frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 14 June 2024
DOI 10.3389/frobt.2024.1386968

Check for updates

OPEN ACCESS

EDITED BY
Yanan Li,
University of Sussex, United Kingdom

REVIEWED BY
Chengzhi Zhu,
South China University of Technology, China
Hassène Gritli,
Carthage University, Tunisia

*CORRESPONDENCE
Ahmad Taher Azar,
✉ aazar@psu.edu.sa,
✉ ahmad.azar@fci.bu.edu.eg

RECEIVED 16 February 2024
ACCEPTED 25 April 2024
PUBLISHED 14 June 2024

CITATION
Jasim Mohamed M, Oleiwi BK, Azar AT and Mahlous AR (2024), Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm.
Front. Robot. AI 11:1386968.
doi: 10.3389/frobt.2024.1386968

Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm

Mohamed Jasim Mohamed[1], Bashra Kadhim Oleiwi[1], Ahmad Taher Azar[2,3,4]* and Ahmed Redha Mahlous[2,3]

[1]Control and Systems Engineering Department, University of Technology-Iraq, Baghdad, Iraq, [2]College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia, [3]Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia, [4]Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

The performance of the robotic manipulator is negatively impacted by outside disturbances and uncertain parameters. The system's variables are also highly coupled, complex, and nonlinear, indicating that it is a multi-input, multi-output system. Therefore, it is necessary to develop a controller that can control the variables in the system in order to handle these complications. This work proposes six control structures based on neural networks (NNs) with proportional integral derivative (PID) and fractional-order PID (FOPID) controllers to operate a 2-link rigid robot manipulator (2-LRRM) for trajectory tracking. These are named as set-point-weighted PID (W-PID), set-point weighted FOPID (W-FOPID), recurrent neural network (RNN)-like PID (RNNPID), RNN-like FOPID (RNN-FOPID), NN+PID, and NN+FOPID controllers. The zebra optimization algorithm (ZOA) was used to adjust the parameters of the proposed controllers while reducing the integral-time-square error (ITSE). A new objective function was proposed for tuning to generate controllers with minimal chattering in the control signal. After implementing the proposed controller designs, a comparative robustness study was conducted among these controllers by altering the initial conditions, disturbances, and model uncertainties. The simulation results demonstrate that the NN+FOPID controller has the best trajectory tracking performance with the minimum ITSE and best robustness against changes in the initial states, external disturbances, and parameter uncertainties compared to the other controllers.

KEYWORDS

neural network, recurrent neural network, set-point controller, proportional integral derivative controller, fractional-order PID controller, 2-link rigid robot manipulator, zebra optimization algorithm

1 Introduction

The field of robotics mainly focuses on problems related to visualization, modeling, and control. Robots are used in many daily tasks and occupations in every aspect of modern life. Robotic manipulators are increasingly required in factories and industries as they play important roles in the operations instead of humans, especially when these operations involve risky, repetitive, and complex activities (Oleiwi et al., 2021). The use of robots has also become necessary to ensure efficient, quick, and accurate operations. Traditional robots are large and bulky since they contain stiff linkages throughout their construction; most industries need upgrades to the current classical robots to lower the building costs, minimize energy consumption brought on by the large actuators, and boost production (Alandoli and Tian, 2020). Since robotic manipulators are well-suited for many applications, particularly in the industrial field, they have been widely used for many years. The trajectory tracking control is an important issue from the viewpoint of automatic control because various applications, such as welding, screwing, moving cars or equipment parts, and painting, demand precise trajectory tracking to accomplish their objectives (Azar and Fernando, 2019). The complexity and non-linearity of a robotic manipulator make it impossible for proportional integral derivative (PID) controllers to provide effective trajectory tracking and constant force/twist control simultaneously. The robotic manipulator also experiences a number of uncertainties, external disturbances, payload variations, and parameter variations during operation (Dachang et al., 2020; Abdulameer and Mohamed, 2022). To design controllers that can handle the dynamics of the manipulator robot for controlling and trajectory tracking, many solutions have been proposed using traditional control systems (Ajeil et al., 2020; Ibraheem et al., 2020; Najm et al., 2020). Sharma et al. (2015) described the design and analysis of a fractional-order PID (FOPID) controller with two degrees of freedom (DOFs) based on the cuckoo search algorithm for a two-link rigid robot manipulator (2-LRRM) with a payload; their results indicated that the suggested strategy improves the performance of the closed-loop system by resolving robustness and disturbance rejection issues. Kumar (2017) proposed an interval type-2 fuzzy proportional derivative plus integral controller based on the genetic algorithm for a 5-DOF redundant robot manipulator. Cao et al. (2021) introduced an invariant control structure for manipulator robot trajectory tracking with input saturation and uncertainty by combining reinforcement learning and non-singular terminal sliding-mode control. Shuyang and John (2021) applied a neural network (NN) as a multilayer perceptron structure based on the iterative process of learning; here, the desired robot joint was employed as the input, and the desired robot motion was related to the output; the movement of the robot with respect to the intended set of joint paths is determined by the iterative learning control. Shojaei et al. (2020) suggested a specific performance-based adaptive neural control system for manipulator robots without considering the input current, acceleration, or velocity; this scheme includes the actuator dynamics under model uncertainty, and an acceleration velocity observer was coupled with a neural adaptive second-order PID controller. Nohooji (2020) addressed the unknown dynamics and outside disturbances of a manipulator robot to design a control approach utilizing NN-based radial basis activation functions and self-tuning PID control. Zhou et al. (2020) and Kareem et al. (2023) introduced fractional-order sliding-mode control using a deep convolutional NN for controlling trajectory tracking in manipulator robots; here, the controller switching gain decreased drastically since the NN corrects the uncertainty of the system without knowledge of the upper boundaries. Four distinguished non-linear control structures were studied by Jenhani et al. (2022) to address the problem of controlling and stabilizing robotic systems to predefined positions. To control the position and velocity of the 2-link robot, classical and adaptive sliding-mode controllers were introduced by Al-Hadithy and Hammoudi (2020) as well as Hameed and Hamoudi (2023). Hamoudi and Rasheed (2023) used particle swarm optimization to study the effectiveness of adaptive and classical backstepping control schemes for non-linear systems.

It is well-known in intelligent control that the NN controller can be used to solve various control issues, particularly when the controlled plant displays non-linearity and/or uncertainties in the model parameters. The advantage of the NN is that it has solid capability for mapping. Conversely, the PID controller is the most widely used controller in the industry because of its robust performance under numerous operating conditions and straightforward design. Therefore, in our proposed controllers, we merge the advantages of NNs with those of the PID and FOPID controllers to obtain hybrid controllers based on the zebra optimization algorithm (ZOA) to control a 2-LRRM.

The following are the main contributions of this work:

1. Six control structures are proposed based on NNs with PID/FOPID operations.
2. ZOA is used to adjust the gains of the proposed control structures on the basis of reducing the integral-time-square error (ITSE) performance index.
3. A comparative robustness study is conducted among the proposed control structures by altering the initial conditions, disturbances, and model uncertainties.
4. A new objective function is proposed for the tuning process to obtain controllers with minimal chattering in the control signals.

The remainder of this paper is structured as follows: the 2-LRRM's mathematical model is described in Section 2. The proposed controller structures are discussed in Section 3. The ZOA is explained in Section 4. The simulation results are shown in Section 5; the robustness tests are presented in Section 6, and the conclusions are presented in Section 7.

2 Mathematical model of the 2-LRRM

The 2-LRRM structure is illustrated in Figure 1. It is composed of two links with lengths $\boldsymbol{l}_{j1}$ and $\boldsymbol{l}_{j2}$ and two mass centers $\boldsymbol{m}_{j1}$ and $\boldsymbol{m}_{j2}$ that are at the distal ends of the links. Encoders determine the angular positions of the links ($\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$) and velocities ($\dot{\boldsymbol{\theta}}_1$ and $\dot{\boldsymbol{\theta}}_2$), while the regulating torque is produced at points A and B by DC motors (Mohan et al., 2018). In robotics, the fundamental control equations are constructed using the dynamic equation of robot motion. In a robotic system, the actuator torque is employed to produce the dynamic motion of the manipulator arm. The dynamic modeling of a robotic system is characterized by the connections between the temporal rates of change and input torques of the robot arm component configurations. The primary goal here is to determine how to calculate the robot's motion equations given the moments and forces exerted on it. Therefore, a part of the robot manipulator's dynamic modeling describes the joint locations, velocities, and accelerations in addition to the functions mapping the forces acting on the structures (Raafat and Raheem, 2017).

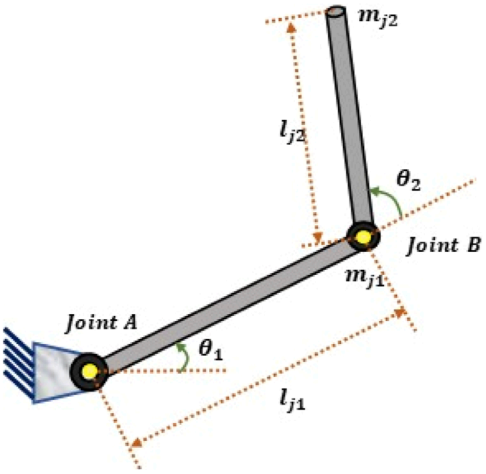

FIGURE 1
Structure of the 2-LRRM.

The equations describing the x and y positions of m_{j1} and m_{j2}, the equation for the kinetic and potential energies, and the two derived coupled non-linear differential equations based on the Euler–Lagrange equation are presented in Lewis et al. (2004) and Raafat and Raheem (2017).

The conventional form can be applied to the manipulator dynamics as indicated in Eqs. (1-10)

$$M(\theta)\ddot{\theta} + V\left(\theta,\dot{\theta}\right) + g(\theta) = T, \tag{1}$$

where $V\left(\theta,\dot{\theta}\right)$ is the Coriolis/centripetal vector, $M(\theta)$ is the inertia matrix that exhibits symmetry, and $g(\theta)$ is the gravity vector.

$$M = \begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix}, \tag{2}$$

$$M_{11} = \left(m_{j1} + m_{j2}\right)l_{j1}^2 + m_{j2}l_{j2}^2 + 2m_{j2}l_{j1}l_{j2}\cos(\theta_2), \tag{3}$$

$$M_{12} = m_{j2}l_{j2}^2 + m_{j2}l_{j1}l_{j2}\cos(\theta_2), \tag{4}$$

$$M_{12} = M_{21} \quad \& \quad M_{22} = m_{j2}l_{j2}^2. \tag{5}$$

The Coriolis/centrifugal vector denoted by V is of the form

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \tag{6}$$

$$V_1 = -m_{j2}l_{j1}, \tag{7}$$

$$V_2 = m_{j2}l_{j1}l_{j2}\dot{\theta}_1^2\sin(\theta_2). \tag{8}$$

The gravity vector $g = [g_{12}g_{21}]^T$ is defined using

$$g_{12} = \left(m_{j1} + m_{j2}\right)gl_{j1}\cos(\theta_1) + m_{j2}gl_{j2}\cos(\theta_1 + \theta_2), \tag{9}$$

$$g_{21} = m_{j2}gl_{j2}\cos(\theta_1 + \theta_2). \tag{10}$$

The 2-LRRM parameter specifications are as listed in Table 1 (Mohamed et al., 2023).

TABLE 1 Specifications of the 2-LRRM.

Parameter	Nominal value
m_{j1}	0.1 kg
m_{j2}	0.1 kg
l_{j1}	0.8 m
l_{j2}	0.4 m
g	9.81 m/s^2

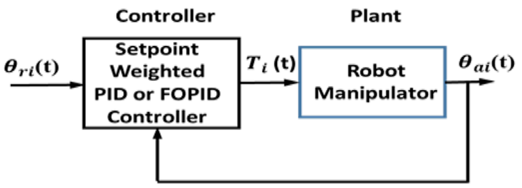

FIGURE 2
Set-point-weighted controller block diagram.

3 Structures of the proposed hybrid controllers

This section provides the descriptions of the proposed controllers below.

3.1 Set-point-weighted PID and FOPID controllers

This section discusses set-point-weighted PID and FOPID controllers. The block diagram of this feedback control system is indicated in Figure 2.

The equation describing the set-point-weighted PID (W-PID) controller is presented in Eq. (11):

$$T_i(t)_{PID} = K_p((1-\beta)\theta_{ri}(t) - \theta_{ai}(t)) + K_i\int(\theta_{ri}(t) - \theta_{ai}(t))\,dt + K_d\frac{d}{dt}((1-\alpha)\theta_{ri}(t) - \theta_{ai}(t)), \tag{11}$$

where $e_{\theta i}(t) = \theta_{ri}(t) - \theta_{ai}(t), 0 < \beta < 1\,,\, 0 < \alpha < 1$.

Here, $e_{\theta i}(t)$ is the error between the required and calculated positions, $\theta_{ri}(t)$ and $\theta_{ai}(\mathrm{t})$, respectively, of the i th link; K_p, K_i, and K_d are the proportional, integral, and derivative gains of the PID controller, respectively; $T_i(t)$ is the control signal (torque) of the i th link; β and α are constants. The equation of the set-point-weighted FOPID (W-FOPID) controller contains two additional parameters (λ, μ) with fractional values, as shown in Eq. (12):

$$T_i(t)_{FOPID} = K_p((1-\beta)\theta_{ri}(t) - \theta_{ai}(t)) + K_iD^{-\lambda}(\theta_{ri}(t) - \theta_{ai}(t)) + K_dD^{\mu}((1-\alpha)\theta_{ri}(t) - \theta_{ai}(t)), \tag{12}$$

where $0 < \lambda < 2\,,\, 0 < \mu < 2$, and D represents the Laplace variable.

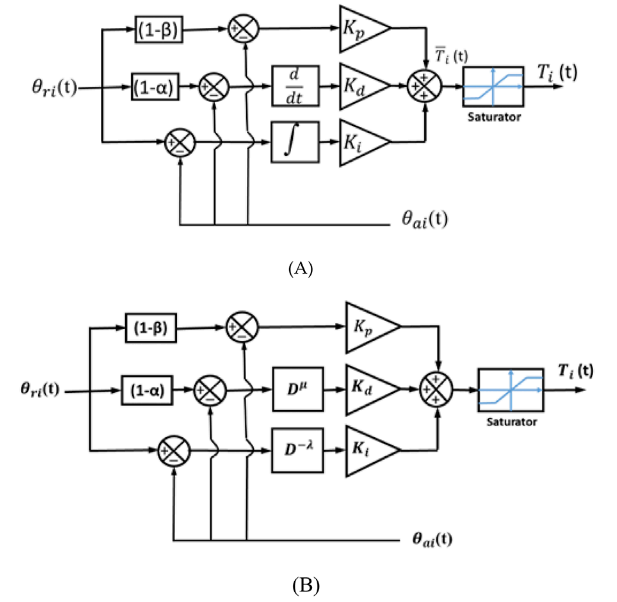

FIGURE 3
Set-point-weighted controller structures: **(A)** W-PID and **(B)** W-FOPID.

In Eqs. (11, 12), α and β are used to adjust the set-point values before comparing with the output to calculate the error signal for each control action. Here, β is used to attenuate the set-point signal before computing the corresponding error for the proportional action and α is used similarly for the derivative action, while the set-point signal attenuation is not used for the integral action. This PID configuration is unconventional, famous, and used in many works.

The set-point-weighted PID and FOPID controller structures are demonstrated in Figure 3.

3.2 Recurrent neural network (RNN)-like PID and FOPID controllers

In these two controllers, RNN-like PID and FOPID are adopted. The block diagram of the feedback control system for this type of hybrid controller is shown in Figure 4. The common structure of the RNN-PID and RNN-FOPID controllers is shown in Figure 5. For the conventional PID controller, the order variables have integer values $\boldsymbol{\lambda = 1}$ and $\boldsymbol{\mu = 1}$. $\boldsymbol{e_{\theta i}(t)}$ is the error between the required and actual positions $\boldsymbol{\theta_{ri}(t)}$ and $\boldsymbol{\theta_{ai}(t)}$, respectively, of the ith link.

In the RNN-PID controller, the input layer has a single neuron $\boldsymbol{e_{\theta i}(t)}$. The first hidden layer has three neurons $\boldsymbol{P}$, $\boldsymbol{I}$, and $\boldsymbol{D}$ that are defined in Eqs (13–15):

$$\boldsymbol{P(t) = K_p e_{\theta i}(t) \;\; or \;\; P(k) = K_p e_{\theta i}(k),} \tag{13}$$

$$\boldsymbol{I(t) = K_i \int e_{\theta i}(t)\,dt \;\; or \;\; I(k) = K_i \sum_{j=0}^{k} e_i(j),} \tag{14}$$

$$\boldsymbol{D(t) = K_d \frac{d}{dt} e_{\theta i}(t) \quad or\, D(k) = K_d \left(e_{\theta i}(k) - e_{\theta i}(k-1)\right)/h,} \tag{15}$$

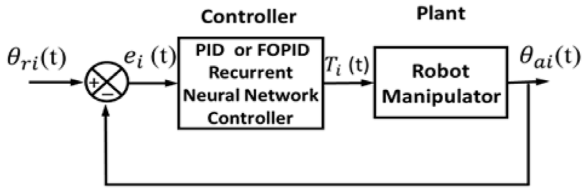

FIGURE 4
Recurrent-neural-network-like PID and FOPID controller block diagrams (RNN-PID and RNN-FOPID).

where, the feedback and processing elements in the second hidden layer are defined in Eqs. (16-17)

$$\begin{bmatrix} \boldsymbol{N_1^1(k)} \\ \boldsymbol{N_2^1(k)} \\ \boldsymbol{N_3^1(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v11} & \boldsymbol{v12} & \boldsymbol{v13} \\ \boldsymbol{v21} & \boldsymbol{v22} & \boldsymbol{v23} \\ \boldsymbol{v31} & \boldsymbol{v32} & \boldsymbol{v33} \end{bmatrix} \begin{bmatrix} \boldsymbol{P(k)} \\ \boldsymbol{I(k)} \\ \boldsymbol{D(k)} \end{bmatrix} \tag{16}$$

and

$$\begin{bmatrix} \boldsymbol{C_1^1(k)} \\ \boldsymbol{C_2^1(k)} \\ \boldsymbol{C_3^1(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{O_1^1(k-1){+}p1\;\; C_1^1(k-1)} \\ \boldsymbol{O_2^1(k-1){+}p2\;\; C_2^1(k-1)} \\ \boldsymbol{O_3^1(k-1)\,{+}p3\, C_3^1(k-1)} \end{bmatrix}. \tag{17}$$

The output of the second hidden layer is given by Eq. (18):

$$\begin{bmatrix} \boldsymbol{O_1^1(k} \\ \boldsymbol{O_2^1(k)} \\ \boldsymbol{O_3^1(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H(N_1^1(k))} \\ \boldsymbol{H(N_2^1(k))} \\ \boldsymbol{H(N_3^1(k))} \end{bmatrix} + \begin{bmatrix} \boldsymbol{vc11} & \boldsymbol{vc12} & \boldsymbol{vc13} \\ \boldsymbol{vc21} & \boldsymbol{vc22} & \boldsymbol{vc23} \\ \boldsymbol{vc31} & \boldsymbol{vc32} & \boldsymbol{vc33} \end{bmatrix} \begin{bmatrix} \boldsymbol{C_1^1(k)} \\ \boldsymbol{C_2^1(k)} \\ \boldsymbol{C_3^1(k)} \end{bmatrix}. \tag{18}$$

The activation function used is a sigmoid function, as shown in Eq. (19):

$$\boldsymbol{H = \frac{2}{\left(1 + e^{-net}\right)} - 1,} \tag{19}$$

where, the feedback elements in the third hidden layer are defined in Eq. (20)

$$\begin{bmatrix} \mathbf{C_1^2(k)} \\ \mathbf{C_2^2(k)} \\ \mathbf{C_3^2(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{O_1^2(k-1){+}pp1\, C_1^2(k-1)} \\ \mathbf{O_2^2(k-1){+}pp2\, C_2^2(k-1)} \\ \mathbf{O_3^2(k-1){+}pp3\, C_3^2(k-1)} \end{bmatrix}. \tag{20}$$

The output of the third hidden layer is expressed in Eq. (21):

$$\begin{bmatrix} \boldsymbol{O_1^2(k)} \\ \boldsymbol{O_2^2(k)} \\ \boldsymbol{O_3^2(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{vv11} & \boldsymbol{vv12} & \boldsymbol{vv13} \\ \boldsymbol{vv21} & \boldsymbol{vv22} & \boldsymbol{vv23} \\ \boldsymbol{vv31} & \boldsymbol{vv32} & \boldsymbol{vv33} \end{bmatrix} \begin{bmatrix} \boldsymbol{O_1^1(k} \\ \boldsymbol{O_2^1(k)} \\ \boldsymbol{O_3^1(k)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{vvc11} & \boldsymbol{vvc12} & \boldsymbol{vvc13} \\ \boldsymbol{vvc21} & \boldsymbol{vvc22} & \boldsymbol{vvc23} \\ \boldsymbol{vvc31} & \boldsymbol{vvc32} & \boldsymbol{vvc33} \end{bmatrix} \begin{bmatrix} \boldsymbol{C_1^2(k)} \\ \boldsymbol{C_2^2(k)} \\ \boldsymbol{C_3^2(k)} \end{bmatrix}. \tag{21}$$

The output of the single neuron in the output layer is given in Eq. (22):

$$\boldsymbol{T_i(k) = w_1\, O_1^2(k) + w_2\, O_2^2(k) + w_3\, O_3^2(k),} \tag{22}$$

where $\boldsymbol{K_p, K_i, K_d, vij, vcij\, vvij, vvcij, w_i, p_i,}$ and $\boldsymbol{pp_i}$ are all parameters.

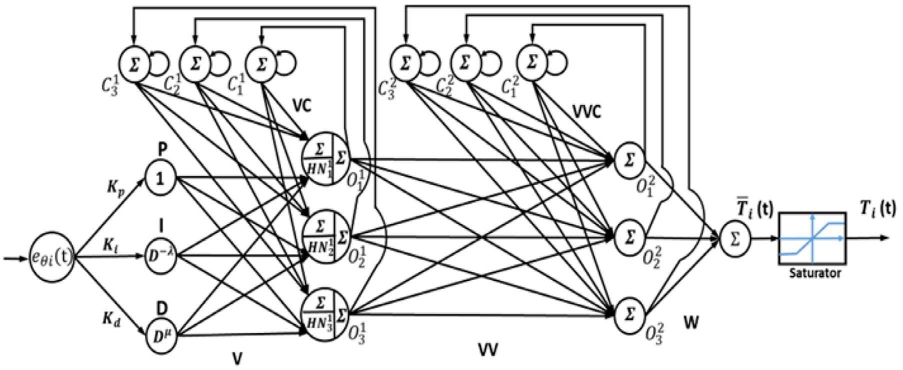

FIGURE 5
Recurrent-neural-network-like PID and FOPID structures (RNN-PID and RNN-FOPID controllers).

In the RNN-FOPID controller, the input layer again has a single neuron $e_{\theta i}(t)$. The first hidden layer has three neurons *P, I,* and *D*, where the order variables $0 < \lambda < 2$ and $0 < \mu < 2$ are fractions instead of integers, and the *P, I,* and *D* neurons of this hidden layer are shown in Eqs (23–25). All the remaining hidden layers and output layer are the same as in the RNN-PID controller.

$$P(t) = K_p e_{\theta i}(t), \tag{23}$$

$$I(t) = K_i D^{-\lambda} e_{\theta i}(t), \tag{24}$$

$$\mathrm{D(t) = K_d D^{\mu} e_{\theta i}(t)}. \tag{25}$$

3.3 NN-based PID and FOPID controllers

In this type of controller, the NN and PID/FOPID controllers both contribute to the production of the control signal. Figure 6 displays the feedback control system block diagram for this type of controller.

The common structure for these two controllers is indicated in Figure 7.

For the conventional PID controller, the order variables have integer values $\lambda = 1$ and $\mu = 1$. There are three neurons in the input layer of the NN + PID controller structure, namely, $e_{\theta i}(k), e_{\theta i}(k-1),$ and $e_{\theta i}(k-2)$ or A, B, and C.

Thus, the first layer of hidden neurons is given by Eq. (26):

$$\begin{bmatrix} N_1^1(k) \\ N_2^1(k) \\ N_3^1(k) \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \\ v31 & v32 & v33 \end{bmatrix} \begin{bmatrix} e_{\theta i}(k) \\ e_{\theta i}(k-1) \\ e_{\theta i}(k-2) \end{bmatrix} + \begin{bmatrix} N_1^1(k-1) \\ N_2^1(k-1) \\ N_3^1(k-1) \end{bmatrix}. \tag{26}$$

The output of the first hidden layer is given by Eq. (27):

$$\begin{bmatrix} O_1^1(k) \\ O_2^1(k) \\ O_3^1(k) \end{bmatrix} = \begin{bmatrix} H(N_1^1(k)) \\ H(N_2^1(k)) \\ H(N_3^1(k)) \end{bmatrix}, \tag{27}$$

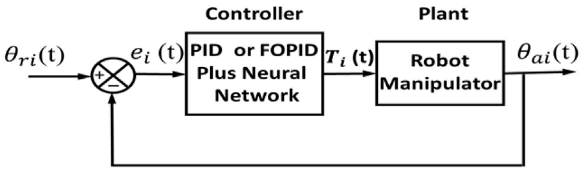

FIGURE 6
Block diagram of a neural network combined with PID and FOPID controllers.

$$\begin{bmatrix} N_1^2(k) \\ N_2^2(k) \\ N_3^2(k) \end{bmatrix} = \begin{bmatrix} vv11 & vv12 & vv13 \\ vv21 & vv22 & vv23 \\ vv31 & vv32 & vv33 \end{bmatrix} \begin{bmatrix} O_1^1(k) \\ O_2^1(k) \\ O_3^1(k) \end{bmatrix} + \begin{bmatrix} N_1^2(k-1) \\ N_2^2(k-1) \\ N_3^2(k-1) \end{bmatrix}. \tag{28}$$

The input and output of the second hidden layer are given by Eqs (28, 29), respectively.

$$\begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix} = \begin{bmatrix} H(N_1^2(k)) \\ H(N_2^2(k)) \\ H(N_3^2(k)) \end{bmatrix}. \tag{29}$$

The activation function used is a sigmoid function, as shown in Eq. (30):

$$H = \frac{2}{(1 + e^{-net})} - 1. \tag{30}$$

The output of the third hidden layer is given by Eq. (31):

$$\begin{bmatrix} O_1^3(k) \\ O_2^3(k) \\ O_3^3(k) \end{bmatrix} = \begin{bmatrix} N_1^3(k) \\ N_2^3(k) \\ N_3^3(k) \end{bmatrix} = \begin{bmatrix} w11 & w12 & w13 \\ w21 & w22 & w23 \\ w31 & w32 & w33 \end{bmatrix} \begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix}. \tag{31}$$

Equations (32–34) define the three control actions of the PID controller, and each of these control actions is added to one of the neuron outputs of the NN, as shown in Eqs. (35–37).

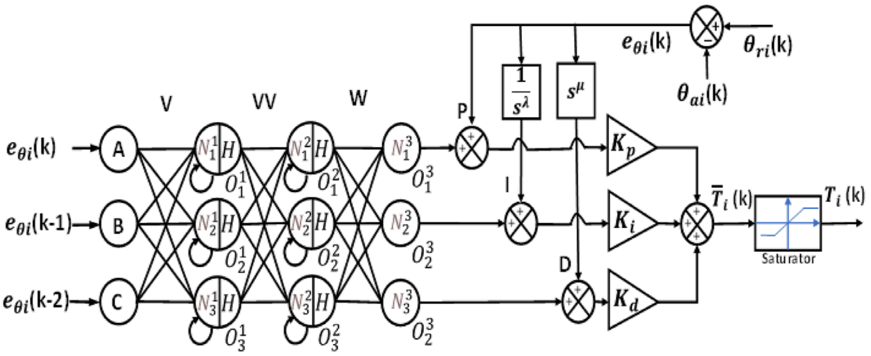

FIGURE 7
Neural network combined with PID and FOPID controller structures (NN + PID and NN + FOPID).

$$P(t) = e_{\theta i}(t) \quad or \quad P(k) = e_{\theta i}(k), \tag{32}$$

$$I(t) = \int e_{\theta i}(t)\,dt \quad or \quad I(k) = \sum_{j=0}^{k} e_{\theta i}(j), \tag{33}$$

$$D(t) = \frac{d}{dt} e_{\theta i}(t) \quad or \quad D(k) = \left(e_{\theta i}(k) - e_{\theta i}(k-1)\right)/h, \tag{34}$$

$$u_1(k) = K_p\left(O_1^3(k) + e_{\theta i}(k)\right), \tag{35}$$

$$u_2(k) = K_i\left(O_2^3(k) + \sum_{j=0}^{k} e_{\theta i}(j)\right), \tag{36}$$

$$u_3(k) = K_d\left(O_3^3(k) + \left(e_{\theta i}(k) - e_{\theta i}(k-1)\right)/h\right). \tag{37}$$

The equation of the control signal is expressed by Eq. (38):

$$\overline{T}_i(k) = u_1(k) + u_2(k) + u_3(k). \tag{38}$$

The structure of the NN + FOPID controller is the same as that of the NN + PID controller, with the difference being fractional-order operations of the integral and derivative functions given by $0 < \lambda < 2$ and $0 < \mu < 2$, respectively, as illustrated in Eqs. (39–44).

$$P(k) = e_{\theta i}(k), \tag{39}$$

$$I(k) = D^{-\lambda} e_{\theta i}(k), \tag{40}$$

$$D(k) = D^{\mu} e_{\theta i}(k), \tag{41}$$

$$u_1(k) = K_p\left(O_1^3(k) + e_{\theta i}(k)\right), \tag{42}$$

$$u_2(k) = K_i\left(O_2^3(k) + D^{-\lambda} e_{\theta i}(k)\right), \tag{43}$$

$$u_3(k) = K_d\left(O_3^3(k) + D^{\mu} e_{\theta i}(k)\right). \tag{44}$$

The equation of the control signal is given by Eq. (45):

$$\overline{T}_i(k) = u_1(k) + u_2(k) + u_3(k). \tag{45}$$

4 Zebra optimization algorithm

The ZOA is a nature-inspired metaheuristic algorithm and is presented mathematically in this section (Trojovská et al., 2022). The actions of zebras in the wild serve as the primary source of inspiration for the ZOA, where the foraging behaviors and defense mechanisms against predator attacks are simulated. The description is provided first, followed by the mathematical modeling of the ZOA steps. Effective real-world optimization issues can be resolved by the ZOA by achieving an appropriate balance between exploration and exploitation.

- Initialization

The population of zebras that provide a solution to the problem can be numerically modeled using a matrix, in addition to the plain where the zebras are located inside the search space. Within the search area, the zebras are positioned randomly at their starting positions. Equation (46) provides the structure of the ZOA population matrix.

$$X = \begin{bmatrix} X_1 \\ X_i \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & x_{1,j} & x_{1,m} \\ x_{i,1} & x_{i,j} & x_{i,m} \\ x_{N,1} & x_{N,j} & x_{N,m} \end{bmatrix}_{N\times m}, \tag{46}$$

where $\boldsymbol{X}$ is the population, N is the total number of initial solutions in the population, and m is the number of variables in each solution; X_i is the ith solution, and $x(i,j)$ is the value of the jth problem variable proposed by the ith solution. The values produced in the form of a vector for the objective function are specified by Eq. (47).

$$F = \begin{bmatrix} F_1 \\ F_i \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ F(X_i) \\ F(X_N) \end{bmatrix}_{N\times 1}, \tag{47}$$

where $\boldsymbol{F}$ represents the vector of objective function values and $\boldsymbol{Fi}$ denotes the value of the objective function for the ith solution. Each iteration involves two updates for the ZOA population members.

Phase 1: Foraging behavior

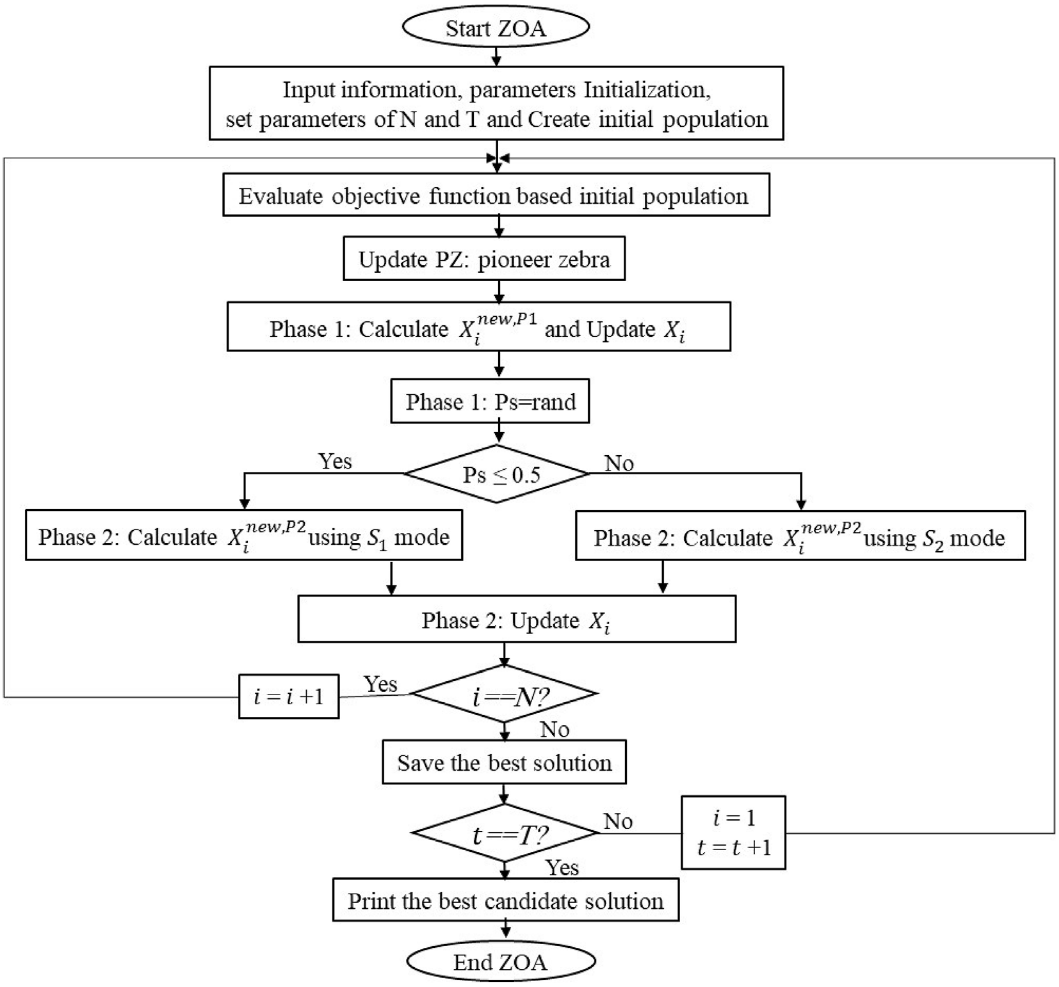
Start ZOA
Input information, parameters Initialization, set parameters of N and T and Create initial population
Evaluate objective function based initial population
Update PZ: pioneer zebra
Phase 1: Calculate $X_i^{new,P1}$ and Update X_i
Phase 1: Ps=rand
Yes
Ps ≤ 0.5
No
Phase 2: Calculate $X_i^{new,P2}$ using S_1 mode
Phase 2: Calculate $X_i^{new,P2}$ using S_2 mode
Phase 2: Update X_i
$i = i + 1$
Yes
$i == N?$
No
Save the best solution
$t == T?$
No
$i = 1$
$t = t + 1$
Yes
Print the best candidate solution
End ZOA

FIGURE 8
Flowchart of the ZOA.

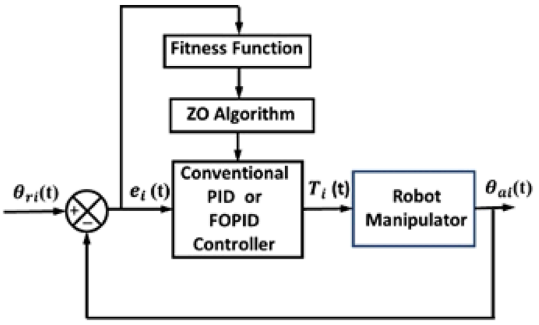

FIGURE 9
Block schematic diagram for tuning of a PID controller.

Zebra behavior models through forage seeking are employed to update the population members during the first phase (Pastor et al., 2006). Equations (48, 49) can be used in the mathematical model to update the positions of the zebras during the foraging phase.

TABLE 2 ITSE values of the suggested controllers for a nominal plant when using two initial positions [0.1745, 0.1745] and [−0.1745, −0.1745].

Controller	ITSE	Controller	ITSE
W-PID	9.4864×10^{-5}	W-FOPID	8.1697×10^{-5}
RNN-PID	9.0165×10^{-5}	RNN-FOPID	8.6837×10^{-5}
NN + PID	9.0259×10^{-5}	NN + FOPID	7.9676×10^{-5}

$$x_{i,j}^{new,P1} = x_{i,j} + r \cdot \left(PZ_j - I \cdot x_{i,j}\right), \quad (48)$$

$$X_i = \begin{cases} X_i^{new,P1}, & F_i^{new,p1}; \\ X_{i,} & else, \end{cases} \quad (49)$$

where $F_i^{new,P1}$ is the objective function value; $x_{i,j}^{new,P1}$ is ith zebra's new status based on the first phase; r is a random number in the interval [0,1]. The best member is $\boldsymbol{PZ}$, the pioneer zebra; its jth dimension is PZ_j; $I = round(1+rand)$ is provided; and rand is a random value within the interval [0,1]. Thus, $I \in \{1, 2\}$, and the population movement changes more noticeably if I = 2.

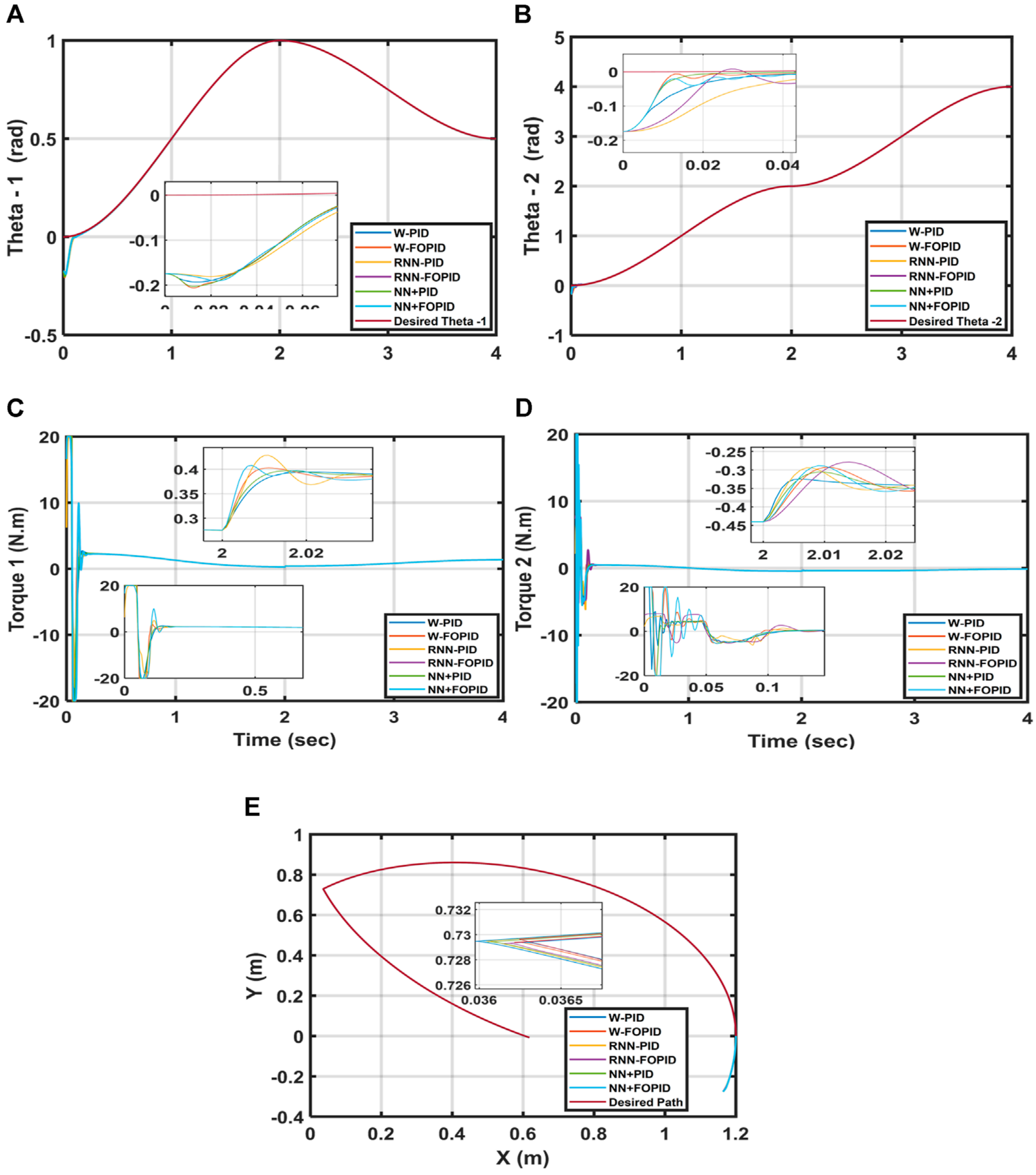

FIGURE 10
(A) Desired and calculated θ_1, **(B)** desired and calculated θ_2, **(C)** torque T_1, **(D)** torque T_2, and **(E)** end-effector trajectories of the 2-LRRM.

Phase 2: Predator defense strategies

The initial strategy for defense involves lions attacking the zebras, but the zebras move away from their current locations to escape (Pastor et al., 2006). Therefore, the mode S1 in Eq. (50) can be used to mathematically represent this strategy. In the second technique, when other predators attack one of the zebras, the other zebras in the herd move toward the attacked zebra in an attempt to confuse and intimidate the predator by forming a protective structure (Caro et al., 2014). Mode S2 in Eq. (50) is used to formally represent the zebra behaviors. When a zebra is updated, its new location is accepted if its objective function has a better value (Kennedy and Kennedy, 2013). This update scenario is represented using Eq. (51).

$$x_{i,j}^{new,P2} = \begin{cases} S_1 : x_{i,j} + R \cdot (2r-1) \cdot \left(1 - \dfrac{t}{T}\right) \cdot x_{i,j}, \; P_s \leq 0.5; \\ S_2 : x_{i,j} + r \cdot \left(AZ_j - I.x_{i,j}\right), \qquad else, \end{cases} \tag{50}$$

$$X_i = \begin{cases} X_i^{new,P2} \; F_i^{new,P2} < F_{i;} \\ X_i, \qquad else, \end{cases} \tag{51}$$

where Ps is the probability of choosing one of two randomly generated strategies in the interval [0, 1], AZ is the status of the attacked zebra and AZ_j is its jth dimension value, $F_i^{new,P2}$

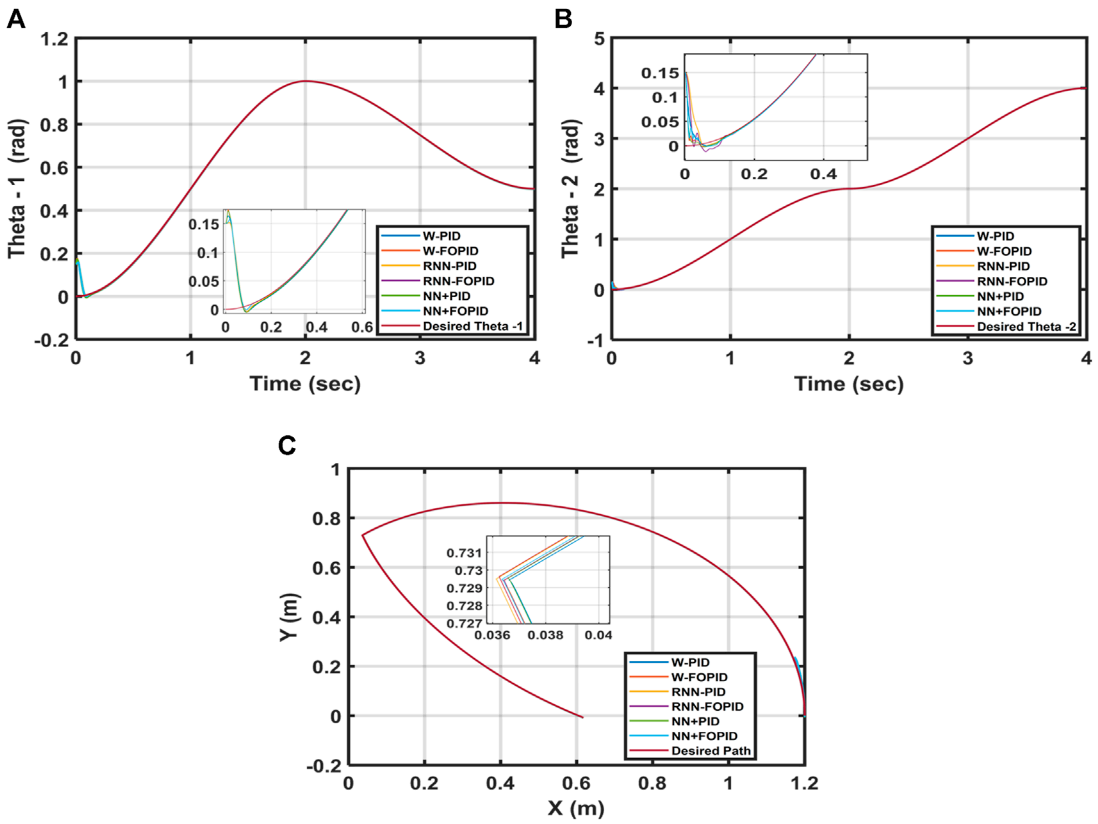

FIGURE 11
(A) Desired and calculated θ_1, **(B)**desired and calculated θ_2, and **(C)**2-LRRM end effector trajectories when the starting point is [0.15, 0.15].

TABLE 3 ITSE values of the proposed controllers for an initial position of [0.15, 0.15].

Controller	ITSE	Controller	ITSE
W-PID	3.6563×10^{-5}	W-FOPID	2.5491×10^{-5}
RNN-PID	2.5536×10^{-5}	RNN-FOPID	2.6496×10^{-5}
NN + PID	3.1708×10^{-5}	NN + FOPID	2.3871×10^{-5}

TABLE 4 ITSE values of the proposed controllers when adding 5% to the masses of both links and setting the starting position to [0.0, 0.0].

Controller	ITSE	Controller	ITSE
W-PID	1.0759×10^{-5}	W-FOPID	0.4012×10^{-5}
RNN-PID	0.1418×10^{-5}	RNN-FOPID	0.3324×10^{-5}
NN + PID	0.7397×10^{-5}	NN + FOPID	0.3139×10^{-5}

is its objective function value, and $X_i^{new,P2}$ is the new status of the *i*th zebra based on the second phase. The iteration contour is denoted by $\boldsymbol{t}$, maximum number of iterations is given by $\boldsymbol{T}$, and constant number $\boldsymbol{R}$ is set to 0.01. The value of its *j*th dimension is $X_i^{new,P2}$. The population members are updated depending on the first and second phases in each ZOA iteration. Until the time the algorithm is completely implemented, the population is updated based on Eqs. (48–51). During subsequent iterations, the best candidate solution is updated and preserved. When the ZOA is fully operational, the best potential answer is made available as the ideal response, as shown in the pseudocode and flowchart representations of the ZOA phases in Figure 8.

4.1 Pseudocode of the proposed ZOA

```
Begin ZOA.
Input: Information regarding the optimization issue.
Calculate the population size ($N$) and total number of iterations ($T$).
Evaluate the objective function based on the initial solution.
For $t = 1{:}\ T$, update $PZ$.
For $i = 1{:}\ N$
Phase 1: Foraging behavior
Use Eq. (48) to determine the new $i$th solution.
Use Eq. (49) to update the $i$th solution.
Phase 2: Predator defense strategies
Ps = rand if $Ps < 0.5$.
Strategy one: Lion-fighting phase
Use mode $S1$ in Eq. (50) to determine the new $i$th solution.
Else
Strategy two: Exploratory phase against other predators
```

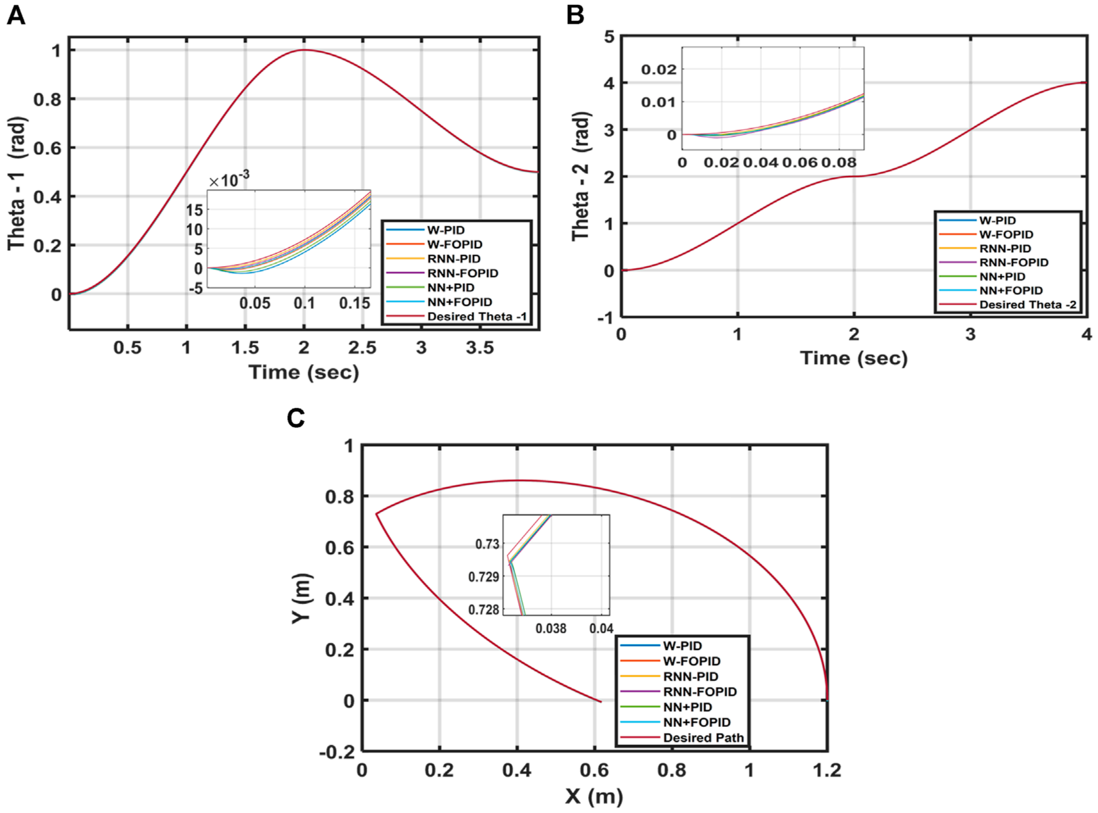

FIGURE 12
Desired and calculated trajectories for **(A)** θ_1, **(B)** θ_2, and **(C)** 5% added masses to both links for a starting position of [0, 0].

TABLE 5 ITSE values of the suggested controllers with added disturbance of sin(50t) to both links and a starting position of [0.15, 0.15].

Controller	ITSE	Controller	ITSE
W-PID	4.7128×10^{-5}	W_FOPID	3.7435×10^{-5}
RNN-PID	4.2479×10^{-5}	RNN-FOPID	4.9497×10^{-5}
NN + PID	4.0179×10^{-5}	NN + FOPID	3.4328×10^{-5}

Determine the *i*th zebra as a new status using mode *S2* in Eq. (50).

End if

Use Eq. (51) to update the *i*th solution.

Ending for $i = 1$: N

Save the best possible candidate solution.

Ending for $t = 1$: T

Display the optimal ZOA solution as the output for the given optimization problem.

Stop ZOA.

The five key components for tuning a PID controller are the fitness function, ZOA optimization method, PID controller, process, and sensor (feedback). Any controller type can be built using various optimum control parameters (Wilson et al., 2018). For the objective function (fitness function) to be minimized, some parameters must be calculated (Oleiwi, 2014). The optimization problem can be expressed using the following concepts that minimize the objective function and applied to the following constraints: $K_p\,min < K_p < K_p\,max$, $K_i\,min < K_i < K_i\,max$, $K_d\,min < K_d < K_d\,max$ (Baruh et al., 2002). The ZOA is used to modify the parameters of each proposed controller to minimize tracking errors between the actual and projected 2-LRRM trajectories. Figure 9 shows a block diagram of a tuned PID controller.

5 Simulation and results

The performances of the 2-LRRM with the proposed controllers for trajectory tracking are examined and discussed in this section. The six proposed controllers, namely W-PID, W-FOPID, RNN-PID, RNN-FOPID, NN + PID, and NN + FOPID controllers, are compared against each other to minimize the performance index when the nominal model is used. Two starting points are used for θ_1 (Theta-1: 0.1745, 0.1745) and θ_2 (Theta-2: −0.1745, −0.1745) to increase the learning of the controllers. The ZOA is used to find the optimal controller's parameters that minimize the ITSE between the calculated and reference trajectories of the 2-LRRM. The settings for the ZOA are population size = 100 and maximum number of iterations = 1000. The optimal solution obtained from the last iteration is regarded as the final solution. The step size for simulation is taken as h = 0.001 s, and the simulation time is taken as 4 s. There are several ways to represent the fractional differentiation and integration components mathematically. The approximation of

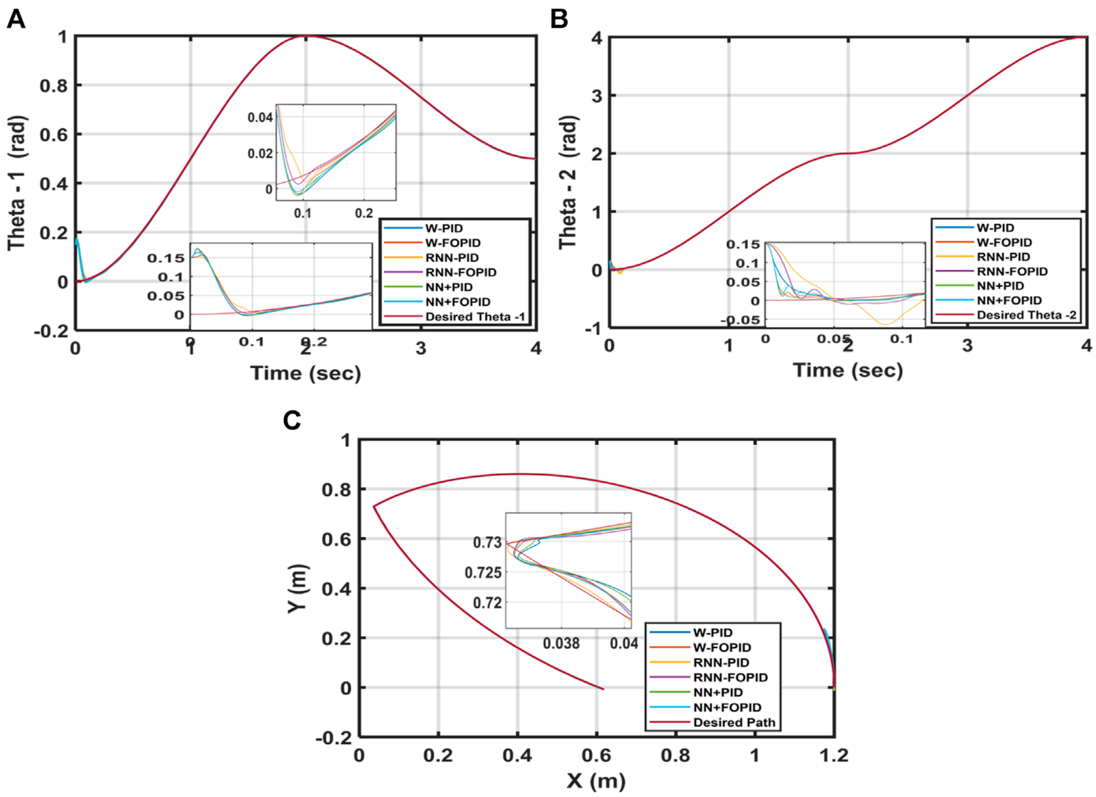

FIGURE 13
Desired and calculated trajectories for **(A)** θ_1, **(B)** θ_2, and **(C)** end effector for an added disturbance of [sin(50t), sin(50t)] and starting point of [0.15, 0.15].

TABLE 6 ITSE between the desired and calculated paths when using a starting point of [0.15, 0.15] with added disturbances of [sin(50t), sin(50t)] and added masses of 5% to both links.

Controller	ITSE	Controller	ITSE
W-PID	4.9523×10^{-5}	W-FOPID	3.7390×10^{-5}
RNN-PID	4.8589×10^{-5}	RNN-FOPID	5.1111×10^{-5}
NN + PID	4.2179×10^{-5}	NN + FOPID	3.5742×10^{-5}

a fractional operator used in the design of the FOPID controller is Oustaloup's approximation of the fifth order (N = 5) with a frequency range of [0.001, 1000] rad/s. Each link's trajectory tracking is determined previously to allow following by the manipulator. The controller with the lowest ITSE value is considered as the best one, and the ITSE is calculated using Eq. (52).

$$ITSE = \int \left(t \times e_1^2(t) + t \times e_2^2(t)\right) dt \tag{52}$$

where $e_1(t)$ and $e_2(t)$ are the differences between the reference trajectories for link1 θ_{ri} and calculated trajectories for link2 θ_i.

One of the important advantages of a NN is its flexibility to capturing complex underlying data structures. In the design of NN controllers, this allows production of the most complex control signals with high frequencies (i.e., chattering). In fact, a chattering signal cannot be applied practically, and the optimal solution obtained is not a feasible solution. Therefore, the objective function is modified as demonstrated in Eq. (53)

$$Objective\,Function\,Value\,or\,Fitness\,Value = ITSE + \rho \times Sn, \tag{53}$$

where Sn is the number of times that the control signal's slope changes signs, and ρ is a small constant number chosen as 10^{-8} in this work.

This modified objective function excludes any solutions that contain high chattering control signals from among the candidate solutions. The desired trajectories θ_{r1} and θ_{r2} for link1 and link2 are given in Eqs (54, 55), respectively:

$$\theta_{r1} = \left\{ \begin{array}{l} 0.75 \times t^2 - 0.25 \times t^3, \quad 0 < t < 2 \\ -1.5 + 3 \times t - 1.125 \times t^2 + 0.125 \times t^3, 2 < t < 4 \end{array} \right\}, \tag{54}$$

$$\theta_{r2} = \left\{ \begin{array}{l} 1.5 \times t^2 - 0.5 \times t^3, \quad 0 < t < 2 \\ 12 - 12 \times t + 4.5 \times t^2 - 0.5 \times t^3, 2 < t < 4 \end{array} \right\}. \tag{55}$$

Now that all details about the simulation and nominal model are available and known, we start applying the ZOA optimizer to adjust the gains of all the suggested controllers based on the nominal model to minimize the ITSE. Since the ZOA is a stochastic algorithm, each controller is simulated 10 times to derive the best results. Table 2 shows the ITSE values for all proposed control structures when applied to a nominal plant and executed with two initial positions.

Overall, the findings show that in terms of the ITSE, the suggested controllers with fractional-order integral and derivative actions perform better than those with corresponding integer-order

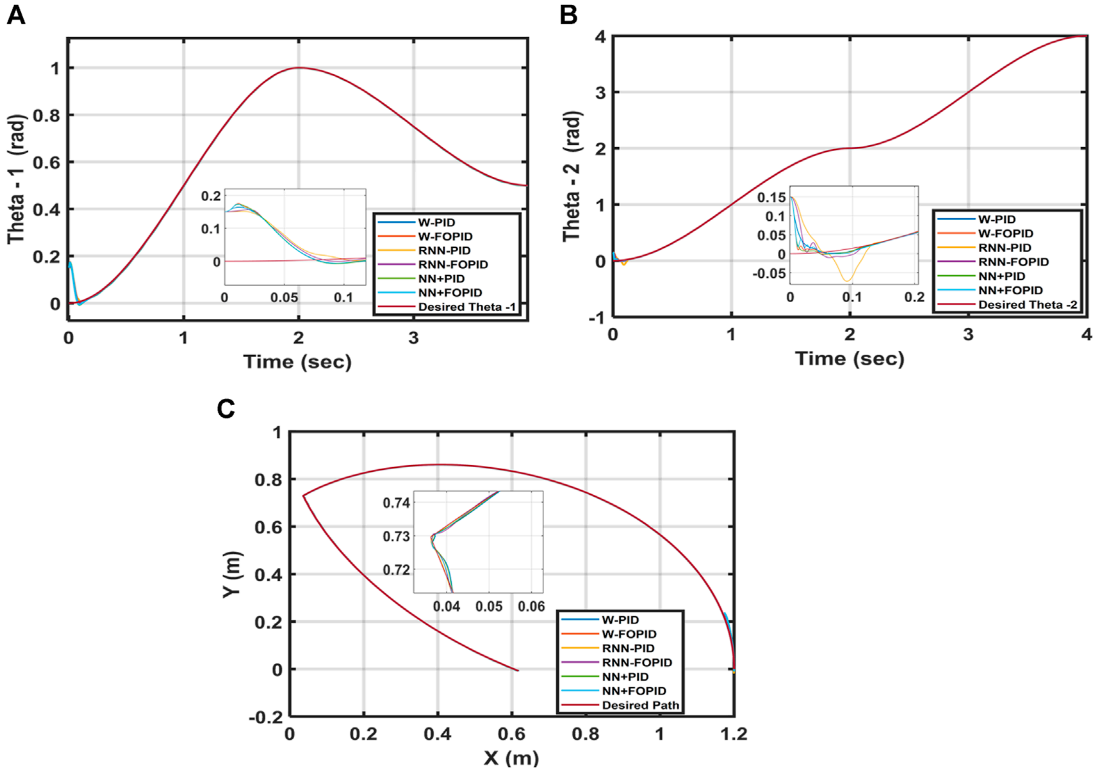

FIGURE 14
Desired and calculated trajectories for **(A)** θ_1, **(B)** θ_2, and **(C)** end effector when using a starting point of [0.15, 0.15], adding a disturbance of sin(50t) to the torques of both links, and adding 5% to the masses of both links.

actions. This is attributed to the fact that the tuning parameters of the controller are increased by the FOPID, which in turn increases the number of DOFs, controller capabilities, and robustness. Despite the results being very close to each other, as seen from Table 2, the findings indicate that the hybrid NN + FOPID structure has the highest ITSE of 7.9676 × 10^{-5} while the W-PID controller has the lowest ITSE of 9.4864 × 10^{-5}. Figure 10 shows the trajectory tracking performances of $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$ (the paths followed by the 2-LRRM's end effectors), and the suggested controllers' torques T_1 and $\boldsymbol{T_2}$.

Based on the results, it is concluded that the NN + FOPID controller performs better than all the other suggested controllers and that it is the best controller among them.

6 Robustness tests

This section presents the results of the proposed controllers that are subjected to robustness tests. To show the capabilities of each controller, the following experiments are implemented in MATLAB without adjusting the gains or retuning the gains of the proposed controllers.

6.1 Initial condition changes

In this test, another set of initial conditions [0.15, 0.15] was considered for [θ_1, θ_2] to evaluate each controller's robustness and test its ability to follow the required trajectory of the 2-LRRM. Table 3 presents the ITSE values for each of the suggested controllers. Figure 11 depicts the trajectory tracking of θ_1 and θ_2 and the end-effector of the 2-LRRM by changing the initial location for all suggested controllers.

Despite altering the starting position, the NN + FOPID controller performs better than the other suggested controllers, where the ITSE = 2.3871 × 10^{-5} of the NN + FOPID controller is the optimal among them. Furthermore, in the trajectory response, NN + FOPID has the least amount of overshoot and fastest settling time for θ_1 as well as a good overshoot and good settling time for θ_2. The W-PID has the worst ITSE of 3.6563 × 10^{-5}, and its responses are poor owing to its large overshoot and lengthy settling times for the θ_1 and θ_2 responses. Furthermore, the trajectory of the 2-LRRM end effector for the NN + FOPID controller still follows the most similar path as the desired trajectory.

6.2 Parameter variations

In this test, the parameter variations of the 2-LRRM model are investigated for the suggested control structures by incrementing the mass of each link by 5%. Table 4 shows the ITSE value of each controller. The RNN-PID controller has the best performance index of ITSE = 0.1418 × 10^{-5}, and the tracked trajectories for θ_1 and θ_2 are closest to the required trajectories of θ_{r1} and θ_{r2} than the other suggested control structures. Consequently, when using the RNN-PID controller, the trajectory followed by the 2-LRRM end effector with parameter variations is very close to the desired trajectory. The second-best controller is the NN + FOPID, with a performance index of ITSE = 0.3139 × 10^{-5} and good trajectory tracking for θ_1 and θ_2; the worst controller is the W-PID with ITSE = 1.0759 × 10^{-5}, and its trajectory tracking has large overshoots and lengthy settling times. Figure 12 shows the trajectories tracked for θ_1 and θ_2 as well as the 2-LRRM end effector when the link masses are changed for each controller.

6.3 Disturbance addition

Another test was conducted to determine the robustness of the proposed control structures by increasing the disturbance terms [sin (50t), sin (50t)] in the control actions $[T_1,\ T_2]$ and setting the initial positions as [0.15,0.15] for $[\theta_1,\ \theta_2]$. Table 5 shows the obtained ITSE values of the proposed controllers. The trajectories tracked for θ_1 and θ_2 as well as the end effector of the 2-LRRM by increasing the disturbance term by sin(50t) N⋅m in both links are demonstrated in Figure 13. From the results, it is concluded that the NN + FOPID controller performs the best in terms of disturbance rejection when compared to the other controllers. The NN + FOPID controller is also the best in terms of the ITSE and has the smallest overshoot during trajectory tracking.

6.4 All tests conducted simultaneously

This combined test is crucial when evaluating robustness since it determines which of the proposed controllers can be used as the best controller. All suggested controllers are subjected to the combined impacts of added disturbance of sin(50t) to the control signals $[T_1,\ T_2]$, increasing the masses of the two links by 5%, as well as altering the starting points to [0.15, 0.15]. Table 6 displays the ITSE value of each controller based on the results attained. Among all the suggested controllers, the lowest ITSE is observed for the NN + FOPID controller. Figure 14 shows the trajectories tracked for θ_1 and θ_2 as well as the end effector of the 2-LRRM for disturbance additions, parameter variations, and initial position changes for all proposed controllers. The NN + FOPID controller still shows optimal results and performs better than all the other proposed controllers because the trajectory followed by the 2-LRRM end effector and tracked the trajectories of θ_1 and θ_2 for the NN + FOPID controller are all close to the required trajectories.

7 Conclusion

In this study, six PID- and FOPID-based control structures are proposed for a 2-LRRM for trajectory tracking; these are named W-PID, W-FOPID, RNN-PID, RNN-FOPID, NN + PID, and NN + FOPID controllers. To optimize the controller parameters, the ZOA was used offline to minimize the performance index ITSE. The MATLAB simulation results show that all proposed controllers have the ability to quickly reduce the errors between the real and desired paths before tracking the required path. By altering the initial state values, adding disturbances to the control signals, and increasing the masses of the two links, the robustness of each suggested controller was examined. According to the results of the nominal system tuning test and all robustness tests, the NN + FOPID has the best control structure among the suggested controllers. The combination of NN with fractional-order integration and differentiation affords high performance and efficient responses, which are reflected in the results. In addition, the modified objective or fitness function allowed ZOA-based tuning of the controller parameters to determine stable responses with fewer fluctuations in the control signals. The trajectory tracking responses for Theta-1 and Theta-2 have the smallest overshoots, shortest settling times, and lowest ITSE values. Moreover, the NN + FOPID controller outperformed all the other controllers in the robustness tests. This work also highlights the capacity of the ZOA for fine-tuning the controller parameters. Finally, as a future work, this study can be extended using other optimization techniques instead of the ZOA, such as the firebug swarm optimization algorithm, chimp optimization algorithm, mayfly optimization algorithm, and gazelle optimization algorithm to adjust the controller gains. In addition, a real robot manipulator equipped with all required hardware may be employed to practically implement and verify the suggested controllers.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material; further inquiries can be directed to the corresponding author.

Author contributions

MJ: conceptualization, formal analysis, methodology, resources, software, writing–original draft, and writing–review and editing. BO: conceptualization, formal analysis, methodology, resources, software, validation, writing–original draft, and writing–review and editing. ATA: conceptualization, formal analysis, investigation, methodology, resources, validation, visualization, and writing–review and editing. AM: funding acquisition, investigation, methodology, resources, validation, visualization, and writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded by Prince Sultan University, Riyadh, Saudi Arabia. This research was also supported by the Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

Acknowledgments

The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia, for support with the article processing charges

(APC) of this publication. The authors specially acknowledge the Automated Systems and Soft Computing Lab (ASSCL) at Prince Sultan University, Riyadh, Saudi Arabia. The authors wish to acknowledge the editor and reviewers for their insightful comments, which have improved the quality of this publication.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdulameer, H. I., and Mohamed, M. J. (2022). Fractional order fuzzy PID controller design for 2-Link rigid robot manipulator. *Int. J. Intelligent Eng. Syst.* 15 (3), 103–117. doi:10.22266/ijies2022.0630.10

Ajeil, F., Ibraheem, I. K., Azar, A. T., and Humaidi, A. J. (2020). Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment. *Int. J. Adv. Robotic Syst.* 17 (3), 172988142092949–15. doi:10.1177/1729881420929498

Alandoli, E. A., and Tian, S. L. (2020). A critical review of control techniques for flexible and rigid link manipulators. *Robotica* 38 (12), 2239–2265. doi:10.1017/S0263574720000223

Azar, A. T., and Fernando, E. S. (2019). Fractional order two degree of freedom pid controller for a robotic manipulator with a fuzzy type-2 compensator. *Proc. Int. Conf. Adv. Intelligent Syst. Inf.* 845, 77–88. Springer International Publishing. doi:10.1007/978-3-319-99010-1_7

Baruh, A. M., and Nenkova, B. (2002). Adaptive neural control with integral-plus-state action. *Cybern. Inf. Technol.* 2 (1), 37–48. Sofia.

Cao, S., Sun, L., Jiang, J., and Zuo, Z. (2021). Reinforcement learning-based fixed-time trajectory tracking control for uncertain robotic manipulators with input saturation. *IEEE Trans. Neural Netw. Learn. Syst.* 34 (8), 4584–4595. doi:10.1109/TNNLS.2021.3116713

Caro, T., Izzo, A., Reiner, R. C., Walker, H., and Stankowich, T. (2014). The function of zebra stripes. *Nat. Commun.* 5 (1), 3535–10. doi:10.1038/ncomms4535

Dachang, Z., Baolin, D., Puchen, Z., and Shouyan, C. (2020). Constant force PID control for robotic manipulator based on fuzzy neural network algorithm. *Complexity* 2020, 1–11. Article ID 3491845. doi:10.1155/2020/3491845

Al-Hadithy, D., and Hammoudi, A. (2020). "Two- link robot through strong and stable adaptive sliding mode controller," in IEEE 13th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, United Kingdom, December 14-17, 2020.

Hameed, A. M., and Hamoudi, A. K. (2023). A 2-link robot with adaptive sliding mode controlled by barrier function. *J. Eur. des Systèmes Automatisés* 56 (6), 1105–1113. doi:10.18280/jesa.560620

Hamoudi, A. K., and Rasheed, L. T. (2023). Design and implementation of adaptive backstepping control for position control of propeller-driven pendulum system. *J. Eur. des Systèmes Automatisés* 56 (2), 281–289. doi:10.18280/jesa.560213

Ibraheem, G. A. R., Azar, A. T., Ibraheem, I. K., and Humaidi, A. J. (2020). A novel design of a neural network-based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization. *Complexity* 2020, 1–18. Article ID 3067024. doi:10.1155/2020/3067024

Jenhani, S., Gritli, H., and Carbone, P. G. (2022). Comparison between some nonlinear controllers for the position control of Lagrangian-type robotic systems. *Chaos Theory Appl.* 4, 179–196. doi:10.51537/chaos.1184952

Kareem, A. A., Oleiwi, B. K., and Mohamed, J. M. (2023). Planning the optimal 3D quadcopter trajectory using a delivery system-based hybrid algorithm. *Int. J. Intelligent Eng. Syst.* 16 (2), 427–439.

Kennedy, S., and Kennedy, V. (2013). *Animals of the masai mara*. Princeton, NJ, USA: Princeton Univ. Press.

Kumar, V. (2017). Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator. *Expert Syst. Appl.* 73, 161–177. doi:10.1016/j.eswa.2016.12.029

Lewis, F. L., Dawson, D. M., and Abdallah, C. T. (2004). *Robot manipulator control theory and practice*. Florida, United States: CRC Press, 1–632. doi:10.1201/9780203026953

Mohamed, M. J., Oleiwi, B. K., Abood, L. H., Azar, A. T., and Hameed, I. A. (2023). Neural fractional order PID controllers design for 2-link rigid robot manipulator. *Fractal Fract.* 7 (9), 693. doi:10.3390/fractalfract7090693

Mohan, V., Chhabra, H., Rani, A., and Singh, V. (2018). An expert 2DOF fractional order fuzzy PID controller for nonlinear systems. *Neural Comput. Appl.* 31 (8), 4253–4270. doi:10.1007/s00521-017-3330-z

Najm, A. A., Ibraheem, I. K., Azar, A. T., and Humaidi, A. J. (2020). Genetic optimization-based consensus control of multi-agent 6-DoF UAV system. *Sensors* 20 (12), 3576. doi:10.3390/s20123576

Nohooji, H. R. (2020). Constrained neural adaptive PID control for robot manipulators. *J. Frankl. Inst.* 357 (7), 3907–3923. doi:10.1016/j.jfranklin.2019.12.042

Oleiwi, B. K., Al-Jarrah, R., Roth, H., and Kazem, B. I. (2015). Integrated motion planning and control for multi objectives optimization and multi robots navigation. *IFAC-PapersOnLine* 48 (10), 99–104. doi:10.1016/j.ifacol.2015.08.115

Oleiwi, B. K., Mahfuz, A., and Roth, H. (2021). "Application of fuzzy logic for collision avoidance of mobile robots in dynamic-indoor environments," in Proceedings of the 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 5-7 January, 2021, 131–136.

Oleiwi, B. K., Roth, H., and Kazem, B. I. (2014). "Multi objective optimization of path and trajectory planning for non-holonomic mobile robot using enhanced genetic algorithm," in *Neural networks and artificial intelligence. ICNNAI 2014. Communications in computer and information science, vol 440.* Editors V. Golovko, and A. Imada (Cham: Springer). doi:10.1007/978-3-319-08201-1_6

Pastor, J., Cohen, Y., and Hobbs, N. T. (2006). "The roles of large herbivores in ecosystem nutrient cycles," in *Large herbivore ecology ecosystem dynamics and conservation conservation biology* (Cambridge, UK: Cambridge Univ. Press), 289–325.

Raafat, S. M., and Raheem, F. A. (2017). "Introduction to robotics-mathematical issues," in *Mathematical advances towards sustainable environmental systems* (Springer), 261–289. doi:10.1007/978-3-319-43901-3_12

Sharma, R., Gaur, P., and Mittal, A. (2015). Performance analysis of two-degree of freedom fractional order PID controllers for robotic manipulator with payload. *ISA Trans.* 58, 279–291. doi:10.1016/j.isatra.2015.03.013

Shojaei, K., Kazemy, A., and Chatraei, A. (2020). An observer-based neural adaptive PID2 controller for robot manipulators including motor dynamics with a prescribed performance. *IEEE/ASME Trans. Mechatronics* 269 (3), 1689–1699. doi:10.1109/TMECH.2020.3028968

Shuyang, C., and John, T. W. (2021). Industrial robot trajectory tracking control using multi-layer neural networks trained by iterative learning control. *Robotics* 10 (1), 50. doi:10.3390/robotics10010050

Trojovská, E., Dehghani, M., and Trojovský, P. (2022). Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access* 10, 49445–49473. doi:10.1109/access.2022.3172789

Wilson, M., Hubel, T. Y., Wilshin, S. D., Lowe, J. C., Lorenc, M., Dewhirst, O. P., et al. (2018). Biomechanics of predator–prey arms race in lion zebra cheetah and impala. *Nature* 554 (7691), 183–188. doi:10.1038/nature25479

Zhou, M., Feng, Y., Xue, C., and Han, F. (2020). Deep convolutional neural network based fractional-order terminal sliding-mode control for robotic manipulators. *Neurocomputing* 416 (27), 143–151. doi:10.1016/j.neucom.2019.04.087