AKADÉMIAI KIADÓ

UNIVERSITY of DEBRECEN

International Review of Applied Sciences and Engineering

14 (2023) 1, 58–67

DOI:
10.1556/1848.2022.00420

ORIGINAL RESEARCH PAPER

PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot

Ammar Al-Jodah[1], Saad Jabbar Abbas[2], Alaq F. Hasan[3], Amjad J. Humaidi[4*], Abdulkareem Sh. Mahdi Al-Obaidi[5], Arif A. AL-Qassar[4] and Raaed F. Hassan[6]

[1] School of Physics, Maths and Computing, Physics, The University of Western Australia, Perth, WA 6907, Australia

[2] AL Rafidain University College, Baghdad, Iraq

[3] Technical Engineering College, Middle Technical University, Baghdad, Iraq

[4] Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

[5] School of Engineering, Faculty of Innovation and Technology, Taylor's University, Malaysia

[6] Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq

Received: November 18, 2021 • Accepted: February 12, 2022
Published online: September 23, 2022

ABSTRACT

The demand for automation using mobile robots has been increased dramatically in the last decade. Nowadays, mobile robots are used for various applications that are not attainable to humans. Omnidirectional mobile robots are one particular type of these mobile robots, which has been the center of attention for their maneuverability and ability to track complex trajectories with ease, unlike their differential type counterparts. However, one of the disadvantages of these robots is their complex dynamical model, which poses several challenges to their control approach. In this work, the modeling of a four-wheeled omnidirectional mobile robot is developed. Moreover, an intelligent Proportional Integral Derivative (PID) neural network control methodology is developed for trajectory tracking tasks, and Particle Swarm Optimization (PSO) algorithm is utilized to find optimized controller's weights. The simulation study is conducted using Simulink and Matlab package, and the results confirmed the accuracy of the proposed intelligent control method to perform trajectory tracking tasks.

KEYWORDS

omnidirectional mobile robot, neural network, PID control, PSO

1. INTRODUCTION

Omnidirectional mobile robots have been utilized in numerous industries for their enhanced features such as a high degree of maneuverability, improved dexterity, and driving capacity. The high maneuverability of these robots has proven to outperform differential wheels mobile robots as they can track complex trajectory that may be challenging or even not feasible for differential wheels mobile robots. Omnidirectional mobile robots have shown the capacity to perform motions in any direction without the need to have extensive space to finish the maneuver. For all these advantages and enhanced features, the omnidirectional mobile robots have been widely adopted in service and industrial applications, such as drug delivery in pharmacies, materials delivery in a factory floor, goods arrangement and packaging in the retail industry, and many more [1].

Omnidirectional mobile robots have the capacity to perform rotational and translational motions at the same time and in an uncoupled type of motion. Various kinds of

*Corresponding author.
E-mail: amjad.j.humaidi@uotechnology.edu.iq

AKJournals

omnidirectional mobile robots were developed in the literature, such as three and four wheels omnidirectional mobile robots. The four-wheeled type can offer much higher maneuverability when compared to the three-wheeled type as it has an extra degree of freedom represented by the fourth wheel. The dynamics model and control methodologies have been proven quite challenging, as many factors affect the mobile robot system. Friction, backlash in the wheels, nonlinear behavior of the DC motors, sensor noise, positioning drift, uncertainty in model parameters have made the control method for these robots require rigorous stability and performance analysis. Several control methodologies have been proposed to take some of these factors into consideration.

In [2], dynamics of a Three-Wheeled Omnidirectional Mobile Robot (TW-OMR) was obtained and then linearized. Moreover, some insights for stability and control was provided. In [3], a proportional-integral control method was proposed to perform point following and tracking tasks. The controller's parameters were tuned using the ant colony optimization algorithm. The methodology was verified in simulation in point following motion and circular trajectory tracking tests. However, complex trajectories were not considered and verified in this study. The use of the linear control method with this highly nonlinear mobile robot system may make this method fail to meet the high maneuverability requirements of complex trajectories. High-speed trajectory tracking controller based on Takagi–Sugeno Fuzzy system was proposed in [4]. Kinematics inversion was utilized in the feedback loop to simplify the control system design. The tracking performance was evaluated with results from a classical PID control, and the proposed method showed a better outcome to follow a simple square trajectory. In [5], a review of the omnidirectional and holonomic mobile robot was provided. Insights into their dynamical and kinematics modeling were discussed. Several control methods such as PID, and Fuzzy control, were developed. However, no simulation was provided to verify these control methods. In [6], a TW-OMR dynamical model was developed. Based on this model, a computed torque control method was proposed to stabilize the mobile robot and provide accurate trajectory tracking performance. A simulation study was conducted to test the control approach, where a simple circular trajectory was utilized. In [7], a full dynamic model was derived for an omnidirectional mobile robot. Output feedback linearization control method was proposed to solve the tracking task.

In [8], the output feedback control structure with a linear controller and observer was used for tracking tasks of a TW-OMR. Disturbances on the system considered as additive uncertainty terms with the control action provide to the wheels. A laboratory-based TW-OMR was used to verify the developed control methodology. The robot was able to follow a simple circular trajectory. In [9], an adaptive backstepping control methodology was proposed to control a Four-Wheeled Omnidirectional Mobile Robot (FW-OMR). Various motions were able to attain using this method, and the performance was verified in the simulation study. In [10], a PI control method was used to control a TW-OMR. A fuzzy logic system was used to tune the PI gains instead of using try and error approach. The effectiveness of the control method was confirmed in simulation by a simple point to point and circular trajectory tracking tests. In [11], a PID control approach was used for trajectory tracking of a TW-OMR. Practical emphases were on estimating the mobile robot velocities from the robot's internal sensors rather than using an external localization system. In [12], a fuzzy control system was proposed to control a FW-OMR, and was directed towards security and surveillance applications. The encoders and sensors of the wheels' motors were used to provide the feedback signal and to estimate the robot location. The control method was verified in an experimental study and was proven effective to follow a circular trajectory. The results were compared with a simple proportional controller. A sliding mode control method was proposed in [13], to achieve trajectory tracking tasks in TW-OMR. Backstepping approach was utilized to facilitate the reachability to the control signal. Simple turning trajectory was used to verify the control method, and small tracking errors were observed. In [14], Wahhab and Al-Araji have presented design based on Convolutional Neural Network Trajectory Tracking (CNNTT) controller to control mobile robot to find optimal path in the presence of obstacles using hybrid swarm optimization. In [15], Al-Araji et al. proposed an adaptive nonlinear controller for trajectory tracking of nonholonomic mobile robot. The controller consists of feedforward multi-layer perceptron and modified Elman neural network. The Elman neural model is trained to work as orientation and position identifier, while the feed-forward multi-layer perceptron is trained off-line and the weights are on-line adapted to generate the actuating torques of mobile motors. In [16], Al-Araji et al., proposed nonlinear neural controller based on optimization algorithm to follow optimal path-tracking of mobile robot. Artificial Bee Colony and Particle Swarm Optimization algorithms are applied for finding the optimal navigation of mobile robot.

It has been noted from the literature that the control methods provided for trajectory tracking tasks in omnidirectional mobile robots are either linear or non-optimized. The performance of those methods is heavily dependent on the designer trial and error approach, which can be infeasible with a large number of parameters. Moreover, most of the control approaches reported in previous studies were only verified using a simple point to point following or circular trajectory. In practice, these robots are expected to follow complex trajectories to perform service or industrial tasks, and it is quite rare that they only follow a simple circular trajectory. Furthermore, the uncertain environment of these robots poses further challenges to their control methodologies. Therefore, in this work, an intelligent Proportional Integral Derivative (PID) based on the neural network control method is proposed for trajectory tracking tasks of a FW-OMR. The learning ability of the neural network is expected to improve the motion accuracy of these robots and make them able to overcome the uncertainties in their environment. The controller's parameters were obtained using the particle swarm optimization algorithm. A simulation study was conducted to verify the proposed

control method, where a complex trajectory was tested and verified. The results have confirmed the accuracy and feasibility of the designed control approach.

2. FOUR-WHEELED OMNIDIRECTIONAL MOBILE ROBOT MODELING

The configuration of the FW-OMR is shown in Fig. 1. There is a 90° between the wheels. It is essential to define the following notations to develop the model of the mobile robot. The robot coordinates are defined by x, y, and θ. The wheels velocities are v_1 to v_4.The linear velocity of the robot is defined by V and V_n, and the angular velocity is represented by ω. Wheels traction forces are given by t_{f1} to t_{f4} .Traction forces in the directions of V and V_n are given by F_v and F_{vn}. The distance between wheels is given by $2L$. The torque of the robot is provided by T. A local coordinate frame is allocated to the center of the robot, and a global coordinate frame is defined to relate the robot motion with its world.

2.1. Robot kinematics

To establish the relations between the robot motions linear and angular motions, the following equation is defined [17]:

$$\begin{bmatrix} V \\ V_n \\ \omega \end{bmatrix} = R \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{1}$$

where $v_x = \dot{x}$, $v_y = \dot{y}$, $\omega = \dot{\theta}$ and R is the orthogonal rotation matrix, and it is given by

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Based on Eq. (1), the following can be defined [18]

$$\dot{x} = -\sin(\theta)V_n + \cos(\theta)V \tag{3}$$

$$\dot{y} = \cos(\theta)V_n + \sin(\theta)V \tag{4}$$

$$\dot{\theta} = \omega \tag{5}$$

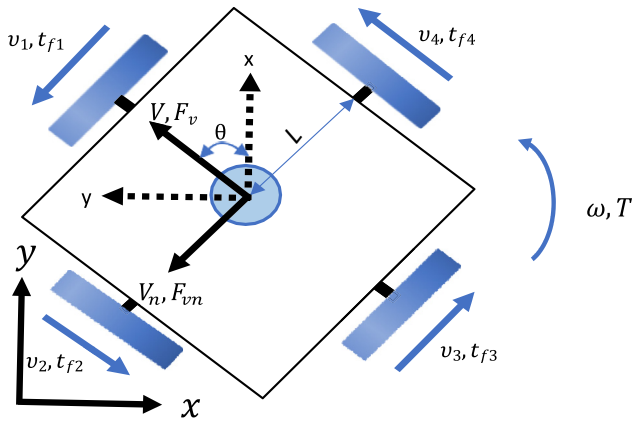

Fig. 1. The configuration of a four-wheeled omnidirectional mobile robot

The transformation from the robot velocities to wheels velocities is given by

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ L & L & L & L \end{bmatrix}^T \begin{bmatrix} V \\ V_n \\ \omega \end{bmatrix} \tag{6}$$

2.2. Robot dynamics

The mobile robot dynamics is given by [19]

$$\dot{V} = \frac{1}{M}(F_v - B_v V - C_v sign(V)) \tag{7}$$

$$\dot{V}_n = \frac{1}{M}(F_{vn} - B_{vn} V_n - C_{vn} sign(V_n)) \tag{8}$$

$$\dot{\omega} = \frac{1}{J}(T - B_\omega \omega - C_\omega sign(\omega)) \tag{9}$$

where M and J are the robot mass, and moment of inertia, respectively; B_ϕ and C_ϕ are the viscous and coulomb friction coefficients, respectively, where $\phi = v,\ v_n$ and ω. The robot torque and forces are given by:

$$F_v = t_{f4} - t_{f2} \tag{10}$$

$$F_{vn} = t_{f1} - t_{f3} \tag{11}$$

$$T = L\left(t_{f1} + t_{f2} + t_{f3} + t_{f4}\right) \tag{12}$$

The traction force on each wheel is estimated by traction torque as $tf_j = T_j/r$. The torque is electrically generated in the motor which can be found by multiplying the motor torque constant with the gear ratio and the current in the motor i.e., $T_j = l\ K_t i_j$. Substituting the torque back in the traction force will result in

$$t_{fj} = lK_t\ i_j/r,\ \ldots j = 1,\ \ldots,\ 4 \tag{13}$$

where l is the gear reduction ratio, K_t is the motor torque constant, i_j is the current of the motor j, and r is the wheel radius. Using the common DC motor model with a negligible inductance, i.e., $u_j = Ri_j + k_v\omega_{mj}$, the i_j can be found as

$$i_j = \left(u_j - k_v\ \omega_{mj}\right)/R,\ \ldots j = 1, \ldots,\ 4 \tag{14}$$

where R is the motor coil resistance, u_j is the voltage of motor j, K_v is the EMF motor constant, and ω_{mj} is the angular velocity of the motor j, and it is given by:

$$\omega_{mj} = lv_j/r,\ \ldots j = 1,\ \ldots,\ 4 \tag{15}$$

3. NEURAL NETWORK PID CONTROL METHOD DESIGN

Neural networks have been used in modeling for their effective function approximation and learning ability. More recently, they have been applied to control various nonlinear systems. Control systems developed based on neural networks are categorized into two schemes. In the first scheme,

the control action is computed directly by the neural network. In the second scheme, the neural network is used to tune the control parameters online. Both schemes require training of the network to perform the required tasks. Back propagation is one of the most famous training methods that is used to train a feedforward neural network in what is called supervised learning. It requires the differentiation of the error signal to update the network's weights. Input and output data set are also necessary for such a learning method, which could be challenging to obtain for some systems. On the other hand, metaheuristic optimization algorithms have shown promising results in solving various engineering problems and recently have been applied to train neural networks. These algorithms are based on minimizing a cost function via tuning some design parameters, and they do not require error differentiation as in the back propagation method. Therefore, they function as global optimizers and do not easily fall in local minimums as in local search methods. Thus, for these advantages, the particle swarm optimization algorithm is adopted in the current study to train the neural network [14, 15].

Every mobile robot is required to follow a trajectory to perform a certain task. The accuracy of following this trajectory is significantly important as that it has a direct effect on the safety of the people around the robot, the battery life, power consumption and heat dissipation, and the wear and tear of the robot parts. The learning ability of the neural network makes it feasible to attain the required motion accuracy with a high level of robustness. Therefore, a neural network proportional-integral-derivative (NN-PID) control method is developed in this work to enhance the trajectory tracking tasks for a FW-OMR. The closed loop control system that involves the proposed control methodology is shown in Fig. 2. The error between the reference and actual position is multiplied by the orthogonal rotational matrix to transfer the error from the world coordinates to the robot coordinates. The PSO algorithm is function as a trainer for the neural network. The cost function using in the PSO is based on the tracking error vector e_σ. The control action is produced from the NN-PID control method, and is represented by four voltages to drive the wheels' motors [16].

The NN-PID control method is firstly proposed by Shu in [14–16, 20]. The NN-PID controller proposed in [14–16, 20] is represented as a neural network with three neurons in the middle layer to mimic the behavior of the continuous PID controllers. The first node in this layer corresponds to the proportional component of the PID. Similarly, the second and third nodes correspond to the integral, and derivative components, respectively. In this work, three

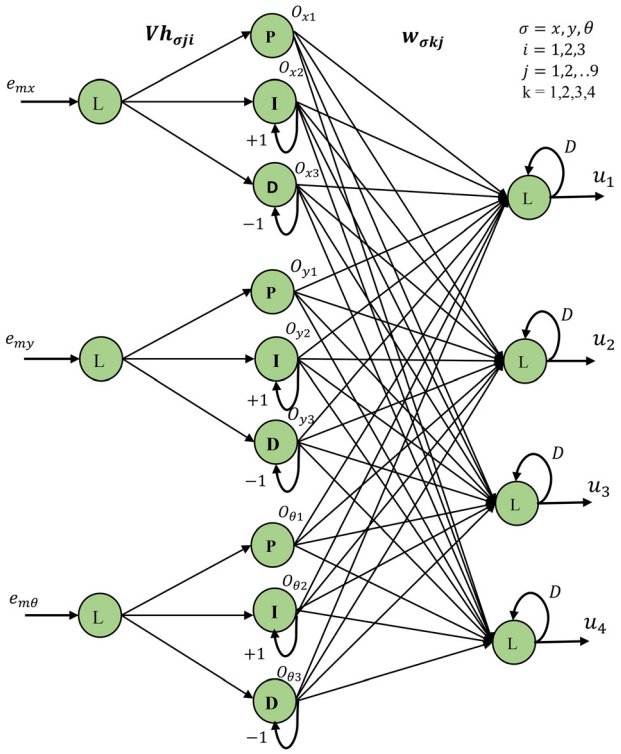

Fig. 3. The structure of the NN-PID controller

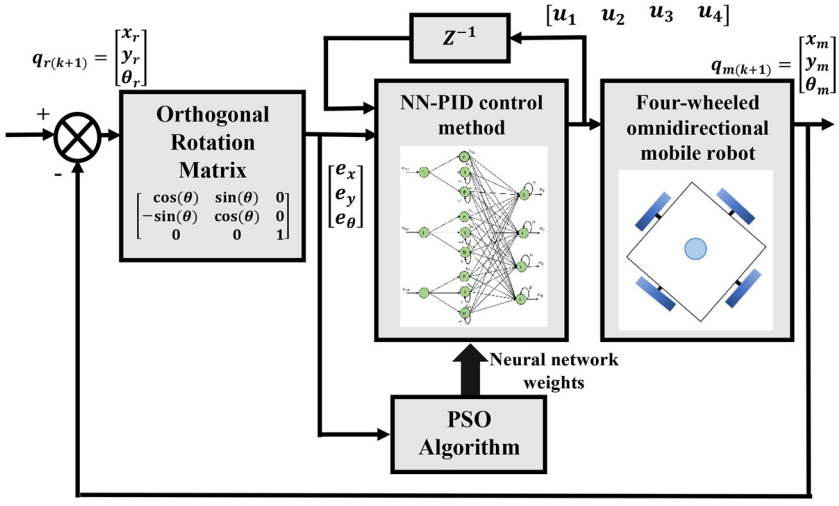

Fig. 2. The structure of the closed loop control system

NN-PID controllers are used, one each coordinate, i.e. NN-PID$_x$, NN-PID$_y$, and NN-PID$_\theta$. A fully connected neural network layer is adopted in this work to map the output of the three NN-PID controllers to the wheel's voltages. The structure of the NN-PID control method is shown in Fig. 3. The input layer consists of three neurons with the error as input. The hidden layer has nine neurons, each three of those represent a NN-PID controller. The output layer has four neurons that represent the four wheel's voltages. A fully connected neural network is utilized to connect the hidden layer with the wheel's voltages in the final layer.

The layers' weights are given by Vh_i, and Wh_{kj}, respectively, where $j = 1,\ 2,\ \ldots,\ 9$, and $k = 1,\ 2,\ 3,\ 4$. The first layer output is given by

$$e_1(T) = e_{mx}(T),\ e_2(T) = e_{my}(T),\ e_3(T) = e_{m\theta}(T) \quad (16)$$

The hidden layer net input to each neuron is given by

$$h_{netj}(T) = Vh_j e_i(T), i = integerDivision((\ j + 1 + remainder(\ j,\ 3)),\ 3) \quad (17)$$

The proportional neurons outputs are given by

$$h_1(T) = sigmoid(h_{net1}(T)) \quad (18)$$

$$h_4(T) = sigmoid(h_{net4}(T)) \quad (19)$$

$$h_7(T) = sigmoid(h_{net7}(T)) \quad (20)$$

The integral neurons outputs are given by

$$h_2(T) = sigmoid(h_{net2}(T)) + h_2(T-1)) \quad (21)$$

$$h_5(T) = sigmoid(h_{net5}(T)) + h_5(T-1)) \quad (22)$$

$$h_8(T) = sigmoid(h_{net8}(T)) + h_8(T-1)) \quad (23)$$

The derivative neurons outputs are given by

$$h_3(T) = sigmoid(h_{net3}(T) - h_3(T-1) \quad (24)$$

$$h_6(T) = sigmoid(h_{net6}(T) - h_3(T-1) \quad (25)$$

$$h_9(T) = sigmoid(h_{net9}(T) - h_3(T-1) \quad (26)$$

The outputs of the final layer are given by

$$o_k(T) = o_k(T-1) + \sum_{j=1}^{9} wh_{kj} h_j(T) \quad (27)$$

The sigmoid function is given by [20]

$$sigmoid(net) = \frac{2}{e^{-net}} - 1 \quad (28)$$

4. NN-PID TRAINING BY PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) was firstly developed in [21, 22], and it has become extremely famous ever since. It is metaheuristic algorithms based on a simplified and efficient set of computational steps. It mimics the social behavior of swarm of particles. It has shown promising results in solving nonlinear continuous problems [23–27]. Moreover, it has shown faster convergence rate, more rapid computations, and extra accurate solutions when compared with other optimization algorithms. The particles positions represent solutions in the search space. Thus, each particle has n-dimensional vector to describe its position, where n is the number of the optimization parameters. The particles communicate within the swarm and follow a learning path

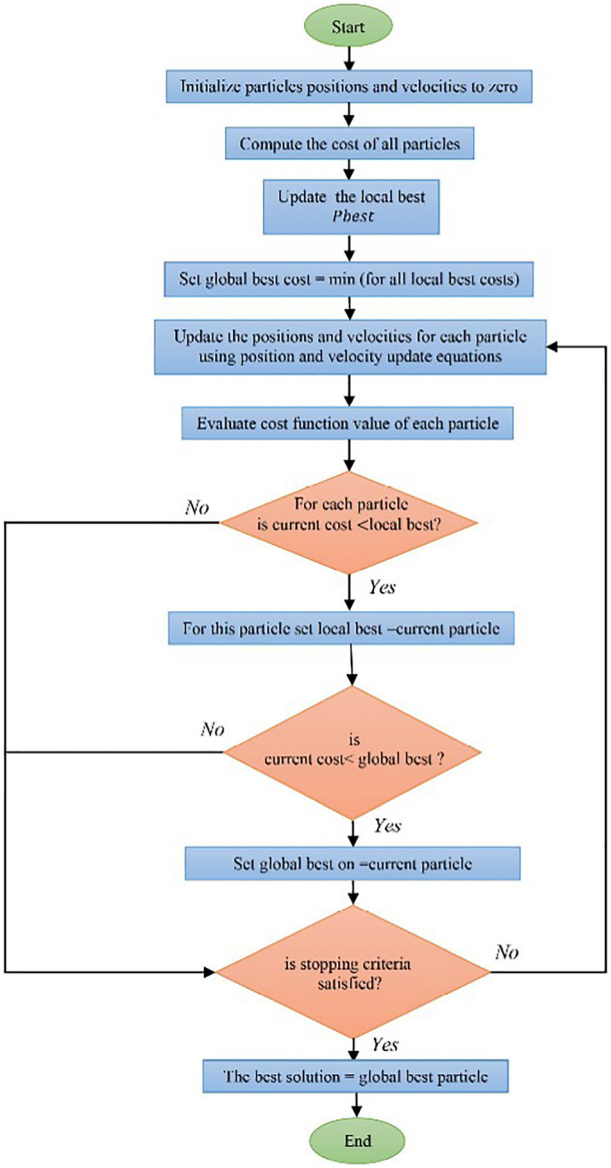

Fig. 4. PSO flowchart

Table 1. PSO parameters

Parameter	Value
maximum number of iterations	1,000
c_1	2
c_2	2
w	1.5
swarm size	30
n	45
parameter range	[–100, 100]

Table 2. Robot parameters [17]

Parameter	Unit	Value
r	m	0.0325
M	kg	2.34
J	kg·m^2	0.0228
R	Ω	4.3111
K_v	V (rad^{-1}/s)	0.0259
K_t	V.s rad^{-1}	0.0259
L	m	0.089
l	/	5
B_v	N.s m^{-1}	0.4978
B_{vn}	N.s m^{-1}	0.6763
B_w	N.m.s rad^{-1}	0.0141
C_v	N	0
C_{vn}	N	0
C_w	N.m	0

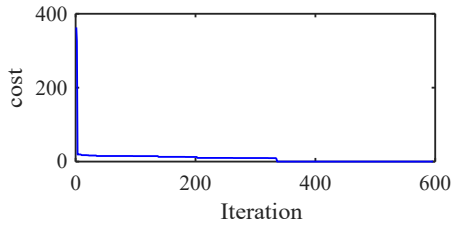

Fig. 5. Cost progression with PSO iterations

Table 3. Input to hidden layer weights

Parameter	Value
K_{h1}	3.183994
K_{h2}	2.164797
K_{h3}	–6.76984
K_{h4}	–0.51265
K_{h5}	2.769091
K_{h6}	–8.1403
K_{h7}	–1.07712
K_{h8}	7.005729
K_{h9}	2.03942

based on their local best seen solution and also based on the best global solution by the swarm.

Each particle update its velocity and position in this process according to the flowing rules [21, 22]:

$$v_i^{k+1} = wv_i^k + c_1 r_1 \left(pbest_i - \xi_i^k\right) + c_2 r_2 \left(Gbest - \xi_i^k\right) \quad (29)$$

$$\xi_i^{k+1} = \xi_i^k v_i^{k+1} \quad (30)$$

where at iteration $k + 1$, v_i^{k+1}, and ξ_i^{k+1} are the particle i's velocity, and position, respectively; w is the inertia; r^1 and r^2 are randomly generated numbers in the range zero to one; c_1, and c_2 are constant to balance the update between the local and global best; b_{besti} and G_{best} are the local and the global best solutions, respectively. The number of the PSO's dimensions n represent how many weights there are in the neural network. The flowchart of the PSO algorithm is shown in Fig. 4. The cost function is based on the error vector, and it is defined as

$$cost = \frac{1}{m} \sum_{i=1}^{m} \left(e_x^2(i) + e_y^2(i) + e_\theta^2(i)\right) \quad (31)$$

where the number of simulation samples is given by m. The PSO parameters are presented in Table 1.

This study can be better improved or extended by incorporating other optimization techniques like the Grey-Wolf Optimization, Social Spider Optimization, Whale-Optimization Algorithm [28–31]. A comparison study can be conducted by comparing one of these recent optimization techniques with PSO algorithm.

5. RESULTS AND DISCUSSION

The simulation was carried out in Simulink and Matlab. The structure of Fig. 2 was adopted in the simulation. The robot physical parameters were assumed as in Table 2. The NN-PID control method was trained using PSO with the parameters given in Table 1. After 600 iteration the neural network was trained. The cost progress with each iteration in shown in Fig. 5. The results of the training are given in Table 3, and Table 4. Unlike previous works, a complex reference trajectory was used in the simulation to verify the effectiveness of the designed control methodology. The reference trajectory is defined as

$$x_r(T) = 3sin\left(6\pi t / t_{final}\right); \quad (32)$$

$$yr(T) = 3sin\left(4\pi t / t_{final}\right); \quad (33)$$

Table 4. Hidden to output layer weights

Weight	Value	Weight	Value	Weight	Value	Weight	Value
Wh_{11}	2.670686	Wh_{21}	1.610139	Wh_{31}	–0.26488	Wh_{41}	–0.41302
Wh_{12}	–4.36445	Wh_{22}	0.903675	Wh_{32}	–3.51607	Wh_{42}	9.555913
Wh_{13}	–11.0002	Wh_{23}	27.50922	Wh_{33}	3.772408	Wh_{43}	–54.2146
Wh_{14}	–13.7829	Wh_{24}	–26.9897	Wh_{34}	5.721002	Wh_{44}	6.367803
Wh_{15}	2.301405	Wh_{25}	–3.54052	Wh_{35}	1.148764	Wh_{45}	0.79572
Wh_{16}	–1.73654	Wh_{26}	–8.72825	Wh_{36}	19.46894	Wh_{46}	30.33683
Wh_{17}	6.857219	Wh_{27}	–0.05555	Wh_{37}	–2.99248	Wh_{47}	–62.7306
Wh_{18}	0.119107	Wh_{28}	3.728767	Wh_{38}	5.861321	Wh_{48}	–0.00976
Wh_{19}	48.98421	Wh_{29}	5.315695	Wh_{39}	45.37744	Wh_{49}	62.72158

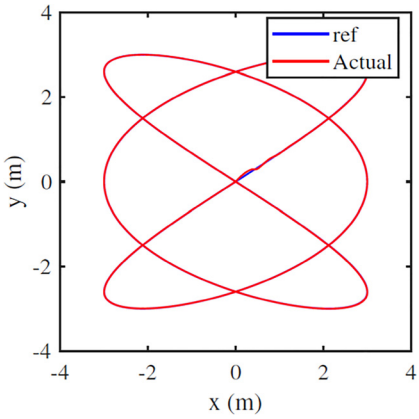

Fig. 6. Trajectory plot

$$thr(T) = tan^{-1}\left(\frac{xr(T) - xr(T-1)}{t+\varepsilon}, \frac{yr(T) - yr(T-1)}{t+\varepsilon}\right) \tag{34}$$

where t is the current time instance and it is given by $t(T) = t(T-1) + \Delta T$, the sampling rate is given $\Delta T = 0.1s$. Moreover, ε is a small number to avoid division by zero.

First, the simulation study was carried out for the point $(x, y, \theta) = (0, 0, \pi/6)$. The corresponding results are given in Figures 6, 7, 8 and 9. The mean square error is calculated and presented in Table 5. It can be noticed from these figures that the mobile robot was able to complete the trajectory tracking with high motion precision. The velocity

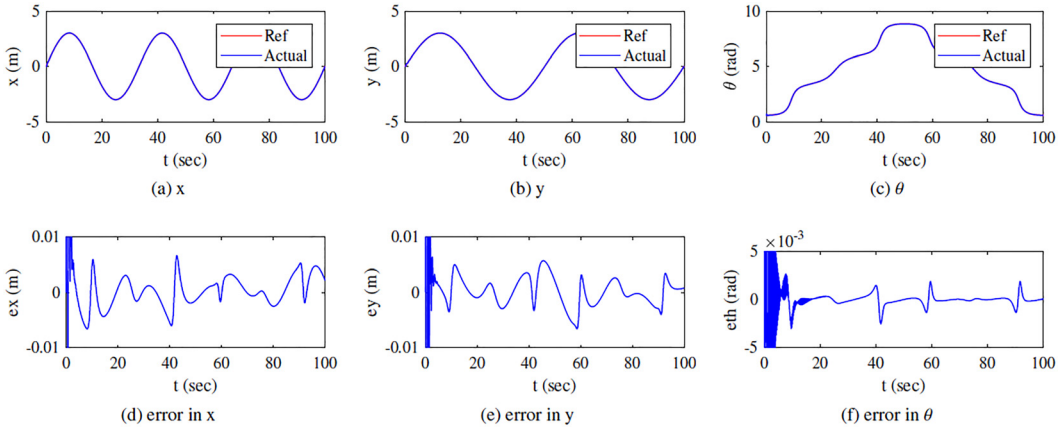

Fig. 7. Tracking errors

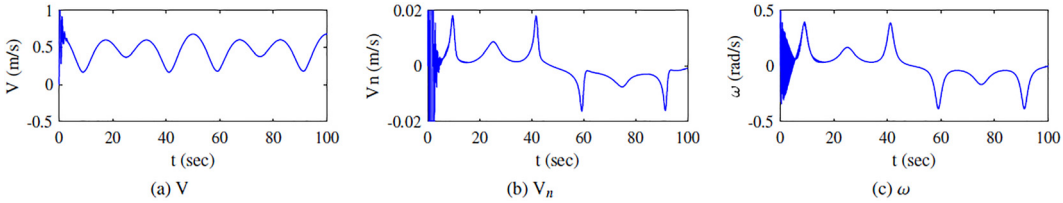

Fig. 8. Robot velocities

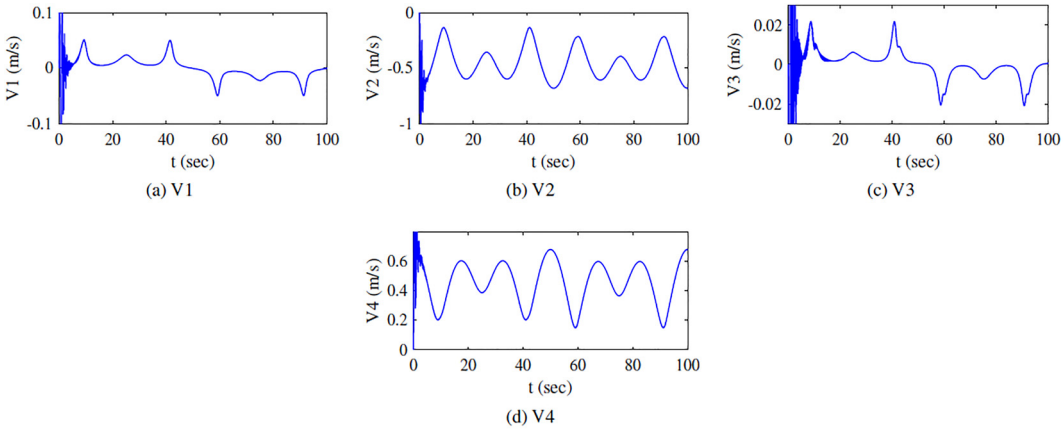

Fig. 9. Wheels velocities

Table 5. Mean square error of trajectory tracking results

Initial point	MSE ex	MSE ey	MSE eth
(0, 0, π/6)	2.3406e−05	2.7623e−05	2.5318e−05
(0.2, 0.6, π/6)	7.1066e−05	7.2527e−04	2.2730e−05

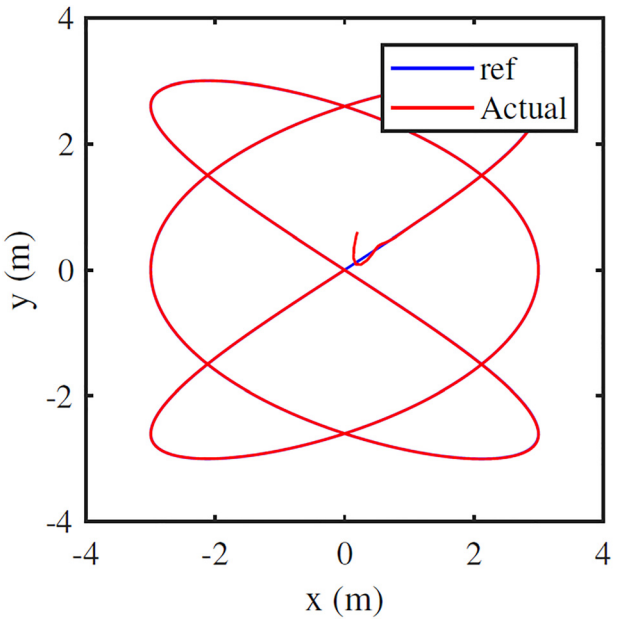

Fig. 10. Trajectory plot

signals of the robot showed smooth change to perform the maneuvers. Higher error and velocity were noticed in the first few seconds of the simulation. These high values are due to the initialization of the control method as the error tends to increase at the start of the simulation. After this initialization phase passes, the control approach continues with a smoothly changing signal to follow the complex trajectory. Further, it was noticed that V3 has small magnitude as compared to the other wheels velocities, as the range of it was limited to $\mp 0.02\ m/s$ unlike the other ones, which have higher values. The reasoning behind this is the redundancy of the degree of freedoms that are offered by this type of mobile robot. Moreover, it means that the mobile robot was able to track the provided complex trajectory without actually needing to rotate the third wheel as much as the other ones. This might be only because of the shape of the given trajectory and other behavior might be noticed in different trajectories. To investigate the robustness of the control methodology, a different initial pose of the robot was tested. In this test the initial point was selected as $(x,y,\theta) = (0.2,0.6,\pi/6)$. The same neural network was used in this simulation without a retraining. The results of this simulation are given in Figures 10, 11, 12 and 13. The mean square error for this simulation is calculated and presented in Table 5. Again, the mobile robot was able to follow the trajectory with high accuracy. The robot started by making a maneuver to orient towards the trajectory and then it moved until it hits the reference trajectory and it continues to follow it. In terms of the robot velocities it was similar as in the previous simulation. However, this time higher oscillation was noticed at the start of the simulation. This was expected as the neural network was not trained to start from this initial pose, therefore higher control action is provided to the motors the drive the wheels to perform the maneuvers rapidly and recover from the unexpected initial pose. The calculated mean square errors in Table 5, show higher values for the second simulation. Again, this behavior is due to the

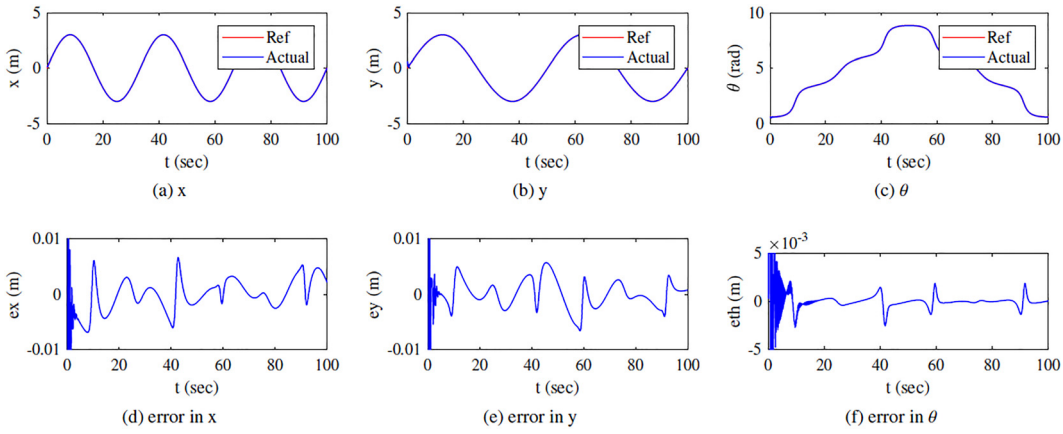

Fig. 11 Tracking errors

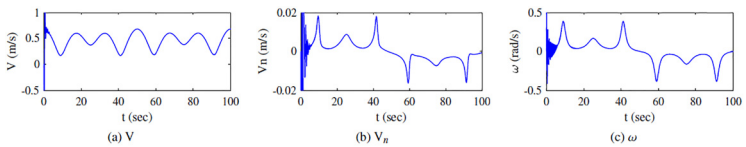

Fig. 12. Robot velocities

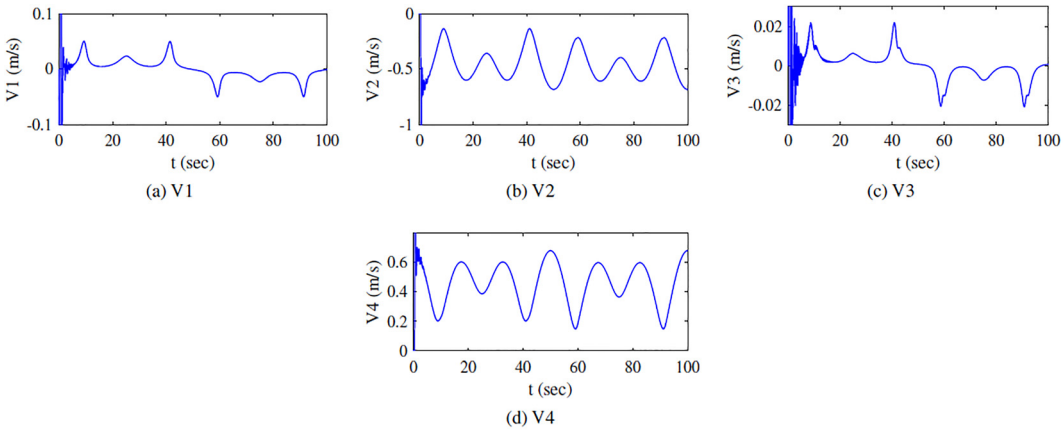

Fig. 13. Wheels velocities

start form an untrained pose. The error increase in the x and y axis was about three times higher than the first case, while the error for the θ was slightly lower. Although the MSE three times increased, by analyzing the error shapes, it can be concluded that most of this error is concentrated at the initial point of the trajectory, and the remaining part of the trajectory was followed by the robot with high accuracy.

6. CONCLUSION

In this work, an optimized intelligent control methodology was proposed for trajectory tracking tasks in four-wheeled omnidirectional mobile robots. Kinematics and dynamics modeling was developed to facilitate the control methodology design and development. A neural network Proportional Integral Derivative (PID) control methodology was designed for the trajectory-tracking tasks. Moreover, controller's neural network was manipulated via particle swarm optimization algorithm. The control approach robustness and effectiveness were verified in a number of simulation studies. The simulation results confirmed the high accuracy of the tracking tasks and the ability of the control method to start from an arbitrary initial point without retraining for the neural network. Future studies could be directed towards including the back propagation methodology in the feedback loop to improve motion accuracy. Moreover, a collision avoidance algorithm is worth investigation to have safe tracking. This study can be extended for future work by introducing other control strategies such as Augmented nonlinear PD controller, sliding mode control, Model reference adaptive control, robust adaptive control [32–39].

REFERENCES

[1] A. S. M. Al-Obaidi, A. A. Al-Qassar, A. R. Nasser, A. Ahmed, A. J. Humaidi, and I. K. Ibraheem, "Embedded design and implementation of mobile robot for surveillance applications," *Indonesian J. Sci. Technol.*, vol. 6, no. 2, pp. 427–40, 2021.

[2] B. Andrea-Novel, G. Bastin, and G. Campion, "Dynamic feedback linearization of nonholonomic wheeled mobile robots," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2527–32.

[3] H. C. Huang, "Intelligent motion control for omnidirectional mobile robots using ant colony optimization," *Appl. Artif. Intelligence*, vol. 27, no. 3, pp. 151–69, 2013. https://doi.org/10.1080/08839514.2013.768877.

[4] C. C. Shing, P. L. Hsu, and S. S. Yen, "T-S fuzzy path controller design for the omnidirectional mobile robot," in *IECON Proceedings (Industrial Electronics Conference)*, 2006, pp. 4142–7. https://doi.org/10.1109/IECON.2006.347314.

[5] K. Watanabe, "Control of an omnidirectional mobile robot, international conference on knowledge-based intelligent electronic systems," *Proc. KES*, vol. 1, no. April, pp. 51–60, 1998. https://doi.org/10.1109/kes.1998.725827.

[6] J. A. Vázquez and M. Velasco-Villa, "Computed-torque control of an omnidirectional mobile robot," in *2007 4th International Conference on Electrical and Electronics Engineering, ICEEE 2007, no. Iceee*, 2007, pp. 274–7. https://doi.org/10.1109/ICEEE.2007.4345021.

[7] A. Betourne and G. Campion, "Dynamic modelling and control design of a class of omnidirectional mobile robots," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996, pp. 2810–5.

[8] H. Sira-Ramírez, C. López-Uribe, and M. Velasco-Villa, "Real-time linear control of the omnidirectional mobile robot," in *Proceedings of the IEEE Conference on Decision and Control*, 2010, pp. 4263–8. https://doi.org/10.1109/CDC.2010.5717804.

[9] C. C. Tsai, Z. R. Wu, Z. C. Wang, and M. F. Hisu, "Adaptive dynamic motion controller design for a four-wheeled omnidirectional mobile robot," in *2010 International Conference on System Science and Engineering, ICSSE 2010, IEEE*, 2010, pp. 233–8. https://doi.org/10.1109/ICSSE.2010.5551786.

[10] H. C. Huang, T. F. Wu, C. H. Yu, and H. S. Hsu, "Intelligent fuzzy motion control of three-wheeled omnidirectional mobile robots for trajectory tracking and stabilization," in *2012 International Conference on Fuzzy Theory and Its Applications, iFUZZY 2012*, IEEE, 2012, pp. 107–12. https://doi.org/10.1109/iFUZZY.2012.6409684.

[11] Y. Song, D. Tan, and M. Miao, "Disturbance analysis and control for an omnidirectional wheeled mobile robot," in *2009 IEEE International Conference on Mechatronics and Automation, ICMA*

2009, 2009, pp. 3241–5. https://doi.org/10.1109/ICMA.2009.5246250.

[12] T. H. S. Li, C. Y. Chen, H. L. Hung, and Y. C. Yeh, "A fully fuzzy trajectory tracking control design for surveillance and security robots," in *Conference Proceedings - IEEE International Conference on Systems*, Man and Cybernetics, 2008, pp. 1995–2000. https://doi.org/10.1109/ICSMC.2008.4811583.

[13] N. Hung, D. H. Kim, H. K. Kim, and S. B. Kim, "Tracking controller design of omnidirectional mobile manipulator system," in *ICCAS-SICE 2009 - ICROS-SICE International Joint Conference 2009, Proceedings*, IEEE, 2009, pp. 539–44.

[14] O. A. R. A. Wahhab and A. S. Al-Araji, "Path planning and control strategy design for mobile robot based on hybrid swarm optimization algorithm," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, pp. 565–79, 2021.

[15] A. S. Al-Araji, M. F. Abbod, and H. S. Al-Raweshidy, "Design of an adaptive nonlinear PID controller for nonholonomic mobile robot based on posture identifier," in *Proceedings - 2011 IEEE International Conference on Control System, Computing and Engineering, ICCSCE*, pp. 337–42, 2011.

[16] A. S. Al-Araji, K. E. Dagher, and B. A. Ibraheem, "An intelligent cognitive system design for mobile robot based on optimization algorithm," in *The 3rd Scientific Conference of Electrical Engineering, SCEE*, pp. 84–9, 2018.

[17] H. P. Oliveira, A. J. Sousa, A. P. Moreira, and P. J. Costa, "Modeling and assessing of omni-directional robots with three and four wheels," in *Contemporary Robotics, IntechOpen*, A. D. Rodi, Ed., Rijeka, 2009. Ch. 12. https://doi.org/10.5772/7796.

[18] A. S. Conceição, A. P. Moreira, and P. J. Costa, "Model's parameters experimental identification of a four wheeled omni-directional mobile robot," *IFAC Proc. Volumes*, vol. 39, no. 15, pp. 230–5, 2006. https://doi.org/10.3182/20060906-3-it-2910.00040.

[19] H. P. Oliveira, A. J. Sousa, A. P. Moreira, and P. J. Costa, "Precise modeling of a four wheeled omni-directional robot," in *Proc. of the 8th conference on autonomous robot systems and competitions*, 2008, pp. 57–62.

[20] H. Shu and Y. Pi, "PID neural networks for time-delay systems," *Comput. Chem. Eng.*, vol. 24, nos 2–7, pp. 859–62, 2000. https://doi.org/10.1016/s0098-1354(00)00340-9.

[21] R. Poli, J. Kennedy, and T. Blackwell, *Particle Swarm Optimization: an Overview, Swarm Intelligence*, Springer, 2007, pp. 33–57.

[22] X. Jian and L. Zhao, "An improved particle swarm optimization algorithm for minlp problems," in *2009 WRI Global Congress on Intelligent Systems*, vol. 1, 2009, pp. 159–62.

[23] A. J. Humaidi and H. M. Badr, "Linear and Nonlinear Active Disturbance Rejection Controllers for single-link flexible joint robot manipulator based on PSO tuner," *J. Eng. Sci. Technol. Rev.*, vol. 11, no. 3, pp. 133–8, 2018.

[24] A. J. Humaidi, S. K. Kadhim, and A. S. Gataa, "Optimal adaptive magnetic suspension control of rotary impeller for artificial heart pump," *Cybernetics Syst.*, vol. 53, no. 1, 2022.

[25] A. A. Al-Qassar, A. Q. Al-Dujaili, A.F. Hasan, A. J. Humaidi, I.K. Ibraheem, and A.T. Azar, "Stabilization of single-axis propeller-powered system for aircraft applications based on optimal adaptive control design," *J. Eng. Sci. Technol.*, vol. 16, no. 3, pp. 1851–69, 2021.

[26] A. J. Humaidi, S. K. Kadhim, and A. S. Gataa, "Development of a novel optimal backstepping control algorithm of magnetic impeller-bearing system for artificial heart ventricle pump," *Cybernetics Syst.*, vol. 51, no. 4, pp. 521–41, 2020.

[27] V. Fathi and G. A. Montazer, "An improvement in RBF learning algorithm based on PSO for real time applications," *Neurocomputing*, vol. 111, pp. 169–76, 2013. https://doi.org/10.1016/j.neucom.2012.12.024.

[28] A. A. Al-Qassar, A. I. Abdu-Alkareem, A. F. Hasan, A. J. Humaidi, I. K. Ibraheem, A. T. Azar, and A. H. Hameed, "Grey-wolf optimization better enhances the dynamic performance of roll motion for tail-sitter VTOL aircraft guided and controlled by STSMC," *J. Eng. Sci. Technol. (Jestec)*, vol. 16, no. 3, pp. 1932–50, 2021.

[29] T. Ghanim, A. R. Ajel, and A. J. Humaidi, "Optimal fuzzy logic control for temperature control based on social spider optimization," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 745, no. 1, 2020, Paper no. 012099.

[30] A. A. Al-Qassar, A. S. M. Al-Obaidi, A. F. Hasan, A. R. Nasser, A. Alkhayyat, and I. K. Ibraheem, "Finite-time control of wing-rock motion for delta wing aircraft based on whale-optimization algorithm," *Indonesian J. Sci. Technol.*, vol. 6, no. 3, pp. 441–56, 2021.

[31] A. A. Al-Azza, A. A. Al-Jodah, and F. J. Harackiewicz, "Spider monkey optimization: a novel technique for antenna optimization," *IEEE Antennas Wireless Propagation Lett.*, vol. 15, pp. 1016–9, 2016.

[32] A. J. Humaidi and H. A. Hussein, "Adaptive control of parallel manipulator in cartesian space," in *Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT* 2019, 8869257.

[33] A. J. Humaidi and A. H. Hameed, "Robustness enhancement of MRAC using modification techniques for speed control of three phase induction motor," *J. Electr. Syst.*, vol. 13, no. 4, pp. 723–41, 2017.

[34] M. Y. Hassan, A. J. Humaidi, and M. K. Hamza, "On the design of backstepping controller for Acrobot system based on adaptive observer," *Int. Rev. Electr. Eng.*, vol. 15, no. 4, pp. 328–35, 2020.

[35] A. H. Hameed, A. Q. Al-Dujaili, A. J. Humaidi, and H. A. Hussein, "Design of terminal sliding position control for electronic throttle valve system: a performance comparative study," *Int. Rev. Automatic Control*, vol. 12, no. 5, pp. 251–60, 2019.

[36] A. J. Humaidi and A. I. Abdulkareem, "Design of augmented nonlinear PD controller of Delta/Par4-like robot," *J. Control Sci. Eng.*, 2019, Paper no. 7689673.

[37] A. J. Humaidi, A. H. Hameed and M. R. Hameed, "Robust adaptive speed control for DC motor using novel weighted E-modified MRAC," in I*EEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI* 2017, 2018, pp. 313–9.

[38] A. J. Humaidi and A. H. Hameed, "Design and comparative study of advanced adaptive control schemes for position control of electronic throttle valve," *Information (Switzerland)*, vol. 10, no. 2, p. 65, 2019.

[39] A. J. Humaidi, S. Hasan, and A. A. Al-Jodah, "Design of second order sliding mode for glucose regulation systems with disturbance," *Int. J. Eng. Technol. (Uae)*, vol. 7, no. 2, pp. 243–7, 2018.