Enhancing Tilt-Integral-Derivative Controller to Motion Control of Holonomic Wheeled Mobile Robot by Using New Hybrid Approach

Sameh Fareed Hasan and Hassan Mohammed Alwan

Department of Mechanical Engineering, University of Technology, Iraq

Email: 21145@student.uotechnology.edu.iq

Abstract. Trajectory tracking of wheeled mobile robot is the most interesting and important subject in robotic systems field. In this paper, a four mecanum wheeled mobile robot (FMWMR) has been considered. This type was considered because it is the most famous holonomic type of wheeled mobile robot (WMR). The main purpose of this work is to design a new hybrid controller for the trajectory tracking of FMWMR based on the kinematic model. The proposed controller consists of Tilt-Integral-Derivative (TID) controller tuned parameters with neural network as well as social spider optimization i.e., (TID-NN-SSO) which is applied on the kinematic model of the FMWMR. The forward and inverse kinematic equations were derived. The TID controller was used for computing the controlled signals which are the angular velocities of each wheel while the neural network (NN) and social spider optimization (SSO) were to compute the parameters of TID controller. MATLAB / Simulink programing was implemented to simulate the results. A comparative study between TID-NN-SSO and TID controller tuned parameters by using particles swarm optimization (TID-PSO) was conducted. The results of the mean square errors (MSE) in (x and y) coordinates as well as the orientation error obtained from TID –NN-SSO are (2.105*10^{-4} m, 1.025*10^{-4} m, 0.0815 rad) and they are less than the MSE that was obtained from TID-PSO which are (0.00567 m, 0.0356 m, 1.22 rad/s). This indicates that TID –NN-SSO is more efficient than TID-PSO.

Keywords. Control, TID controller, neural network, SSO, PSO, kinematics.

1. Introduction

In the recent years, there has been great interest in research on the control of wheeled mobile robot (WMR). The main purpose of WMR motion control is to make the robot tracking stability within specified path [1]. Autonomous robot can be defined as the machine that is able to work with the help of artificial intelligence instead of human help. The WMR is classified as non-holonomic and holonomic depending on location, number as well as of the wheels on the robot chassis. The non-holonomic WMR has two degrees of freedom whereas the holonomic type has three degrees of freedom [2]. The four-mecanum-wheeled mobile robot (FMWMR) is considered a holonomic WMR. The FMWMR has high maneuver and mobility and it can be tracked with narrow spacing due to its lateral movement. These features are considered to be the main difference between the holonomic and non-holonomic WMR. There are several studies dealing with the motion control of WMR. In [3], a comparative study was conducted between linear quadratic regulator (LQR) controller and PI controller for tracking control of FMWMR. The magnitudes of the errors obtained from LQR were...
less than the errors from PI controller and that proves the efficiency of LQR controller. In [4], adaptive sliding mode controller (ASMC) was implemented based on dynamic model of FMWMR. The authors used the proposed controller for computing the controlled torques. They compared the results that were obtained from ASMC and the results from SMC and they concluded that the results from ASMC were better. In [5], back-stepping controller (BSC) was applied for tracking control of FMWMR. They tested the stability of the proposed controller by using Lyapunov theorem. The magnitudes of BSC gains parameters were selected by try and error i.e., (without tuning). In [6], type-1 fuzzy logic control (T1FLC) was used for making control on the kinematic model of FMWMR. In [7], a model predictive control was selected in order to control the speed of WMR with three-mecanum wheels. The author examined the performance of their proposed control by choosing line as well as square paths tracking. In [8], the kinematic model of FMWMR was studied. First, the kinematic model was derived theoretically and the authors used stepper motors for the experimental work and the results of the wheels angular speeds and the robot speeds were presented. The obtained results were good. In [9], PID-T1FLC controller was used for the trajectory control of FMWMR based on kinematic model. T1FLC was implemented for tuning the PID controller gains magnitudes. The authors used brushless DC motor with encoder connected with each mecanum wheel and used STM32F4 as the microcontroller in the experimental work. In [10], back-stepping controller was adopted to control the dynamic model of Nubot robot with mecanum wheels. The stability of the controller was tested by using Lyapunov method. In [11], fuzzy-PID control was presented to control the kinematic model of mecanum wheeled mobile robot. The fuzzy made adjustment on PID gains values while PID made control based on the robot wheels rotational speeds. A comparative study between PID and Fuzzy-PID controllers was made and the Fuzzy-PID gave the best results. In [12], T1FLC was proposed to control the position as well as the orientation of the FMWMR. Solid-works programing was used for drawing the robot and MATLAB/ Simulink was used to simulate the results. In this work, a new hybrid controller consisting of TID –NN-SSO is implemented to make motion control on FMWMR. TID controller is used in order to control the speed of the robot wheels while NN and SSO are used to compute the parameters of TID controller. A comparative study between this new hybrid controller and TID-PSO is implemented and the results indicate that the new hybrid controller gives better results.

2. Kinematic model of FMWMR
Holonomy can be defined as the feature of the WMR that constrains the movement of robot. It relies on the kind of the wheel. Differential drive robot is considered to be non-holonomic type while mecanum wheeled mobile robot is holonomic type. In this work, FMWMR is chosen and its configuration is shown in figure 1:

![Figure 1. FMWMR Configuration.](image-url)
The main idea of using mecanum wheels is to make the robot move in a narrow space with high maneuver and stability due to its lateral movement characteristics. Each wheel includes passive rollers which are inclined by 45°. Also, each wheel is fitted with the platform of the robot with respect to local frame \( \{R\} \). The parameters of each wheel are: \( (A_i) \) represents the wheel-\( i \) center, \( (\delta) \) represents the measured angle between \( X_R \) and point \( A_i \), \( (\Omega) \) represents the angle between wheel plane and the roller rotation axis, \( (D) \) represents the centroid of robot, \( (\xi) \) represents the measured angle between the wheel axis as well as vector \( DA_i \), \( (r) \): is the radius of each mecanum wheel and \( (\dot{\psi}_i) \) rotation speed of the wheel. These parameters are illustrated in figure 2:

![Figure 2. Representation of mecanum Wheel-i parameters.](image)

In the derivation of the kinematic models, it is considered the case of no slipping i.e., (pure rolling) and the formula describing the velocity of \( (A_i) \) is:

\[
V_{A_i} = x_R \cos \left[ \frac{\pi}{2} - \left( \frac{\pi}{2} - (\delta + \xi) + (\frac{\pi}{2} - \Omega) \right) \right] + y_R \cos \left[ \left( \frac{\pi}{2} - (\delta + \xi) + (\frac{\pi}{2} - \Omega) \right) \right] + \dot{\theta} \cos \left[ \left( \frac{\pi}{2} - (\delta + \xi) + (\frac{\pi}{2} - \Omega) \right) \right]
\]

(1)

After some straightforward calculations, \( (V_{A_i}) \) is represented as:

\[
[\sin(\delta_i + \xi_i + \Omega_i), \ - \cos(\delta_i + \xi_i + \Omega_i), \ - \dot{\theta} \cos(\xi_i + \Omega_i)] R_{R_i} (\theta) \cdot \dot{P}_o = r, \psi_i, \cos \Omega_i
\]

(2)

Where; \( i=1, 2, 3, 4 \) and \( (\dot{P}_o) \) refers to the robot velocity with respect to the universal coordinate and \( R_{R_\theta} (\psi) \) is defined as rotation matrix. The velocity of the robot vector \( (\dot{P}_o) \) including two translation speeds in \( (x, y) \) directions and one rotational speed about \( (z-axis) \). This can be represented with respect to universal and local frames as:

\[
\dot{P}_o = \begin{bmatrix} \dot{x}_o \\ \dot{y}_o \\ \dot{\theta}_o \end{bmatrix} ; \dot{P}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix}
\]

(3)

The velocity in the universal coordinate can be converted to local coordinate as below:

\[
\dot{P}_R = R_{R_i} (\theta) \cdot \dot{P}_o
\]

(4)

The rotation matrix can be represented as:
The radius of each wheel is equal so that \( r_1 = r_2 = r_3 = r_4 = r \). The inverse kinematic equation is obtained as:

\[
\begin{bmatrix}
\psi_1 \\
\psi_2 \\
\psi_3 \\
\psi_4 \\
\end{bmatrix} = \left( \frac{1}{r} \right) \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33} \\
a_{41} & a_{42} & a_{43} \\
\end{bmatrix} \begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
x_O \\
y_O \\
\theta_O \\
\end{bmatrix}
\]

(5)

where: \( a_{11} = a_{12} = a_{13} = 1 \), \( a_{22} = a_{31} = a_{32} = a_{43} = -1 \), \( a_{22} = a_{33} = a_{44} = 1 \), \( a_{33} = a_{44} = \sqrt{2\sin(\frac{\pi}{4} - \delta)} \), \( \delta = \tan^{-1}\left(\frac{h_1}{h_2}\right) \).

The representation of Jacobian matrix \( (H) \) can be illustrated as:

\[
J = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33} \\
a_{41} & a_{42} & a_{43} \\
\end{bmatrix}
\]

(6)

The formula that represents the forward kinematics is:

\[
\begin{bmatrix}
x_O \\
y_O \\
\theta_O \\
\end{bmatrix} = \left( \frac{\sqrt{2}}{2} \right) r J_o \begin{bmatrix}
\psi_1 \\
\psi_2 \\
\psi_3 \\
\psi_4 \\
\end{bmatrix}
\]

(7)

where \( J_o = (J_o^T J_o)^{-1} J_o^T \) is defined as pseudo inverse of matrix \( [J_o] \).

3. Controller design of FMWMR

In this work, a comparison between two controllers based on the kinematic model of FMWMR has been adopted. The first controller consists of TID controller with tuned parameters by PSO while the second controller is considered to be a new hybrid controller consisting of TID controller with tuned parameters by neural network (NN) and SSO. This can be seen in the next sections.

4. Design of TID controller

A TID controller is a tunable feedback control system. The structure of TID controller is like the structure of PID controller but the proportional gain is replaced by tilted gain \( (K_T) \) that has transfer function \( \frac{1}{s^\alpha} \), and the equation that presents the TID controller is [13]:

\[
U(t) = \frac{K_T}{1 + \frac{1}{s^\alpha}} + K_i \frac{1}{s} + K_D s
\]

(8)

Where \( U(t) \) is a controlled signal, \( K_T \) is tilted gain, \( K_i \) is derivative gain, \( K_i \) is an integral gain and \( (n) \) is nonzero real number. The basic structure of TID controller is shown in figure 3:

**Figure 3. Basic structure of TID controller [13].**
The TID controller provides high flexibility for control parameters. It has predominant properties such as simple tuning, improved feedback control and variation of parameters with little effect on the response of the system. In this work, TID controller is applied in order to control the speed of the robot wheels, hence, four TID are used. The parameters of TID i.e., \((K_p, K_d, K_i, n)\) are tuned by using two cases. In the first case, particle swarm optimization (PSO) is used while in the second case the neural network (NN) as well as social spider optimization (SSO) is used and that can be seen in the next sections.

5. Particle swarm optimization (PSO)
PSO is a technique proposed by Elbert as well as Kennedy in 1997. PSO relies on the motion behavior of the animals such as birds. It is simple and fast in reaching a solution but it may fail in the local minima. Every particle in the swarm has apposition as well as velocity in the solution space. They try to update their position and velocity with respect to environment change and that leads the particles to the optimum solutions. The formula that represents the updated velocity and position are [14]:

\[
\begin{align*}
v_{i,t+1} &= v_i + w + c_1 \times \text{rand} \times (pbest_{i,t} - x_{i,t}) + c_2 \times \text{rand} \times (gbest_{i,t} - x_{i,t}) \\
x_{i,t+1} &= x_{i,t} + v_{i,t+1}
\end{align*}
\]

(9) (10)

Where; \((v_{i,t+1})\) represents the speed of \(i^{th}\) particle in iteration \((t+1)\), \((w)\) represents inertia weight factor, \(c_1\) and \(c_2\) are the acceleration and the learning coefficients, respectively, \((x_{i,t+1})\) is the location of the \(i^{th}\) particle in iteration \((t+1)\), (rand) represents random value between \([0-1]\), (pbest) is the best position of each particle and (gbest) is the global best position of the swarm. All parameters of PSO are illustrated in Table 1:

| Parameters       | Value   |
|------------------|---------|
| No. of particles | 10      |
| No. of populations | 20   |
| No. of iterations | 50    |
| \(C_1=C_2\)     | 2       |
| Objective Function | \(\sqrt{e_x^2 + e_z^2 + e_{th}^2}\) |
| Weight (w)       | 0.98    |

The main purpose of using PSO is for tuning the TID parameters and the basic structure of TID-PSO can be seen in figure 4:

![Figure 4. Structure of TID controller tuned parameters by PSO.](image-url)

6. Framework of neural network
Neural network (NN) is considered to be a type of the mathematical algorithms used to emulate the biological counterparts. The types of NN can be divided into: feed-forward as well as recurrent-neural networks. In the feed-forward NN, the information is allowed in single path while in the recurrent NN, the information goes back to previous neurons [15]. In this work, the proposed NN is feed-forward consisting of three layers. The first layer represents the input layer which includes four inputs. The inputs in the NN are the errors in the angular velocities of the wheels while the second layer is a hidden layer. The hidden layer consists of five neurons and the third layer is the output layer including twelve outputs which represent the parameters of TID controller except the parameter (n). The type of the activation function used for the hidden layer is binary sigmoidal and its equation is [15]:

$$f(\text{net}) = \frac{1}{1 + e^{-\lambda \text{net}}}$$  \hspace{1cm} (11)

Where; ($\lambda$) is steepness parameter and its value is equal to one. The basic structure of the NN is shown in figure 5:

![Figure 5. Basic structure of NN.](image)

In the present work, the neural network (NN) is used for computing the parameters of the TID controller except of (n) parameter. The (n) parameter and the initial values of the weights of NN are computed from social spider optimization (SSO) as can be seen in the next section.

7. Social spider optimization (SSO):
SSO is an optimization algorithm that has been used to solve complex problems. A cooperative behavior of spiders is considered the main factor in the work of SSO. In this algorithm, two different genders which are male and female are represent the search area of the solution space. Each spider represents the position of the solution and according to the type of the gender, every individual has been conducted by evolutionary operators. This is considered good featuring over some optimization methods such as PSO. The formula used for calculating female spiders [N_f] is [16]:

$$N_f = \text{floor} [(0.9 - \text{rand} 0.25).N]$$ \hspace{1cm} (12)

Where (rand) represents random number between [0-1] and floor is considered as mapping from real to integer number and (N) is the total number of the spiders. From equation (12) that is maintained above, the numbers of male spiders are calculated as:

$$N_M = N - N_F$$ \hspace{1cm} (13)

The weight of each spider can be calculated as:

$$w_i = \frac{J(s_i) - \text{worst}}{\text{best} - \text{worst}}$$ \hspace{1cm} (14)

Where; $J(s_i)$ represents the fitness obtained from the position of the spider ($s_i$) depending on the objective function. The parameter (R) represents a mating operation in SSO and it is defined as the radius that is based on the size of search space and it is calculated as below:
\[ R = \frac{\sum_{j=1}^{n} (p_{j}^{\text{high}} - p_{j}^{\text{low}})}{2n} \]  \hspace{1cm} (15)

Where; \( p_{j}^{\text{high}} \) and \( p_{j}^{\text{low}} \) are the upper as well as the lower bond of the initial parameters. The magnitudes of the SSO parameters are shown in Table 2:

| Parameters          | Value                  |
|---------------------|------------------------|
| No. of spiders      | 10                     |
| No. of iterations   | 50                     |
| Objective Function  | \( \sqrt{e_{x}^2 + e_{y}^2 + e_{th}^2} \) |

The basic structure of the TID–NN-SSO can be depicted in figure 6 as below:

**Figure 6.** Basic structure of TID-NN-SSO.

8. Results and Discussions

In this section, the results of TID-PSO and the results of TID–NN-SSO are simulated by using MATLAB package and a comparison between them has been done. The efficiency of both controllers are tested by using infinity trajectory and the equations of this trajectory are described as:

\[ X_d = 1 \cdot \sin (0.2t) \]  \hspace{1cm} (16)
\[ Y_d = 1 \cdot \sin (0.4t) \]  \hspace{1cm} (17)
\[ \theta_d = \tan^{-1} \left( \frac{\dot{Y}_d}{\dot{X}_d} \right) \]  \hspace{1cm} (18)

The parameters of the mobile robot can be listed in Table 3 as below [17]:

| 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)        |
8.1. Results of TID-PSO

The actual robot started its motion with initial position $q = [0,0,0]^T$ and the performance of infinity tracking is illustrated in figure 7 as below:

![Infinity tracking of FMWMR.](image)

From the above figure, small deviation from the desired path can be seen and the performance of the tracking is considered good. The change of the position as well as the orientation error with the time are shown in figure 8.

![Posture error](image)

Figure 8. Posture error (a) Error in X-coordinate, (b) Error in Y-coordinate, (c) Orientation error.
From figure 8, the maximum magnitude of the error in X-coordinate is (0.221 m) and this magnitude has been decreased to (0.0357 m) after sixteen seconds. After that, it converged to small until the end of simulation time. The mean square error (MSE) in the X-coordinate is (0.00567 m) while the maximum magnitude of the error in Y-coordinate is (0.815 m). After about four seconds, this value was reduced to (0.0356 m) at the twenty five seconds. Later, the error reached to small value at the end of the simulation time and the MSE in the Y-coordinate was (0.0021 m). The behavior of the velocities of the robot can be depicted in figure 9:

![Figure 9](image)

**Figure 9.** (a) Linear velocity in X-coordinate, (b) Linear velocity in Y-coordinate, (c) Angular velocity

The angular velocities of each wheel are illustrated in figure 10:

![Figure 10](image)

**Figure 10.** (a) Angular velocity of wheel 1 (rad/s), (b) Angular velocity of wheel 2 (rad/s), (c) Angular velocity of wheel 3 (rad/s), (d) Angular velocity of wheel 4 (rad/s).
The actions described in figure 10 show the controlled angular velocity of the maximum beak equals to ± 1.4 rad/s. This indicates that the values are acceptable. The magnitudes of the TID gains that are tuned by PSO are:

\[ K_{T1} = 101.203; \quad K_{D1} = 78.0259; \quad K_{I1} = 25.0236; \quad n_1 = 3.25; \quad K_{T2} = 56.102; \quad K_{D2} = 10.203; K_{I2} = 18.905; K_{I2} = 9.521; \]
\[ n_2 = 4.568 \]
\[ K_{T3} = 35.6687; \quad K_{D3} = 15.798; \quad n_3 = 1.902 \]
\[ K_{D4} = 6.502; \quad K_{T4} = 14.201; \quad K_{D4} = 11.359; K_{I4} = 2.057; n_4 = 5.258. \]

8.2. Results of TID – NN-SSO

The actual robot started its motion with initial position \( q = [0,0,0]^T \) and the performance of infinity tracking is illustrated in figure 11 as:

![Figure 11. The performance of Infinity tracking.](image)

Figure 11 above presents an excellent tracking performance which is better than TID-PSO performance. The efficiency of the new hybrid controllers show i convergence of the position as well as orientation errors as depicted in figure 12:

![Figure 12. Posture error (a) in X-coordinate, (b) in Y-coordinate, (c) Orientation error.](image)
From figure 12, the maximum magnitude of error in (X and Y) coordinates were (0.0375 m) and (-0.02 m), respectively. The beginning of the motion was until about 2 seconds, after that the magnitudes of the error converged to very close to zero. The MSE in X-coordinate was (2.105*10^{-4} m) and MSE in Y-coordinate was (1.025*10^{-4} m). On the other hand, the maximum magnitude of the orientation error was (-0.458 rad) fluctuating with small magnitudes until twenty seconds. Then, the magnitudes were reducible to very small value and these values of error in (X and Y) coordinates as well as the values of the orientation error were less than the magnitudes of the error obtained from TID-PSO. The behavior of the robot velocities is depreciated in figure 13:

![Graphs showing robot velocities](image)

**Figure 13.** (a) Linear velocity in X-coordinate; (b) Linear velocity in Y-coordinate; (c) Angular velocity.

The angular velocities of each wheel are illustrated in figure 14:
(c)

Figure 14. (a) Angular velocity of wheel 1 (rad/s); (b) Angular velocity of wheel 2 (rad/s), (c) Angular velocity of wheel 3 (rad/s); (d) Angular velocity of wheel 4 (rad/s).

The actions described in figure (14) show that the controlled angular velocity were smooth and the maximum beak equals to ± 0.78 rad/s. This indicates that the values are small and required small power to drive. Also, this value of (± 0.78 rad/s) is less than the magnitudes of the controlled wheels speeds obtained from TID-PSO.

The magnitudes of TID parameters obtained from NN-SSO are:

\[ K_{T1} = 110.428; \quad K_{D1} = 61.590; \quad K_{i1} = 28.692; \quad n_1 = 2.528 \quad K_{T2} = 44.225; \quad K_{D2} = 14.589; \quad K_{i2} = 16.511; \quad K_{D3} = 17.106; \quad n_2 = 4.789 \]

\[ K_{T3} = 38.157; \quad K_{D3} = 9.815; \quad K_{i3} = 4.210; \quad n_3 = 2.589 \quad K_{T4} = 9.2617; \quad K_{D4} = 7.885; \quad K_{i4} = 1.9057; \quad n_4 = 3.698. \]

9. Conclusions

In this work, a new hybrid controller consisting of TID-NN-SSO based on the kinematic model of the FMWMR has been implemented. Four TID controllers were used in order to control the speed of the robot wheels while the NN and SSO were used for computing the parameters of TID controller. A comparative study between the new hybrid controller and the TID-PSO has been presented. The simulation results have shown that the new hybrid controller is better than TID-PSO, because the MSE of the (ex, ey, eθ), obtained from the new hybrid controller were (2.105*10^-4 m, 1.025*10^-4 m, 0.0815 rad), respectively. On the other hand, the values of the MSE of the (ex, ey, eθ) obtained from TID-PSO were (0.00567 m, 0.0356 m, 1.22 rad/s), respectively. Moreover, the behavior of the controlled angular velocities of each wheel of the robot obtained from TID–NN-SSO were smooth and their magnitudes did not exceed (± 0.78 rad/s). This value is considered very suitable and less than the magnitudes obtained from TID-PSO which were about (± 1.4 rad/s).

10. References

[1] Spyros G Tzafestas 2014 Introduction to Mobile Robot Control, First Edition (Elsevier)
[2] Gregor Klancar, Andrej Zdesar, Saso Blazic and Igor Škrjanc 2017 Wheeled Mobile Robotics From Fundamentals Towards Autonomous Systems (Oxford, Elsevier)
[3] Sergio Morales and Cesar Delgado 2018 LQR Trajectory Tracking Control of an Omnidirectional Wheeled Mobile Robot (IEEE 2nd Colombian Conference on Robotics and Automation (CCRA))
[4] Veer Alakshendra and Shital S.Chiddarwar 2015 Design of Robust Adaptive Controller for a Four Wheel Omnidirectional Mobile Robot (International Conference on Advances in Computing, Communications and Informatics) pp 63-68
[5] Zhening Gao, Yuanxin Yang, Yanping Du, Yuan Zhang , Zhaohua Wang and Weidong Xu 2017 Kinematic Modeling and Trajectory Tracking Control of a Wheeled Omni-directional Mobile Logistics Platform (Asia-Pacific Engineering and Technology Conference) pp 69-175
[6] Jungje Park, Sudae Kim, Jungmin Kim and Sungshin Kim 2010 Driving Control of Mobile Robot with Mecanum Wheel Using Fuzzy Inference System (International Conference on Control, Automation and Systems) pp 2519-2523
[7] Chengcheng Wang, Xiaofeng Liu, Xianqiang Yang, Fang Hu Aimin Jiang and Chenguang Yang 2018 Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy (Applied Science) pp 1-15
[8] Eka Maulama, M. Aziz Muslim and Veri Hendrayawan 2015 Inverse Kinematic Implementation of Four-Wheels Mecanum Drive Mobile Robot Using Stepper Motors (International Seminar on Intelligent Technology and Its Applications) pp 51-54
[9] Ehsan Malajjerdi, Hadi Kalani and Mohsen Malajjerdi 2018 Self-Tuning Fuzzy PID Control of a Four-Mecanum Wheel Omni-directional Mobile Platform (26th Iranian Conference on Electrical Engineering) pp 816-820
[10] Qingzhu Cui, Xun Li, Xiangke Wang and Meng Zhang 2012 Backstepping Control Design on
the Dynamics of Omni-Directional Mobil Robot (Applied Mechanics and Materials) vol 203 pp 51-56

[11] Addzrull Hi-Fi Syam Ahmad Jamil, Rosli Omar, Sazali Yaacob and Mohd Firdaus Mokhtar 2017 Development of Fuzzy PID Controller for Mecanum Wheel Robot (International Journal of Applied Engineering Research) vol 12 no 24 pp 14478-14483

[12] Pouya Jamali, S Masoud Tabatabaei, Omid Sohrabi and Navid Seifipour 2013 Software Based Modeling, Simulation and Fuzzy Control of a Mecanum Wheeled Mobile Robot (International Conference on Robotics and Mechatronics) pp 200-204

[13] Rajendra Kumar Khadanga, Sasmita Padhy, Sidhartha Panda and Amit Kumar 2018 Design and Analysis of Tilt Integral Derivative Controller for Frequency Control in an Islanded Microgrid: A Novel Hybrid Dragonfly and Pattern Search Algorithm Approach (Arabian Journal for Science and Engineering)

[14] Singiresu S Rao 2009 Engineering Optimization Theory and Practice, Fourth Edition (John Wiley & Sons, Inc)

[15] R S Burns 2001 Advanced Control Engineering. Boston (Butterworth Heinemann)

[16] Alberto Luque-Chang, Erik Cuevas, Fernando Fausto, Daniel Zald-var and Marco Pérez 2018 Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives (International Journal of Engineering Mathematical, Hindawi) vol 2 pp 1-29

[17] Igor Zeidis and Klaus Zimmermann 2019 Dynamics of a Four-Wheeled Mobile Robot with Mecanum Wheels (Journal of Applied Mathematics and Mechanics) vol 3 pp 1-22