Journal of Mechanical Engineering Research and Developments
ISSN: 1024-1752
CODEN: JERDFO
Vol. 43, No.5, pp.400-414
Published Year 2020

Design of Hybrid Controller for the Trajectory Tracking of Wheeled Mobile Robot with Mecanum Wheels

Sameh F. Hasan*, Hassan M. Alwan

Department of Mechanical Engineering, University of Technology, Iraq

*Corresponding Author Email: 21145@student.uotechnology.edu.iq

ABSTRACT: A Mecanum wheels mobile robot (MWMR) is consider being the most famous holonomic type of the wheeled mobile robot (WMR). This robot is getting importance due to its applications in industries, healthcare as well as in other sectors of service. The objective of this work is to design a novel hybrid controller for the trajectory tracking of four mecanum wheeled mobile robot i.e.,(FMWMR). First, the kinematic and dynamic models have been derived with tacking into account the effect of friction in the dynamic model. The new hybrid control consists from Backstepping-Type2 fuzzy logic control- social spider optimization (BSC-T2FLC-SSO). The backstepping controller (BSC) is used for computing the torque that exerted from each motor while type-2 fuzzy logic control (T2FLC) and social spider optimization (SSO) are used for computing the gains parameters of BSC. MATLAB programing has been used for simulate the equations and for presenting the results of the torques and the results of positioning error in (x,y) direction as well as orientation error. Circular trajectory tracking has been selected to test the performance of the new hybrid controller. A comparative study between the results that obtained from the new hybrid controller and the results from backstepping controller has been done. It was found that the magnitudes of the errors in position and orientation of the robot from the new hybrid controller was less than the magnitudes of the error in position and orientation that obtained from backstepping controller and this comparative results showed the effective and robust of BSC-T2FLC-SSO.

KEYWORDS: Mecanum wheels robot, Backsteeping controller, Fuzzy logic, Social spider optimization.

INTRODUCTION

Mobile robots can be defined as automatic machines that can developed many activates such us navigation, mapping as well as moving in determined space [1]. The classifications of the (WMR) depending on the type, number and location of the wheels on the chassis of robot [2]. The most commonly WMR configurations are differential drive, omnidirectional and bicycle drive. Omnidirectional robots characterized by high mobility as well as maneuverability. Also omnidirectional robots characterized by their lateral movement that the nonholonomic WMR cannot be achieved it [3]. There are many researches that studied and analysis the models and tracking of the omnidirectional robots. In [4], the authors used backstepping controller with particle swarm optimization for tracking control of non-holonomic WMR. They used different trajectories for testing the proposed controller. In [5], LQR controller was used for trajectory tracking of FMWMR. The (R and Q) are selected from experimental work.In [6], BSC was implemented for the trajectory of FMWMR and the magnitudes of gain parameters are chosen by tray and error.

The authors selected circular path for testing the performance of the robot. In [7], the authors adopted PID- Type 1Fuzzy logic control based on kinematic model of FMWMR while in [8], robust adaptive sliding mode controller for the FMWMR was presented. In [9], PID and LQR controllers are applied for the tracking of FMWMR. In [10], the kinematics and typical actuator dynamics of a nonholonomic differential drive WMR are derived. Two controllers which are the outer controller (kinematic controller) and the inner loop controller are investigated. In[11], The tracking problem of the robot was investigated by using a simple time-varying controller. In [12], linear feedback control was used with pole placement for the tracking control of non-holonomic WMR. The objective of this work is to design a novel hybrid controller which is consist of BSC-T2FLC-SSO for the trajectory tracking of FMWMR. MATLAB programing is used to simulate the equations as well as to present the results of the torques as well as positions and orientation errors. The results from this new hybrid controller compared with

the results that obtained from backstepping controller. The results showed the effective and accuracy of this new hybrid controller.

KINEMATIC MODEL

Kinematics represents the motion of the objects without taking into account the forces that affected on them. Omnidirectional WMR is consider to be holonomic robot with 3 degree of freedom in the plane motion. The main purpose of using mecanum wheels is make the robot from moving in lateral or sideways movement. Each wheel having set of passive roller which is angled 45° about the circumference of the hub. The mecanum wheel is fixed on the robot platform with local frame {R} and the parameters that are related to the robot can be defined as: (D) is a point represents the centroid of the robot, (A_i) is a point represents the center of the i-wheel, (δ) is the angle between the X_R and the point A, (ξ):- The angle between the wheel axis and the vector DA, (r):- the radius of the wheel, (Ω) is the angle between wheel plane and roller rotation axis and ( ) Rotation speed of the wheel and this can be show in figure (1) as below:

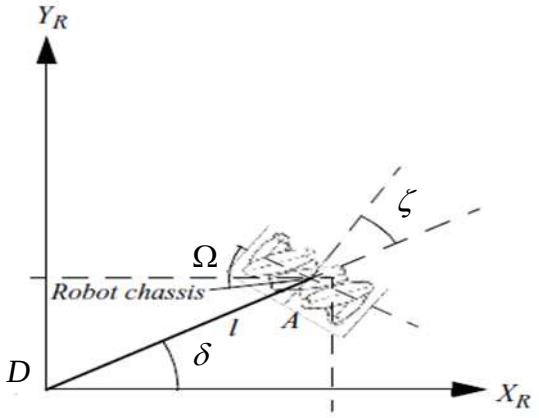


Figure 1. Mecanum wheel [1].

It is assumed that the kinematic model is derived for pure rolling i.e., (without slipping) condition. It can see that the velocity of point (A_i) is equal to ($r_i \dot{\Psi}_i$) from figure (1) that is shown above while its magnitude equal to ($r\dot{\psi}_i \cos\Omega_i$) along roller axis. The translation velocities of the robot with respect to frame {R} are [$\dot{x}_R \ \dot{y}_R$] while ($\dot{\theta}$) is the rotational speed about (z_R). The velocity of wheel center (A_i) with respect to roller contact axis direction can be represented as:-

$$V_{A_i} = \dot{x}_R \cos\left[\frac{\pi}{2} - (\frac{\pi}{2} - (\delta + \xi) + (\frac{\pi}{2} - \Omega)\right] + \dot{y}_R \cos\left[(\frac{\pi}{2} - (\delta + \xi) + (\frac{\pi}{2} - \Omega)\right] + l\dot{\theta}\cos\left[\delta + \left[(\frac{\pi}{2} - (\delta + \xi) + (\frac{\pi}{2} - \Omega)\right]\right] \quad \text{Equation (1)}$$

After some forward substitution equation (1) will become as below:

$$[\sin(\delta_i + \zeta_i + \Omega_i) \quad -\cos(\delta_i + \zeta_i + \Omega_i) \quad -l\cos(\zeta_i + \Omega_i)] \bullet {}^{R}\boldsymbol{R}_{O}(\theta) \bullet \dot{P}_O = r_i \dot{\psi}_i \cos\Omega_i \quad \text{Equation (2)}$$

Where i=1,2,3,4 and ($\dot{P}_O$) is the velocity of the robot in the global coordinate and ${}^{R}R_{o}(\theta)$ is the rotation matrix. The robot velocity in global coordinate {O}as well as in the local coordinate {R} is described as:

$$\dot{P}_O = \begin{bmatrix} \dot{x}_O \\ \dot{y}_O \\ \dot{\theta}_O \end{bmatrix} ; \dot{P}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} \quad . \qquad \text{Equation (3)}$$

It can convert the velocity representation from global to local frame as below:

$$\dot{P}_R = {}^{R}\boldsymbol{R}_O(\theta) \bullet \dot{P}_O \qquad \text{Equation (4).}$$

The rotation matrix can be expressed as:

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

The FMWMR is shown in figure (2) below:

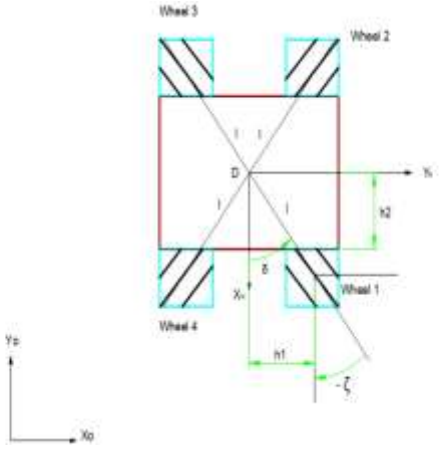


Figure 2. WMR with four mecanum wheels.

Equation (2) can be applied for each wheel with tacking into account that the radius of each is equal as well as the distance from each wheel to point (D) as show in figure (2) are equal so that, r_i=r. After some forward calculations it can obtain the equation of inverse kinematics as:

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

Where: $a_{11} = a_{12} = a_{21} = a_{42} = 1$, $a_{22} = a_{31} = a_{32} = a_{41} = -1$, $a_{22} = a_{31} = a_{32} = a_{41} = -1$, $a_{14} = a_{24} = a_{34} = a_{44} = \sqrt{2}\, l \sin(\frac{\pi}{4} - \delta)$
Where $\delta = \tan^{-1}\left(\frac{h1}{h2}\right)$. The Jacobian matrix (H) can be defined as:

$$H = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \\ a41 & a42 & a43 \end{bmatrix} \bullet \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{Equation (6)}$$

The equation of forward kinematics can be obtained as below:

$$\begin{bmatrix} \dot{x}_O \\ \dot{y}_O \\ \dot{\theta}_O \end{bmatrix} = -\left(\frac{\sqrt{2}}{2}\right) r H^{+} \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} \quad \text{Equation (7)}$$

Where $H^{+}=(H^{T}H)^{-1} H^{T}$ is the pseudo inverse of H.

DYNAMIC MODEL

The dynamic model of the robot that shown in figure (2) is derived by Lagrange equation. It is consider that there is an offset between the centroid of the robot i.e., (point D) and the center of gravity i.e. (point D/). The velocities of points (D) and (D/) relative to local frame {R} are determined as below:

$$R v_D = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_O \\ \dot{y}_O \end{bmatrix} = \begin{bmatrix} \dot{x}_O \cos\theta \\ -\dot{y}_O \sin\theta \end{bmatrix}; \; R v_{D'} = R v_D + \dot{\theta} k_R \times r_{D'/D} \quad \text{Equation (8)}$$

$$R v_{D'} = \left(\dot{x}_O \cos\theta + y_O \sin\theta + \dot{\theta} d_2\right) i_R + \left(-\dot{x}_O \sin\theta + y_O \cos\theta - \dot{\theta} d_1\right) j_R \quad \text{Equation (9)}$$

Where (d_1)and (d_2)are the offset distances between D and D/. The kinetic energy (T) of the robot is computed as:

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

m_b: platform mass. m_w:wheel mass. I_b: platform moment of inertia, I_ψ: wheel moment of inertia,
$\dot{\psi}$: wheel rotational speed. The Lagrangian (L) can be expressed as L=T-U. (U) is the potential energy and its equal to zero because the robot is consider to be a plane motion and that lets to make (L=T) . It is assumed that $I_{w1}= I_{w2}= I_{w3}= I_{w4}=I_w$. The Lagrange equation can be expressed as:

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

i=1,2,3. (q_i) and (F_i) are represented the generalized coordinate and generalized torque/force respectively and can be represented as:
The $q = [q_1 \quad q_2 \quad q_3]^T = [x \quad y \quad \theta]^T$ generalized (torque/force) evaluated as :

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

(τ_i) is the torque of each wheel and (τ_{fi}) is the frictional torque. Both frictional torques and forces evaluated as below:

$$\tau_{fi} = f_i * r \quad \text{Equation (13)}$$

$$f_{si} = \mu (m_b + 4m_w) * g \quad \text{Equation (14)}$$

Where (μ) is the coefficient of friction and (g) is the gravitational acceleration. The frictional forces can be written as:

$$f_i = (f_{si} \operatorname{sgn}(\dot{\psi}_i)) \quad \text{Equation (15)}$$

The parameter ($\frac{\partial \dot{\psi}_i}{\partial q_i}$) that is found in equation (12) is evaluated from equation (5). The dynamic equation of motion of the robot expressed as:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+B^T S f=\frac{1}{r}B^T \tau \qquad \text{Equation (16)}$$

Where:- [M] is the inertia matrix, [C] is the Coriolis and centripetal matrix, [f] is the frictional matrix,[B] is input transformation matrix. After some forward calculations, the representation of each matrix is shown below:

$$\tau=\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix} \quad ; \quad S=\begin{bmatrix}\text{sgn}(\dot{\psi}_1) & 0 & 0 & 0\\ 0 & \text{sgn}(\dot{\psi}_2) & 0 & 0\\ 0 & 0 & \text{sgn}(\dot{\psi}_3) & 0\\ 0 & 0 & 0 & \text{sgn}(\dot{\psi}_4)\end{bmatrix}; \quad f=\begin{bmatrix}f_1\\ f_2\\ f_3\\ f_4\end{bmatrix}$$

$$C=\begin{bmatrix}0 & 0 & m_b\dot{\theta}(d_1\cos\theta-d_2\sin\theta)\\ 0 & 0 & m_b\dot{\theta}(d_1\sin\theta+d_2\cos\theta)\\ 0 & 0 & 0\end{bmatrix}; M=\begin{bmatrix}m_{11} & m_{12} & m_{13}\\ m_{21} & m_{22} & m_{23}\\ m_{31} & m_{32} & m_{33}\end{bmatrix}$$

Where:

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

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

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

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

CONTROLLER DESIGN

In this paper, a new hybrid controller consist of BSC-T2FLC-SSO is implemented. The backstepping control is applied on the nonlinear acceleration equation to evaluate the controlled torques (τ) while T2FLC and SSO are used to evaluate the optimal gains parameters.

Backstepping Controller

Recall to equation (16) it can make the acceleration in one side and the other parameters on the other as shown below:-

$$\ddot{q}=M^{-1}\left[\frac{1}{r}B^T\tau-C\dot{q}-B^T S f\right] \qquad \text{Equation (17)}$$

The state vector of generalized coordinate can be written as below:

$$X=\begin{bmatrix}X_1^T & X_2^T\end{bmatrix}^T=\begin{bmatrix}q^T & \dot{q}^T\end{bmatrix}^T=\begin{bmatrix}x & y & \theta & \dot{x} & \dot{y} & \dot{\theta}\end{bmatrix}^T$$

Based on backstepping controller, It is considered that $(\dot{X}_1)$ is to be a subsystem and let $(\dot{X}_1)$ is equal to (u_1) where (u_1) is represent the virtual input. The equation of the tracking error can be written as:

$$e_1 = X_1 - q_d = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_O \\ y_O \\ \theta \end{bmatrix} - \begin{bmatrix} x_{O,d} \\ y_{O,d} \\ \theta_d \end{bmatrix}$$ Equation (18).

The $q_d(t)=[\ x_{0,d}(t)\ ,\ y_{O,d}(t)\ ,\ \theta_d]^T$ is the desired tracking of the robot. The differentiating of equation (18) can be represented as:

$\dot{e}_1 = \dot{X}_1 - \dot{q}_d = u_1 - \dot{q}_d$ Equation (19). Now the Lyapunov function is considering for checking the stability as:

$$V_1 = \frac{1}{2} e_1^T K_1 e_1$$ Equation (19)

Where $K_1 \in R^{3\times3}$ symmetric and positive. The equation (20) is derivative with respect of time and it can obtain:

$$\dot{V}_1 = \frac{1}{2}\dot{e}_1^T K_1 e_1 + \frac{1}{2} e_1^T K_1 \dot{e}_1 = e_1^T K_1 \dot{e}_1 = e_1^T K_1 (u_1 - \dot{q}_d)$$ Equation (20)

For making the system stable, it can choose $u_1 = \dot{q}_d - e_1$ and it can obtain:

$$\dot{V}_1 = -e_1^T K_1 e_1 \le 0$$ Equation (21)

It can see from equation (22) that the system asymptotically stable. Anew error must be introduced as:

$e_2 = X_2 - u_1$ Equation (23). The derivative of equation (23) can be written as below:

$\dot{e}_1 = \dot{X}_1 - \dot{q}_d = X_2 - \dot{q}_d + u_1 - u_1 = e_2 - e_1$ Equation (24)

$$\dot{e}_2 = M^{-1}\left[\frac{1}{r} B^T \tau - C\dot{q} - B^T S f\right] - \ddot{q}_d + (e_2 - e_1)$$ Equation (25)

After that, Lyapunov function is used for checking the stability and that can see as below:

$$V_2 = V_1 + \frac{1}{2} e_2^T K_2 e_2 = \frac{1}{2} e_1^T K_1 e_1 + \frac{1}{2} e_2^T K_2 e_2$$ Equation (26)

$K_2 \in R^{3\times3}$ is positive definite and symmetric.

The derivative of equation (26) becomes as below:

$$\dot{V}_2 = e_1^T K_1 \dot{e}_1 + e_2^T K_2 \dot{e}_2 = e_1^T K_1 (e_2 - e_1) + e_2^T K_2 \left[(e_2 - e_1) + M^{-1}(\frac{1}{r} B^T \tau - C\dot{q} - B^T S f) - \ddot{q}\right]$$ *Equation* (27)

The computed torque equation can be represented as:

$$\tau = rB(B^T B)^{-1} M\left[\left[M^{-1}\left[-C\dot{q} - B^T S f\right] - \ddot{q} + (e_2 - e_1) - e_2 - K_2^{-1} K_1 e_1\right]\right]$$ Equation (28)

After substitute equation (28) into equation (27) it can be obtained:

$$\dot{V}_2 = -e_1^T K_1 e_1 - e_2^T K_2 e_2 \le 0$$ Equation (29)

It can see from equation (29) that the system is negatively definite and that lead the system to be asymptotically stable. The magnitudes of $\mathbf{K_1}$ and $\mathbf{K_2}$ are computed by using Type2- Fuzzy logic i.e.(T2FLC) as well as social spider optimization i.e.(SSO) and that will be represented in the next sections.

Type-2 Fuzzy Logic Controller (T2FLC)

In this work, A Type-2 Fuzzy Logic i.e. (T2FLC) is adopted to compute the parameters of the controller gain (**K1**) i.e. (K_1,K_2,K_3). A T2FLC has been applied in many control process because of its capability for uncertainty handling. The T2FLC is introduce as an extension theorem of type-1 fuzzy control i.e.(T1FLC) by Zadeh in 1975[13]. The T2FLC can be discriminative by membership functions(MFs) in which the (MFs) grade is a set between [0,1]. The (MFs) of (T2FLC) are 3-dimienstions as well as a foot point of uncertainty (FOU) is included. An extra degree of freedom was provided by (FOU) for handling the uncertainty.
The Simulink diagram of the torque controller is shown as:

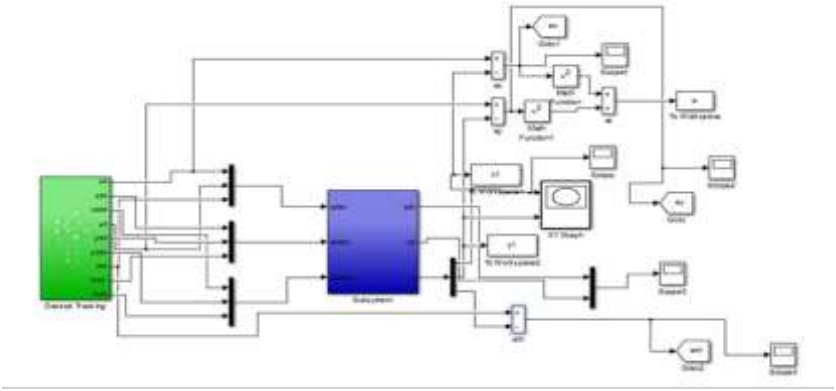


Figure 3. Simulink diagram of trajectory tracking

It can be seen from figure (3) that is maintained above there are two subsystems. The subsystem that is in green color including the equation of the desired tracking. While the subsystem in the blue color including the simulation and the mathematical models of the T2FLC, BSC as well as the dynamic equation and that can be show as below:

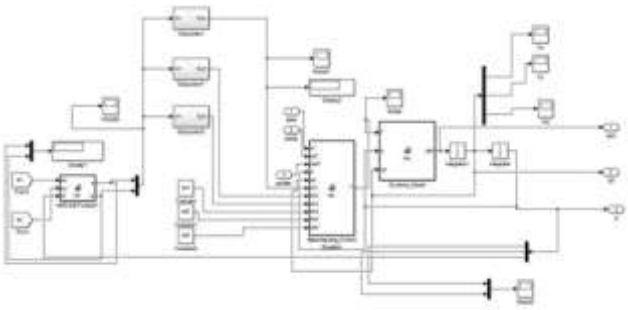


Figure 4. Simulink diagram of the hybrid controller.

Each subsystem in the gray color includes two FLC blocks. One block for representing inner (MFs) and the other for the outer (MFs). The output from each block is combined and multiplied by a scale factor equal to (0.5) to achieve a "reducer-type" that is relating to T2FLC and that can be seen as below:

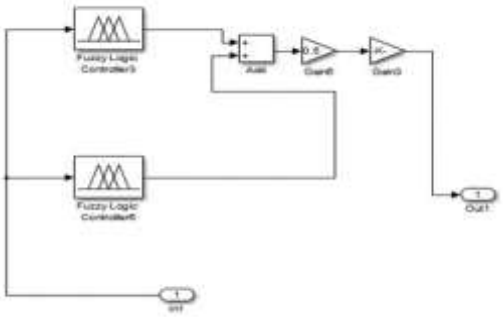


Figure 5. Type-2 FLC representation in MATLAB.

In this work, there are two inputs delivered to T2FLC which are (L) and (α). (L) is definined as the distance from the actual to reference position while (α) defined as the deviation angle between the actual and the reference position of the robot and that can see below:

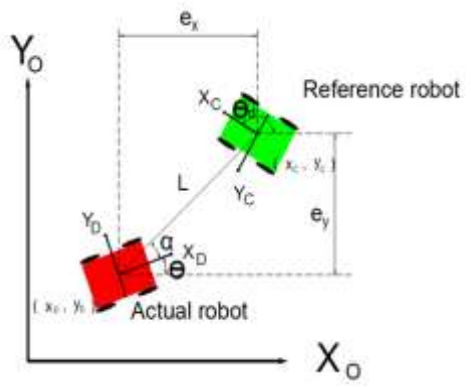


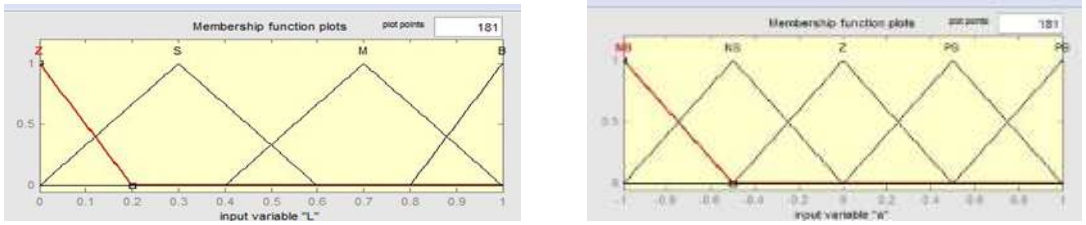
Figure 6. Robot configuration with (L) and (α) parameters.

The values of (L) and (α) can be determined from the following formula as below:

$$\alpha = \tan^{-1}(e_y / e_x) \qquad \text{Equation (30)}$$

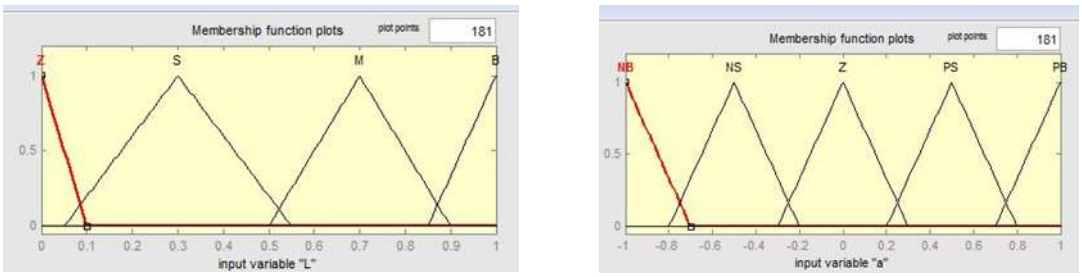
$$L = \sqrt{(e_x)^2 + (e_y)^2} \qquad \text{Equation (31)}$$

The first input (L) has 4 triangle (MFs) for inner and outer (MFs) which are: Zero(Z), small (S), Medium(M) and big(B) while the second input (α) has 5 triangle (MFs) for inner and outer (MFs) which are: negative big (NB), negative small (NS), zero(Z), positive small (PS) and positive big (PB). The (MFs) for each input can be seen as below:



(a) (b)

Figure 7. Outer (MFs) (a) for the input L and (b) for the input (α).



(a) (b)

Figure 8. Inner (MFs) (a) for the input L and (b) for the input (α).

The output (K) has 3triangle (MFs) which are: small(S), medium(M) and big(B) and they illustrated as below:

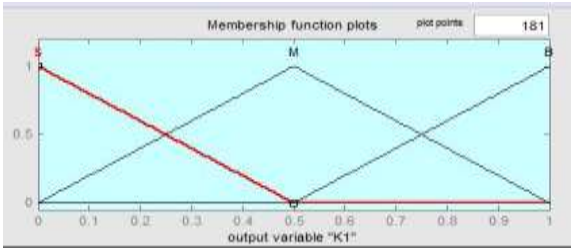


Figure 9. MFs of the controller gain parameter.

The rules that are related to each output i.e. (gains) can be seen as below in tables 1,2,3:

Table 1. Rules of the gain parameter (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 2. Rules of the gain parameter (K_2).

L \ α	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 3. Rules of the gain parameter (K_3).

L \ α	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 formula that used to represent the strategy of the defuzzification is the centroid of area (COA) which is computed as:

$$COA = \frac{\int_z \mu_A(z).z\,d(z)}{\int_z \mu_A(z)\,d(z)} \qquad \text{Equation (32)}$$

Where (Z) is the quantity of output and μ_A (z) is the compiled MFs output. Now, the gain parameters (K_1,K_2,K_3) are computed by using T2FLC but there values are normalized i.e.(there magnitudes are between [0-1]) so that , the output from each T2FLC are multiplied by a scale factor which is represented by a gain in MATLAB/ Simulink and this think is referred in figure (5). The optimum vale of each scale factor is as well as the other gains magnitudes i.e.(K_4,K_5,K_6) are computed by using SSO and that will be illustrated in the next section.

Social Spider Optimization (SSO)

In this work, the SSO is adopted for computing the gains parameters of BSC which are (K_4,K_5,K_6) as well as to optimizing the outputs values from T2FLC. The SSO approach depending on the collaborative conduct of the spiders. The SSO consist of two different spiders which are: Female as well as Male. Relay on gender, the behavior of each individual is controlled by evolutionary methods which are simulate the behavior of the colony and that

consider the main advantages of SSO over other optimizations algorithms such us particle swarms (PSO) and artificial bee colony (ABC) which is prevent SSO from local minima and premature convergence. In this algorithm, the number of female spiders (N_f) are between (65-90%) from the total number (N). The equation that used for evaluating (N_f) is given as[14]:

$$N_f = floor\ [(0.9 - rand\ 0.25).N] \quad \text{Equation (33)}$$

Where (rand) is a random number between $[0,1]$ and floor is mapping from real to integer number.

The spiders male number computed as: $N_m = N - N_f$. The equation that used for evaluating the weight of each spider is written as:

$$w_i = \frac{J(s_i) - worst_s}{best_s - worst_s} \quad \text{Equation (34)}$$

Where $J(s_i)$ can be defined as the value of the fitness which obtained from the position of the spider (s_i) with respect to the objective function (J_f). The range (R) of the mating operation in the SSO can be defined as the radius that is relay on the search space size and it is evaluating as below:

$$r = \frac{\sum_{j=1}^{n} (p_j^{high} - p_j^{low})}{2n} \quad \text{Equation (35)}$$

Where (P_j^{high}) and (P_j^{low}) are the lower and upper bond of the initial parameters.

RESULTS AND DISCUSSIONS

To demonstrate the effective and performance of the new hybrid controller design for the FMWMR, MATLAB programing has been used to simulate the equations and review the results. In this work, a comparative study has been done between the results that obtained from the new hybrid controller i.e.BSC-T2FLC-SSO and the results of back-stepping controller. The parametric values of the robot can be seen in table (6) and are taken from [15] as below:

Table 4. Robot Parameters [15].

Parameters	Value (unit)
Mass of robot platform (m_b)	3.1 (Kg)
Mass of each wheel (m_w)	0.35 (Kg)
Mass moment of inertia of the platform (I_b)	0.032 (Kg m^2)
Mass moment of inertia of each wheel (I_w)	$6.25*10^{-4}$ (Kg m^2)
Radius of each wheel (r)	0.05
The distance (h_1) and (h_2)	0.15 (m)
Coefficient of friction (μ)	0.02
The offset distance (d_1 , d_2)	0.02 (m)
The distance (l)	0.25 (m)

In this simulation, a circular trajectory is considered for the FMWMR to be followed with robot constant angular velocity (W_d =0.2 rad/s)and the radius of the circle equal to (0.5) m. The equations that has been used for the desired tracking can be demonstrated as below:

$X_d = 0.5*\cos(\omega_d * t)$; $Y_d = 0.5*\sin(\omega_d * t)$.

The robot started from the initial position which is represented the center of the circle i.e.$q=[0,0,0]^T$. In this section, there are two cases are taking into account. The first case represents the results that obtained from BSC only while the second case represents the results that obtained from BSC-T2FLC-SSO and that can be seen as below:

Case One

In this case, it has been consider only the results that obtained from backstepping controller and that represented as below:

Equation (28) that is maintained earlier is programing and simulated in MATLAB/Simulink. The values of the gains i.e.(K_1 and K_2) are chosen by trial and error and the values of parameters equal to:

$$K_1 = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 100 \end{bmatrix} \quad K_2 = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 35 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$ and the results are obtained as below:

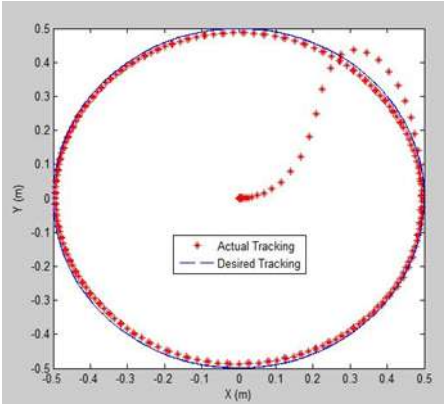


Figure 10. Circular tracking.

From figure (10) that is maintained above , it can see that the tracking performance is good by means of the proposed controller but there is some deviations from the desired tack due to the error between the desired and actual tracking.

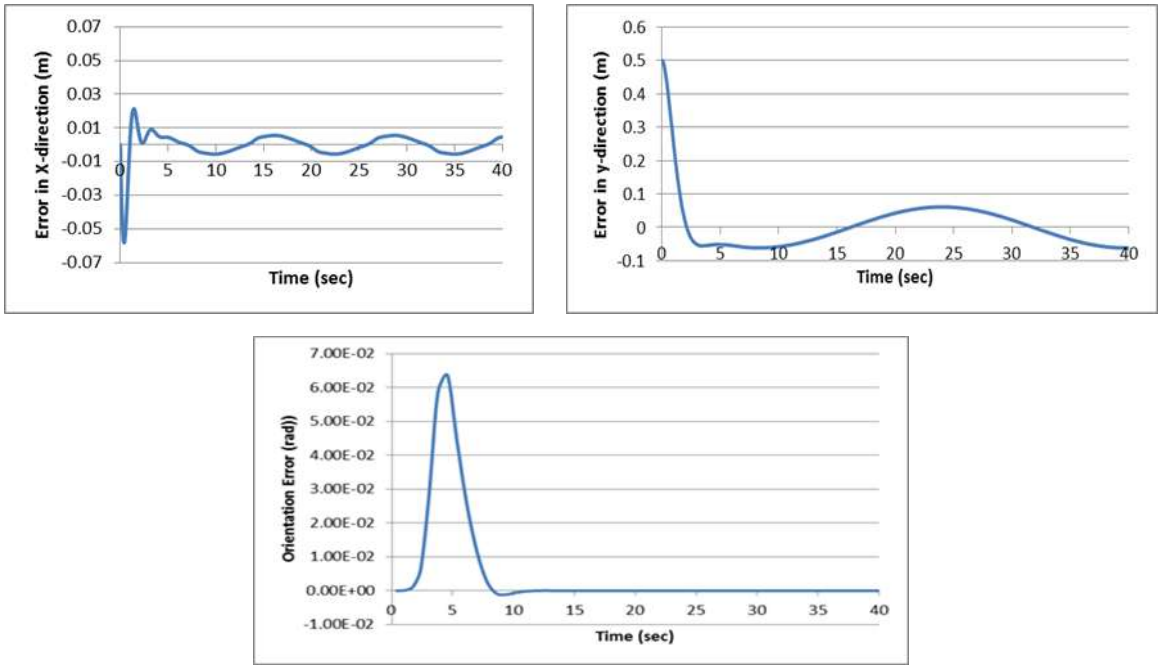


Figure 11. (a) Error in x-direction, (b) Error in y-direction (c) Orientation error.

Figure (11) shows the errors of the trajectory tracking for the position and orientation of the FMWMR. It can see that the magnitude of the error in x-direction i.e.(e_x) is beginning from value (-0.06) m and after 5 seconds, the magnitudes of the error became fluctuating between (0.1) and (-0.1) m and that continues until the final simulation time i.e.,(40 second) while the magnitude of the error in y-direction i.e.(e_y) is beginning from (0.5) m and reduced after 5 seconds to (-0.035) m and from 5 to 40 seconds the magnitudes of the error are change between (-0.035to

0.058) m and it is reached to (-0.044) m at the final simulation time while the maximum value of the orientation error i.e.(e_θ) is equal to (0.062) rad/s at about the 5th second and reduced to (-0.009) rad/s at about the 9th second and from the (10-40) seconds the magnitude of the error is approximately equal to zero.

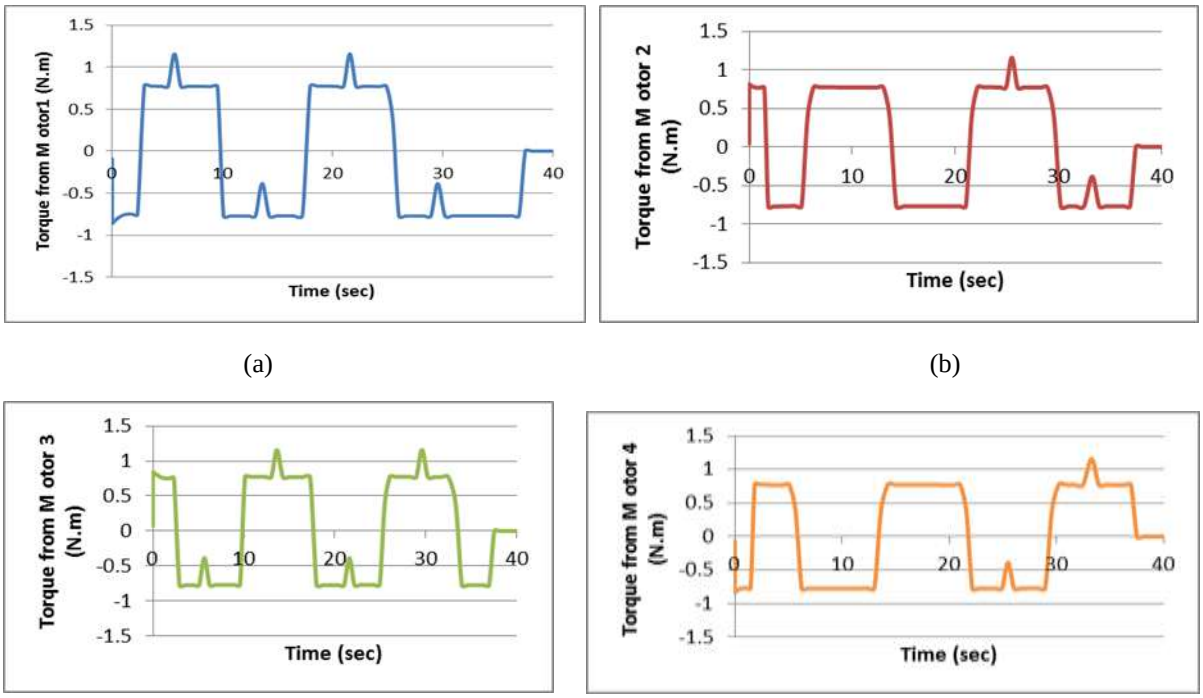


Figure 12. Magnitudes of the torque generated from each motor.

Figure (12) represents the torques that is generated from the FMWMR motors with some sharp spikes. The values of the torques are occur between (1.2 to -0.72) N.m and this value are consider to be somewhat large and it is consider to be undesirable.

Case Two:

In this case, it has been considering the results that obtained from BSC-T2FLC-SSO. The values of the parameters gain ($\mathbf{K_1}$) that obtained from T2FLC which are (k_1,k_2,k_3) which are optimized by SSO with a range (0-200) are

$$\mathbf{K_1}=\begin{bmatrix} 28.120 & 0 & 0 \\ 0 & 32.3758 & 0 \\ 0 & 0 & 13.815 \end{bmatrix}.$$

While the values of the values of the parameters are (c) parameters gain ($\mathbf{K_2}$) which are (k_4,k_5,k_6) that obtained from SSO. The initial given in the range (0-100) while the optimum values (d) that obtained from SSO are

$$\mathbf{K_2}=\begin{bmatrix} 23.428 & 0 & 0 \\ 0 & 12.218 & 0 \\ 0 & 0 & 5.079 \end{bmatrix}.$$

The numbers of spiders that are selected are (20) and the number of iterations that tacking into account are (50). The objective function that is used in SSO can be expressed as:

$$J_f = \sqrt{e_x^{\,2} + e_y^{\,2}}$$

Equation (36)

The results that obtained from this hybrid controller are shown as below:

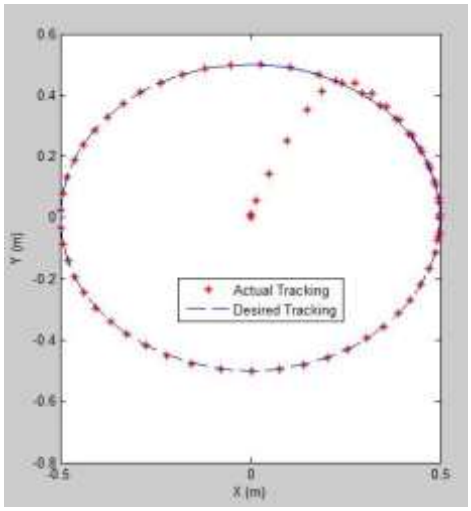


Figure 13. Circular tracking.

It can see from figure (13) that is maintained above the actual tracking of FMWMR is approximately coincide with the desired tracking and that indicates that the performance of the hybrid controller is very good and better than the performance of back-stepping controller.

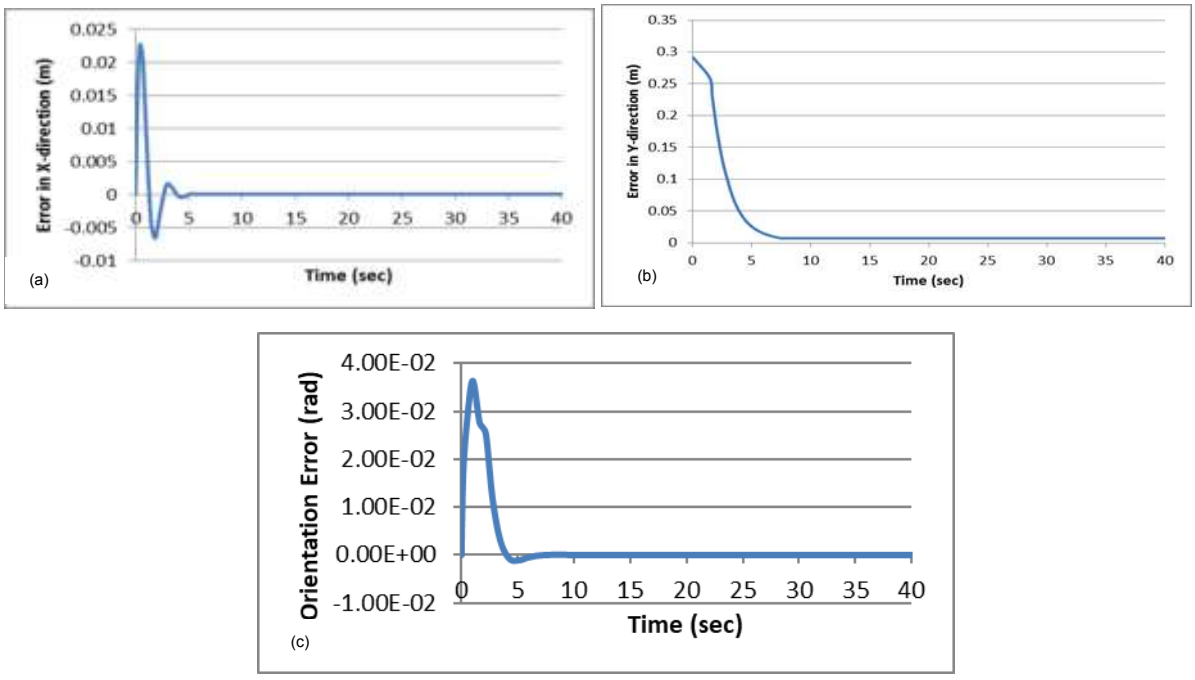


Figure 14. (a) Error in x-direction, (b) Error in y-direction (c) Orientation error.

From Figure (14) it can see that the values of the errors until about 5 seconds are varying with time but with small relatively values because the actual initial position of the robot is at $q=[0\ 0\ 0]^T$ and this (5 seconds) represent the time that the robot requires for reaching the circle and at the period between (5-10) seconds the values of the errors are reducible to a value so closed from zero i.e.,($1.2*10^{-4}$ m) and remain at this value until the end of the simulation i.e.,(40 seconds) while in the case of back-stepping controller, the errors values are fluctuating between some values which are considered to be larger than the error values of the hybrid controller and that indicates that the new hybrid controller gives very little errors as compared with errors of back-stepping controller.

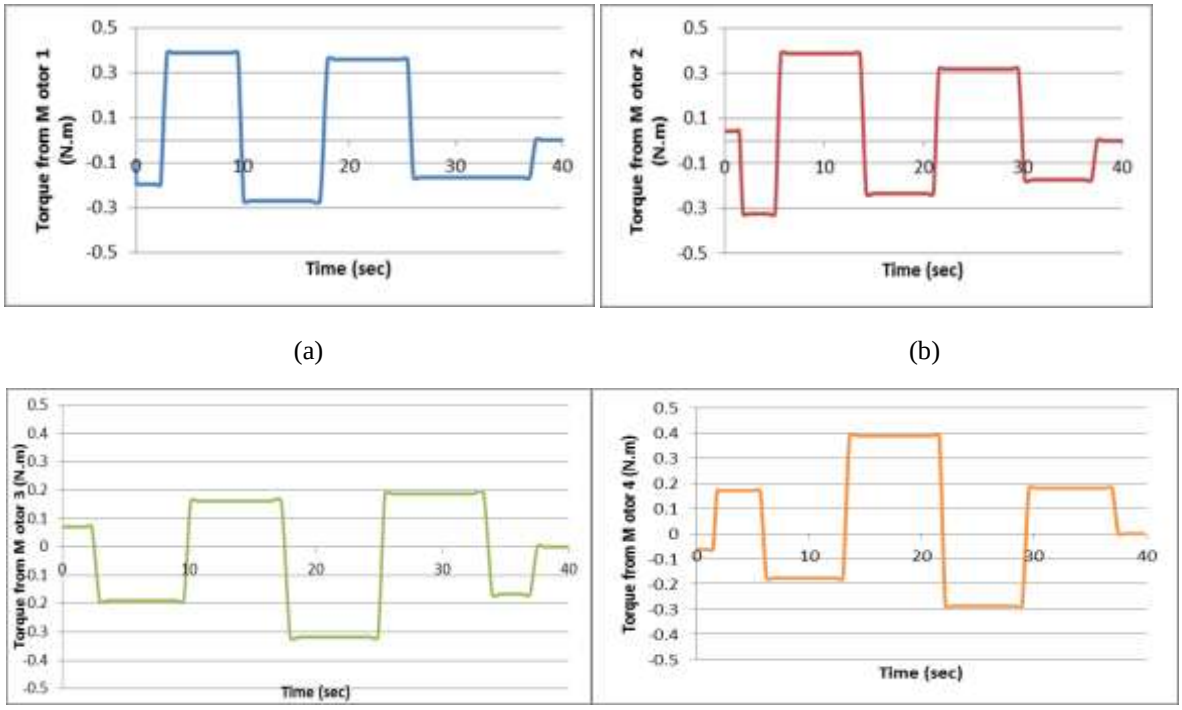


Figure 15. Magnitudes of the torque generated from each motor.

From figure (15) it can show that the behavior of the torque that generated from each motor is smooth without shark spikes with magnitudes between (0.4 to -0.32) N.m and that indicate the behavior of the torques are better than the behavior that obtained from back-stepping controller as well as the magnitudes of the torques that obtained from this new hybrid controller are less than the magnitudes that results from back-stepping.

CONCLUSIONS

The study of trajectory tracking of WMR is considered an important and interesting topic In this work a new hybrid controller consist of backstepping-Type 2 fuzzy logic control- social spider optimization i.e.,(BSC-T2FLC-SSO) is proposed for the trajectory tracking of FMWMR. First both kinematics and dynamic models have been derived with tacking into account the effect of friction in the dynamic model. Backstepping controller has been applied for the nonlinear acceleration equation to compute the magnitudes of the torque that are generated from each motor of the robot. T2FLC has been used to compute the magnitudes the parameters of gain **(K_1)** but the output from fuzzy logic will be in normalized values i.e.,(the value is between [0-1]) so that, SSO has been used to de-normalizing the output values from T2FLC as well as to compute the optimum values of another gain parameter which is (**K_2**). MATLAB programing has been used to simulate the equations and for obtaining the results. A comparison study between the new hybrid controller and backstepping controller has been adopted and the results show that this new hybrid controller i.e.,(BSC-T2FLC-SSO) gives results better than the results that obtained from backstepping controller because this hybrid controller minimized the tracking errors of the FMWMR to a small values which are about ($1.2*10^{-4}$) m as well as reducing the magnitudes of the torques that generated from each motor which are between (-0.4 to 0.4) N.m and that lead to enhancement the performance and stability of the robot.

REFERENCES

[1] R. Siegwart, I.R. Nourbakhsh, "Introduction to Autonomous Mobile Robots", Cambridge, 2nd edition, MIT Press, UK, 2004.

[2] G. Klancar, A. Zdesar, S. Blazic and I. Škrjanc, "Wheeled Mobile Robotics from Fundamentals Towards Autonomous Systems", Oxford, Elsevier, 2017.

[3] S.G. Tzafestas, "Introduction to Mobile Robot Control", First edition, Elsevier, 2014.

[4] K.K. Younus and N.H. Hadi, "Optimum Path Tracking and Control for A Wheeled Mobile Robot (WMR)", Journal of Mechanics of Continua and Mathematical Sciences, Vol.15, Pp.73-95, 2020.

[5] S. Morales and C. Delgado, "LQR Trajectory Tracking Control of an Omnidirectional Wheeled Mobile Robot", IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), 2018.

[6] Z. Gao, Y. Yang, Y. Du, Y. Zhang, Z. Wang and W. Xu, "Kinematic Modeling and Trajectory Tracking Control of a Wheeled Omni-directional Mobile Logistics Platform", Asia-Pacific Engineering and Technology Conference, Pp. 169-175,2017.

[7] E. Malayjerdi, H. Kalani, and M. Malayjerdi, "Self-Tuning Fuzzy PID Control of a Four-Mecanum Wheel Omni-directional Mobile Platform", 26th Iranian Conference on Electrical Engineering, Pp. 816-820,2018.

[8] V. Alakshendra and S.S. Chiddarwar, "Design of robust adaptive controller for a four-wheel omnidirectional mobile rob"ot, International Conference on Advances in Computing, Communications and Informatics (ICACCI), Pp. 63-68, 2015.

[9] A.N. Amudhan, P. Sakthivel, A.P. Sudheer and T.K.S. Kumar, "Design of controllers for omnidirectional robot based on the system identification technique for trajectory tracking", IOP Conf. Series: Journal of Physics, Pp. 1-9, 2019.

[10] N. Leenaa, and K.K. Sajub, "Modelling and trajectory tracking of wheeled mobile robots". J. Procedia Technology. Vol. 24, Pp. 538 – 545, 2016.

[11] M. Maghenem, "Global tracking-stabilization control of mobile robots with parametric uncertainty". International Federation of Automatic Control, Vol. 50, Pp. 4114–4119, 2017.

[12] S. Nurmaini, "Differential-Drive Mobile Robot Control Design based-on Linear Feedback Control Law".IAES International Conference on Electrical Engineering, Computer Science and Informatics, S755. Pp. 1-7, 2017.

[13] Oscar Castillo and Patricia Melin, "Type-2 Fuzzy Logic: Theory and Applications", Springer, 2008.

[14] A. Luque-Chang, E. Cuevas, F. Fausto, D. Zald-var and M. Pérez, "Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives", International Journal of Engineering Mathematical, Hindawi ,Vol. 2, Pp.1-29, 2018.

[15] I. Zeidis and K. Zimmermann, "Dynamics of a four-wheeled mobile robot with Mecanum wheels", Journal of Applied Mathematics and Mechanics, Vol. 3, Pp. 1-22, 2019.

[16] R.M. Hussein," Effect of Friction on the Dynamical Analysis of Three-Link Planar Robot Arm by Using Lagrange Approach". Eng. & Tech. Journal, Vol. 35, Part A, No. 6, 2017.

[17] H.M. Alwan and S.F. Hasan," Evaluation of Friction Forces in the Joints of Gough-Stewart Manipulator". Eng. & Tech. Journal, Vol.34, Part (A), No.6,2016.