Engineering and Technology Journal

Journal homepage: engtechjournal.org

Modeling and Control of Wheeled Mobile Robot With Four Mecanum Wheels

Sameh F. Hasana [a*], **Hassan. M. Alwan** [b]

[a] University of Technology, Baghdad , Iraq , 21145@student.uotechnology.edu.iq

[b] University of Technology, Baghdad , Iraq , 20071@uotechnology.edu.iq

*Corresponding author.

Submitted: 15/11/2020 Accepted: 06/03/2021 Published: 25/05/2021

KEYWORDS

FMWMR, Control, Backstepping, Fuzzy logic, Trajectory tracking, PSO.

ABSTRACT

This work presents a driving control for the trajectory tracking of four mecanum wheeled mobile robot (FMWMR). The control consists of Backstepping-Type 1 Fuzzy Logic-Particle swarm optimization i.e.,(BSC-T1FLC-PSO). The kinematic and dynamic models have been derived. Backstepping controller (BSC) is used for finding controlled torques that generated from robot motors while Type-1 fuzzy logic control (T1FLC) as well as particle swarm optimization (PSO) used for finding the appropriate values of gain parameters of BSC. Square trajectory has been selected to test the performance of the control system of FMWMR. MATLAB/ Simulink is used to simulate the results. It has been concluded from the results that obtained from this control system there is a good matching between the simulated and the desired trajectories.

How to cite this article: S. F. Hasan and H. M. Alwan, "Modeling and Control of Wheeled Mobile Robot With Four Mecanum Wheels," Engineering and Technology Journal, Vol. 39, Part A, No. 05, pp. 779-789, 2021.
DOI: https://doi.org/10.30684/etj.v39i5A.1926

1. Introduction

The most invitingly type of mobile robots are the wheeled mobile robot (WMR) due to their capability for fast maneuver and effective control approaches [1]. WMR can be used in a lot of applications which are very important in our life such us in transportations, health and in industry [2]. In spite of benefit properties of WMR that are maintained above but there is weakness specially when the robot is moving in a narrow space. So that, WMR with Mecanum wheels can overcome this drawback due to their lateral movement ability as well as higher maneuver [3]. There are some studies that are concerned with approaches that are used for control and tracking of WMR. In [4], backstepping controller was used for the trajectory tracking of FMWMR. The backstepping gain parameters had been chosen by trial and error. In [5], a nonlinear PID controller used for the tracking of non-holonomic WMR. In [6], the authors studied the inverse kinematics of FMWMR with stepper motors. In [7], Fractional order PID controller based on modified PSO was adopted as a control system for tracking of non-holonomic WMR while in [8], BSC was used as a controller action of FMWMR. LQR controller was Implemented for the tracking of FMWMR, and its results compared with PI controller [9]. Model predictive controller had been selected as a controller system for omnidirectional WMR [10]. Two controllers had been adopted for FMWMR tracking which are PID and LQR controllers [11].

In this paper, hybrid controller design consist of BSC-T1FLC-PSO have been used for the tracking of FMWMR. MATLAB programing is selected for programing the equations and to present the results. The results show that there is a good matching between the desired and simulation tracking.

2. MODELING OF MOBILE ROBOT

In this section, the kinematics as well as the dynamic models of FMWMR are derived. First, the kinematic model has been derived. Kinematic model of WMR describes the mapping between the robot velocities with wheels' angular velocities. The WMR with Mecanum wheels is a holonomic robot due to its ability to move in the lateral direction and this feature is very important especially when the robot moves in narrow spacing or in the case of the obstacle's avoidance. The FMWMR consist of platform as well as four Mecanum wheels. Each of this wheel contains a set of passive rollers which are in 45° about the hub of the robot. The robot parameters can be defined as:

Points (A) and (D) represent the centroid of the robot and the wheel center, respectively. (ξ_i):- The angle between the wheel axis and the vector (DA). (δ_i) is the measured angle between (A_i) and X_R. (Ω_i) is the angle between the roller rotation axis and wheel plane. (r_i) is the wheel (i) radius and ($\dot{\Omega}_I$)is the rotational speed of each wheel and these parameters are illustrated in Figure 1 as shown below.

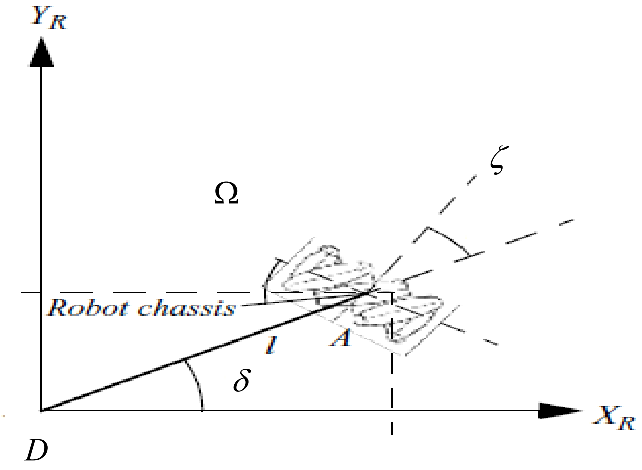


Figure 1: Parameters of Mecanum wheel [1].

In the derivation of kinematic model, pure rolling condition i.e., (without slipping) is considered. The velocity of (A_i) is equal to ($r_i\ \dot{\Psi}_i$) while its value along the roller axis equal to ($r\dot{\psi}_i \cos\Omega_i$). According to the local frame {R}, the robot translation velocities are [$\dot{x}_R\ \dot{y}_R$] and ($\dot{\theta}$) can be defined as the robot rotational speed about (Z_R). The velocity of
(A_i) for each wheel along the axis of roller contact can be expressed as:

$$V_{A_i} = \dot{x}_R \cos\left[\frac{\pi}{2} - (\frac{\pi}{2} - (\delta_i + \xi_i) + (\frac{\pi}{2} - \Omega_i)\right] + \dot{y}_R \cos\left[(\frac{\pi}{2} - (\delta_i + \xi_i) + (\frac{\pi}{2} - \Omega_i)\right] \tag{1}$$
$$+ l\dot{\theta}\cos\left[\delta_i + \left[(\frac{\pi}{2} - (\delta_i + \xi_i) + (\frac{\pi}{2} - \Omega_i)\right]\right]$$

After straight forward calculations, it can obtain the following equation:

$$\left[\sin(\delta_i + \zeta_i + \Omega_i) \quad -\cos(\delta_i + \zeta_i + \Omega_i) \quad -l\cos(\zeta_i + \Omega_i\right] \bullet {}^{R}R_{O}(\theta) \bullet \dot{P}_O = r_i\,\dot{\psi}_i \cos\Omega_i \tag{2}$$

Where i=1,2,3,4 as well as ($\dot{P}_0$) and ($R_{R_0(\theta)}$) are the velocity vector of the robot in terms of global coordinate and the rotation matrix, respectively. The robot velocity vector in terms of global {O) and local {R} frames can be expressed as below:

$$\dot{P}_O = \begin{bmatrix} \dot{x}_O \\ \dot{y}_O \\ \dot{\theta}_O \end{bmatrix} ; \dot{P}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} \tag{3}$$

The relation between the global and local velocity vector can be described as:

$$\dot{P}_R = {}^R R_O(\theta) \bullet \dot{P}_O \tag{4}$$

The representation of rotation matrix is:

$${}^R R_O = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The configuration of FMWMR can be seen in Figure 2 as below:

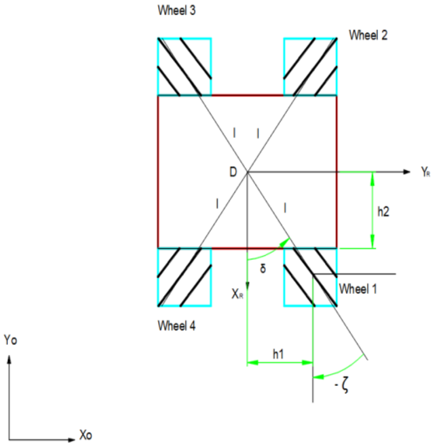


Figure 2: FMWMR Configuration.

The radius of each wheel is equal so that (r_i=r) and the inverse kinematics equation can be written as:

$$\begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} = -\left(\frac{1}{r}\right) \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \\ a41 & a42 & a43 \end{bmatrix} \bullet \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \dot{x}_O \\ \dot{y}_O \\ \dot{\theta}_O \end{bmatrix} \tag{5}$$

Where:

$$a_{11} = a_{12} = a_{21} = a_{42} = 1,\ a_{22} = a_{31} = a_{32} = a_{41} = -1,\ a_{22} = a_{31} = a_{32} = a_{41} = -1,\ a_{14} = a_{24} = a_{34} = a_{44} = \sqrt{2}\, l \sin(\frac{\pi}{4} - \delta)$$

The magnitude of (δ) can be evaluated as δ=$\tan^{-1}\left(\frac{h1}{h2}\right)$. Forward kinematic equation can be expressed as:

$$\begin{bmatrix} \dot{x}_O \\ \dot{y}_O \\ \dot{\theta}_O \end{bmatrix} = -\left(\frac{\sqrt{2}}{2}\right) r\, H^+ \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} \tag{6}$$

Where $H^+ = (H^T H)^{-1} H^T$ and the matrix [H] has been represented as:

$$H = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \\ a41 & a42 & a43 \end{bmatrix} \bullet \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (7)$$

Secondly, the dynamic model has been derived by applying Lagrange equation. It is considered that the robot centroid point (D) not coincide with robot center of gravity point (D$^{/}$). With respect to frame {R}, the velocities of points (D) and (D$^{/}$) can be expressed as:

$$^{R}v_{D} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_O \\ \dot{y}_O \end{bmatrix} = \begin{bmatrix} \dot{x}_O \cos\theta \\ -\dot{y}_O \sin\theta \end{bmatrix};\; ^{R}v_{D^{\backslash}} = {}^{R}v_{D} + \dot{\theta} k_R \times r_{D^{,}/D} \qquad (8)$$

The parameters (d$_1$) and (d$_2$) represent the distance between point (D) and (D$^{/}$). Lagrangian (L) is determined as:

$$L=T-U \qquad (9)$$

(U) is the potential energy and its magnitude is zero due to the plane movement of the robot and for this reason (L=T). (T) is the kinetic energy and the mathematical model that used for representing it0 can be written as:

$$T = \frac{1}{2}\left[m_b v^{T}_{D/} v_D + I_b \dot{\theta} + \sum_{i=1}^{4} m_{wi} (r\dot{\psi}_i)^2 + \sum_{i=1}^{4} I_{wi} \dot{\psi}_i^{\;2} \right] \qquad (10)$$

Where m$_b$: platform mass, I$_b$: platform moment of inertia, m$_w$: wheel mass, and I$_w$: wheel moment of inertia.

Each wheel has the same moment of inertia and that leads to I$_{w1}$= I$_{w2}$= I$_{w3=}$ I$_{w4}$=I$_w$. The representation of Lagrange equation is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i \qquad (11)$$

F$_i$ and $\dot{q}$ that are maintained in equation (11) represented the generalized forces and generalized coordinates respectively and they described as: $q = [q_1 \quad q_2 \quad q_3]^T = [x \;\; y \;\; \theta]^T$

The evaluation of generalized (Force/ Torque) can be expressed as:

$$F_i = \sum_{i=1}^{4} \left(\tau_i - \tau_{fi}\right) \frac{\partial \dot{\psi}_i}{\partial \dot{q}_i} \qquad (12)$$

Where (τ_{fi}) is frictional torque and (τ_i) is the torque of wheel (i). The equations of calculation the frictional forces and torques can be written as:

$$f_{si} = \mu\left(\frac{m_b}{4} + m_{wi}\right) * g \qquad (13)$$

$$\tau_{fi} = f_{si} * r \qquad (14)$$

Where (μ) can be defined as coefficient of friction and (g) is the gravitational acceleration. The formula that described the motion of the robot is:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B^T S f = \frac{1}{r} B^T \tau \qquad (15)$$

Where: - [M] is the matrix of inertia, [C] is the matrix of centripetal and Coriolis, [B] is the input transformation matrix, [S] is the matrix that indicate the direction of friction and [f] is representing the frictional effect. The elements of each matrix can be shown as below:

$$C=\begin{bmatrix} 0 & 0 & m_b\,\dot{\theta}(d_1\cos\theta-d_2\sin\theta) \\ 0 & 0 & m_b\,\dot{\theta}(d_1\sin\theta+d_2\cos\theta) \\ 0 & 0 & 0 \end{bmatrix};M=\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\quad \tau=\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix}\quad ;\ S=\begin{bmatrix} \mathrm{sgn}(\dot{\psi}_1) & 0 & 0 & 0 \\ 0 & \mathrm{sgn}(\dot{\psi}_2) & 0 & 0 \\ 0 & 0 & \mathrm{sgn}(\dot{\psi}_3) & 0 \\ 0 & 0 & 0 & \mathrm{sgn}(\dot{\psi}_4) \end{bmatrix};\ f=\begin{bmatrix} f_1\\ f_2\\ f_3\\ f_4\end{bmatrix}$$

Where:

$$m_{11}=m_{22}=4\left(m_w+\frac{I_w}{r^2}\right)+m_b\,,\ m_{21}=m_{12}=0$$

$$m_{13}=m_{31}=m_b(d_1\sin\theta+d_2\cos\theta),\ m_{23}=m_{32}=m_b(-d_1\cos\theta+d_2\sin\theta)$$

$$m_{33}=m_b\,({d_1}^2+{d_2}^2)+I_b+8\,(m_w+\frac{I_w}{r^2})l^2\sin^2(\frac{\pi}{4}-\delta)$$

$$B=\begin{bmatrix} (-\cos\theta+\sin\theta) & (-\cos\theta-\sin\theta) & -\sqrt{2}\,l\sin(\frac{\pi}{4}-\delta) \\ (-\cos\theta-\sin\theta) & (\cos\theta-\sin\theta) & -\sqrt{2}\,l\sin(\frac{\pi}{4}-\delta) \\ (\cos\theta-\sin\theta) & (\cos\theta+\sin\theta) & -\sqrt{2}\,l\sin(\frac{\pi}{4}-\delta) \\ (\cos\theta+\sin\theta) & (-\cos\theta+\sin\theta) & -\sqrt{2}\,l\sin(\frac{\pi}{4}-\delta) \end{bmatrix}$$

3. FOUR MECANUM WHEELED MOBILE ROBOT CONTROL

In this work, a hybrid controller consist of BSC-T1FLC-PSO has been used for trajectory tracking of FMWMR. Backstepping controller used for computing the controlled torques that are generated from WMR motors while T1FLC and PSO are applied for evaluating the proper gains values of BSC and that be described as below:

I. Backstepping Controller (BSC):

BSC is applied on equation (15) and that can be shown as below:

$$\ddot{q}=M^{-1}\left[\frac{1}{r}\,B^T\,\tau-C\,\dot{q}-B^T\,S\,f\right] \tag{16}$$

The state vectors of generalized coordinated can express as:

$$X=[X_1^T \qquad X_2^T]^T=[q^T \quad q^T]^T$$

In BSC, $(\dot{X}_1)$ is considered as a subsystem and it is defined as a virtual input (u_1). The tracking error equation can be represented as:

$$e_1=X_1-q_d\ \begin{bmatrix} e_x\\ e_y\\ e_\theta\end{bmatrix}=\begin{bmatrix} x_O\\ y_O\\ \theta\end{bmatrix}-\begin{bmatrix} x_{O,d}\\ y_{O,d}\\ \theta_d\end{bmatrix} \tag{17}$$

Where $q_d(t)$ is defined as robot desired tracking. The differentiation of equation (17) is written as:

$$\dot{e}_1=\dot{X}_1-\dot{q}_d=u_1-\dot{q}_d \tag{18}$$

To check the stability of the nonlinear control system, Lyapunov function is used and it's expressed in the vector form as:

$$V_1=\frac{1}{2}e_1^T K_1\,e_1 \tag{19}$$

The $K_1\in R^{3\times 3}$ symmetric and positive. The time derivative of equation (19) is:

$$\dot{V}_1 = \frac{1}{2}\dot{e}_1^T \; K_1 \, e_1 + \frac{1}{2} e_1^T K_1 \, \dot{e}_1 = e_1^T \, K_1 \, \dot{e}_1 = e_1^T \, K_1 (\text{u}_1\text{-} \dot{q}_d) \tag{20}$$

For making Eq (21) stable, (u_1) must replace by ($\dot{q}_d$ -e_1) and that leading to:

$$\dot{V}_1 = -e_1^T K_1 \, e_1 \;\leq 0 \tag{21}$$

From the above equation, it can see that the system is asymptotically stable. The equation that represents the velocity n be written as:

$$e_2 = X_2 - u_1 \tag{22}$$

The time derivative of equation (22) is:

$$\dot{e}_1 = \dot{X}_1 \text{-} \dot{q}_d = \text{X}_2\text{-} \dot{q}_d \text{+u}_1\text{-u}_1\text{=e}_2\text{-e}_1 \tag{23}$$

$$\dot{e}_2 = M^{-1}\left[\frac{1}{r} B^T \tau - C\dot{q} - B^T S f\right] - \ddot{q}_d + (e_1 - e_2) \tag{24}$$

It is necessary to check the stability of the system and there for Lyapunov function has been used as below:

$$V_2 = V_1 + \frac{1}{2} e_2^T K_2 \, e_2 = \frac{1}{2} e_1^T K_1 \, e_1 + \frac{1}{2} e_2^T K_2 \, e_2 \tag{25}$$

The derivate of equation (25) is:

$$\dot{V}_2 = e_1^{\,T} \, K_1 \, \dot{e}_1 + e_2^{\,T} \, K_2 \, \dot{e}_2 = e_1^{\,T} K_1 (e_2 - e_1) + e_2^{\,T} K_2 \left[(e_2 - e_1) + M^{-1} (\frac{1}{r} B^T \tau - C\dot{q} - B^T S f) - \ddot{q} \right] \tag{26}$$

The equation of the controlled torque is represented as below:

$$\tau = rB(B^T B)^{-1} M \left[\left[M^{-1} \left[-C\dot{q} - B^T S f \right] - \ddot{q} + (e_2 - e_1) - e_2 - K_2^{-1} K_1 \, e_1 \right] \right] \tag{27}$$

Now, substitute (27) in (26) and it can obtain:

$$\dot{V}_2 = -e_1^T \, K_1 \, e_1 - e_2^T \, K_2 \, e_2 \;\leq 0 \tag{28}$$

From (28), it can see that the system is asymptotically stable.

The parameters of BSC gains (**K_1** and **K_2)** are evaluating by T1FLC and PSO and the detail of the evaluating is described in the next sections.

II. Type-1 Fuzzy Logic Control (T1FLC):

In this work, T1FLC has been implemented to compute the controller gains parameters of BSC which is $K_1 \in R^{3\times 3}$ i.e.,(K_1,K_2,K_3). T1FLC is a computing framework and it relies on fuzzy sets, if then fuzzy rules as well as fuzzy reasoning. In this work, Mamdani membership function (MF_s) is selected as inference unit of fuzzy logic control (FLC) [12]. The input variables to FLC are (L) and (α) where (L) can be defined as the distance from the desired to actual WMR position and (α) is the deviation angle between the desired and actual WMR and that illustrated as in Figure 3.

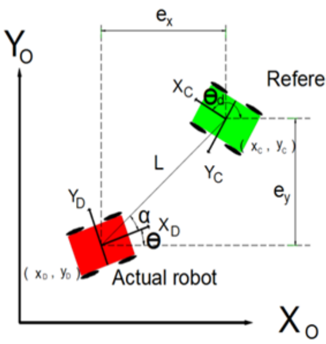


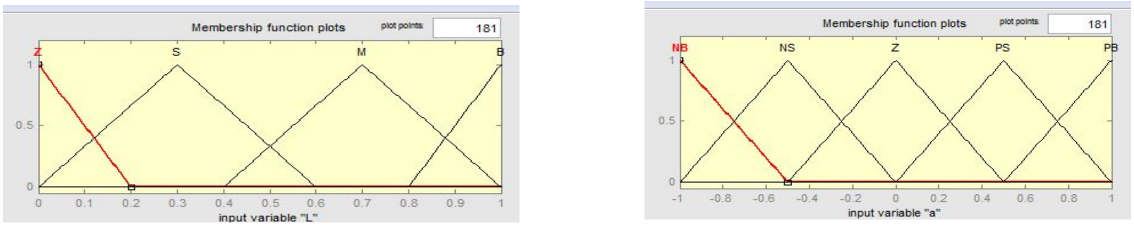
Figure 3: FMWMR with (α) and (L) parameters

The equations that used for evaluating (L) and (α) are represented as:

$$\alpha = \tan^{-1}(e_y / e_x) \tag{29}$$

$$L = \sqrt{(e_x)^2 + (e_y)^2} \tag{30}$$

The variable (α) has five triangle MF$_s$ which are positive big (PB),), positive small (PS), zero(Z), negative small (NS) and negative big (NB). While variable (L) has four MF$_s$ which are Zero(Z), small (S), Medium(M) and big(B). The output from the FLC are the three gains parameters of BSC i.e., (K$_1$, K2, K3) and their MF$_s$ are: big(B), medium(M) and small(S). The representation of the above MF$_s$ for the inputs and outputs parameters in MATLAB programing can be seen in Figures 4 and 5.



(*a*) (*b*)

Figure 4: MF$_s$ (a) for parameter (L) and (b) for parameter (α)

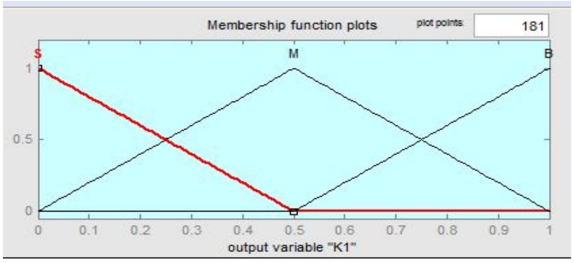


Figure 5: MF$_s$ for each gain parameter

It can be summarized the fuzzy rules that relating between the inputs and output variables in Tables I, II, and III.

TABLE I: Fuzzy Rules of gain (K$_1$).

L \ α	NB	NS	Z	PS	PB
Z	S	S	S	S	S
S	M	M	S	M	M
M	B	M	M	M	B
B	B	B	B	B	B

TABLE II: Fuzzy Rules of gain (K2).

α L	NB	NS	Z	PS	PB
Z	M	M	S	M	M
S	S	S	S	S	S
M	M	M	S	M	M
B	B	B	B	B	B

TABLE III: Fuzzy Rules of gain (K$_3$).

α L	NB	NS	Z	PS	PB
Z	B	M	S	M	B
S	B	M	S	M	B
M	B	M	S	M	B
B	B	M	S	M	B

The defuzzification formula that has been used in this work is the centroid of area (COA) and its mathematical equation is:

$$COA = \frac{\int_z \mu_A(z).z\,d(z)}{\int_z \mu_A(z)\,d(z)} \tag{31}$$

Now, the parameters (K$_1$, K$_2$, K$_3$) are evaluate from T1FLC but their values are in normalized mode i.e., (their magnitudes are between [0-1]) and so that, the output from FLC has been de-normalizing and for this reason PSO has been adopted for de-normalizing the output from FLC as well as computing the another set of BSC gains i.e., (K$_4$, K5, K6) and that has been described in the next section.

III. Particle Swarm Optimization (PSO)

PSO can be defined as a stochastic optimization that has been used for solving the problems. PSO is impelled by the intelligent behavior of animals such as flocks of birds. It is fast as well as simple algorithm. Each individual in the swarm is called particle. Particles can update their velocities and positions with respect to the change of the environment and trying to reach the optimum or best solution. The mathematical model that used for updating the velocity as well as the position is [13]:

$$v_{i,.t+1} = w * v_i + c_1 * rand * (pbest_{i,t} - x_{i,t}) + c_2 * rand * (gbest_{i,t} - x_{i,t}) \tag{32}$$

$$x_{i,t+1} = x_{i,t} + v_{i,t+1} \tag{33}$$

Where ($v_{i,t+1}$) is represent the speed of particle ith in iteration (t+1), $(x_{i,t+1})$ is defined as the location of the particle ith in iteration (t+1), (w) is inertia weight factor, c$_1$ and c$_2$ are the acceleration and learning coefficients respectively, (rand) is a random value between [0-1], (pbest) represents the best position of each particle and (gbest) defined as the global best position of the swarm. The parameters that are relating to PSO can be illustrated in Table IV.

TABLE IV:Parameters of PSO

Parameters	**Value**
No. of particles	10
No. of variables	6
No. of iterations	50
C$_1$=C$_2$	2
Objective Function	$\sqrt{e_x^2 + e_y^2 + e_\theta^2}$

Weight (w)	0.98

In this work, the main purpose of using PSO is to find the optimum de-normalizing values of gains parameters (K_1, K_2, K_3) that are computed from T1FLC as well as computing the second set of BSC gains parameters which are (K_4,K_5,K_6). The values of the gains (K_1,…, K_6) are within range (0-150).

4. RESULTS AND DISCUSSIONS

The adopted controller has been tested with square trajectory by using MATLAB programing. The dimension of the square path is equal to (2.25 meter) and the robot moving with constant desired velocity equal to (0.1 m/s). The parameters of FMWMR are taken from [14]:

M=3.1 kg, M_w=0.35 kg, I_b=0.032 kg m^2, I_w=6.24*10^{-4} kg m^2, r=0.05 m, h_l=h_2=0.15 m, μ=0.02, d_1=d_2=0.02 m and l=0.25. The magnitude of gain (**K₁**) that including (k_1, k_2, k_3) that computed from T1FLC based on PSO are:

$$\mathbf{K_1}=\begin{bmatrix} 112.2149 & 0 & 0 \\ 0 & 73.5671 & 0 \\ 0 & 0 & 44.2837 \end{bmatrix}$$ while the magnitude of (**K₂**) which including (k_4, k_5, k_6) that are computed from PSO are

$$\mathbf{K_2}=\begin{bmatrix} 54.3361 & 0 & 0 \\ 0 & 40.3458 & 0 \\ 0 & 0 & 18.1154 \end{bmatrix}$$

The robot started its movement from the initial position q= $[0, 0, 0]^T$ and the results of the adopted controller can be showed as below:

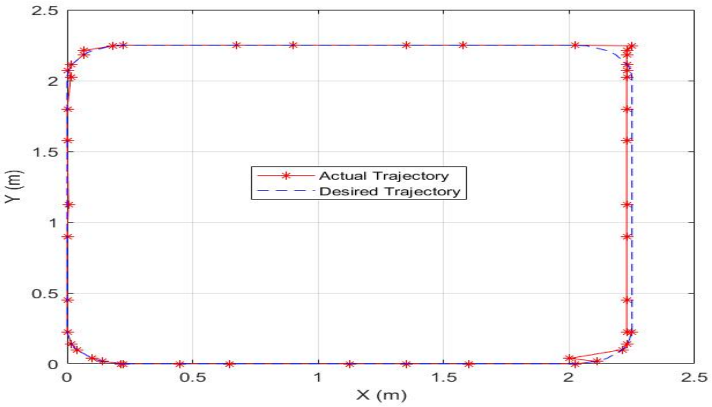
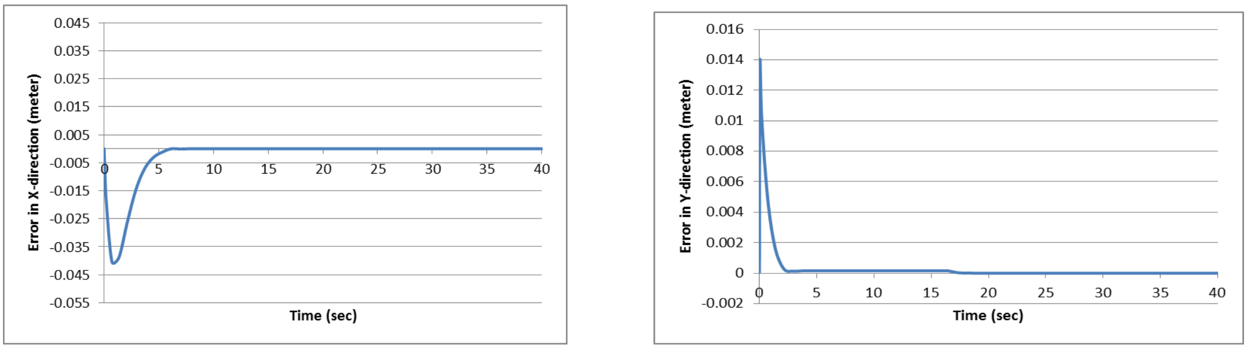


Figure 6: Square Trajectory Tracking

From Figure 6 explains the performance of the trajectory tracking of the robot. It can see that there is a good matching between the actual and desired tracking and that indicate the proposed controller has effective performance. The behavior of the errors in (x and y) coordinates can be seen in Figure 7.



(a) (b)

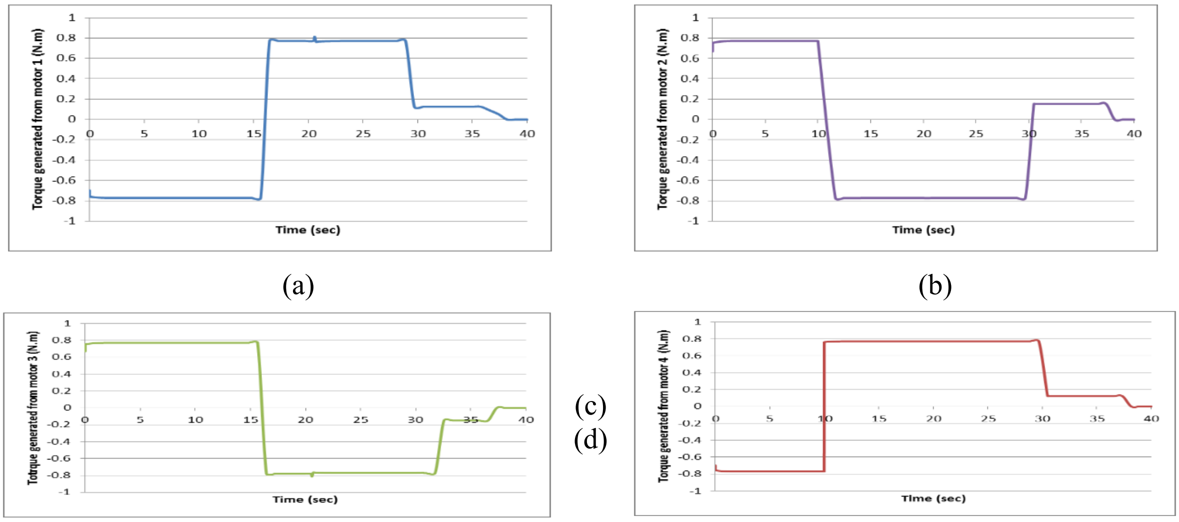
Figure 7: (a) Error in x- coordinate, (b) Error in y- coordinate

The maximum value of the error in (x) coordinate is equal to (-0.0377 m) at the beginning of the simulation and after about seven seconds the magnitude of the error is approximately converging to zero while the maximum value of the error in the (y) coordinate (0.01389 m) and after four seconds the error is converging to zero. The values of the mean square error (MSE) for each state error component can be seen in Table V.

TABLE V: mean square error.

Mean Square Error	Value (m)
x- direction	$2.3787*10^{-4}$
y-direction	$1.4435*910^{-4}$

The magnitudes of the torques that generated from the robot motors can illustrated in Figure 8.

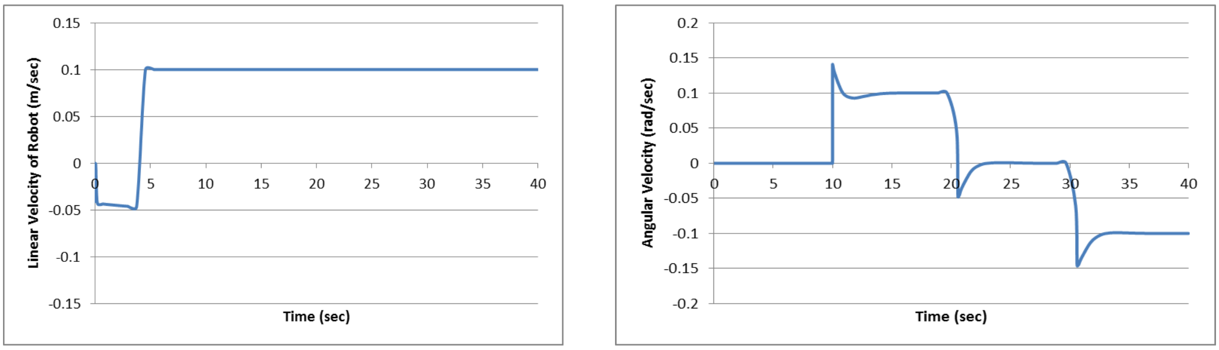


(a) (b)

(c)
(d)

Figure 8: The magnitudes of the torques.

It can b seen from Figure 8 (a), (b), (c) and (d) that the magnitudes of the torques is between (0.8) to (-0.8) N.m. The linear and angular velocity of the robot is shown in Figure 9.



(a) (b)

Figure 9: (a) Linear velocity of robot (m/sec), (b) Angular velocity (rad/sec)

From the results that are shown in Figures 7, 8 and 9, it is concluded that the magnitudes of the errors are small in the beginning of the simulation and it is converge to zero after small period of time and the magnitudes of the torques and velocities of the robot are smooth and acceptable, and all these results indicate the robust and effective of the proposed controller.

5. CONCLUSIONS

In this work, a hybrid controller consisting of BSC-T1FLC-PSO controller for trajectory tracking of FMWMR has been presented. BSC controller is used for computing the controlled torques while T1FLC as well as PSO are adopted for evaluating the optimum values of BSC gains parameters. The dynamic model of FMWMR has been derived by using Lagrange method. It has been proved the stability of the hybrid controller by using Lyapunov theory. MATLAB programing is used for simulating the results of the proposed controller for the case of the square tracking. The magnitudes of the MSE for (x and y) coordinates are ($2.3787*10^{-4}$) m and ($1.4435*910^{-4}$) m respectively and these values of the error are small, and the magnitudes of the error are converging to zero after about five seconds. The magnitudes of the torques as well as the robot linear and angular velocities are presented, and they are smooth and acceptable. Also, the performance of the trajectory tracking is presented and there is a good matching between the desired and actual tracking and that indicates the BSC-T1FLC-PSO controller is effective and its able to track the FMWMR for any path. For future work, it is good for making a comparison with another hybrid controller such as BSC-T1FLC.

References

[1] G. Klancar, A. Zdesar, S. Blazic , I. Škrjanc, Wheeled mobile robotics from fundamentals towards autonomous systems, 2017.

[2] S. G. Tzafestas, Introduction to mobile robot control, First edition, Elsevier, 2014.

[3] R. Siegwart , I. R. Nourbakhsh , Introduction to autonomous mobile robots, Cambridge, 2nd edition, MIT Press, UK, 2004.

[4] Q. Cui, X. Li, X. Wang, M. Zhang, Backstepping control design on the dynamics of Omni-directional Mobil Robot, App. Mecha. Mater., 203 (2012) 51-56. https://doi.org/10.4028/www.scientific.net/AMM.203.51

[5] K. E. Dagher , A. Al-Araji, Design of a nonlinear PID neural trajectory tracking controller for mobile robot based on optimization algorithm, J. Eng. Technol., 32 (2014) 973-985. https://doi.org/10.30684/etj.32.4A.13

[6] E. Maulama, M. A. Muslim , V. Hendrayawan, Inverse Kinematic Implementation of four-wheels Mecanum drive mobile robot using stepper motors, Int. Semin. intel. Technol. Appl., (2015) 51-54. https://doi.org/10.1109/ISITIA.2015.7219952

[7] I. K. Ibraheem , G. A. Ibraheem, Motion control of an autonomous mobile robot using modified particle swarm optimization based fractional order PID controller, Eng. Technol. J., 34 (2016) 2406-2419. https://doi.org/10.30684/etj.34.13A.4

[8] Z. Gao, Y. Yang, Y. Du, Y. Zhang, Z. Wang, W. Xu, Kinematic modeling and trajectory tracking control of a wheeled Omni-directional mobile logistics platform, DEStech. trans. eng. technol. Res., 198 (2017)169-175. https://doi.org/10.12783/dtetr/apetc2017/10906

[9] S. Morales , C. Delgado, LQR trajectory tracking control of an omnidirectional wheeled mobile robot, IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), 2018.

[10] C. Wang, X. Liu, X. Yang, F.Hu, A. Jiang , C. Yang, Trajectory tracking of an Omni-directional wheeled mobile robot using a model predictive control strategy, Appl. Sci. (2018) 1-15. http://dx.doi.org/10.3390/app8020231

[11] A. N. Amudhan, P. Sakthivel, A. P. Sudheer , T. K. S. Kumar, Design of controllers for omnidirectional robot based on the system identification technique for trajectory tracking, J. Phys. Conf. Ser., 1240 (2019) 012146. http://dx.doi.org/10.1088/1742-6596/1240/1/012146

[12] R. S. Burns, Advanced control engineering, Boston, Butterworth Heinemann, 2001.

[13] D. Wang, D. Tan , L. Liu, Particle swarm optimization algorithm: an overview, Soft Compute, 22 (2018) 387-408. https://doi.org/10.1007/s00500-016-2474-6

[14] I. Zeidis , K. Zimmermann, Dynamics of a four-wheeled mobile robot with Mecanum wheels, J. Appl. Math. Mech., 3 (2019) 1-22. https://doi.org/10.1002/zamm.201900173