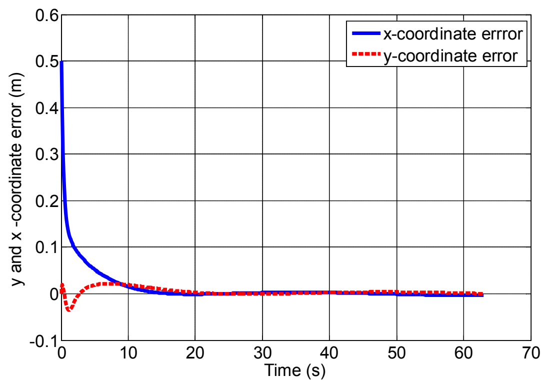

Figure 10: x-y Coordinate trajectory tracking error

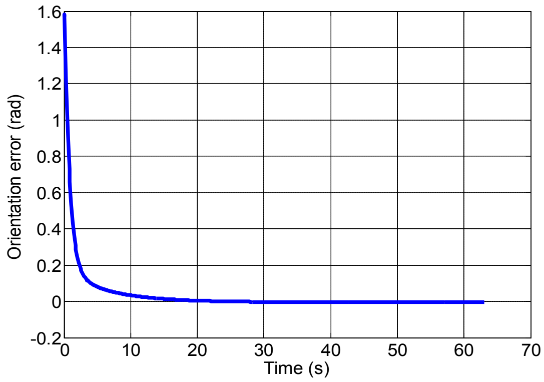

Figure 11: Orientation trajectory tracking error

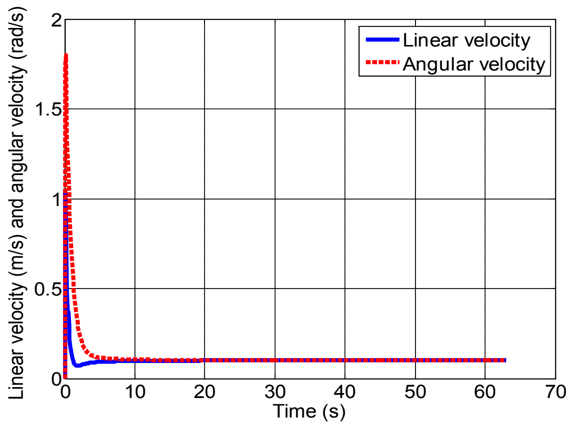

Figure 12: Linear and angular velocity of mobile robot

5.2 *Infinity Trajectory-1*

The lemniscuses or infinity is not easy tracking case which has changeable radius and rotation. The following equations describe the infinity trajectory used in this work:

$$x_r(t) = 0.75 + 0.75 \ \sin(\tfrac{2\pi t}{50}) \quad (45)$$

$$y_r(t) = 0.75 \ \sin(\tfrac{4\pi t}{50}) \quad (46)$$

$$\theta_r(t) = atan(\dot{y}_r/\dot{x}_r) \quad (47)$$

5.2.1 Case study-1

The actual MR initial position is, $q(0) = [\,0.75, 0.3, \frac{\pi}{5}]^T$ and the virtual MR initiates from, $q_r(0) = [0.75, 0, 1.2041]^T$. The simulation results of infinity trajectory tracking are shown in **Figs. 13-16**, where it is clearly that the tracking does not coincide with the desired trajectory always but it is still near from it. This performance considers an expectable tracking for such difficult type of trajectories.

MSE=

X_r = (0.000506006129832323),

Y_r= (3.64979166783888e-05),

Θ_r= (0.0242296480345646)

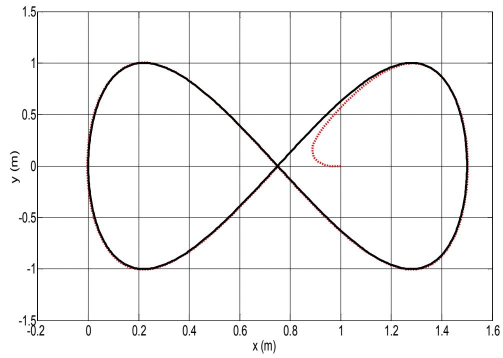

Figure 13: Infinity trajectory tracking

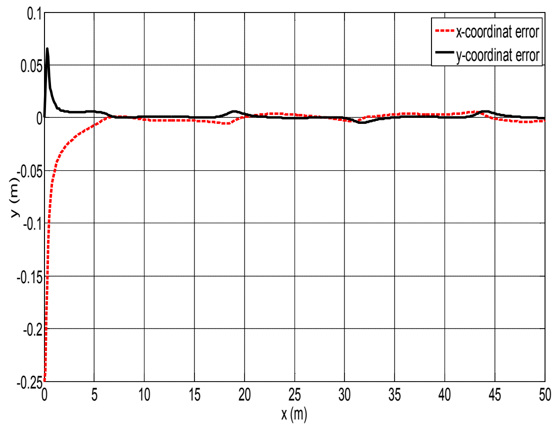

Figure 14: x-y Coordinate trajectory tracking error

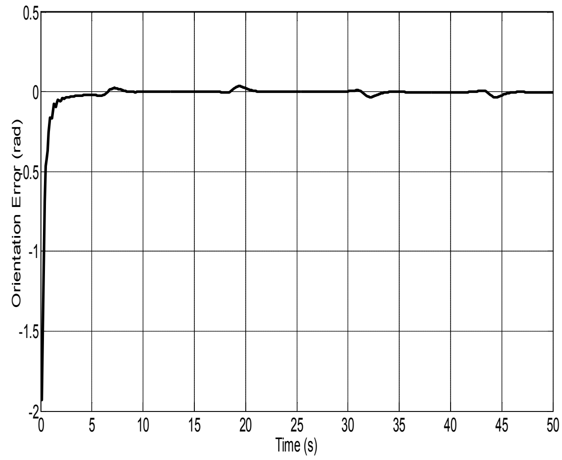

Figure 15: Orientation trajectory tracking error

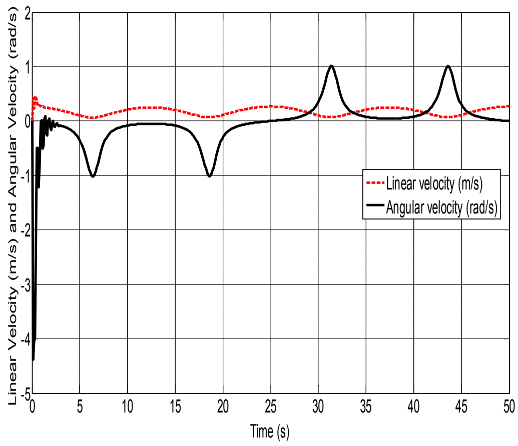

Figure 16: Linear and angular velocity of mobile robot

5.3 *Infinity Trajectory_2*

The actual MR position is, $q(0) = [\,0.75, 0.3, \frac{\pi}{5}]^T$ and its virtual position is$q_r(0) = [0.75, 0, 1.2041]^T$. The simulation results of infinity trajectory tracking are shown in Figs. 17-20, where it is clearly that the tracking doses not coincide with the desired trajectory always but it is still near from it. This performance considers an expectable tracking for such difficult type of trajectories.

MSE=
X_r= (0.000165902230086451),
Y_r= (2.37791020615571e-05),
Θ_r= (0.00807532487502962)

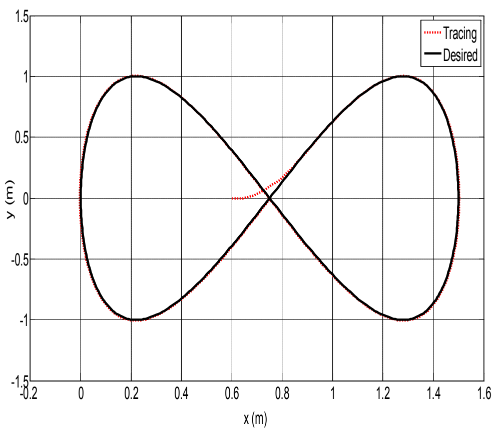

Figure 17: Infinity trajectory tracking

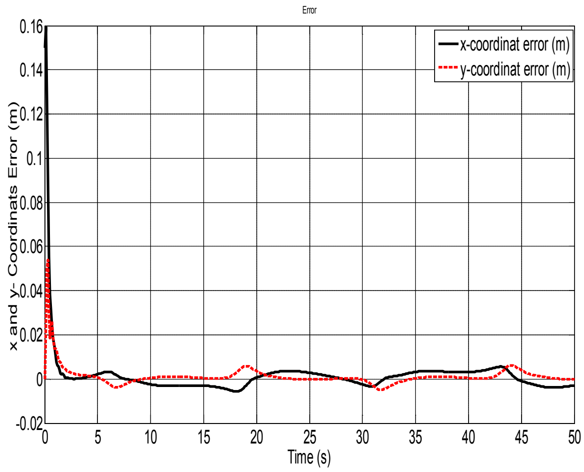

Figure 18: x-y Coordinate trajectory tracking error

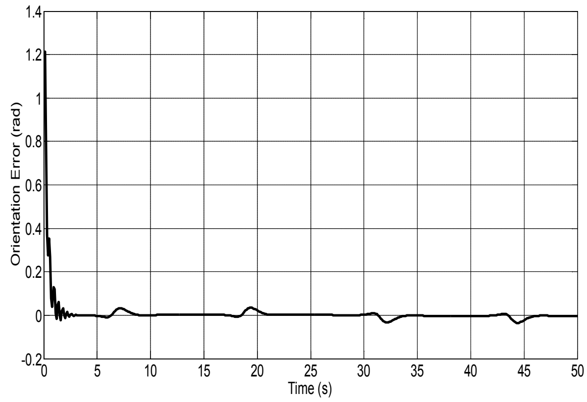

Figure 19: Orientation trajectory tracking error

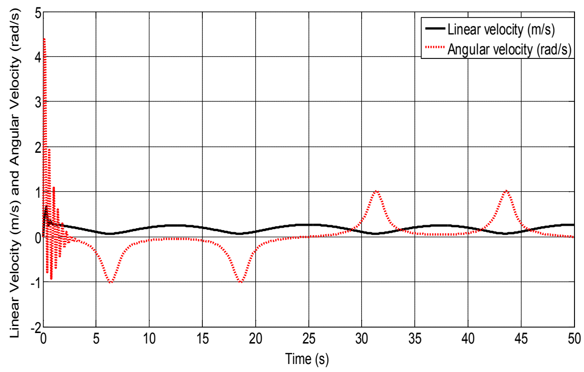

Figure 20: Linear and angular velocity of mobile robot

5.4 *Linear Trajectory-1*

Simulation results are also achieved for line as a desired trajectory. The desired line trajectory can be described using the following equation:

$$\begin{bmatrix} x_r(t) \\ y_r(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} 1 + 0.894 * v_r * t \\ 2 + 0.4475 * v_r * t \\ 0.463 \end{bmatrix} \qquad (48)$$

5.4.1 Case study-1

The actual MR position is , $q(0) = [0, 0, 0]^{\mathrm{T}}$.The simulation results of line trajectory-1 tracking are shown in **Figs. 21-24**, while reference robot initial position is $q_r(0) = [\,1, 2, 0.463]^{\mathrm{T}}$.

MSE=

X_r= (0.0353419615751516),
Y_r= (0.182919646927685),
Θ_r= (0.0691019109460619)

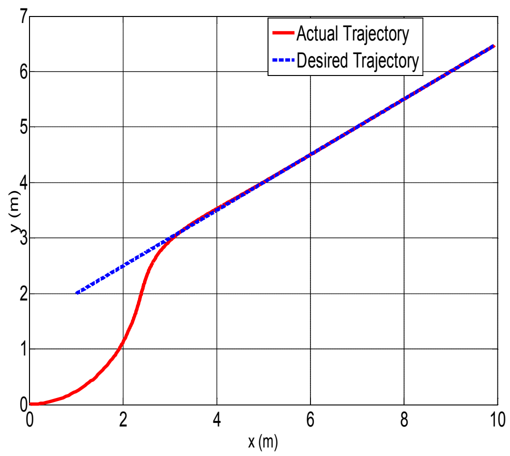

Figure 21: Line trajectory tracking

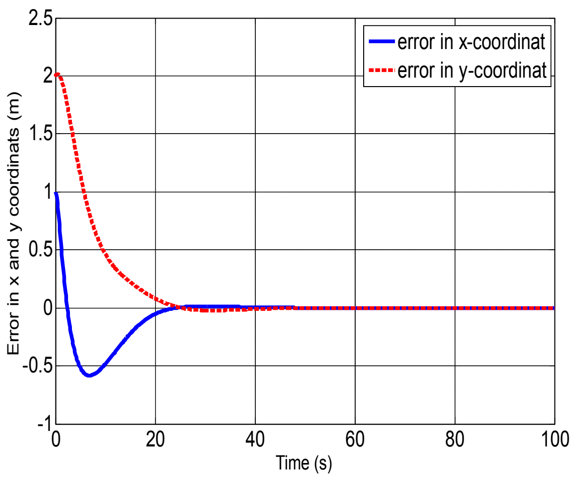

Figure 22: Linear and angular velocity of mobile robot

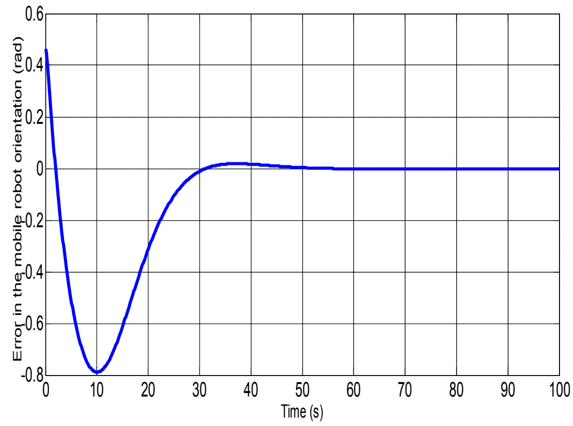

Figure 23: Orientation trajectory tracking error

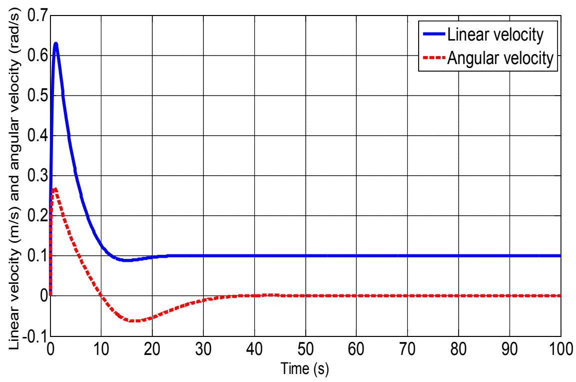

Figure 24: Linear and angular velocity of mobile robot

5.5 *Linear Trajectory_2*

The robot initial position is $\boldsymbol{q}(0) = [2, 0, \pi]^{\mathrm{T}}$, the simulation results of line trajectory-2 tracking are shown in **Figs. 25-28**, while reference robot starts form the initial posture $\boldsymbol{q_r}(0) = [\,1, 2, 0.463]^{\mathrm{T}}$.

MSE=
X_r= (0.00368783782668100),
Y_r= (0.0243976564719479),
Θ_r= (0.0406777214466495)

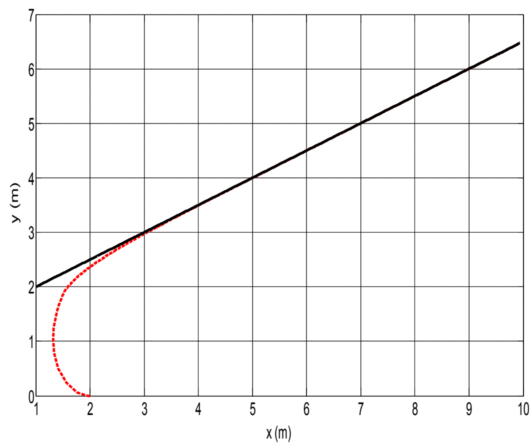

Figure 25: Line trajectory tracking

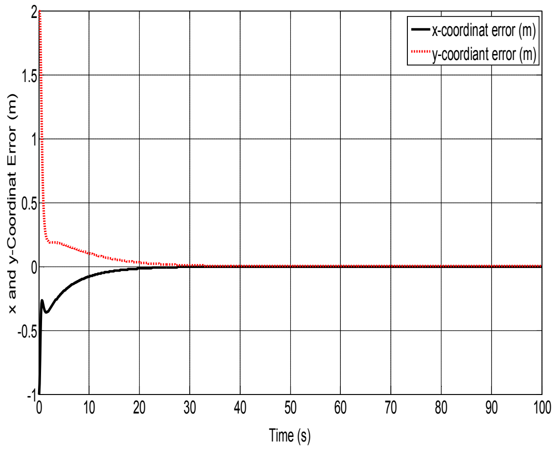

Figure 26: Linear and angular velocity of mobile robot

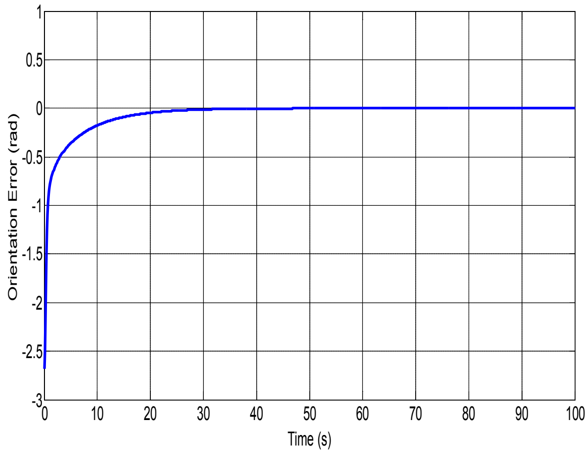

Figure 27: Orientation trajectory tracking error

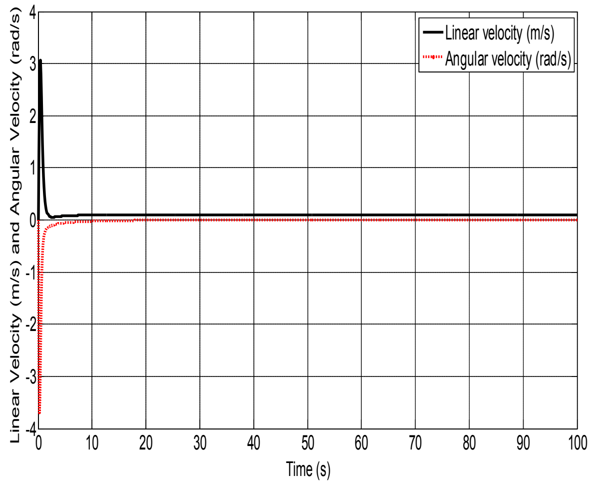

Figure 28: Linear and angular velocity of mobile robot

5.6 *validation of the PID-Back stepping controller*

In order to determine the improvement of the implementation of the adopted controller with another controller using fractional order PID [27], **Table 3** show comparison between the two cited controllers for different shapes in x, y and θ with improvement percentage. It is obvious that the adopted controller of PID – Back stepping shows good improvement in comparison with the controller of [27] in spite of the difference between the actual and virtual position for MR.

Table 3: MSE in X coordinate, Y coordinate and Θ orientation for the present work and work by [27]

Shape infinity	**Fractional order PID FOPID[27]**	**Present work PID – Back stepping**	**Improvement%**
X Y Θ	4.069 E -04 5.2178 E -04 0.0079	1.689 E -04 2.3779 E -05 0.00807	58.5% 95% -2.1%
Actual position **Virtual position**	$[0.75\ \ 0.3\ \ \pi/5]^T$ $[0.844\ \ 0.24\ \ 1.204]^T$	$[0.75\ 0.3\ \pi/6\ ]^T$ $[0.75\ \ 0\ \ 1.204]^T$	
Shape Circular	**FOPID[27]**	**PID – Back stepping**	**Improvement%**
X Y Θ	1.268 E -04 6.38 E -04 0.053	2.168 E -04 3.085 E 04 0.024	-70.9% 51.7% 54.7%
Actual position **Virtual position**	$[1.4\ \ \ 0\ \ \ 0]^T$ $[1.5\ \ \ 0\ \ \ 0\ ]^T$	$[1.5\ \ \ 0\ \ \ 0]^T$ $[2\ \ \ 0\ \ \ 0]^T$	
Shape Line	**FOPID[27]**	**PID – Back stepping**	**Improvement%**
X Y Θ	0.0074 0.0064 0.0639	0.00368 0.02439 0.04	50.25% -2.81% 59.75%
Actual position **Virtual position**	$[1.4\ \ 1.4\ \ \pi/8\ ]^T$ $[1\ \ \ 2\ \ 0.462]^T$	$[2\ \ \ 0\ \ \ \pi]^T$ $[1\ \ 2\ \ 0.463]^T$	

6. Conclusions

The PID-Back stepping controller based on optimization method for differential driving wheeled mobile robot has been demonstrated in this paper and it is concluded that:

- The adopted controller which consists of a back-stepping technique and PID controller modifies the output of the nonlinear kinematic trajectory tracking controller.
- The proposed controller stability is investigated using Lyapunov method.
- Optimal value of gains for both back-stepping and PID controller have been used through using the optimization method (PSO).
- Simulation tests have been conducted to various shape of trajectories (circular, infinity, and linear) with different initial conditions using Matlab program and the results show a reasonable accuracy of the developed controller in comparison with [27].
- The results show the proposed controller minimized the tracking errors.

References

[1] Aydin, S.;Kilic, I.; Temeltas, H. , "Using lindebuzo gray clustering neural networks for solving the motion equations of a mobile robot". Arab". J. Sci. Eng. 36(5), 795-807 (2011).

[2] Al-Arji A. ,"Development of kinematic path-tracking controller Design for Real Mobile Robot based on optimal Back-stepping Slice Genetic robust algorithm technique". Arab. J. Sci. Eng. 39(12), 8825-8835 (2014).

[3] Al-Arji A. ,"Design of on-line Nonlinear Kinematic Trajectory Tracking controller for Mobile Robot based on optimal back-stepping Technique". Iraqi J. comp. comm. Cont. Syst. Eng. 14(2), 25-36(2014).

[4] Cai, N.; Cao, J-w.; Ma, H-Y.; Wang, c-x ," Swarm stability analysis of nonlinear dynamical multi-agent systems via relative Lyapunov function". Arab". J. Sci. Eng. 39(3), 2427-2434 (2014).

[5] Kolmanovsky, I. ; McClamroch, N. H. , "Developments in non-holonomic control problems". IEEE Control Systems Magazine 15 (6), 20-36(1995).

[6] Jiang, Z.; Nijmeijer, P. H. ,"Tracking control of mobile robots: a case study in back stepping". Automatica 33 (7), 1393–1399(1997).

[7] Pourboghrat,F.; Karlsson, M.P. ,"Adaptive control of dynamic mobile robots with non-holonomic constraints". Computers & Electrical Engineering 28 (4), 241–253(2002).

[8] Brockett, R.W. ,"Asymptotic stability and feedback stabilization in Differential Geometric Control Theory". Birkhkuser, Boston, MA, 181–191(1983).

[9] Morin, P. ; Samson, C. ,"Control of non-holonomic mobile robots based on the transverse function approach". IEEE Transactions on Robotics, 25 (5), 1058–1073(2009).

[10] Lizarraga, D., "Obstructions to the existence of universal stabilizers for smooth control systems". Mathematics of Control, Signals, and Systems, MCSS (16), 255–277(2004).

[11] Pourboghrat, F. , "Exponential stabilization of non-holonomic mobile robots". Computers & Electrical Engineering 28 (5), 349–359(2002).

[12] Kanayama,Y. ; Nilipour,A. ; Lelm,C.; ,"A locomotion control method for autonomous vehicles". Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Washington, DC, USA,(2), 1315–1317(1988).

[13] Buccieri, D. ; Perritaz, D. ; Mullhaupt,P. ; Jiang, Z.-P.; Bonvin, D. ,"Velocity-scheduling control for a unicycle mobile robot: theory and experiments". IEEE Transactions on Robotics 25 (2), 451–458(2009).

[14] LaValle, S. M. ,"Planning Algorithms". Cambridge University Press, Cambridge, UK, (2006).

[15]Lepetić,M. ; Klančar, G. ; Škrjanc, I. ; Matko, D . ; Potočnik ,B. ,"Time optimal path planning considering acceleration limits". Robotics and Autonomous Systems 45 (3–4), 199–210(2003).

[16]Pozna, C. ; Troester , F . ; Precup, R. E.; Tar, J. K.; Preitl,S. ,"On the design of an obstacle avoiding trajectory: method and simulation". Mathematics and Computers in Simulation 79 (7), 2211–2226(2009).

[17] Klančar,G.; Matko,D.;Blažič,S. ,"A control strategy for platoons of differential drive wheeled mobile robot". Robotics and Autonomous Systems 59 (2), 57–64(2011).

[18] Knayama,Y.;Kimura,Y.;Miyazaki,F.;Noguchi,T.; "A stable tracking control method for an autonomous mobile robot". Proceedings 1990 IEEE International Conference on Robotics and Automation, Los Alamitos, CA, USA,(1), 384–389 and 1722-1727(1990).

[19] Samson, C. ," Time-varying feedback stabilization of car like wheeled mobile robot". International Journal of Robotics Research 12 (1), 55–64(1993).

[20] Klančar, G.; Škrjanc, I. , "Tracking-error model-based predictive control for mobile robots in real time". Robotics and Autonomous Systems 55 (6), 460–469(2007).

[21]Guechi, E.-H. ; Lauber, J. ; Dambrine, M. ; Klančar, G.; Blažič, S. ,"Control design for non-holonomic wheeled mobile robots with delayed outputs". Journal of Intelligent & Robotic Systems, 60 (3), 395–414(2010).

[22] Li, T. H. S.; Chang, S. J. ,"Autonomous fuzzy parking control of a car-like mobile robot". IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 33 (4), 451–465(2003).

[23] Hou, Z.-G.; Zou, A.-M.; Cheng, L.;Tan,M. ,"Adaptive control of an electrically driven non-holonomic mobile robot via back stepping and fuzzy approach". IEEE Transactions on Control Systems Technology 17 (4), 803–815(2009).

[24] Wai,R.-J.;Liu,C.-M. ,"Design of dynamic petri recurrent fuzzy neural network and its application to path-tracking control of non-holonomic mobile robot". IEEE Transactions on Industrial Electronics, 56 (7), 2667–2683(2009).

[25] V. Gazi and K. M. Passino ,"Swarm stability and optimization". German: springer science + business media (2011).

[26] R. Fierro and F. L. Lewis, "control of a Nonholonomic MR :Back Stepping Kinematics into DYNAMICS, " Jour. & Robotic System, Vol 4 ,No. 3, 1997. PP.149-163.

[27] Salah Mahdi Swadi,"trajectory Tracking Control of Autonomous Chaotic Wheeled Mobile robot ".PhD. Thesis April 2016 University of Technology – Mech. Eng. Dept.

Nomenclature

$A(q)$	Vector associated with constrains.
$B(q)$	Input transformation matrix.
C	Origin of the reference frame of MR.
CG	Center of gravity of MR.
D	Center of gravity offset from driving axis.
e	Pose errors vector.
$F(\dot{q})$	Surface friction vector.
$G(q)$	Gravitational vector.
K_1,k_2,k_3..	Control gains.
K_d	Derivative gain.
K_i	Integral gain.
K_p	Proportional gain.
2L	Distance between two wheels of MR.
M	Mass of mobile robot.
M(q)	Symmetric positive definite inertia matrix.
q	pose vector of the actual MR.
$S(q)$	Transformation matrix of MR velocity.
u(t)	The output of the PID controller in time domain.
Vr	Linear velocity of the reference MR.
Vm(q,q.)	Centripetal and carioles matrix.
[Xc ,Yc]	Coordinates of vehicle mass center.

Greek symbols

λ	Vector of constrain forces.
ω	Angular velocity of MR
ω_r	Angular velocity of the reference MR.
τ	Input torque vector.
τ_d	Bounded unknown disturbances torque.
θ	Orientation with respect to the inertia basis.

السيطرة على حركة انسان آلي متحرك بعجلات بأستخدام طريقة مثلى (PSO)

موفق علي توفيق

قسم الهندسة الميكانيكية - الجامعة التكنولوجية، بغداد، العراق، 20040@uotechnology.edu.iq

الخلاصة – تعد طريقة استخدام المسيطر (PID) للسيطرة على حركة مسارات (WMR) غير كفوءة لكون النظام غير خطي. ولذلك فان البحث الحالي يقدم طريقة تعشيق او تركيب طريقة الـ (Back-stepping) مع الـ (PID) للحصول على مسيطر فعال وكفوء لكي يتعامل مع النظام اللاخطي. تم تطبيق مسارات مختلفة شائعة الاستخدام مثل (∞ و الدائرة والخط المستقيم) ليتم تتبعها من قبل (WMR) لفحص فعالية نظام السيطرة.كما تم اختبار المسيطر من خلال مقارنة مسار (WMR) مع المسارات المطلوبة وتم احتساب معدل مربع الخطأ لاحداثيات X ، Y ، θ .تم استخدام طريقة (Practical Swarm Optimization) لايجاد (Control Gain) للوصول الى اقل نسبة خطأ. اظهرت نتائج النمذجة لتتبع الاثر اداءا جيد للمسيطر مع المسارات المطلوبة.

الكلمات الرئيسية – مسيطر PID , طريقة الرجوع المتدرج, انسان الي متحرك, ذكاء اصطناعي امثل.