



OPEN ACCESS

EDITED BY

Javier Moreno-Valenzuela,
National Polytechnic Institute, Mexico

REVIEWED BY

Önder Tutsoy,
Adana Science and Technology
University, Türkiye
Ying Feng,
South China University of Technology, China
Phuong Nam Dao,
Hanoi University of Science and
Technology, Vietnam

*CORRESPONDENCE

Ahmad Taher Azar,
✉ aazar@psu.edu.sa
Ahmed Redha Mahlous,
✉ armahlous@psu.edu.sa

RECEIVED 28 August 2024

ACCEPTED 28 February 2025

PUBLISHED 02 May 2025

CITATION

Olewi BK, Jasim M, Azar AT, Ahmed S and
Mahlous AR (2025) Bat optimization of hybrid
neural network-FOPID controllers for robust
robot manipulator control.
Front. Robot. AI 12:1487844.
doi: 10.3389/frobt.2025.1487844

COPYRIGHT

© 2025 Olewi, Jasim, Azar, Ahmed and
Mahlous. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Bat optimization of hybrid neural network-FOPID controllers for robust robot manipulator control

Bashra Kadhim Olewi¹, Mohamed Jasim¹,
Ahmad Taher Azar^{2,3*}, Saim Ahmed^{2,3} and
Ahmed Redha Mahlous^{2,3*}

¹Department of Control and System Engineering, University of Technology, Baghdad, Iraq, ²College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia, ³Automated Systems and Computing Lab (ASCL), Prince Sultan University, Riyadh, Saudi Arabia

The position and trajectory tracking control of rigid-link robot manipulators suffers from problems such as poor accuracy, unstable performance, and response caused by unidentified loads and outside disturbances. In this paper, three control structures have been proposed to control a multi-input, multi-output coupled nonlinear three-link rigid robot manipulator (3-LRRM) system and effectively solve the signal chattering in the control signal. To overcome these problems, three hybrid control structures based on combinations between the benefits of fractional order proportional-integral-derivative operations (FOPID) and the benefits of neural networks are proposed for a 3-LRRM. The first hybrid control scheme is a neural network- (NN) like fractional order proportional-integral plus an NN-like fractional order proportional derivative controller (NN-FOIPD) and the second control scheme is an NN plus FOPID controller (NN + FOPID). In contrast, the third control scheme is the Elman NN-like FOPID controller (ELNN-FOPID). The bat optimization algorithm (BOA) is applied to find the best parameter values of the proposed control scheme by minimizing the performance index of the integral time square error (ITSE). MATLAB software is used to carry out the simulation results. Using the simulation tests, the performance of the suggested controllers is compared without retraining the controller parameters. The robustness of the designed control schemes' performance is assessed utilizing uncertainties in system parameters, outside disturbances, and initial position changes. The results show that the NN-FOIPD structure demonstrated the best performance among the suggested controllers.

KEYWORDS

trajectory tracking, neural network, neural network controller, PIPD controller, PID controller, FOPID controller, bat optimization algorithm, 3-link rigid robotic manipulator

1 Introduction

One of the most promising technologies is robots, which are extensively employed in numerous applications in industries like drilling, medical procedures (Chen et al., 2017), transportation (Wu et al., 2016), assembly (Fani et al., 2018), and manufacturing (Sadrfaridpour et al., 2016). The three main categories of robotic manipulator research are robot manipulator control,

optimal trajectory planning for industrial manipulators (Ayten et al., 2016), and robotic system design (Zemiti et al., 2007; Ayten et al., 2017). Among these fields of study, robotic manipulator control plays an important and vital role in tracking a reliable and accurate path to the starting and finishing points of the work.

Robotic systems are time-varying nonlinear systems because of external disturbances, modeling uncertainties, and the nonlinear behavior of the system dynamics. Therefore, obtaining robotic systems that track specified paths successfully and accurately is challenging. Many sophisticated control techniques have been developed to overcome these undesirable uncertainties for various types of robotic systems (Tutsoy et al., 2023). These techniques include fractional-order control (Viola and Angel, 2015; Dhivakaran et al., 2022; Kaushik et al., 2024), H-infinity control (Makarov et al., 2016), adaptive control (Huang, 2016; Tutsoy et al., 2024), neural network control (Sun et al., 2017; Rajchakit et al., 2019), and sliding mode control (SMC) (Dumlu et al., 2017; Basci et al., 2017).

Using a fractional-order scheme, several methods have been integrated with other notable schemes to improve the performance. The fractional-order SMC (FOSMC) was proposed by Dumlu (2018) to illustrate a robotic manipulator's ability to track position practically. Ren et al. (2023) used a unique composite position predictive control method that combines disturbance preview and motion profile techniques to achieve accurate and smooth trajectory regulation. Xia et al. (2019) proposed a control strategy combining the modified terminal sliding mode with the double power reaching law to precisely and quickly track tasks of rigid robotic manipulators. Using fractional calculus, Ullah et al. (2015) proposed a new fuzzy FOSMC for a servo actuation system. The results showed that the suggested method's tracking error is less than that of a traditional SMC. For the doubly fed induction generator, Ebrahimkhani (2016) developed an innovative, reliable FOSMC with a fractional-order estimator. The outcomes have validated the suggested controller's robustness and efficacy in dealing with the effects of changing disturbances and parameters. Andualem and Gebremichael (2020) suggested using a hyperbolic tangent function instead of a signum function in a fuzzy gain scheduling terminal SMC for controlling and tracking the UR5 robot manipulator.

Recently, several research papers have proposed control structures and hybrid controllers for a 2-LRRM that deals with the path-tracking problem. Kumar et al. (2019) suggested a self-regulated FOPID controller with a backtracking search algorithm. Mohamed et al. (2023) suggested six control structures as a hybrid controller for a 2-LRRM that addresses the trajectory tracking problem by combining the advantages of PID controllers with integer and fractional orders, neural networks, and the Gorilla Forces Troops Optimization algorithm. Six neural network-based control structures with PID and fractional-order PID (FOPID) controllers were proposed by Mohamed et al. (2024) for operating a 2-LRRM for trajectory tracking. These controllers are referred to as NN + PID, NN + FOPID, recurrent neural network-like PID, set-point-weighted FOPID, set-point-weighted PID, and RNN-like FOPID. The parameters of the suggested controllers' conditions, disturbances, and model uncertainties were modified using the Zebra Optimization algorithm. Three control structures for 3-LRRM were proposed by Mohamed et al. (2023) utilizing a neural network in conjunction with PID actions. The Coot Optimization algorithm

is used to modify the parameters of the suggested controllers. Based on the social behavior of spider monkeys, Spider Monkey Optimization (SMO) is another optimization technique used to control robotic manipulators (Abualigah et al., 2024; Agrawal et al., 2023). The PID controller adjusts with SMO to find the best control parameters in the robotic manipulator in order to enhance the manipulator version (Agrawal et al., 2021). Wang et al. (2020) proposed an adaptive PID control algorithm for multi-degree-of-freedom (DOF) industrial robots, leveraging a fuzzy neural network framework to enhance trajectory tracking performance. Their research addresses the complexities inherent in industrial robot control systems, characterized by nonlinearity, time-varying dynamics, and strong coupling effects. By establishing a functional relationship between control error and reaching degree, the authors achieved self-adaptive adjustments of PID parameters, significantly improving control accuracy and stability compared to traditional PID methods. The effectiveness of their approach was validated through simulations and joint experiments using the ADAMS virtual simulation system, demonstrating superior performance in trajectory tracking tasks. In the study conducted by Sau (2020), a comparative analysis of fractional order controllers was performed specifically for a three-link robotic manipulator system. The research highlights the significance of fractional order control strategies in enhancing the dynamic performance and stability of robotic systems. By employing fractional calculus, the study demonstrates that these controllers can provide improved tuning flexibility compared to traditional integer-order controllers. This flexibility allows for more precise adjustments to the system's response characteristics, which is crucial in applications requiring high precision and reliability.

It is evident that each of the methods discussed above has advantages and disadvantages in the majority of studies. They were employed at the expense of accuracy and time due to their complexity and time-consuming computations. Previous investigations indicated that most of the studies did not resolve the signal chattering in the control signal and addressed the suggested controllers separately. In this paper, three control structures have been proposed to control a multi-input, multi-output coupled, nonlinear 3-LRRM system and effectively solve the signal chattering in the control signal. It is well known that the FOPID controller is more robust and flexible than the conventional PID controller. On the other hand, the neural network has high flexibility in mapping complex data. Hence, a hybrid controller consisting of FOPID operations and a neural network will perform better than the FOPID controller alone.

The aim of this work is to design hybrid controllers combining the benefits of FOPID operations and the benefits of neural networks-based BOA for 3-LRRM by solving the nonlinear problems of compensating robot manipulator control with disturbances and uncertainties and achieving precise trajectory tracking.

This work's primary contributions are as follows:

- 1 Develop three structures of hybrid control based on combinations between the benefits of FOPID operations and the benefits of neural networks. These hybrid control structures are neural network-like FOPID controllers (NN-FOPIDs), neural network plus FOPID controllers (NN +

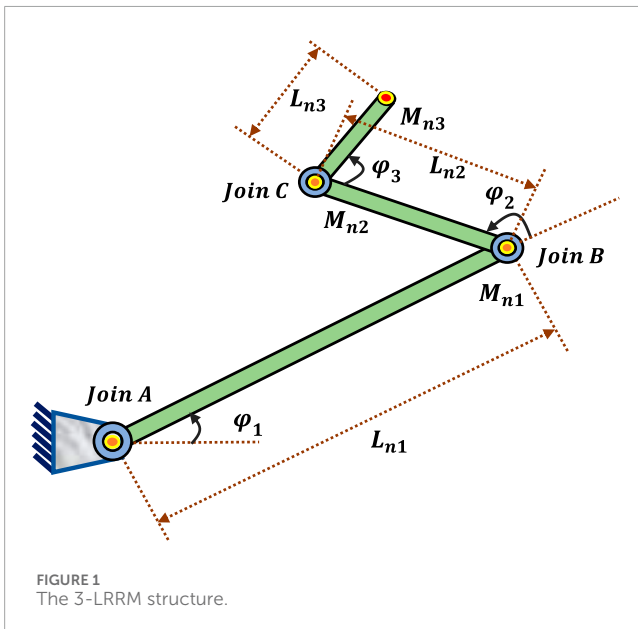
FOPIDs), and Elman neural network-like FOPID controllers (ELNN-FOPIDs).

- 2 Apply the bat optimization algorithm (BOA) to determine the optimal parameters for each suggested controller in order to further improve the performance.
- 3 Conduct a comparative analysis among the suggested controllers concerning changing the starting conditions, outside disruptions, parameter uncertainty, and all combined effects.
- 4 A new objective function suggestion to fine-tune the suggested controller to generate the control signal with a minimum amount of chattering.

The remaining sections of this work are organized as follows: [Section 2](#) explains the 3-LRRM's dynamical system. [Section 3](#) illustrates the suggested controllers. [Section 4](#) displays the BOA, and [Section 5](#) displays the simulation's results. The robustness test is provided in [Section 6](#). The conclusion is given in [Section 7](#).

2 System modeling formulation

A robotic manipulator is constructed of several links joined by joints at each end. Planar robotic manipulators can only move in one plane ([Abdulameer and Mohamed, 2022](#)). This research considers three rotating joints on a planar robotic manipulator, and it is assumed that each joint is actuated ([Kumar et al., 2019](#)). [Figure 1](#) shows the 3-LRRM structure. The manipulator's dynamic motion equation is used to develop the fundamental control equations for robots. The torques produced by actuators in a robotic system cause the dynamic movement of the manipulator arms ([Raafat and Raheem, 2017](#)). The following is the dynamic model of a 3-LRRM.



The 3-LRRM's dynamic model is defined by Lagrange dynamics ([Lewis et al., 2004](#)). [Equations 1–6](#) are the equations for the x and y positions of links 1, 2, and 3:

$$x_1 = L_{n1} \cos(\varphi_1), \quad (1)$$

$$y_1 = L_{n1} \sin(\varphi_1), \quad (2)$$

$$x_2 = L_{n1} \cos(\varphi_1) + L_{n2} \cos(\varphi_1 + \varphi_2), \quad (3)$$

$$y_2 = L_{n1} \sin(\varphi_1) + L_{n2} \sin(\varphi_1 + \varphi_2), \quad (4)$$

$$x_3 = L_{n1} \cos(\varphi_1) + L_{n2} \cos(\varphi_1 + \varphi_2) + L_{n3} \cos(\varphi_1 + \varphi_2 + \varphi_3), \quad (5)$$

$$y_3 = L_{n1} \sin(\varphi_1) + L_{n2} \sin(\varphi_1 + \varphi_2) + L_{n3} \sin(\varphi_1 + \varphi_2 + \varphi_3). \quad (6)$$

Kinetic energy (*KinE*) is indicated in [Equation 7](#):

$$KinE = \frac{1}{2} M_{n1} v_1^2 + \frac{1}{2} M_{n2} v_2^2 + \frac{1}{2} M_{n3} v_3^2. \quad (7)$$

where L_{n1} , L_{n2} , and L_{n3} and M_{n1} , M_{n2} and M_{n3} are the lengths and masses of the links. φ_1 , φ_2 , φ_3 and v_1 , v_2 , v_3 represent the positions and velocities of the links, respectively. The velocities can be written as [Equation 8](#).

$$v_1 = (\dot{x}_1^2 + \dot{y}_1^2)^{0.5}, \quad v_2 = (\dot{x}_2^2 + \dot{y}_2^2)^{0.5}, \quad v_3 = (\dot{x}_3^2 + \dot{y}_3^2)^{0.5} \quad (8)$$

The kinetic energy (*KinE*) is presented in [Equation 9](#):

$$KinE = \frac{1}{2} M_{n1} (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} M_{n2} (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} M_{n3} (\dot{x}_3^2 + \dot{y}_3^2). \quad (9)$$

The potential energy (*PotE*) is defined in [Equations 10, 11](#):

$$PotE = \sum_{i=1}^3 M_{ni} g h_i(\varphi), \quad (10)$$

where g represents the acceleration and $h_i(\varphi)$ represents the height.

$$PotE = M_{n1} g L_{n1} \sin(\varphi_1) + M_{n2} g (L_{n1} \sin(\varphi_1) + L_{n2} \sin(\varphi_1 + \varphi_2)) + M_{n3} g (L_{n1} \sin(\varphi_1) + L_{n2} \sin(\varphi_1 + \varphi_2) + L_{n3} \sin(\varphi_1 + \varphi_2 + \varphi_3)) \quad (11)$$

Next, the Lagrangian (L) was determined using the Lagrange dynamic, as shown in [Equation 12](#):

$$L = KinE - PotE. \quad (12)$$

The Euler–Lagrange equation is expressed by [Equation 13](#):

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\varphi}_i} \right] - \frac{\partial L}{\partial \varphi_i} = F\varphi_i, \quad (13)$$

where $F\varphi_i$ represents the torque exerted on link i .

The basic principle dynamics of the manipulator are illustrated in [Equations 14–24](#) ([Lewis et al., 2004](#)):

$$Q(\varphi)\ddot{\varphi} + P_c(\varphi, \dot{\varphi}^2) + R(\varphi, \dot{\varphi}, \ddot{\varphi}) + G(\varphi) = \tau, \quad (14)$$

where the inertia matrix is denoted by $Q(\varphi)$, centrifugal term P_c , and Coriolis term R .

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix}, \quad (15)$$

$$Q_{11} = (M_{n1} + M_{n2} + M_{n3})L_{n1}^2 + (M_{n2} + M_{n3})L_{n2}^2 + M_{n3}L_{n3}^2 + 2M_{n3}L_{n1}L_{n3}\cos(\varphi_2 + \varphi_3) + 2(M_{n2} + M_{n3})L_{n1}L_{n2}\cos(\varphi_2) + 2M_{n3}L_{n2}L_{n3}\cos(\varphi_3) \quad (16)$$

$$Q_{12} = (M_{n2} + M_{n3})L_{n2}^2 + M_{n3}L_{n3}^2 + M_{n3}L_{n1}L_{n3}\cos(\varphi_2 + \varphi_3) + (M_{n2} + M_{n3})L_{n1}L_{n2}\cos(\varphi_2) + 2M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (17)$$

$$Q_{13} = M_{n3}L_{n3}^2 + M_{n3}L_{n1}L_{n3}\cos(\varphi_2 + \varphi_3) + M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (18)$$

$$Q_{21} = M_{n2}L_{n2}^2 + M_{n3}L_{n2}^2 + M_{n3}L_{n3}^2 + M_{n3}L_{n1}L_{n3}\cos(\varphi_2 + \varphi_3) + M_{n2}L_{n1}L_{n2}\cos(\varphi_2) + M_{n3}L_{n1}L_{n2}\cos(\varphi_2) + 2M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (19)$$

$$Q_{22} = M_{n2}L_{n2}^2 + M_{n3}L_{n2}^2 + M_{n3}L_{n3}^2 + 2M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (20)$$

$$Q_{23} = M_{n3}L_{n3}^2 + M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (21)$$

$$Q_{31} = M_{n3}L_{n3}^2 + M_{n3}L_{n1}L_{n3}\cos(\varphi_2 + \varphi_3) + M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (22)$$

$$Q_{32} = M_{n3}L_{n3}^2 + M_{n3}L_{n2}L_{n3}\cos(\varphi_3), \quad (23)$$

$$Q_{33} = M_{n3}L_{n3}^2. \quad (24)$$

The centrifugal term (P_c) is defined as illustrated in Equations 25–28:

$$P_c = \begin{bmatrix} P_{c1} \\ P_{c2} \\ P_{c3} \end{bmatrix}, \quad (25)$$

$$P_{c1} = -L_{n1}(M_{n3}L_{n3}\sin(\varphi_2 + \varphi_3) + M_{n2}L_{n2}\sin(\varphi_2) + M_{n3}L_{n2}\sin(\varphi_2)) \times \dot{\varphi}_2^2 - M_{n3}L_{n3}(L_{n1}\sin(\varphi_2 + \varphi_3) + L_{n2}\sin(\varphi_3))\dot{\varphi}_3^2 \quad (26)$$

$$P_{c2} = L_{n1}(M_{n3}L_{n3}\sin(\varphi_2 + \varphi_3) + M_{n2}L_{n2}\sin(\varphi_2) + M_{n3}L_{n2}\sin(\varphi_2)) \times \dot{\varphi}_1^2 - M_{n3}L_{n2}L_{n3}\sin(\varphi_3)\dot{\varphi}_3^2 \quad (27)$$

$$P_{c3} = M_{n3}L_{n3}(L_{n1}\sin(\varphi_2 + \varphi_3) + L_{n2}\sin(\varphi_3))\dot{\varphi}_1^2 + M_{n3}L_{n2}L_{n3}\sin(\varphi_3)\dot{\varphi}_2^2 \quad (28)$$

The Coriolis term R is defined as presented in Equations 29–32:

$$R = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}, \quad (29)$$

$$R_1 = -2L_{n1}(M_{n3}L_{n3}\sin(\varphi_2 + \varphi_3) + (M_{n2} + M_{n3})L_{n2}\sin(\varphi_2)) \times \dot{\varphi}_1\dot{\varphi}_2 - 2M_{n3}L_{n3}(L_{n1}\sin(\varphi_2 + \varphi_3) + L_{n2}\sin(\varphi_3))\dot{\varphi}_2\dot{\varphi}_3 - 2M_{n3}L_{n3}(L_{n1}\sin(\varphi_2 + \varphi_3) + L_{n2}\sin(\varphi_3))\dot{\varphi}_1\dot{\varphi}_3 \quad (30)$$

$$R_2 = -2M_{n3}L_{n2}L_{n3}\sin(\varphi_3)\dot{\varphi}_1\dot{\varphi}_3 - 2M_{n3}L_{n2}L_{n3}\sin(\varphi_3)\dot{\varphi}_2\dot{\varphi}_3, \quad (31)$$

$$R_3 = 2M_{n3}L_{n2}L_{n3}\sin(\varphi_3)\dot{\varphi}_1\dot{\varphi}_2. \quad (32)$$

The potential energy term, G , is defined in Equations 33–36:

$$G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}, \quad (33)$$

$$G_1 = (M_{n1} + M_{n2} + M_{n3})gL_{n1}\cos(\varphi_1) + (M_{n2} + M_{n3})gL_{n2} \times \cos(\varphi_1 + \varphi_2) + M_{n3}gL_{n3}\cos(\varphi_1 + \varphi_2 + \varphi_3), \quad (34)$$

$$G_2 = (M_{n2} + M_{n3})gL_{n2}\cos(\varphi_1 + \varphi_2) + M_{n3}gL_{n3}\cos(\varphi_1 + \varphi_2 + \varphi_3), \quad (35)$$

$$G_3 = M_{n3}gL_{n3}\cos(\varphi_1 + \varphi_2 + \varphi_3). \quad (36)$$

Using forward kinematic (Kumar et al., 2019), the joint angles φ_{r1} , φ_{r2} , and φ_{r3} provide the coordinates for 3-LRRM's end effector, as indicated by Equations 37, 38:

Regarding the reference trajectory,

$$x_r = L_{n1}\cos(\varphi_{r1}) + L_{n2}\cos(\varphi_{r1} + \varphi_{r2}) + L_{n3}\cos(\varphi_{r1} + \varphi_{r2} + \varphi_{r3}), \quad (37)$$

$$y_r = L_{n1}\sin(\varphi_{r1}) + L_{n2}\sin(\varphi_{r1} + \varphi_{r2}) + L_{n3}\sin(\varphi_{r1} + \varphi_{r2} + \varphi_{r3}). \quad (38)$$

where φ_{r1} , φ_{r2} and φ_{r3} are the desired trajectories.

Table 1 (Mohamed et al., 2023) provides a detailed description of the robot parameter settings used in this work.

TABLE 1 Nominal values of 3-LRRM parameters.

Parameter name	Nominal parameter value
L_{n1}	0.8 M
L_{n2}	0.4 M
L_{n3}	0.2 M
M_{n1}	0.1 Kg
M_{n2}	0.1 Kg
M_{n3}	0.1 Kg
g	9.81 M/S ²

3 Proposed controller structures

Detailed information regarding the proposed hybrid neural network controllers' structures is presented in this section. The closed-loop block diagram of the NN-like FOPID is given in Figure 2, where $\varphi_{ri}(t)$ is the desired position, $\varphi_{ai}(t)$ is the actual position, $e_i(t)$ is the error position, and $\varphi_{ri}(t)$ is the control signal.

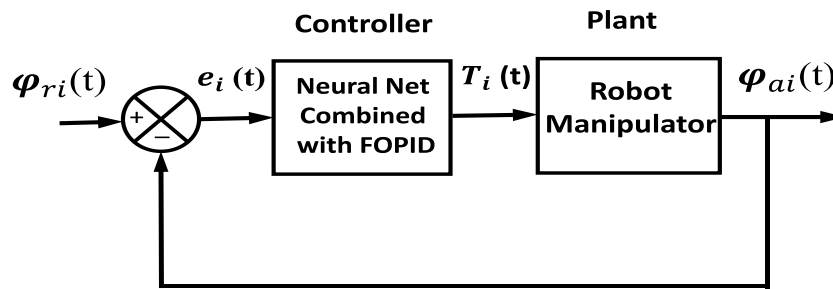


FIGURE 2
Schematic diagram of combined NN with FOPI controller.

3.1 Neural network-like FOPI and FOPD controller

The fundamental structure of the NN-FOPIPD controller is demonstrated in Figure 3, where the variables λ and μ are fractional values in the range of 0–2.

The difference between the desired position $\varphi_{ri}(t)$ and actual position $\varphi_{ai}(t)$ of the i link is known as the error $e_{\varphi i}(t)$. The equations describing the controller from the error signal $e_{\varphi i}(t)$ to the control signal $\bar{T}_i(k)$ are shown below in Equations 39–48.

The single-processing element input layer is represented as $e_{\varphi i}(t)$.

The first hidden layer has four processing elements: proportional operation P_1 , fractional integral operation I, proportional operation P_2 , and fractional derivative operation D, as follows:

$$P_1(t) = K_{P1} e_{\varphi i}(t) \text{ or } P_1(k) = K_{P1} e_{\varphi i}(k), \quad (39)$$

$$I(t) = K_I D^{-\lambda} e_{\varphi i}(t) \text{ or } I(k) = K_I D^{-\lambda} e_{\varphi i}(k), \quad (40)$$

$$P_2(t) = K_{P2} e_{\varphi i}(t) \text{ or } P_2(k) = K_{P2} e_{\varphi i}(k), \quad (41)$$

$$D(t) = K_D D^{\mu} e_{\varphi i}(t) \text{ or } D(k) = K_D D^{\mu} e_{\varphi i}(k), \quad (42)$$

Where:

$$\begin{bmatrix} NET_1^1(k) \\ NET_2^1(k) \\ NET_3^1(k) \\ NET_4^1(k) \end{bmatrix} = \begin{bmatrix} z11 & z12 & 0 & 0 \\ z21 & z22 & 0 & 0 \\ 0 & 0 & z33 & z34 \\ 0 & 0 & z43 & z44 \end{bmatrix} \begin{bmatrix} P_1(k) \\ I(k) \\ P_2(k) \\ D(k) \end{bmatrix}, \quad (43)$$

$$\begin{bmatrix} Out_1^1(k) \\ Out_2^1(k) \\ Out_3^1(k) \\ Out_4^1(k) \end{bmatrix} = \begin{bmatrix} H(NET_1^1(k)) \\ H(NET_2^1(k)) \\ H(NET_3^1(k)) \\ H(NET_4^1(k)) \end{bmatrix} + \begin{bmatrix} z13 \times Out_1^1(k-1) \\ z14 \times Out_2^1(k-1) \\ z23 \times Out_3^1(k-1) \\ z24 \times Out_4^1(k-1) \end{bmatrix} + \begin{bmatrix} z31 \times Out_1^2(k-1) \\ z32 \times Out_2^2(k-1) \\ z41 \times Out_3^2(k-1) \\ z42 \times Out_4^2(k-1) \end{bmatrix}, \quad (44)$$

Where:

$$H = \frac{4}{(1 + e^{-net})} - 2, \quad (45)$$

$$\begin{bmatrix} Out_1^2(k) \\ Out_2^2(k) \end{bmatrix} = \begin{bmatrix} zz11 & zz12 \\ zz21 & zz22 \end{bmatrix} \begin{bmatrix} Out_1^1(k) \\ Out_2^1(k) \end{bmatrix} + \begin{bmatrix} zz13 \times Out_1^2(k-1) \\ zz23 \times Out_2^2(k-1) \end{bmatrix} + \begin{bmatrix} zz14 \times \bar{T}(k-1) \\ zz24 \times \bar{T}(k-1) \end{bmatrix}, \quad (46)$$

$$\begin{bmatrix} Out_2^2(k) \\ Out_3^2(k) \end{bmatrix} = \begin{bmatrix} zz31 & zz32 \\ zz41 & zz42 \end{bmatrix} \begin{bmatrix} Out_2^1(k) \\ Out_3^1(k) \end{bmatrix} + \begin{bmatrix} zz33 \times Out_2^2(k-1) \\ zz43 \times Out_3^2(k-1) \end{bmatrix} + \begin{bmatrix} zz34 \times \bar{T}(k-1) \\ zz44 \times \bar{T}(k-1) \end{bmatrix}, \quad (47)$$

$$\begin{aligned} \bar{T}_i(k) &= \bar{T}_i(k-1) + wz_1 \times Out_1^1(k) + wz_2 \times Out_2^2(k) + wz_3 \\ &\times Out_3^2(k) + wz_4 \times Out_4^2(k), \end{aligned} \quad (48)$$

3.2 Design of NN+FOPID

This controller contains two parts, as shown in Figure 4. The first part is the neural network; the input layer has three input elements, $e_{\varphi i}(k)$, $e_{\varphi i}(k-1)$ and $e_{\varphi i}(k-2)$ or A, B, and C elements. The second part of the controller is the FOPID controller with a filter; both are merged and connected with the system in cascade.

The equations describing the controller from the error signal $e_{\varphi i}(t)$ to control signal $\bar{T}_i(k)$ are shown below in Equations 49–62.

Where:

$$\begin{bmatrix} NET_1^1(k) \\ NET_2^1(k) \\ NET_3^1(k) \end{bmatrix} = \begin{bmatrix} z11 & z12 & z13 \\ z21 & z22 & z23 \\ z31 & z32 & z33 \end{bmatrix} \begin{bmatrix} e_{\varphi i}(k) \\ e_{\varphi i}(k-1) \\ e_{\varphi i}(k-2) \end{bmatrix} + \begin{bmatrix} NET_1^1(k-1) \\ NET_2^1(k-1) \\ NET_3^1(k-1) \end{bmatrix}, \quad (49)$$

The first hidden layer's output is defined in Equation 50:

$$\begin{bmatrix} Out_1^1(k) \\ Out_2^1(k) \\ Out_3^1(k) \end{bmatrix} = \begin{bmatrix} H(NET_1^1(k)) \\ H(NET_2^1(k)) \\ H(NET_3^1(k)) \end{bmatrix}, \quad (50)$$

$$\begin{bmatrix} NET_1^2(k) \\ NET_2^2(k) \\ NET_3^2(k) \end{bmatrix} = \begin{bmatrix} zz11 & zz12 & zz13 \\ zz21 & zz22 & zz23 \\ zz31 & zz32 & zz33 \end{bmatrix} \begin{bmatrix} Out_1^1(k) \\ Out_2^1(k) \\ Out_3^1(k) \end{bmatrix} + \begin{bmatrix} NET_1^2(k-1) \\ NET_2^2(k-1) \\ NET_3^2(k-1) \end{bmatrix}. \quad (51)$$

The second hidden layer's output is indicated in Equation 52:

$$\begin{bmatrix} Out_1^2(k) \\ Out_2^2(k) \\ Out_3^2(k) \end{bmatrix} = \begin{bmatrix} H(NET_1^2(k)) \\ H(NET_2^2(k)) \\ H(NET_3^2(k)) \end{bmatrix}. \quad (52)$$

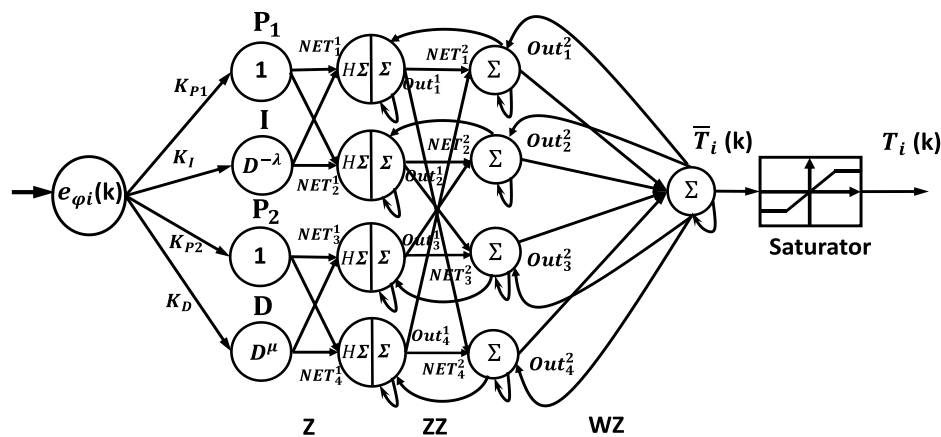


FIGURE 3
Neural controller PIPD controller.

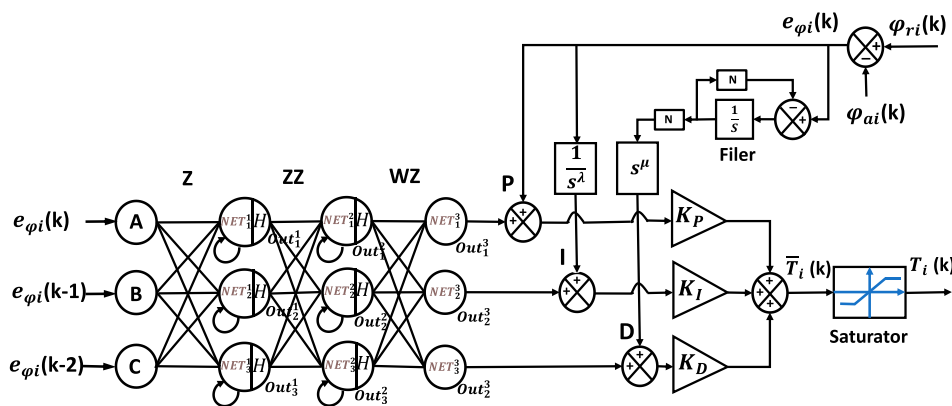


FIGURE 4
Neural network plus FOPID controller.

As can be noticed in Equation 52, the activation function is a sigmoid.

$$H = \frac{2}{(1 + e^{-net})} - 1. \quad (53)$$

Equation 54 displays the output of the third and last hidden layer.

$$\begin{bmatrix} Out_1^3(k) \\ Out_2^3(k) \\ Out_3^3(k) \end{bmatrix} = \begin{bmatrix} NET_1^3(k) \\ NET_2^3(k) \\ NET_3^3(k) \end{bmatrix} = \begin{bmatrix} wz11 & wz12 & wz13 \\ wz21 & wz22 & wz23 \\ wz31 & wz32 & wz33 \end{bmatrix} \begin{bmatrix} Out_1^2(k) \\ Out_2^2(k) \\ Out_3^2(k) \end{bmatrix}. \quad (54)$$

The three FOPID control operations are displayed in Equations 55–61.

$$P(t) = e_{\phi i}(t) \text{ or } P(k) = e_{\phi i}(k). \quad (55)$$

$$I(t) = D^{-\lambda} e_{\phi i}(t) \text{ or } I(k) = D^{-\lambda} e_{\phi i}(k). \quad (56)$$

$$f_{\phi i}(s) = \frac{N}{s+N} e_{\phi i}(s); \text{ The filter of derivative.} \quad (57)$$

$$D(t) = D^{\mu} f_{\phi i}(t) \text{ or } D(k) = D^{\mu} f_{\phi i}(k). \quad (58)$$

$$u_1(k) = K_P (Out_1^3(k) + P(k)). \quad (59)$$

$$u_2(k) = K_I (Out_2^3(k) + I(k)). \quad (60)$$

$$u_3(k) = K_D (Out_3^3(k) + D(k)). \quad (61)$$

Equation 62 illustrates the control signal equation.

$$\bar{T}_i(k) = u_1(k) + u_2(k) + u_3(k). \quad (62)$$

3.3 Design of the Elman NN-FOPID

The Elman neural network FOPID controller's structural layout is depicted in Figure 5.

The difference between a required position $\phi_{ri}(t)$ and an actual position $\phi_{ai}(t)$ of the i link is known as the error $e_{\phi i}(t)$. The equations

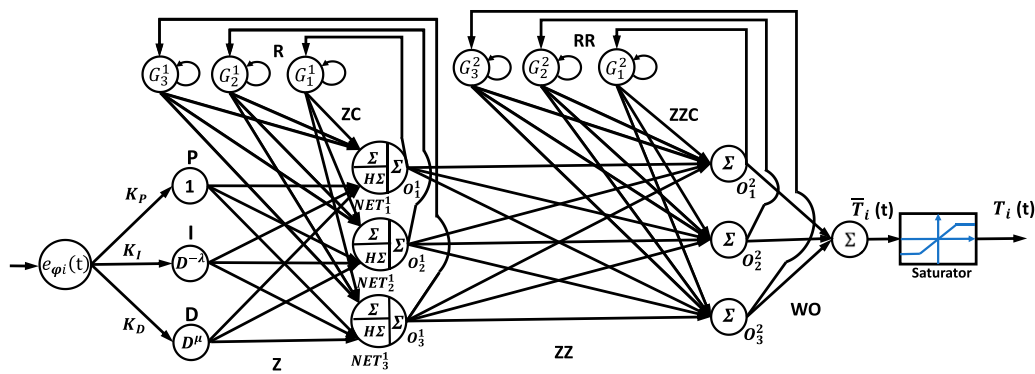


FIGURE 5
The structure of the Elman neural network-like FOPID controller.

describing the controller from the error signal $e_{\varphi_i}(t)$ to control signal $\bar{T}_i(k)$ are shown below in Equations 63–72.

The representation of the input layer with a single input element is $e_{\varphi_i}(t)$.

The first hidden layer has three processing elements (proportional operation P, integral operation I, and derivative operation D); the input to each node is the error $e_{\varphi_i}(t)$ multiplied by a certain gain, while the activation functions represent one of the operations P, I, and D, and the output of each node is the result of its operation as described in Equations 63–65:

$$P(t) = K_P e_{\varphi_i}(t) \text{ or } P(k) = K_P e_{\varphi_i}(k), \quad (63)$$

$$I(t) = K_I D^{-\lambda} e_{\varphi_i}(t) \text{ or } I(k) = K_I D^{-\lambda} e_{\varphi_i}(k), \quad (64)$$

$$D(t) = K_D D^{\mu} e_{\varphi_i}(t) \text{ or } D(k) = K_D D^{\mu} e_{\varphi_i}(k), \quad (65)$$

$$\begin{bmatrix} NET_1^1(k) \\ NET_2^1(k) \\ NET_3^1(k) \end{bmatrix} = \begin{bmatrix} \sum_1^1(k) \\ \sum_2^1(k) \\ \sum_3^1(k) \end{bmatrix} = \begin{bmatrix} z11 & z12 & z13 \\ z21 & z22 & z23 \\ z31 & z32 & z33 \end{bmatrix} \begin{bmatrix} P(k) \\ I(k) \\ D(k) \end{bmatrix}, \quad (66)$$

and

$$\begin{bmatrix} G_1^1(k) \\ G_2^1(k) \\ G_3^1(k) \end{bmatrix} = \begin{bmatrix} O_1^1(k-1) + r1 \times G_1^1(k-1) \\ O_2^1(k-1) + r2 \times G_2^1(k-1) \\ O_3^1(k-1) + r3 \times G_3^1(k-1) \end{bmatrix}. \quad (67)$$

The second hidden layer's output is indicated in Equation 68:

$$\begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix} = \begin{bmatrix} H(\sum_1^2(k)) \\ H(\sum_2^2(k)) \\ H(\sum_3^2(k)) \end{bmatrix} + \begin{bmatrix} zc11 & zc12 & zc13 \\ zc21 & zc22 & zc23 \\ zc31 & zc32 & zc33 \end{bmatrix} \begin{bmatrix} G_1^1(k) \\ G_2^1(k) \\ G_3^1(k) \end{bmatrix}. \quad (68)$$

According to Equation 69, the activation function is a sigmoid function.

$$H = \frac{2}{(1 + e^{-net})} - 1. \quad (69)$$

$$\begin{bmatrix} G_1^2(k) \\ G_2^2(k) \\ G_3^2(k) \end{bmatrix} = \begin{bmatrix} O_1^2(k-1) + rr1 \times G_1^2(k-1) \\ O_2^2(k-1) + rr2 \times G_2^2(k-1) \\ O_3^2(k-1) + rr3 \times G_3^2(k-1) \end{bmatrix}. \quad (70)$$

The output of the third hidden layer is displayed in Equation 71:

$$\begin{bmatrix} O_1^3(k) \\ O_2^3(k) \\ O_3^3(k) \end{bmatrix} = \begin{bmatrix} zz11 & zz12 & zz13 \\ zz21 & zz22 & zz23 \\ zz31 & zz32 & zz33 \end{bmatrix} \begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix} + \begin{bmatrix} zzc11 & zzc12 & zzc13 \\ zzc21 & zzc22 & zzc23 \\ zzc31 & zzc32 & zzc33 \end{bmatrix} \begin{bmatrix} G_1^2(k) \\ G_2^2(k) \\ G_3^2(k) \end{bmatrix}. \quad (71)$$

The output layer consists of a single node, as shown in Equation 72:

$$\bar{T}_i(k) = wo_1 \times O_1^3(k) + wo_2 \times O_2^3(k) + wo_3 \times O_3^3(k), \quad (72)$$

where $K_P, K_I, K_D, zij, zcij, zzcij, wo_i, r_i$, and rr_i , all are design parameters.

4 Bat optimization algorithm

This algorithm is inspired by the diverse emission rates and pulse intensities displayed by microbats during echolocation (Yang and He, 2013). Three primary rules can be used to explain the bat optimization algorithm: The first rule is to use the echolocation phenomenon to find the best distance to the food. According to the second rule, the bats fly at random with a fixed frequency and velocity at a particular search space location. However, the bats' loudness and wavelength can vary based on their overall distance from their current location and the distance from the food. Lastly, the bat algorithm's third rule is to linearly reduce the bat loudness factor's behavior (Nor'Azlan et al., 2018). The following is the BOA procedure (Sambariya and Paliwal, 2016):

Step 1: Set initial values of the algorithm, including the issue's dimension (dim), the highest value of iterations ($Iter$), the size of the population (N), and the minimum and maximum frequencies (F_{min} and F_{max}), respectively. Additional variables are the bats' initial velocity (v), their loudness

(A_i), their pulse emission rate (Y), and their initial pulse emission rate (ro).

Step 2: Equation 73 can be used to generate a random initial population of bats, which is a feasible solution for each bat position.

$$x_{ij} = x_j^l + r \text{ and}() * (x_j^u - x_j^l). \quad (73)$$

The dimensions' upper and lower bounds, j , are represented by x_j^l and x_j^u , respectively, and $i = 1, \dots, N, j = 1, \dots, \text{dim}$,

Step 3: Calculate the cost function (MSE) for every bat using Equation 74, and save the results in the cost vector.

$$MSE(i) = \frac{1}{N} \sum_{i=1}^N (R - Y)^2, \quad (74)$$

where the desired signal is represented by R and the signal produced by Y .

Step 4: Determine the bat with the smallest (MSE) value as the ($F_{\min}$) among all bats in the vector is the best bat position (x^*).

Step 5: As indicated in Equations 75–77, update the it^{th} bat's frequency, velocity, and position.

$$F_i = F_{\min} + (F_{\max} - F_{\min}) * r \text{ and}(), \quad (75)$$

$$v_i^{ii} = v_i^{ii-1} + (x_i^{ii-1} - x^*) * F_i, \quad (76)$$

$$x_i^{ii} = x_i^{ii-1} + v_i^{ii}, \quad (77)$$

where $ii = 1, \dots, \text{Iter}$.

TABLE 2 ITSE and each control's sign change at two initial positions (0.15, −0.55, −0.85) rad and (−0.15, −0.85, −1.15) rad when the nominal plant is applied.

Type of controller	ITSE	Number of slope sign Change in all control signals
NN-FOIPD	2.49695×10^{-5}	78
NN + FOPID	4.31033×10^{-5}	64
ELNN-FOPID	8.09808×10^{-5}	164

Step 6: When a random number is less than the it^{th} bat's pulse emission rate (Y_i), a local search is conducted. A new solution is generated using a random walk for the bat in position i to improve the diversity of possible solutions, as indicated in Equation 78.

$$x_i^{\text{new}} = x^* + \epsilon * \langle A_i \rangle^{ii}. \quad (78)$$

At the ii iteration, the total bats' average noise level is represented by $\langle A_i \rangle^{ii}$, and ϵ is a scaling factor that is generated at random within the range $[-1, 1]$.

Step 7: Utilizing Equation 74, determine the cost function value (MSE_i) for the i^{th} bat and enter the result in the variable.

Step 8: If the i^{th} bat's loudness (A_{ii}) is greater than a randomly generated number, its updated value of the objective or cost function is less than its previous value. Subsequently, the loudness, pulse rate, and updated cost function are computed, as shown in Equations 79–81, respectively.

$$\text{Cost}i = \text{Cost}N, \quad (79)$$

$$A_{ii} = \alpha * A_{ii}, \quad (80)$$

$$Yi = ro * (1 - \exp(-\beta * ii)), \quad (81)$$

where $[0, 1]$ is the range of constants α and β .

Step 9: Using Equations 82–83, the minimum frequency ($F_{\min}$) and best bat position (x^*) for the i^{th} bat can be updated if its cost $\text{Cost}Ni$ is smaller than $F_{\min}$.

$$x^* = x_{ij}, \quad (82)$$

$$F_{(\min)} = \text{cost}Ni. \quad (83)$$

Step 10: Proceed to Step 11 if the maximum number of bats (N) has been attained. If not, proceed to Step 5.

Step 11: End the procedure if the largest number of iterations (Iter) has been achieved. If not, repeat steps 5 through 11.

5 Simulation results

The trajectory tracking performance of the 3-LRRM nominal model with the proposed controllers is shown below. The proposed controllers for the 3-LRRM tracking problem are implemented using the MATLAB code. The torque limits for each link are set at (−200 to 200) N-m, the simulation period is taken 10 s, and the simulation's step size is 1 msec. Moreover, the 11th-order Oustaloup's approximation ($N = 5$), having a range of frequencies of (0.001, 1000) rad/s, is used in the fraction operator design. Each link's controller is tuned via its parameters to track the required link. The overall performance goodness of all links will determine to what extent the accuracy of the manipulator end effector can maintain the required path. The basic objective for each link controller is to follow the desired path for this link with reduced error and match the desired path more quickly. In this work, the sum of ITSE for each link is used as the classical objective function in tuning all proposed controllers. The objective function is stated in Equation 84:

$$\min J = \int (t \times e_1(t)^2 + t \times e_2(t)^2 + t \times e_3(t)^2) dt. \quad (84)$$

TABLE 3 Specifications of the controller for each link when using a nominal system with starting positions ($x_1 = 0.15$ rad, $x_2 = -0.55$ rad, and $x_3 = -0.85$ rad).

Controller Type	No of link	Rise Time	Settling Time	Overshoot %	ITSE $\times 10^{-5}$
NN-FOIPD	L1	0.040	0.060	5.778	0.34448
	L2	0.084	0.084	0.0014	0.41969
	L3	0.039	0.106	8.1747	0.42887
NN + FOPID	L1	0.039	0.457	32.889	1.50855
	L2	0.026	0.371	39.995	1.06976
	L3	0.026	0.093	18.209	0.24544
ELNN-FOPID	L1	0.052	0.219	21.086	0.93473
	L2	0.046	0.229	25.885	1.15868
	L3	0.053	0.366	21.697	1.49175

The suggested controllers' parameters are adjusted using the BOA algorithm in order to minimize the ITSE for each link. The controller is assessed according to the total tracking error ITSE for the three links between the reference paths and the calculated paths. Two initial positions $(-0.15, -0.85, -1.15)$, $(0.15, -0.55, -0.85)$ rad for Phi-1, Phi-2, and Phi-3, respectively, are taken into consideration in simulation to strengthen the training process for all suggested controllers. The fitness of a candidate solution is determined by adding the ITSE from these two executions. The BOA algorithm setting is a population size of 200, and the maximum value of iterations is 1500. Each suggested controller's performance is assessed using ITSE computation. The controller that produces the lowest amount of ITSE is the best. Before discussing the outcomes of each suggested controller, we observed through numerous tests that the resulting control signal was typically highly chattering and inapplicable. This indicates that the design is unreliable and that the controller generates a control signal that the actuator cannot apply. In fact, neural networks have a high capability to map complicated underlying data structures, which is one of their key advantages. This capability produces the most complex high-frequency control signals in neural network controller design (i.e., chattering phenomenon). A chattering signal is actually impractical to use. Consequently, to solve this issue, the objective function is changed as follows:

The classical objective function is to minimize the ITSE, as expressed [Equations 85, 86](#):

$$t \times e(t)^2 = t \times e_1(t)^2 + t \times e_2(t)^2 + t \times e_3(t)^2, \quad (85)$$

$$\min J = \int t \times e(t)^2 dt. \quad (86)$$

The new objective function is as follows as presented in [Equation 87](#):

$$\min J = \int t \times e(t)^2 dt + Co \times \sigma. \quad (87)$$

σ is a small number chosen as 10^{-8}

Co is a count of sign alterations in the control signal's slope.

In the competition between the candidate solutions, this modified objective function will eliminate the solution that exhibits a high chattering control signal.

The desired trajectories φ_{r1} , φ_{r2} and φ_{r3} for link 1, link 2, and link 3 are provided in [Equations 88–90](#), respectively:

$$\varphi_{r1} = \sin(0.2\pi t). \quad (88)$$

$$\varphi_{r2} = \sin\left(0.2\pi t - \frac{\pi}{4}\right). \quad (89)$$

$$\varphi_{r3} = \sin(0.2\pi t - \pi/2). \quad (90)$$

where the initial condition for $\varphi_{r1} = 0$, $\varphi_{r2} = -0.7$, and $\varphi_{r3} = -1$ rad, and $x(0) = 1.0769$, $y(0) = -0.4580$ m.

Now, all of the nominal model and simulation data are available. First, to minimize the ITSE, we use the BOA to tune all the parameters of each proposed controller when using nominal model parameters. Because of the stochastic nature of the BOA, ten simulation runs were performed to get the best results for each controller simulation. [Table 2](#) displays the performance index ITSE value for each suggested controller when a nominal plant is applied with two initial positions. [Table 3](#) lists the features of each link's control for all proposed controllers: settling time, rise time, maximum overshoot, and ITSE value when using the initial positions of 0.15 rad, -0.55 rad, and -0.85 rad. [Figure 6](#) displays the trajectory tracking for each link of the proposed controller and its control signal, as well as end-effector x-y plots. [Figures 6A–C](#) show the desired and actual values for Phi-1, Phi-2, and Phi-3, respectively. [Figures 6D,E](#) show their control signals for Torque-1, Torque-2, and Torque-3, respectively, as well as the desired and actual path of the end effector in [Figure 6G](#).

From these results, it can be seen that the NN-FOIPD controller structure gave smoother and faster convergence to the reference trajectory. In addition, the NN-FOIPD gave the

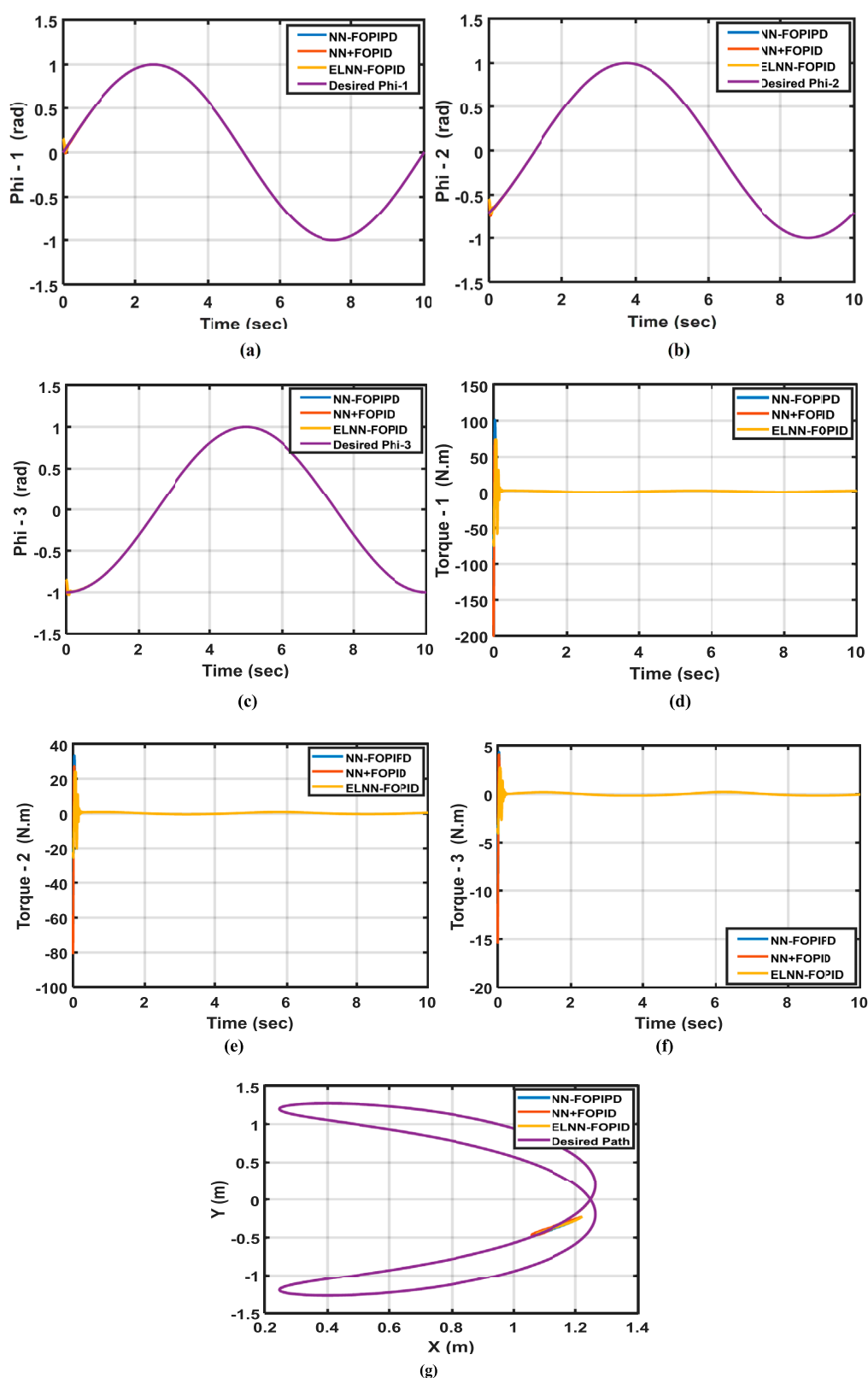


FIGURE 6

The desired and actual values for (a) Φ_1 , (b) Φ_2 , and (c) Φ_3 . The control signals for (d) Control torque-1, (e) Control torque-2, (f) Control torque-3, and (g) the desired and actual end-effector paths.

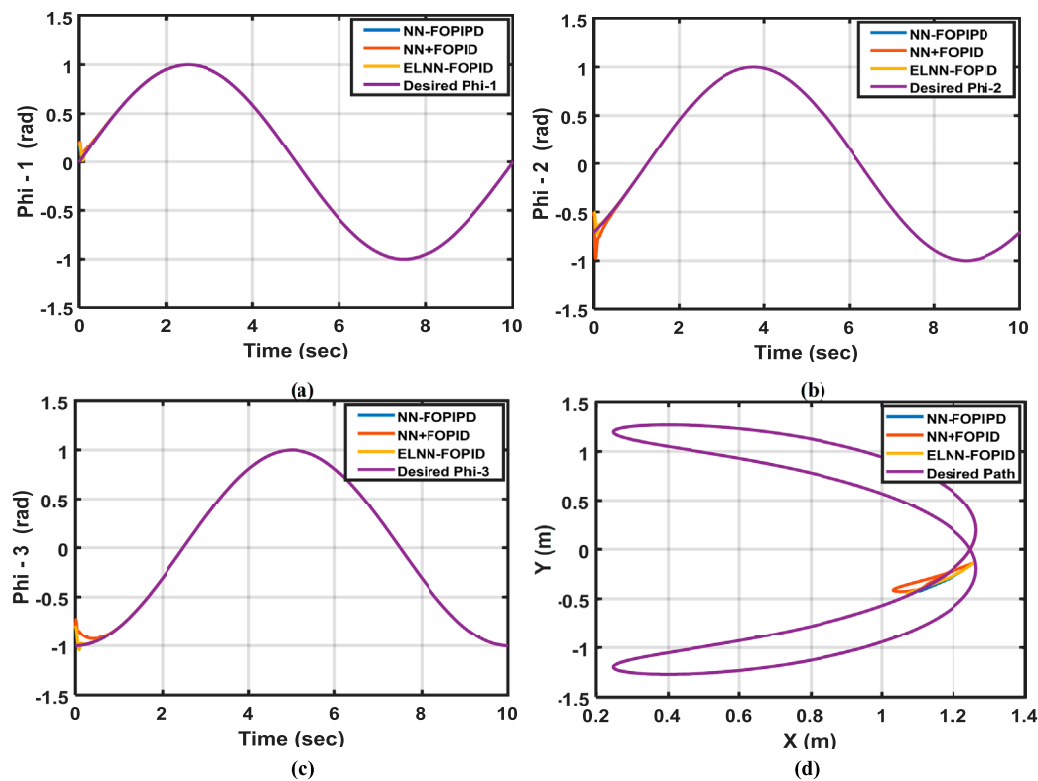


FIGURE 7
The desired and actual trajectories for (a) Φ_1 , (b) Φ_2 , and (c) Φ_3 , respectively, and the (d) desired and actual end-effector paths when the initial positions of the links are (0.2, −0.5, −0.8) rad, respectively.

TABLE 4 ITSEs of the proposed schemes when the links' initial values are (0.2, −0.5, −0.8) rad.

Controller	ITSE
NN-FOIPD	3.430154×10^{-5}
NN + FOPID	110.2063×10^{-5}
ELNN-FOPID	8.278723×10^{-5}

TABLE 5 ITSE of the proposed scheme with starting conditions of 0 rad, −0.7 rad, and 1 rad, and disturbances added to all control signals in the period 2–6 s sin (100t).

Control scheme	ITSE
NN-FOIPD	0.062988
NN + FOPID	0.065961
ELNN-FOPID	0.050069

smallest overshoot, shortest settling time, and a closer trajectory to the reference trajectory with the lowest values of ITSE compared with the other controllers. The ELNN-FOPID controller yielded a value that was marginally higher than the NN-FOIPD controller and relatively close to the NN-FOIPD performance.

The NN + FOPID controller has the lowest performance for each suggested controller. The NN-FOIPD controller performs the best.

6 Robustness tests

In this section, each suggested controller's ability, efficiency, and robustness will be demonstrated by evaluating its robustness without changing its parameters.

6.1 Change initial position

The robustness of the proposed controllers can be evaluated by altering the starting positions to (0.2, −0.5, and 0.8) rad for Φ_1 , Φ_2 , and Φ_3 , respectively. In Figure 7, the trajectory of the 3-LRRM end effector is depicted when the initial positions of all links are changed, in addition to tracking Φ_1 , Φ_2 , and Φ_3 trajectories. The ITSE of the proposed control schemes is given in Table 4.

As can be observed, the NN-FOIPD controller has the smallest ITSE value, fastest response, and least settling time. Consequently, when Φ_1 , Φ_2 , and Φ_3 's starting positions are changed, the NN-FOIPD controller performs better than the other suggested controllers.

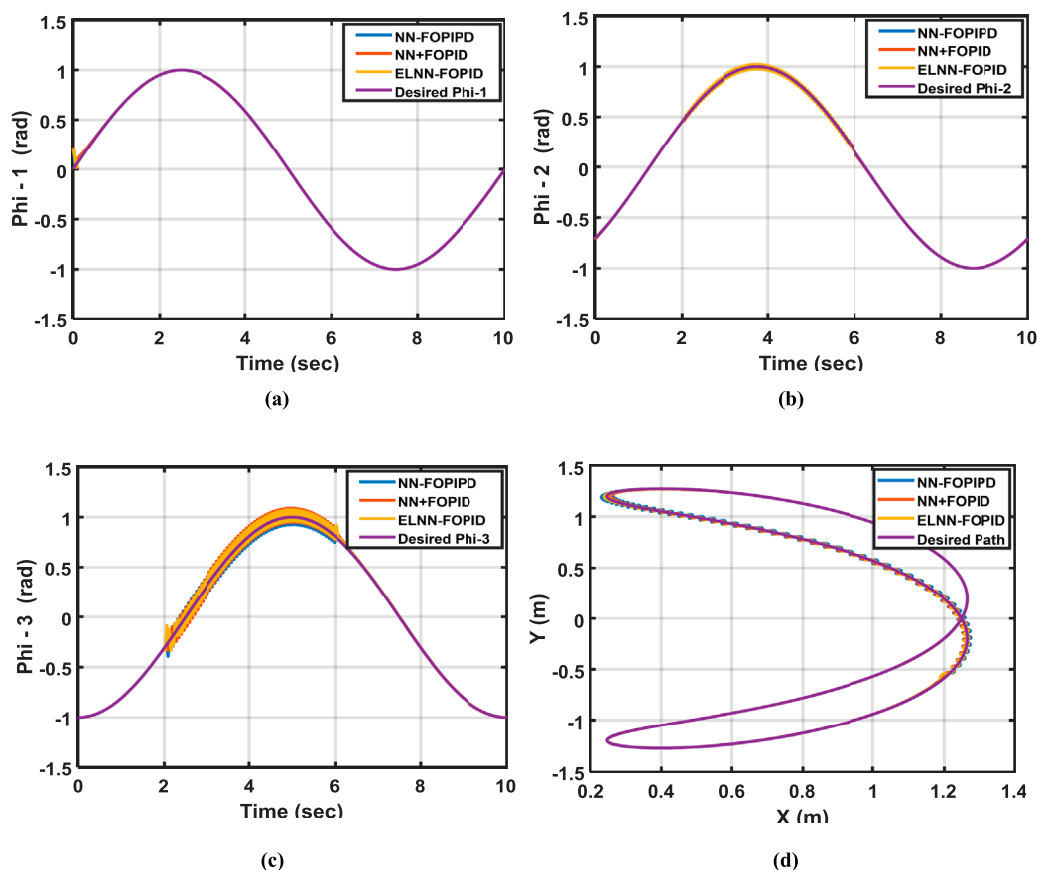


FIGURE 8

The desired and actual trajectories for (a) Phi-1, (b) Phi-2, and (c) Phi-3, respectively, and the (d) desired and calculated paths of the end effector when applying the external disturbance $\sin(100t)$ N-M to all link controllers while the starting position is $(0.0, -0.7, -1)$ rad.

TABLE 6 The ITSE values for the proposed controllers when the mass of Link-3 is increased by 10% and the initial position is $(0.0, -0.7, 1)$ rad.

Proposed schemes	ITSE
NN-FOIPD	0.00099×10^{-5}
NN + FOPID	0.30056×10^{-5}
ELNN-FOPID	0.00179×10^{-5}

6.2 Disturbance addition

The capability of disturbance rejection for each suggested controller has been analyzed. This test is implemented after including disturbance term $\sin(100t)$ N-m for the duration of 2–6 s in the controller output of each link. In this test, the starting position is taken to be $(0, -0.7, -1)$ rad. Table 5 displays the achieved ITSE results for each proposed controller. Figure 8 shows how the actual trajectory of the 3-LRRM end effector tracks the desired trajectory; in addition, it shows the trajectory tracking of Phi-1, Phi-2, and Phi-3 when the external disturbance is considered is $\sin(100t)$ N-m in each link.

The results clearly show that the suggested NN-FOIPD and ELNN-FOPID controllers perform nearly equally well. However, the ELNN-FOPID gives the lowest value of ITSE and a slightly better performance than the NN-FOIPD. The NN + FOPID controller gives a worse performance. Therefore, the ELNN-FOPID outperforms the other proposed controllers for disturbance rejection.

6.3 Parameter variations

In industry, a manipulator's main job is picking and placing different components with varying weights utilizing its end effector. In order to conduct this test, link 3's masses will be increased by 10%, and the starting position $(0.0, -0.7, -1)$ rad will be used. This test explores the proposed controller's performance when parameter variation occurs in the system. Table 6 lists the calculated ITSEs for each suggested controller. The desired and actual trajectory tracking of Phi-1, Phi-2, and Phi-3 in relation to changing the weights for all controllers is displayed in Figure 9.

Based on the findings, the ITSE for the NN-FOIPD controller is the lowest value among other controllers when parameter variation occurs. However, the NN-FOIPD gave a fast response that was closest to the required trajectory for Phi-1, Phi-2, and Phi-3 when

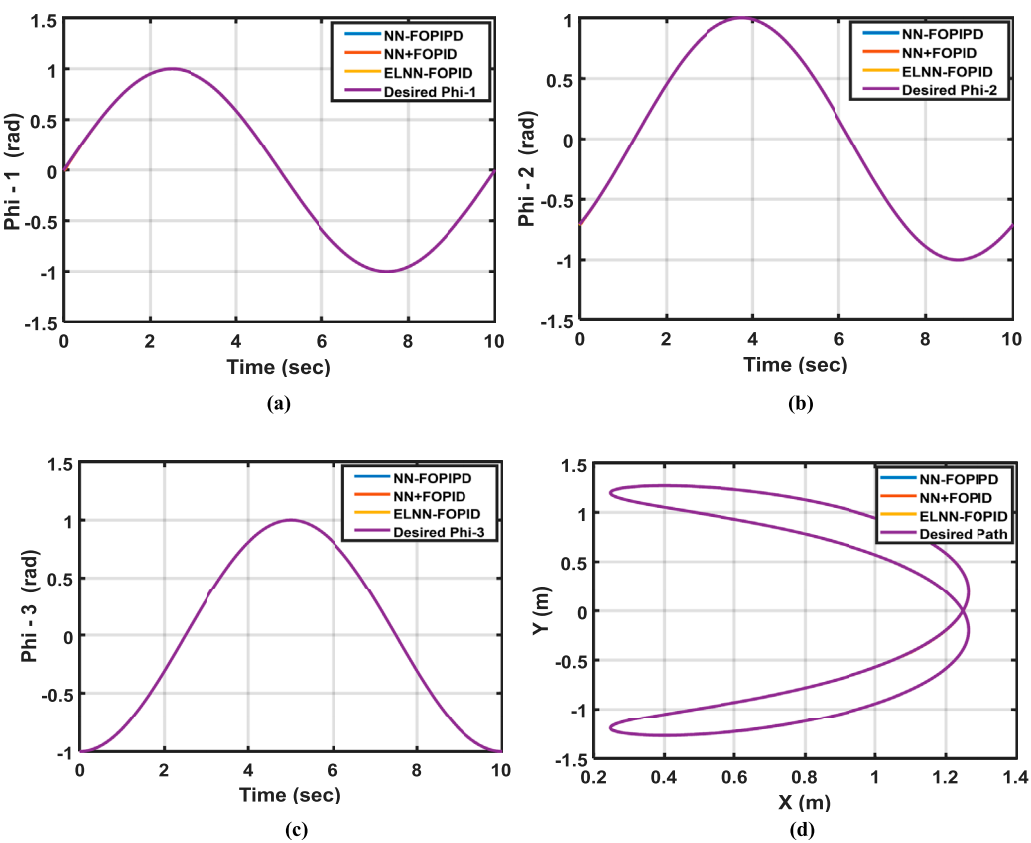


FIGURE 9 (a) The desired and calculated Phi-1, (b) Phi-2, (c) Phi-3, and (d) the desired and calculated paths of the end effector when 10% is added to the mass of link3 with starting positions for all links are (0, -0.7, -1) rad.

TABLE 7 The values of ITSE for all controllers when utilizing the initial position (0.2, -0.5, and -0.8) rad, adding external disturbance term $\sin(100t)^{\ast}(1, 1, 1)$, and changing the third link mass by 10% simultaneously.

Control scheme	ITSE
NN-FOIPD	0.0630227
NN + FOPID	0.066909
ELNN-FOPID	0.050358

compared to other proposed controllers. Conversely, the highest (and worst) ITSE value was caused by applying the NN + FOPID controller.

6.4 All previous tests combined

The efficiency of the suggested controllers is determined by combining all the effects on the control system. In this crucial test, all the proposed controllers are simultaneously subjected to the following effects: altering the initial position, adding a disturbance, and changing the parameters of the system model. Table 7 contains a list of the obtained ITSE values. Figure 10

shows the desired and actual Phi-1, Phi-2, and Phi-3 trajectories and the end-effector trajectories when the system is subjected to the above effects. Of the proposed controllers, the ELNN-FOPID has the lowest ITSE value even after changing the model parameters, adding a disturbance, and shifting the starting positions. The ELNN-FOPID controller demonstrates the fastest response, shortest settling times, closeness-to-required trajectories, and lowest energy consumption, as demonstrated by its Phi-1, Phi-2, and Phi-3 trajectories. The NN-FOIPD controller comes in second order in performance and accuracy. On the other hand, the NN + FOPID controller has the worst performance because of its maximum settling time and highest ITSE value. We can conclude that the ELNN-FOPID controller is the best and outperforms in all states.

7 Conclusion

A robotic manipulator’s performance is negatively affected by parameter uncertainty and outside disturbances because it is a highly coupled, complex, nonlinear, and multi-input multi-output system. Consequently, the controllers designed for these systems must be able to handle their complexity. The goal of this work is to solve

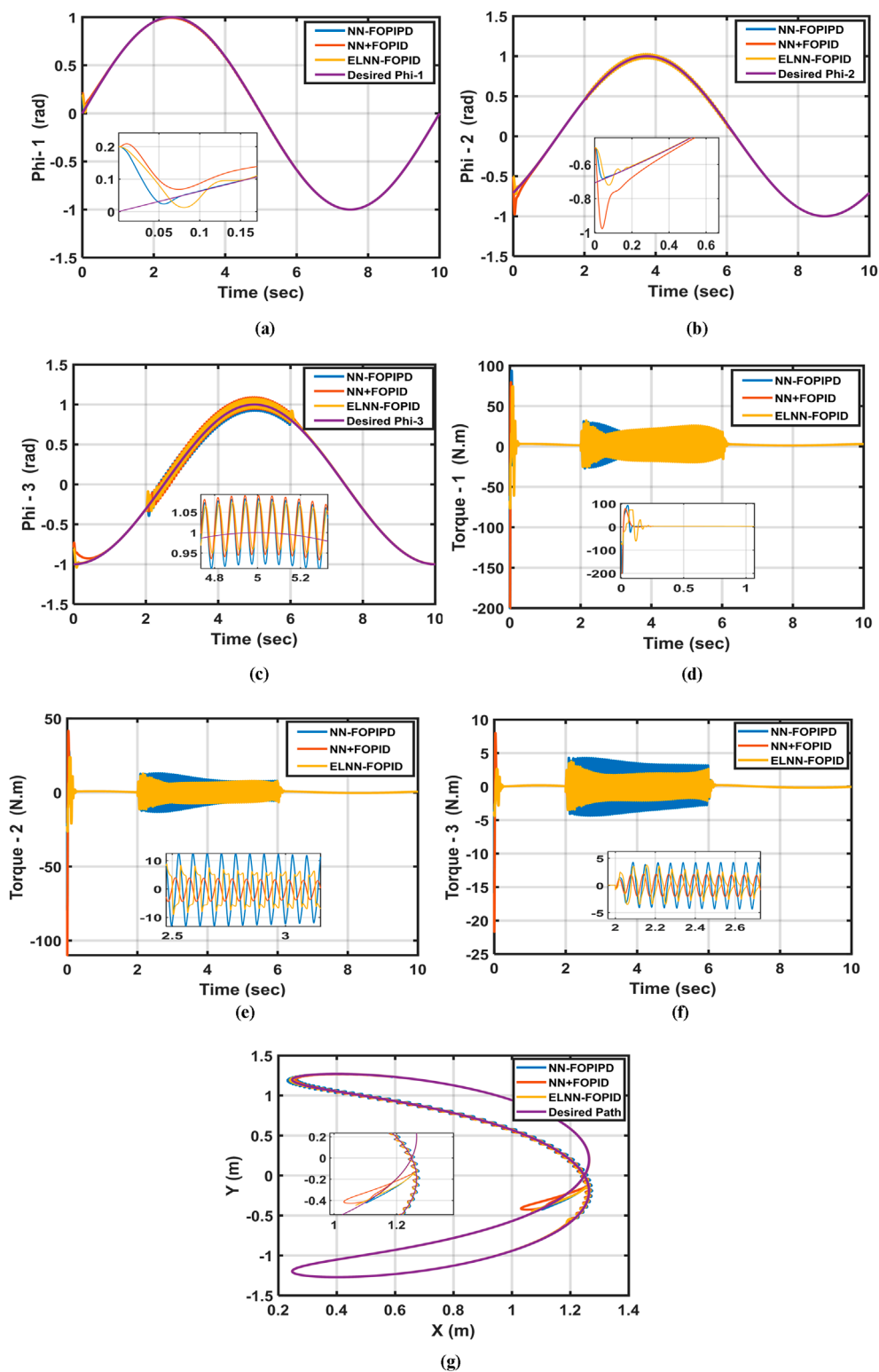


FIGURE 10

Desired and actual trajectories (a) Phi-1, (b) Phi-2, (c) Phi-3 as well as the corresponding control signals, (d) Control torque-1, (e) Control torque-2, (f) Control torque-3, and (g) actual path traced by the end effector when using starting position (0.2, -0.5, -0.8) rad, adding disturbance [sin (100t)] N-M for all links, and a 10% increase in link 3's mass.

the nonlinear problems of compensating robot manipulator control with disturbances and uncertainties and achieving precise position by designing hybrid controllers that combine the advantages of neural network-based BOA and a PID controller utilizing BOA for 3-LRRM. The parameters of these controllers are adjusted in accordance with the definition of the objective function as the ITSE. All suggested controllers are for reference trajectory tracking when all effects are present, like altering initial positions, adding disturbance, and model parameter uncertainty. The simulation's outcomes showed that the suggested NN-FOIPD controller is most effective for tracking trajectories, rejecting disturbances, adjusting parameters, and having the lowest performance index ITSE value. In other words, the NN-FOIPD controller exhibits superior tracking, stability, and robustness compared to the other suggested controllers. Future work and extension of the research idea can address the following reference ideas: [Nam Dao et al. \(2024\)](#), [Nam Dao et al. \(2025\)](#), [Köhler et al. \(2021\)](#), and [Nguyen et al. \(2024\)](#).

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 authors.

Author contributions

BO: conceptualization, formal analysis, methodology, resources, software, writing—original draft, and writing—review and editing. MM: conceptualization, formal analysis, investigation, methodology, resources, writing—original draft, and writing—review and editing. AA: conceptualization, formal analysis, investigation, methodology, validation, writing—original draft, and writing—review and editing. SA: formal analysis, investigation, methodology, resources, validation, visualization, and writing—review and editing. ARM: formal analysis, investigation, methodology, resources, validation, visualization, and writing—review and editing.

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
- Abualigah, L., Alshatti, S. M., Ikotun, A. M., Abu Zitar, R., Alsoud, A. R., Khodadadi, N., et al. (2024). "Spider monkey optimizations: Application review and results." in *Metaheuristic Optimization Algorithms*. Morgan Kaufmann. 117–131. doi:10.1016/B978-0-443-13925-3.00003-0
- Agrawal, A., Garg, D., Popli, D., Banerjee, A., Raj, A., and Dikshit, I. (2023). A review of spider monkey optimization: Modification and its biomedical application. *International Journal on Interactive Design and Manufacturing (IIJDeM)*, 1–22. doi:10.1007/s12008
- Agrawal, A., Goyal, V., and Mishra, P. (2021). Comparative study of fuzzy PID and PID controller optimized with spider monkey optimization for a robotic manipulator system. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, 14 (4), 1173–1181. doi:10.2174/2213275912666191107104635
- Andualem, W., and Gebremichael, T. (2020). Trajectory tracking control of UR5 robot manipulator using fuzzy gain scheduling terminal sliding mode controller. *J. Mechatronics Robotics* 4, 13–135. doi:10.3844/jmrsp.2020.113.135
- Ayten, K. K., Sahinkaya, M. N., and Dumlu, A. (2016). Optimum trajectory generation for redundant/hyper-redundant manipulators. *IFAC PapersOnLine* 49 (21), 493–500. doi:10.1016/j.ifacol.2016.10.651
- Ayten, K. K., Sahinkaya, M. N., and Dumlu, A. (2017). Real time optimum trajectory generation for redundant/hyper-redundant serial industrial manipulators. *Int. Journal of advanced robotic systems* 14, 1–14. doi:10.1177/1729881417737241
- Basci, A., Caan, K., Orman, K., and Derdiyok, A. (2017). Trajectory tracking control of a four-rotor unmanned aerial vehicle based on continuous sliding mode controller. *Elektron. Ir. Elektrotechnika* 23 (3), 12–19. doi:10.5755/j01.eie.23.3.18325
- Chen, Y., Xu, W., Li, Z., Song, S., Lim, C. M., Wang, Y., et al. (2017). Safety-enhanced motion planning for flexible surgical manipulator using neural dynamics. *IEEE Trans. Control Syst. Technol.* 25 (5), 1711–1723. doi:10.1109/TCST.2016.2628806

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 Computing Lab (ASCL), Prince Sultan University, Riyadh, Saudi Arabia.

Acknowledgments

This article is derived from a research grant funded by the Research, Development, and Innovation Authority (RDIA), Kingdom of Saudi Arabia, with grant number 13382-psu-2023-PSNU-R-3-1-EI-. The authors would like to acknowledge the support of Prince Sultan University, Riyadh, Saudi Arabia, for paying the article processing charges of this publication. This research is supported by the Automated Systems and Computing Lab (ASCL), Prince Sultan University, Riyadh, Saudi Arabia. In addition, 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.

- Dhivakaran, P. B., Vinodkumar, A., Vijay, S., Lakshmanan, S., Alzabut, J., El-Nabulsi, R. A., et al. (2022). Bipartite synchronization of fractional-order memristor-based coupled delayed neural networks with pinning control. *Mathematics* 10 (19), 3699. doi:10.3390/math10193699
- Dumlu, A. (2018). Practical position tracking control of a robotic manipulator based on fractional order sliding mode controller. *Elektron. Ir. Elektrotechnika* 24 (5). doi:10.5755/j01.eie.24.5.21838
- Dumlu, A., Erentürk, K., Kaleli, A., and Ayten, K. K. (2017). A comparative study of two model-based control techniques for the industrial manipulator. *Robotica* 35 (10), 2036–2055. doi:10.1017/S0263574716000709
- Ebrahimi-khani, S. (2016). Robust fractional order sliding mode control of doubly-fed induction generator (DFIG)-based wind turbines. *ISA Trans.* 63, 343–354. doi:10.1016/j.isatra.2016.03.003
- Fani, S., Ciotti, S., Catalano, M. G., Grioli, G., Tognetti, A., Valenza, G., et al. (2018). Simplifying telerobotics: wearability and teleimpedance improves human-robot interactions in teleoperation. *IEEE Robotics and Automation Mag.* 25 (1), 77–88. doi:10.1109/MRA.2017.2741579
- Huang, H. C. (2016). Fusion of modified bat algorithm soft computing and dynamic model hard computing to online self-adaptive fuzzy control of autonomous mobile robots. *IEEE Trans. Ind. Inf.* 12 (3), 972–979. doi:10.1109/TII.2016.2542206
- Kaushik, K., Kumar, A., Karthikeyan, K., Khan, A., and Abdeljawad, T. (2024). New results on controllability analysis of nonlinear fractional order integrodifferential Langevin system with multiple delays. *Results Control Optim.* 14, 100363. doi:10.1016/j.rico.2023.100363
- Köhler, J., Soloperto, R., Müller, M. A., and Allgöwer, F. (2021). A computationally efficient robust model predictive control framework for uncertain nonlinear systems. *IEEE Trans. Automatic Control* 66 (2), 794–801. doi:10.1109/TAC.2020.2982585
- Kumar, V., Rana, K. P. S., and Kler, D. (2019). Efficient control of a 3-link planar rigid manipulator using self-regulated fractional-order fuzzy PID controller. *Applied Soft Computing Number C* 82, 105531. doi:10.1016/j.asoc.2019.105531
- Lewis, F. L., Dawson, D. M., and Abdallah, C. T. (2004). *Robot manipulator control theory and practice*. New York: Marcel Dekker, USA: CRC Press, 1–632.
- Makarov, M., Grossard, M., Rodriguez-Ayerbe, P., and Dumur, D. (2016). Modeling and preview H infinity control design for motion control of elastic-joint robots with uncertainties. in *IEEE Trans. Ind. Electron.*, 63 (10), pp. 6429–6438. doi:10.1109/TIE.2016.2583406
- Mohamed, J., Oleiwi, B., Abood, L., Azar, A. T., and Hameed, I. (2023). Neural fractional order PID controllers design for 2-link rigid robot manipulator. *Fractal Fract.* 7 (9), 693. doi:10.3390/fractalfract7090693
- Mohamed, J. M., Oleiwi, B. K., Azar, A. T., and Hameed, I. A. (2024b). Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator. *Heliyon* 10 (13), e32661. doi:10.1016/j.heliyon.2024.e32661
- Mohamed, J. M., Oleiwi, B. K., Azar, A. T., and Mahlous, M. R. (2024a). Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm. *Front. Robot. AI. Sec. Robot Learn. Evol.* 11. doi:10.3389/frobt.2024.1386968
- Nam Dao, P., Nguyen, Q. P., and Vu, M. H. (2025). Adaptive optimal coordination control of perturbed Bilateral Teleoperators with variable time delays using Actor-Critic Reinforcement Learning algorithm. *Math. Comput. Simul.* 229, 151–175. doi:10.1016/j.matcom.2024.09.007
- Nam Dao, P., Nguyen, V. Q., and Duc, H. A. N. (2024). Nonlinear RISE based integral reinforcement learning algorithms for perturbed Bilateral Teleoperators with variable time delay. *Neurocomputing* 605, 128355. doi:10.1016/j.neucom.2024.128355
- Nguyen, T. Ha, Nguyen, V. C., Dang, Q. B., and Dao, P. N. (2024). An efficient Min/Max Robust Model Predictive Control for nonlinear discrete-time systems with dynamic disturbance. *Chaos, Solit. and Fractals* 180, 114551. doi:10.1016/j.chaos.2024.114551
- Nor'Azlan, N., Selamat, N., and Yahya, N. (2018). Multivariable PID controller design tuning using bat algorithm for activated sludge process. *IOP Conf. Ser. Mater. Sci. Eng.* 342 (1), 012030–012039. doi:10.1088/1757-899x/342/1/012030
- Raafat, S. M., and Raheem, F. A. (2017). *Introduction to robotics-mathematical issues, mathematical advances towards sustainable environmental systems*. Cham: Springer, 261–289.
- Rajchakit, G., Pratap, A., Raja, R., Cao, J., Alzabut, J., and Huang, C. (2019). Hybrid control scheme for projective lag synchronization of riemann-liouville sense fractional order memristive BAM NeuralNetworks with mixed delays. *Mathematics* 7 (8), 759. doi:10.3390/math7080759
- Ren, S., Han, L., Mao, J., and Li, J. (2023). Optimized trajectory tracking for robot manipulators with uncertain Dynamics: a composite position predictive control approach. *Electronics* 12, 4548. doi:10.3390/electronics12214548
- Sadrifaridpour, B., Saiedi, H., and Wang, Y. (2016). “An integrated framework for human-robot collaborative assembly in hybrid manufacturing cells,” in *2016 IEEE int. Conf. Automation science and engineering (CASE 2016)*. Fort Worth, TX, 462–4670. doi:10.1109/COASE.2016.7743441
- Sambariya, D., and Paliwal, D. (2016). Design of PIDA controller using bat algorithm for AVR power system. *Adv. Energy Power* 4 (1), 1–6. doi:10.13189/aep.2016.040101
- Sau, P. C. (2020). “Comparative study of fractional order controllers for three-link robotic manipulator system,” 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), IEEE 528–533. doi:10.1109/PARC49193.2020.236669
- Sun, C., Gao, H., He, W., and Yu, Y. (2017). “Fuzzy neural network control of a flexible robotic manipulator using assumed mode method,” in *IEEE trans. Neural networks and learning systems*, 1–14. doi:10.1109/TNNLS.2017.2743103
- Tutsoy, O., Asadi, D., Ahmadi, K., Nabavi-Chashmi, S. Y., and Iqbal, J. (2024). Minimum distance and minimum time optimal path planning with bioinspired machine learning algorithms for faulty unmanned air vehicles. *IEEE Trans. Intelligent Transp. Syst.* 25, 9069–9077. doi:10.1109/tits.2024.3367769
- Tutsoy, O., Asadi, D., Ahmadi, K., and Nabavi-Chasmi, S. Y. (2023). Robust reduced order thau observer with the adaptive fault estimator for the unmanned air vehicles. *IEEE Trans. Veh. Technol.* 72 (2), 1601–1610. doi:10.1109/tvt.2022.3214479
- Ullah, N., Shaoping, W., Khattak, M. I., and Shafi, M. (2015). Fractional order adaptive fuzzy sliding mode controller for a position servo system subjected to aerodynamic loading and nonlinearities. *Aerosp. Sci. Technol.* 43, 381–387. doi:10.1016/j.ast.2015.03.020
- Viola, J., and Angel, L. (2015). Identification, control and robustness analysis of a robotic system using fractional control, in *IEEE Lat. Am. Trans.* 13 (5), pp. 1294–1302. doi:10.1109/TLA.2015.7111982
- Wang, J., Zhu, Y., Qi, R., Zheng, X., and Li, W. (2020). Adaptive PID control of multi-DOF industrial robot based on neural network. *Journal of Ambient Intelligence and Humanized Computing* 11, 6249–6260. doi:10.1007/s12652-020-01693-w
- Wu, J., Yue, X., and Li, W. (2016). Integration of hardware and software designs for object grasping and transportation by a mobile robot with navigation guidance via a unique bearing-alignment mechanism. *IEEE/ASME Trans. Mechatronics* 21 (1), 576–583. doi:10.1109/TMECH.2015.2429681
- Xia, Y., Xie, W., and Ma, J. (2019). Research on trajectory tracking control of manipulator based on modified terminal sliding mode with double power reaching law. *Int. J. Adv. Robotic Syst.* 16 (3). doi:10.1177/1729881419847899
- Yang, X. S., and He, X. (2013). Bat algorithm: literature review and applications. *Int. J. Bio-Inspired Comput.* 5 (3), 141–149. doi:10.1504/ijbic.2013.055093
- Zemiti, N., Morel, G., Ortmaier, T., and Bonnet, N. (2007). Mechatronic design of a new robot for force control in minimally invasive surgery. *IEEE/ASME Trans. Mechatronics* 12 (2), 143–153. doi:10.1109/TMECH.2007.892831