Heliyon 10 (2024) e32661

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

# Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator

Mohamed Jasim Mohamed [a], Bashra Kadhim Oleiwi [a], Ahmad Taher Azar [b,c,d,*], Ibrahim A. Hameed [e,**]

[a] Department of Control and System Engineering, University of Technology, Iraq
[b] College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia
[c] Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, 11586, Saudi Arabia
[d] Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt
[e] Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Larsgardsvegen, 2, 6009, Alesund, Norway

ARTICLE INFO

*Keywords:*
Neural network
PID controller
3-Link rigid robotic manipulator
Nonlinear system
Coot optimization algorithm
Trajectory tracking

ABSTRACT

Robotic manipulators are nonlinear systems, multi-input multi-output, highly coupled and complicated whose performance is negatively impacted by external disturbances and parameter un-certainties. Therefore, the controllers designed for such systems must be capable of managing their complexity. The main aim of this study is to tackle the trajectory tracking issue of the three-Link Rigid Robot Manipulator (3-LRRM) based on designing three control structures using a combi-nation Neural Network (NN) with Proportional, Integral and Derivative (PID) actions named Neural Controller Like PIPD (NN-PIPD) controller, Neural Network plus PID (NN + PID) controller NN + PID controller and Elman Neural Network Like PID (ELNN-PID) controller. The parameters of the proposed controllers are adjusted utilizing the Coot Optimization Algorithm (COOA) in order to reduce the Integral Time Square Error (ITSE). A novel objective function for tuning process to produce a controller with minimum value of the chattering in the control signal is proposed. The performance of the proposed controllers is evaluated in terms of disturbance rejection, model uncertainty, fluctuating initial conditions and reference trajectory tracking. According to the simulation results proved that the suggested NN-PIPD controller outperforms all other proposed controller structures for tracking performance, stability, and robustness. As a result of the com-parison analysis the optimal controller was considered to be an NN-PIPD controller for tracking trajectory, rejecting disturbances, and parameter variation with mini-mizing ITSE of 0.001777.

## 1. Introduction

Robot manipulators have been used in industrial applications increasingly in recent years. Basically, industrial robot manipulators

* Corresponding author. College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia.
** Corresponding author.
*E-mail addresses:* 60098@uotechnology.edu.iq (M.J. Mohamed), bushra.k.oleiwi@uotechnology.edu.iq (B.K. Oleiwi), aazar@psu.edu.sa, ahmad.azar@fci.bu.edu.eg (A.T. Azar), aazar@psu.edu.sa, ibib@ntnu.no (I.A. Hameed).

https://doi.org/10.1016/j.heliyon.2024.e32661
Received 22 January 2024; Received in revised form 26 May 2024; Accepted 6 June 2024
Available online 17 June 2024

are tools for handling and positioning [1]. Numerous applications have been found, including those for forging, handling hazardous radioactive materials, spray painting, automatic assembly lines, loading and unloading cargo, and military use [2,3]. Consequently, an efficient robot manipulator can regulate both its own motion and the effects of forces that exerts to the environment. Industrial robot manipulators are actually nonlinear systems MIMO and susceptible for variations in payload, nonlinear friction, and external disturbances. Therefore, without a thorough understanding of the robot system, developing a precise controller is challenging [4]. Consequently, the challenge of regulating the location and orientation of a planar robotic manipulator has drawn attention from researchers [5]. Several research studies have been carried out on the design of various controller structures in order to solve the path tracking problem of 2-link robot manipulator (2LRM). The authors of [6] proposed several types of control strategies utilizing neural networks and PID controllers based fractional order and integer-based gorilla optimization algorithm to reduce the ITSE. In Ref. [7] the authors concentrated on using an adaptive fractional order fuzzy sliding mode proportional integral and derivative (FOFSMPID) controller based cuckoo search optimization technique. The extensive research has shown that the performance of the FOFSMPID controller is significantly better than others controllers. The authors of [8] described four structures of control that consisted of F/IOFPID controllers based most valuable player algorithm. The FOFPD-FOPID has lowest ITSE and best structure. Non-singular rapid terminal sliding mode control approach with a wavelet neural networks observer was proposed by the authors in Ref. [9]. Stability analysis utilizing the Lyapunov criterion shows that this control technique is stable. A FOPID controller was presented by the authors in Ref. [10]. The Interval Type II Fuzzy Fractional-Order Proportional Integral Differential (IT2FO-FPID) controller-based whale optimization algorithm (WOA) was developed by the authors of [11]. A Non-Linear Fractional order PID controller (NLF-PID) was designed by the authors in Ref. [12]. A genetic algorithm based on multiple-objectives optimization is suggested for adjusting the gains of the proposed controller. Compared with other proposed controllers in different scenarios, the results showed that NLF-PID provided reliable and efficient control. The authors in Ref. [13] proposed a fractional order controller. The effectiveness method with respect to good tracking performance is demonstrated by simulation results and comparison with the calculated torque method. The FOFPID controller based Cuckoo search algorithm to modify the controller parameters was studied by the authors in Ref. [14]. Simulation results showed FOFPID controller is better in trajectory tracking compared to traditional PID and fuzzy PID controllers. The authors of [15] worked on optimizing the gains of FOPID controller using Bat algorithm. The authors of [16] designed a PID controller using a fuzzy neural network method. The efficiency of the proposed strategy is demonstrated through the numerical simulation results [17]. suggested using type-3 (T3) fuzzy logic systems (FLSs) (also known as T3-FLSs) to estimate and control the symmetrical perturbations and the dynamics of the robotic manipulators. The superiority of the proposed controller is shown by multiple simulations. Several research studies have been carried out on the design of various controller structures in order to solve the path tracking problem of 3-link robot manipulator (3LRM). In Ref. [18] the authors presented the development of Fuzzy Logic and Adaptive Neuro-Fuzzy Inference System (ANFIS) controllers. The ANFIS Controller continuously performed better than the Fuzzy Logic Controller. A fractional-order self-tuned fuzzy PID (FOSTFPID) controller was presented in Ref. [19]. FOSTFPID, fractional-order fuzzy PID, and integer-order self-tuning fuzzy PID controllers' performances were compared. All adjusted by the cuckoo search method. The comparison findings showed that FOSTFPID was clearly superior. A fuzzy fractional order adaptive impedance controller was presented by the authors in Ref. [20] in order to maintain force error in the tracking dynamic phase and prevent power overflow in the connection stage. The authors in Ref. [21] designed and three distinct position control approaches named PID, PD, and FLC controllers. The FLC showed minimum overshoot. On the other hand, the rising time and settling time performance of the PID and PD controllers were better. The authors in Ref. [22] introduced a Self-Regulated Fractional-Order Fuzzy Proportional–Integral–Derivative (SRFOFPID) controller. The study's findings showed that the SRFOFPID controller's degree of freedom and robustness are increased by fractional operators. The authors of [23] provided a sliding mode adaptive fractional fuzzy control. By using fractional-order adaptation laws to update fuzzy controller parameters. The integer-order Lyapunov stability criterion is applied, and fractional-order integral Lyapunov functions are suggested to aid in the stability analysis. The authors of [24] presented and evaluated two optimization methods for figuring out the best gains for a PID controller. an improved artificial bee colony and the basic artificial bee colony with multi-elite guiding. The Fuzzy-PID controller design of a 4 DOF industrial arm robot manipulator was utilized by the authors of [25]. The authors of [26] introduced PID, tilt integral derivative (TID), and fractional-order proportional integral derivative (FOPID) controllers based grey wolf optimization (GWO) and particle swarm optimization (PSO) to optimize the controller gains. When compared to FOPID-PSO and other controllers, the FOPID controller optimized by GWO yields lower ITAE, resulting in superior performance [27–42]. introduced fuzzy based control algorithms for nonlinear system [43–46]. introduced safe and fast tracking on a robot manipulator and fuzzy technique. As is clear, most of the techniques mentioned above in most studies have their pros and cons. They were used in terms of complexity and time-consuming calculations at the expense of accuracy and time.

In this study the main goal is to solve the trajectory tracking issue of the 3-LRRM based on designing three control structures. The interesting element and motivation in the proposed structures is to combine the power of neural network representation of linear and non-linear functions with the power and robustness of the PID controller for the purpose of obtaining a hybrid controller with both features.

The main contributions of this work are.

1. Designing three control structures consisting of Neural Network (NN) with Proportional, Integral and Derivative (PID) actions named Neural Controller Like PIPD (NN-PIPD) controller, Neural Network plus PID (NN + PID) controller NN + PID controller and Elman Neural Network Like PID (ELNN-PID) controller for 3-LRRM trajectory tracking problem.
2. Finding the optimal parameters of the proposed controllers based on Coot Optimization Algorithm (COOA) to give the best desired performance.

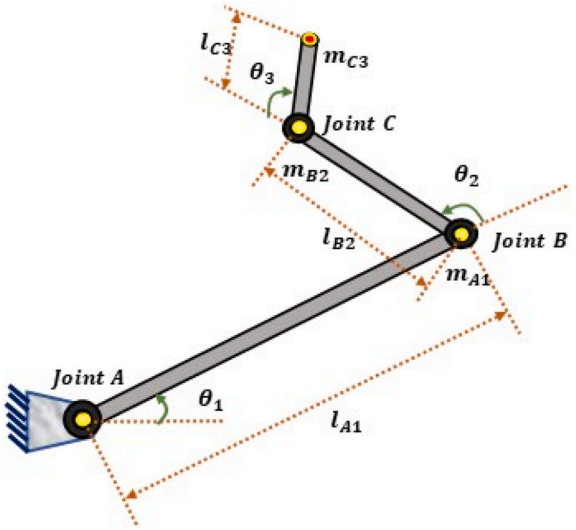


**Fig. 1.** Three-LRRM's structural diagram.

3. Suggesting a new objective function for tuning process to produce a controller with the least amount of chattering in the control signal.
4. Implementing a comparison study among the proposed controllers in terms of changing initial conditions, adding external disturbances, changing system's parameters, and adding all uncertainty effects together in order to select the best controller., changing system's parameters, and adding all uncertainty effects together in order to select the best controller.

The remaining parts of the article are structured in the following: The 3-LRRM's dynamic model is explained in Section 2. The suggested controllers are outlined in Section 3. Section 4 displays the COOA. The simulation results are shown in Section 5. Section 6 provides the conclusion.

## 2. Three-LRRM dynamic model

The robot manipulator consists of a set of links connected to one another by joints. A robot manipulator that is planar can only move in one plane. Many industrial robotic systems for assisted and medical automation use prototypes of planar robot manipulators. This study considers the planar robot manipulator with three links as revolute joints, assuming that every joint is taken to be actuated. A 3-LRRM system with 3 DOF is demonstrated in Fig. 1. The system's 1st link is connected to a rigid bottom by a frictionless shaft, and the 2nd link is connected to the 1st link's end by a frictionless ball bearing. Ball bearings that don't cause friction are used to connect the second and 3rd links [19]. The fundamental control equations in robotics are constructed using the manipulator's dynamic equation of motion. The manipulator's arm in a robot system moves dynamically due to the torques generated by the actuators [47]. A 3-LRRM has the following dynamic model.

The 3-LRRM is described by the Lagrange dynamic model [48].

The following describes the *x*- and *y*-position equations for link1, as indicated in equations (1) and (2) [48]:

$$x_1 = l_{A1}\, cos\,(\theta_1) \tag{1}$$

$$y_1 = l_{A1}\, sin\,(\theta_1) \tag{2}$$

In the same way, link2's x- and y-position equations are provided by equations (3) and (4)

$$x_2 = l_{A1}\, cos\,(\theta_1) + l_{B2}\, cos\,(\theta_1 + \theta_2) \tag{3}$$

$$y_2 = l_{A1}\, sin\,(\theta_1) + l_{B2}\ \ sin\,(\theta_1 + \theta_2) \tag{4}$$

Additionally, link3's x- and y-position equations are provided by equations (5) and (6)

$$x_3 = l_{A1}\, cos\,(\theta_1) + l_{B2}\, cos\,(\theta_1 + \theta_2) + l_{C3}\, cos(\theta_1 + \theta_2 + \theta_3) \tag{5}$$

$$y_3 = l_{A1}\, sin\,(\theta_1) + l_{B2}\ \ sin\,(\theta_1 + \theta_2) + l_{C3}\, sin(\theta_1 + \theta_2 + \theta_3) \tag{6}$$

where, $l_i, \theta_i$ , $x_i$ and $y_i$ are the length, angle, x-position and y-position of the link *i*.

The following is the definition of Kinetic Energy (*KE*), as expressed in equation (7):

$$KE = \frac{1}{2}\, m_{A1}\, v_1^2 + \frac{1}{2}\, m_{B2}\, v_2^2 + \frac{1}{2}\, m_{C3}\, v_3^2 \tag{7}$$

where, $v_i$ and $m_i$ ,are the velocity and mass at the end of the link *i*. The velocities for $m_{A1}$ , $m_{B2}$ and $m_{C3}$ are represented by the variables $v_1$, $v_2$ and $v_3$ , respectively, and can be computed in equation (8):

$$v_1 = \sqrt{\dot{x}_1^2 + \dot{y}_1^2}, v_2 = \sqrt{\dot{x}_2^2 + \dot{y}_2^2}, v_3 = \sqrt{\dot{x}_3^2 + \dot{y}_3^2} \tag{8}$$

Thus, the Kinetic Energy (KE) is defined in equation (9):

$$KE = \frac{1}{2} m_{A1} \left(\dot{x}_1^2 + \dot{y}_1^2\right) + \frac{1}{2}\, m_{B2}\, \left(\dot{x}_2^2 + \dot{y}_2^2\right) + \frac{1}{2}\, m_{C3}\, \left(\dot{x}_3^2 + \dot{y}_3^2\right) \tag{9}$$

Additionally, the Potential Energy (*PE*) can be expressed as presented in equations 10 and 11:

$$PE = \sum_{i=1}^{3} m_i\, g\, h_i(\theta) \tag{10}$$

where, $g$ is the gravity and $h_i$ is the height of mass of link *i*.

$$PE = m_{A1}\, g\, l_{A1}\, sin(\theta_1) + m_{A2}\, g(l_{A1}\ sin(\theta_1) + l_{B2}\, sin(\theta_1 + \theta_2)) + m_{C3}\, g(\, l_{A1}\, sin(\theta_1) + l_{B2}\, sin(\theta_1 + \theta_2) + l_{C3}\, sin(\theta_1 + \theta_2 + \theta_3)) \tag{11}$$

After that, the Lagrangian L is explained in equation (12) using the Lagrange dynamic:

$$L = KE - PE \tag{12}$$

The following is the Euler Lagrange Equation (13):

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_i}\right] - \frac{\partial L}{\partial \theta_i} = F\theta_i \tag{13}$$

whereas the torque applied onto *i-th* link is denoted by $F\theta_i$ or $\tau_i$.

The standard form for these manipulator dynamics is as indicated in equations 14–24 [49]:

$$D(\theta)\ddot{\theta} + P\left(\theta, \dot{\theta}^2\right) + R\left(\theta, \dot{\theta}_i\dot{\theta}_j\right) + G(\theta) = \tau \tag{14}$$

where, $D(\theta)$ is the inertia matrix.

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \tag{15}$$

$$D_{11} = (m_{A1} + m_{B2} + m_{C3}) l_{A1}^2 + (m_{B2} + m_{C3}) l_{B2}^2 + m_{C3} l_{C3}^2 + 2m_{C3} l_{A1} l_{C3}\ \ cos\left(\theta_2 + \theta_3\right] + 2(m_{B2} + m_{C3}) l_{A1} l_{B2}\, cos(\theta_2) + 2m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{16}$$

$$D_{12} = (m_{B2} + m_{C3}) l_{B2}^2 + m_{C3} l_{C3}^2 + m_{C3} l_{A1} l_{C3}\, cos(\theta_2 + \theta_3) + (m_{B2} + m_{C3}) l_{A1} l_{B2}\, cos(\theta_2) + 2m_{C3} l_{B2} lC_3\, cos\ (\theta_3) \tag{17}$$

$$D_{13} = m_{C3} l_3^2 + m_{C3} l_{A1} l_{C3}\, cos(\theta_2 + \theta_3) + m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{18}$$

$$D_{21} = m_{B2} l_{B2}^2 + m_{C3} l_{B2}^2 + m_{C3} l_{C3}^2 + m_{C3} l_{A1} l_{C3}\, cos(\theta_2 + \theta_3) + m_{B2} l_{A1} l_{B2}\, cos(\theta_2) + m_{C3} l_{A1} l_{B2}\, cos(\theta_2) + 2m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{19}$$

$$D_{22} = m_{B2} l_{B2}^2 + m_{C3} l_{B2}^2 + m_{C3} l_3^2 + 2m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{20}$$

$$D_{23} = m_{C3} l_{C3}^2 + m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{21}$$

$$D_{31} = m_{C3} l_{C3}^2 + m_{C3} l_{A1} l_{C3}\, cos(\theta_2 + \theta_3) + m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{22}$$

$$D_{32} = m_{C3} l_3^2 + m_{C3} l_{B2} l_{C3}\, cos\ (\theta_3) \tag{23}$$

$$D_{33} = m_{C3} l_{C3}^2 \tag{24}$$

$P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$, is the term for centrifugal force, defined in equations $(25 - 27)$: (25)

$$P_1 = -\, l_{A1}(m_{C3} l_{C3}\, sin(\theta_2 + \theta_3) + m_{B2} l_{B2}\, sin(\theta_2) + m_{C3} l_{B2}\, sin(\theta_2))\dot{\theta}_2^2 - m_{C3} l_{C3}(l_{A1}\, sin(\theta_2 + \theta_3) + l_{B2}\, sin(\theta_3))\dot{\theta}_3^2 \tag{26}$$

**Table 1**
Three-LRRM's parameters.

| Parameters | Nominal value |
|---|---|
| $l_{A1}$ | 0.8 m |
| $l_{B2}$ | 0.4 m |
| $l_{C3}$ | 0.2 m |
| $m_{A1}$ | 0.1 kg |
| $m_{B2}$ | 0.1 kg |
| $m_{C3}$ | 0.1 kg |
| $g$ | 9.81 m/ $s^2$ |

$$P_2 = l_{A1}(m_{C3}l_{C3}\ sin(\theta_2+\theta_3)+m_{B2}l_{B2}\ sin(\theta_2)+m_{C3}l_{B2}\ sin(\theta_2))\dot{\theta}_1^2 - m_{C3}l_{B2}l_{C3}\ sin\ (\theta_3)\dot{\theta}_3^2$$

$$P_3 = m_{C3}l_{C3}(l_{C1}\ sin(\theta_2+\theta_3)+l_{B2}\ sin(\theta_3))\dot{\theta}_1^2 + m_{C3}l_{B2}l_{C3}\ sin\ (\theta_3)\dot{\theta}_2^2 \tag{27}$$

$$R = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}, \text{is the term in Coriolis that is defined in equations } (28-31) \tag{28}$$

$$R_1 = -2l_{A1}(m_{C3}l_{C3}\ sin(\theta_2+\theta_3)+(m_{B2}+m_{C3})lB_2\ sin(\theta_2))\dot{\theta}_1\dot{\theta}_2 - 2m_{C3}l_{C3}(l_{A1}\ sin(\theta_2+\theta_3)+l_{B2}\ sin(\theta_3))\dot{\theta}_2\dot{\theta}_3$$
$$-2m_{C3}l_{C3}(l_{A1}\ sin(\theta_2+\theta_3)+l_{B2}\ sin(\theta_3))\dot{\theta}_1\dot{\theta}_3 \tag{29}$$

$$R_2 = -2m_{C3}l_{B2}l_{C3}\ sin(\theta_3)\dot{\theta}_1\dot{\theta}_3 - 2m_{C3}l_{B2}l_{C3}\ sin\ (\theta_3)\dot{\theta}_2\dot{\theta}_3 \tag{30}$$

$$R_3 = 2m_{C3}l_{B2}l_{C3}\ sin\ (\theta_3)\dot{\theta}_1\dot{\theta}_2 \tag{31}$$

Potential energy terms are defined in $G$, as expressed in equations 32–35:

$$G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \tag{32}$$

$$G_1 = (m_{A1}+m_{B2}+m_{C3})gl_{A1}\ cos(\theta_1)+(m_{B2}+m_{C3})gl_2\ cos(\theta_1+\theta_2)+m_{C3}gl_{C3}\ cos\ (\theta_1+\theta_2+\theta_3) \tag{33}$$

$$G_2 = (m_{B2}+m_{C3})gl_{B2}\ cos(\theta_1+\theta_2)+m_{C3}gl_{C3}\ cos\ (\theta_1+\theta_2+\theta_3) \tag{34}$$

$$G_3 = m_{C3}gl_{C3}\ cos\ (\theta_1+\theta_2+\theta_3) \tag{35}$$

Using forward kinematic [30], the coordinates of the 3-LRRM's end-effector will be found regarding the joint angles $\theta_{r1}$, $\theta_{r2}$ and $\theta_{r3}$, as shown in the following equations 36 and 37:

For a reference trajectory

$$x_r = l_{A1}\ cos(\theta_{r1})+l_{B2}\ cos(\theta_{r1}+\theta_{r2})+l_{C3}\ cos(\theta_{r1}+\theta_{r2}+\theta_{r3}) \tag{36}$$

$$y_r = l_{A1}\ sin(\theta_{r1})+l_{B2}\ sin(\theta_{r1}+\theta_{r2})+l_{C3}\ sin(\theta_{r1}+\theta_{r2}+\theta_{r3}) \tag{37}$$

where, $\theta_{r1}$, $\theta_{r2}$ *and* $\theta_{r3}$ are the desired trajectories and $x_r, y_r$ is the desired coordinate of end effector

Table 1 contains a list of the 3-LRRM parameter values that were utilized in this work [19].

## 3. Three-LRRM feedback linearization

The feedback linearization of the 3-LRRM is employed by applying the nonlinear feedback control law, to determine the necessary joint torques, as demonstrated in equations 38–54 [50].

$$D(\theta)\ddot{\theta}+P\left(\theta,\dot{\theta}^2\right)+R\left(\theta,\dot{\theta}_i\dot{\theta}_j\right)+G(\theta)=\tau \tag{38}$$

$$\begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix}\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}+\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}+\begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}+\begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}=\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{39}$$

$$\ddot{\theta} = D^{-1}(\theta)\left(\tau - P\left(\theta, \dot{\theta}^2\right) - R\left(\theta, \dot{\theta}_i\dot{\theta}_j\right) - G(\theta)\right) \tag{40}$$

where,

$$D^{-1}(\theta) = \frac{1}{det(D)}\begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \tag{41}$$

$$\det(\mathrm{D}) = (D_{11}D_{22}D_{33} - D_{11}D_{23}D_{32} - D_{12}D_{21}D_{33} + D_{12}D_{23}D_{31} + D_{13}D_{21}D_{32} - D_{13}D_{22}D_{31}) \tag{42}$$

$$Q_{11} = D_{22}D_{33} - D_{23}D_{32} \tag{43}$$

$$Q_{12} = -(D_{12}D_{33} - D_{13}D_{32}) \tag{44}$$

$$Q_{13} = D_{12}D_{23} - D_{13}D_{22} \tag{45}$$

$$Q_{21} = -(D_{21}D_{33} - D_{23}D_{31}) \tag{46}$$

$$Q_{22} = D_{11}D_{33} - D_{13}D_{31} \tag{47}$$

$$Q_{23} = -(D_{11}D_{23} - D_{13}D_{21}) \tag{48}$$

$$Q_{31} = D_{21}D_{32} - D_{22}D_{31} \tag{49}$$

$$Q_{32} = -(D_{11}D_{32} - D_{12}D_{31}) \tag{50}$$

$$Q_{33} = D_{11}D_{22} - D_{12}D_{21} \tag{51}$$

$$\ddot{\theta}_1 = \frac{1}{\det(D)}(Q_{11}\tau_1 + Q_{12}\tau_2 + Q_{13}\tau_3 - Q_{11}(P_1 + R_1 + G_1) - Q_{12}(P_2 + R_2 + G_2) - Q_{13}(P_3 + R_3 + G_3)) \tag{52}$$

$$\ddot{\theta}_2 = \frac{1}{det(D)}(Q_{21}\tau_1 + Q_{22}\tau_2 + Q_{23}\tau_3 - Q_{21}(P_1 + R_1 + G_1) - Q_{22}(P_2 + R_2 + G_2) - Q_{23}(P_3 + R_3 + G_3)) \tag{53}$$

$$\ddot{\theta}_3 = \frac{1}{det(D)}(Q_{31}\tau_1 + Q_{32}\tau_2 + Q_{33}\tau_3 - Q_{31}(P_1 + R_1 + G_1) - Q_{32}(P_2 + R_2 + G_2) - Q_{33}(P_3 + R_3 + G_3)) \tag{54}$$

Now let

$$x_1 = \theta_1, x_2 = \theta_2, x_3 = \theta_3, x_4 = \dot{\theta}_1,\ x_5 = \dot{\theta}_2, x_6 = \dot{\theta}_3 \tag{55}$$

$$\dot{x}_1 = x_4, \dot{x}_2 = x_5,\ \dot{x}_3 = x_6,\ \dot{x}_4 = \ddot{\theta}_1,\ \dot{x}_5 = \ddot{\theta}_2, \dot{x}_6 = \ddot{\theta}_3 \tag{56}$$

The general form of state space is presented in equation (57) and the details are represented by equations 58–87:

$$\dot{x} = f(x) + g(x)\tau \tag{57}$$

where,

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix}, g(x) = \begin{bmatrix} g_{11}(x) & g_{12}(x) & g_{13}(x) \\ g_{21}(x) & g_{22}(x) & g_{23}(x) \\ g_{31}(x) & g_{32}(x) & g_{33}(x) \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{58}$$

And

$$f_1(x) = \frac{1}{det(D)}(-Q_{11}(P_1 + R_1 + G_1) - Q_{12}(P_2 + R_2 + G_2) - Q_{13}(P_3 + R_3 + G_3)) \tag{59}$$

$$f_2(x) = \frac{1}{det(D)}(-Q_{21}(P_1 + R_1 + G_1) - Q_{22}(P_2 + R_2 + G_2) - Q_{23}(P_3 + R_3 + G_3)) \tag{60}$$

$$f_3(x) = \frac{1}{det(D)}(-Q_{31}(P_1 + R_1 + G_1) - Q_{32}(P_2 + R_2 + G_2) - Q_{33}(P_3 + R_3 + G_3)) \tag{61}$$

$$g_{11}(x)=\frac{Q_{11}}{det(D)}, g_{12}(x)=\frac{Q_{12}}{det(D)}, g_{13}(x)=Q_{13}\Big/det(D) \tag{62}$$

$$g_{21}(x)=\frac{Q_{21}}{det(D)}, g_{22}(x)=\frac{Q_{22}}{det(D)}, g_{23}(x)=Q_{23}\Big/det(D) \tag{63}$$

$$g_{31}(x)=\frac{Q_{31}}{det(D)}, g_{32}(x)=\frac{Q_{32}}{det(D)}, g_{33}(x)=Q_{33}\Big/det(D) \tag{64}$$

$$\tau=g^{-1}(x)(-f(x)+u) \tag{65}$$

$$g^{-1}(x)=\frac{1}{det\,(g)}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{66}$$

$$det(g)=\left(g_{11}g_{22}g_{33}-g_{11}g_{23}g_{32}-g_{12}g_{21}g_{33}+g_{12}g_{23}g_{31}+g_{13}g_{21}g_{32}-g_{13}g_{22}g_{31}\right) \tag{67}$$

$$k_{11}=g_{22}g_{33}-g_{23}g_{32} \tag{68}$$

$$k_{12}=-\left(g_{12}g_{33}-g_{13}g_{32}\right) \tag{69}$$

$$k_{13}=g_{12}g_{23}-g_{13}g_{22} \tag{70}$$

$$k_{21}=-\left(g_{21}g_{33}-g_{23}g_{31}\right) \tag{71}$$

$$k_{22}=g_{11}g_{33}-g_{13}g_{31} \tag{72}$$

$$k_{23}=-\left(g_{11}g_{23}-g_{13}g_{21}\right) \tag{73}$$

$$k_{31}=g_{21}g_{32}-g_{22}g_{31} \tag{74}$$

$$k_{32}=-\left(g_{11}g_{32}-g_{12}g_{31}\right) \tag{75}$$

$$k_{33}=g_{11}g_{22}-g_{12}g_{21} \tag{76}$$

$$\tau_1=\left(-k_{11}f_1(x)-k_{12}f_2(x)-k_{13}f_3(x)+k_{11}u_1+k_{12}u_2+k_{13}u_3\right)/\,det\,(g) \tag{77}$$

$$\tau_2=\left(-k_{21}f_1(x)-k_{22}f_2(x)-k_{23}f_3(x)+k_{21}u_1+k_{22}u_2+k_{23}u_3\right)/\,det\,(g) \tag{78}$$

$$\tau_3=\left(-k_{31}f_1(x)-k_{32}f_2(x)-k_{33}f_3(x)+k_{31}u_1+k_{32}u_2+k_{33}u_3\right)/\,det\,(g) \tag{79}$$

$$\dot{x}_1=x_4, \dot{x}_2=x_5, \dot{x}_3=x_6, \dot{x}_4=u_1, \dot{x}_5=u_2, \dot{x}_6=u_3 \tag{80}$$

$$e_1=x_{1d}-x_1, e_2=x_{2d}-x_2, e_3=x_{3d}-x_3 \tag{81}$$

$$e_4=\dot{x}_{1d}-x_4, e_5=\dot{x}_{2d}-x_5, e_6=\dot{x}_{3d}-x_6 \tag{82}$$

$$\dot{e}_1=\dot{x}_{1d}-\dot{x}_1, \dot{e}_2=\dot{x}_{2d}-\dot{x}_2, \dot{e}_3=\dot{x}_{3d}-\dot{x}_3 \tag{83}$$

$$\dot{e}_4=\ddot{x}_{1d}-\dot{x}_4, \dot{e}_5=\ddot{x}_{2d}-\dot{x}_5, \dot{e}_6=\ddot{x}_{3d}-\dot{x}_6 \tag{84}$$

$$\dot{e}_1=e_4, \dot{e}_2=e_5, \dot{e}_3=e_6, \dot{e}_4=\ddot{x}_{1d}-u_1 \tag{85}$$

$$\dot{e}_5=\ddot{x}_{2d}-u_2, \dot{e}_6=\ddot{x}_{3d}-u_3 \tag{86}$$

$$u_1=\ddot{x}_{1d}-V_1, u_2=\ddot{x}_{2d}-V_2, u_3=\ddot{x}_{3d}-V_3 \tag{87}$$

where, $V_1$ , $V_2$ and $V_3$ are proposed controllers.

## 4. Proposed hybrid PID Neural Controllers

The PID controller is one of the most popular control methods and implemented in different applications, it is reliable and easy to

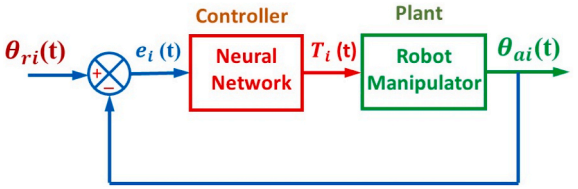


**Fig. 2.** General block diagram for neural network combined with PID controller.

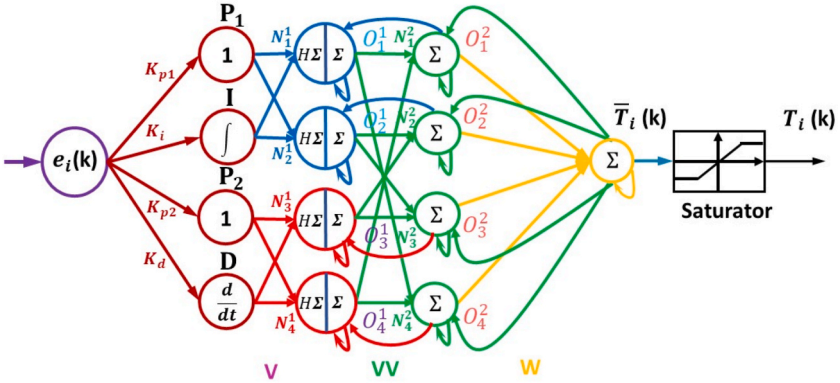


**Fig. 3.** The structure of Neural PIPD controller.

understand. The following equation (88) is a representation of the PID controller:

$$u(t) = K_p\, e(t) + K_d \frac{d}{dt} e(t) + K_i \int_0^t e(\tau) d\tau \tag{88}$$

where, the control action is represented by $u(t)$, the error by $e(t)$ and the change in error by $\dot{e}(t)$, and the proportional gain is represented by $Kp$. As a result, A control action is generated that is proportionate to the error signal $e(t)$. The essential component for decreasing the steady-state error is K*i*. by continuously integrating the error signal $e(t)$ and K*d* is the derivative part to generate the control signal proportionate to the error change with time, improving transient response and causing output overshoot damping [51]. The proposed controllers' structures that combine the three actions proportional, integral, derivative and neural network in different ways will be illustrated below. To demonstrate the procedure for establishing the design of these controllers, the major parts and structures of the suggested controllers will be described in the following sections. The general block diagram of the proposed hybrid controllers using neural network combined with PID controller is presented in figure (2).

The proposed method has a good merit where the neural network has a good capability in mapping linear and nonlinear data and the three operations of PID controller are known have a good robustness capability. Therefore, the combinations of these two tools give integration between each other and give superior results. In addition, the method has a good merit where it is clear in application, not need good an expert of the underlying system, can be used the constraints if they are presents on the system's states and control signals and can be used for other systems by same idea as well as competes other methods in the performance and robustness.

### 4.1. Neural Controller Like PIPD (NN-PIPD) controller

Fig. 3 depicts the architecture of this neural PID controller.

The difference between the *ith* link's required position $\theta_{ri}(t)$ and actual position $\theta_{ai}(t)$ is known as $e_{\theta i}(t)$. The input layer with a one node is $e_{\theta i}(t)$. The first hidden layer with four nodes $P_1$, I, $P_2$ and D as following as illustrated in equations 89–92;

$$P_{1n}(t) = K_{p1n}\, e_{\theta n}(t) \quad \text{or} \quad P_{1n}(k) = K_{p1n} e_{\theta n}(k) \tag{89}$$

$$I_n(t) = K_{in} \int e_{\theta n}(t)\, dt \ \text{ or } \ I_n(k) = K_{in} \sum_{j=0}^{k} h \times\ e_{\theta n}(j) \tag{90}$$

$$P_{2n}(t) = K_{p2n}\, e_{\theta n}(t) \text{ or } P_{2n}(k) = K_{p2n}\, e_{\theta n}(k) \tag{91}$$

$$D_n(t) = K_{dn} \frac{d}{dt} e_{\theta n}(t) \text{ or } \quad D_n(k) = K_{dn}\, (e_{\theta n}(k) - e_{\theta n}(k-1)\,) \Big/ h \tag{92}$$

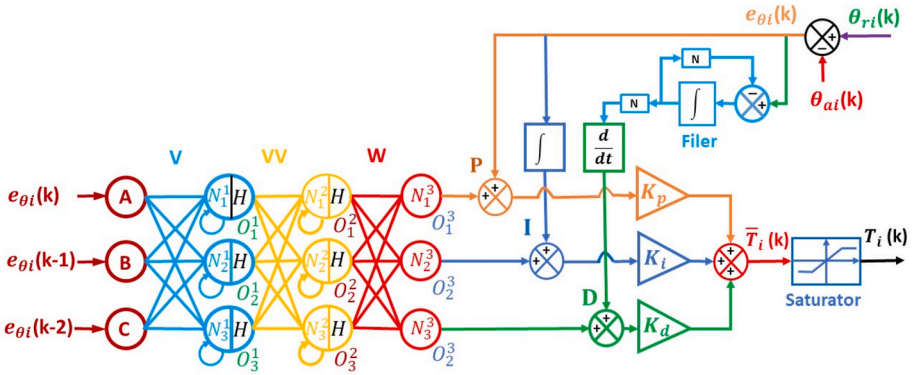


**Fig. 4.** Neural controller + PID controller.

where the number of links $n = 1, 2, 3$ , and the sum of the second hidden layer is defined in equations 93 and 94;

$$\begin{bmatrix} N_1^1(k) \\ N_2^1(k) \\ N_3^1(k) \\ N_4^1(k) \end{bmatrix} = \begin{bmatrix} v11 & v12 & 0 & 0 \\ v21 & v22 & 0 & 0 \\ 0 & 0 & v33 & v34 \\ 0 & 0 & v43 & v44 \end{bmatrix} \begin{bmatrix} \mathrm{P}_1(k) \\ \mathrm{I}(k) \\ \mathrm{P}_2(k) \\ \mathrm{D}(k) \end{bmatrix} \tag{93}$$

$$\begin{bmatrix} O_1^1(k) \\ O_2^1(k) \\ O_3^1(k) \\ O_4^1(k) \end{bmatrix} = \begin{bmatrix} H(N_1^1(k)) \\ H(N_2^1(k)) \\ H(N_3^1(k)) \\ H(N_4^1(k)) \end{bmatrix} + \begin{bmatrix} v13 \times O_1^1(k-1) \\ v14 \times O_2^1(k-1) \\ v23 \times O_3^1(k-1) \\ v24 \times O_4^1(k-1) \end{bmatrix} + \begin{bmatrix} v31 \times O_1^2(k-1) \\ v32 \times O_2^2(k-1) \\ v41 \times O_3^2(k-1) \\ v42 \times O_4^2(k-1) \end{bmatrix} \tag{94}$$

where the activation function is expressed in equation (95),

$$H = \frac{4}{(1+e^{-net})} - 2 \tag{95}$$

And the sum of the third hidden layer and the output are defined in equations 96–98;

$$\begin{bmatrix} O_1^2(k) \\ O_4^2(k) \end{bmatrix} = \begin{bmatrix} vv11 & vv12 \\ vv21 & vv22 \end{bmatrix} \begin{bmatrix} O_1^1(k) \\ O_4^1(k) \end{bmatrix} + \begin{bmatrix} vv13 \times O_1^2(k-1) \\ vv23 \times O_4^2(k-1) \end{bmatrix} + \begin{bmatrix} vv14 \times \overline{T}(k-1) \\ vv24 \times \overline{T}(k-1) \end{bmatrix} \tag{96}$$

$$\begin{bmatrix} O_2^2(k) \\ O_3^2(k) \end{bmatrix} = \begin{bmatrix} vv31 & vv32 \\ vv41 & vv42 \end{bmatrix} \begin{bmatrix} O_2^1(k) \\ O_3^1(k) \end{bmatrix} + \begin{bmatrix} vv33 \times O_2^2(k-1) \\ vv43 \times O_3^2(k-1) \end{bmatrix} + \begin{bmatrix} vv34 \times \overline{T}(k-1) \\ vv44 \times \overline{T}(k-1) \end{bmatrix} \tag{97}$$

$$\overline{T_i}(k) = \overline{T_i}(k-1) + w_1 \times O_1^2(k) + w_2 \times O_2^2(k) + w_3 \times O_3^2(k) + w_4 \times O_4^2(k) \tag{98}$$

### 4.2. Neural network + PID (NN + PID) controller

The suggested hybrid controller's structure consists of proportional, integral and derivative actions. Each output of these action is added to a one output of the neural network to produce the control signal as shown in Fig. 4.

In this structure the input layer has three nodes $e_{\theta i}(\mathrm{k}), e_{\theta i}(\mathrm{k}-1)$, and $e_{\theta i}(\mathrm{k}-2)$ or A, B and C nodes, as displayed in equation (99). where;

$$\begin{bmatrix} N_1^1(k) \\ N_2^1(k) \\ N_3^1(k) \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \\ v31 & v32 & v33 \end{bmatrix} \begin{bmatrix} e_{\theta i}(k) \\ e_{\theta i}(k-1) \\ e_{\theta i}(k-2) \end{bmatrix} + \begin{bmatrix} N_1^1(k-1) \\ N_2^1(k-1) \\ N_3^1(k-1) \end{bmatrix} \tag{99}$$

The output of first hidden layer is presented in equation (100);

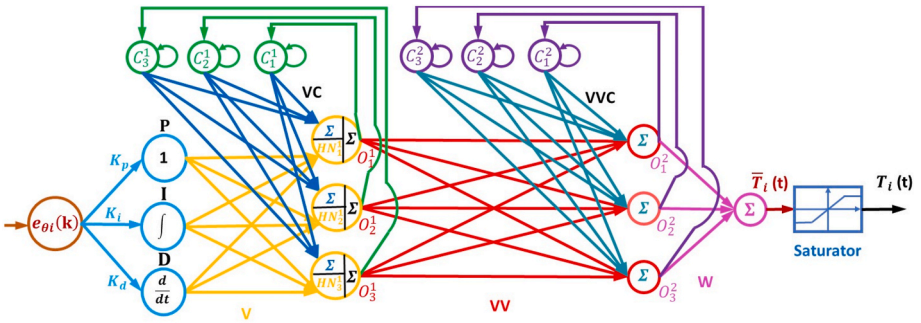


**Fig. 5.** Elman neural like PID controller.

$$\begin{bmatrix} O_1^1(k) \\ O_2^1(k) \\ O_3^1(k) \end{bmatrix} = \begin{bmatrix} H(N_1^1(k)) \\ H(N_2^1(k)) \\ H(N_3^1(k)) \end{bmatrix} \tag{100}$$

$$\begin{bmatrix} N_1^2(k) \\ N_2^2(k) \\ N_3^2(k) \end{bmatrix} = \begin{bmatrix} vv11 & vv12 & vv13 \\ vv21 & vv22 & vv23 \\ vv31 & vv32 & vv33 \end{bmatrix} \begin{bmatrix} O_1^1(k) \\ O_2^1(k) \\ O_3^1(k) \end{bmatrix} + \begin{bmatrix} N_1^2(k-1) \\ N_2^2(k-1) \\ N_3^2(k-1) \end{bmatrix} \tag{101}$$

The output of the second hidden layer is defined in equation (102)

$$\begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix} = \begin{bmatrix} H(N_1^2(k)) \\ H(N_2^2(k)) \\ H(N_3^2(k)) \end{bmatrix} \tag{102}$$

The activation function is a sigmoid function as shown in equation (103)

$$H = \frac{2}{(1+e^{-net})} - 1 \tag{103}$$

The output of the third hidden layer is indicated in equation (104);

$$\begin{bmatrix} O_1^3(k) \\ O_2^3(k) \\ O_3^3(k) \end{bmatrix} = \begin{bmatrix} N_1^3(k) \\ N_2^3(k) \\ N_3^3(k) \end{bmatrix} = \begin{bmatrix} w11 & w12 & w13 \\ w21 & w22 & w23 \\ w31 & w32 & w33 \end{bmatrix} \begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix} \tag{104}$$

Now the three control actions of PID are presented in equations 105–108;

$$P(t) = e_{\theta i}(\mathrm{t}) \text{ or } P(k) = e_{\theta i}(k) \tag{105}$$

$$I(t) = \int e_{\theta i}(t)\, dt \text{ or } I(k) = \sum_{j=0}^{k} h \times e_{\theta i}(j) \tag{106}$$

The filter of derivative is presented in equation (107)

$$f_{\theta i}(s) = \frac{N}{s+N} e_{\theta i}(s) \tag{107}$$

$$D(t) = \frac{d}{dt} f_{\theta i}(t) \text{ or } D(k) = \left(f_{\theta i}(k) - f_{\theta i}(k-1)\right) \Big/ h \tag{108}$$

$$u_1(k) = K_p \left(O_1^3(k) + P(k)\right) \tag{109}$$

$$u_2(k) = K_i \left(O_2^3(k) + I(k)\right) \tag{110}$$

$$u_3(k) = K_d \left(O_3^3(k) + D(k)\right) \tag{111}$$

The equation of control signal is as presented equation (112);

$$\overline{T}_i(k) = u_1\ (\mathrm{k}) + u_2\ (\mathrm{k}) + u_3\ (\mathrm{k}) \tag{112}$$

### 4.3. Elman Neural Network Like PID controller (ELNN-PID)

In this proposed hybrid controller, the proportional, integral and derivative actions are combined with Elman neural network to produce the control signal as shown in Fig. 5.where, $e_{\theta i}(t)$ is the difference between the *i*th link desired and calculated positions $\theta_{ri}(t)$ and $\theta_{ai}(t)$, respectively. $e_{\theta i}(t)$ is the input layer with a single node. The neuron's outputs from the first hidden layer, which has three nodes *P, I,* and *D*, are as indicated in equations 113–115.

$$P(t) = K_p e_{\theta i}(t) \text{ or } P(k) = K_p e_{\theta i}(k) \tag{113}$$

$$I(t) = K_i \int e_{\theta i}(t)\ dt \ \text{ or } \ I(k) = K_i \sum_{j=0}^{k} h \times e_i(j) \tag{114}$$

$$D(t) = K_d\ \frac{d}{dt} e_{\theta i}(t) \text{ or } \ D(k) = K_d\ (e_{\theta i}(k) - e_{\theta i}(k-1)\ ) \Big/ h \tag{115}$$

where;

$$\begin{bmatrix} N_1^1(k) \\ N_2^1(k) \\ N_3^1(k) \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \\ v31 & v32 & v33 \end{bmatrix} \begin{bmatrix} P(k) \\ I(k) \\ D(k) \end{bmatrix} \tag{116}$$

and

$$\begin{bmatrix} C_1^1(k) \\ C_2^1(k) \\ C_3^1(k) \end{bmatrix} = \begin{bmatrix} O_1^1(k-1) + p1 \times C_1^1(k-1) \\ O_2^1(k-1) + p2 \times C_2^1(k-1) \\ O_3^1(k-1) + p3 \times C_3^1(k-1) \end{bmatrix} \tag{117}$$

The output of the second hidden layer is presented in equation (118);

$$\begin{bmatrix} O_1^1(k) \\ O_2^1(k) \\ O_3^1(k) \end{bmatrix} = \begin{bmatrix} H(N_1^1(k)) \\ H(N_2^1(k)) \\ H(N_3^1(k)) \end{bmatrix} + \begin{bmatrix} vc11 & vc12 & vc13 \\ vc21 & vc22 & vc23 \\ vc31 & vc32 & vc33 \end{bmatrix} \begin{bmatrix} C_1^1(k) \\ C_2^1(k) \\ C_3^1(k) \end{bmatrix} \tag{118}$$

The activation function is a sigmoid function as shown in equation (119)

$$H = \frac{2}{(1 + e^{-net})} - 1 \tag{119}$$

$$\begin{bmatrix} C_1^2(k) \\ C_2^2(k) \\ C_3^2(k) \end{bmatrix} = \begin{bmatrix} O_1^2(k-1) + pp1 \times C_1^2(k-1) \\ O_2^2(k-1) + pp2 \times C_2^2(k-1) \\ O_3^2(k-1) + pp3 \times C_3^2(k-1) \end{bmatrix} \tag{120}$$

The output of the third hidden layer is indicated in equation (121);

$$\begin{bmatrix} O_1^2(k) \\ O_2^2(k) \\ O_3^2(k) \end{bmatrix} = \begin{bmatrix} vv11 & vv12 & vv13 \\ vv21 & vv22 & vv23 \\ vv31 & vv32 & vv33 \end{bmatrix} \begin{bmatrix} O_1^1(k \\ O_2^1(k) \\ O_3^1(k) \end{bmatrix} + \begin{bmatrix} vvc11 & vvc12 & vvc13 \\ vvc21 & vvc22 & vvc23 \\ vvc31 & vvc32 & vvc33 \end{bmatrix} \begin{bmatrix} C_1^2(k) \\ C_2^2(k) \\ C_3^2(k) \end{bmatrix} \tag{121}$$

The output layer with a one node as expressed in equation (122).

$$\overline{T_i}(k) = w_1 \times O_1^2(k) + w_2 \times O_2^2(k) + w_3 \times O_3^2(k) \tag{122}$$

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

## 5. Coot optimization algorithm (COOA)

The COOA is a metaheuristic algorithm and inspired by coots' forage for food in the wild. According to COOA, the individuals in

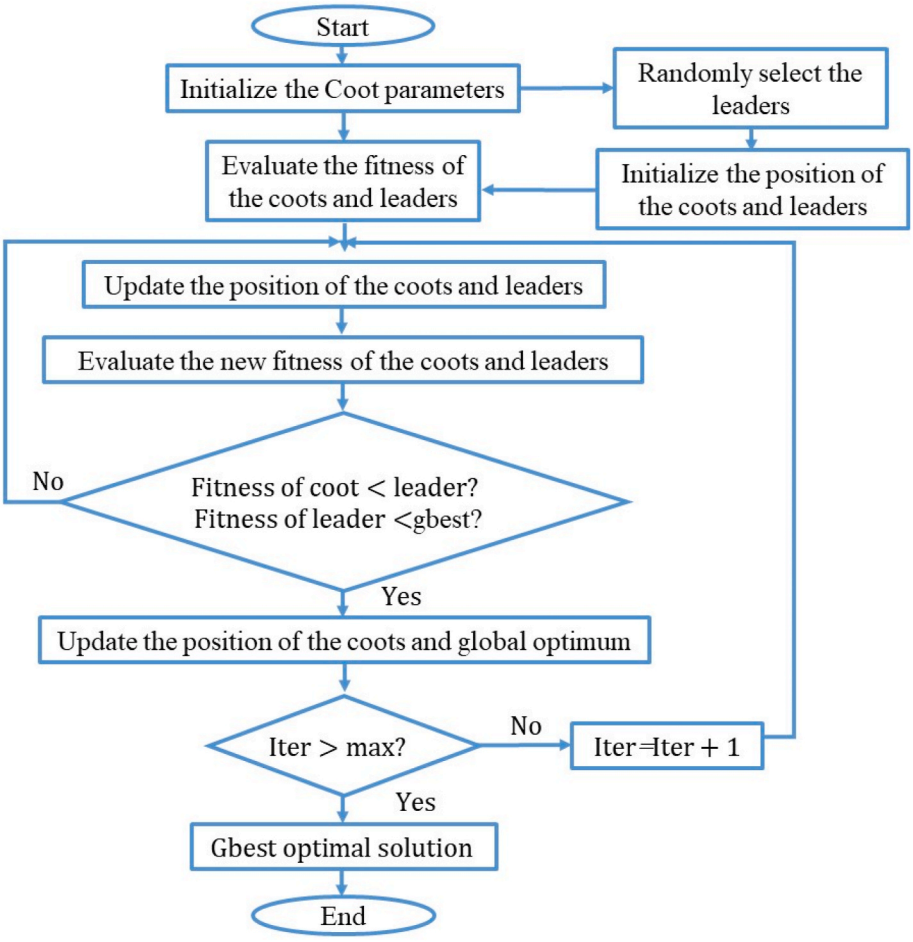


**Fig. 6.** Cooa flowchart.

population are separated into members and leaders. High-quality solutions represent leaders, and low-quality solutions represent members. The following provides a detailed expression of the three primary COOA steps [52,53].

- Initial Solution Set Generation.

In the population each coot k is considered as a solution Ak, as described equation (123):

$$A_k = A^{min} + r_1\left(A^{max} - A^{min}\right); i = 1, \ldots., N_{po,} \tag{123}$$

where $r_1$ represents the random number between 0 and 1, and $A^{max}$ and $A^{min}$ are the upper and lower bounds, respectively, of each existing solution. $N_{PO}$, the population size, is represented in equation (123) as two groups: the good group, which has leaders from $N_L$, and the bad group, which has members from $N_M$.

In the population each member is evaluated by their fitness function, and each member is then assigned to an appropriate group according to the fitness value that was determined. by *Ldm*, where $m = 1, \ldots, N_L$, represents each solution in the leader group, and *Mrj,* where $j = 1, \ldots, N_M$, represents each solution in the member group.

- Member Group Update.

In order to update each solution in the member set one of the following three techniques is used, as indicated in equations 124–126 [54]:

$$Mr_j^{new} = Mr_j + FG_1 \cdot r_2 \cdot \left(A_{r\,d} - Mr_j\right), \tag{124}$$

$$Mr_j^{new} = 0.5 \times \left(\left(Mr_{j-1} + Mr_j\right)\right), \tag{125}$$

$$Mr_j^{new} = Ld_{rd} + 2 \cdot r_3 \cdot cos(2 \cdot \pi \cdot r') \cdot \left(Ld_{rd} - Mr_j\right), \tag{126}$$

where r′ represents the random number between −1 and 1, and $r_2$ and $r_3$ are random numbers between 0 and 1. $Ld_{rd}$ is the leader group's randomly selected solution. The random solution $A_{r\ d}$ and iteration function $FG_1$ expressed by equations 127 and 128:

$$A_{rd} = A^{min} + r_4\left(A^{max} - A^{min}\right) \tag{127}$$

$$FG_1 = 1 - G \times \left(\frac{1}{G^{max}}\right), \tag{128}$$

where the random number $r_4$ falls between 0 and 1. The current iteration and the maximum iteration number are denoted by $G$ and $G^{max}$, respectively.

- The Leader Group update

COOA uses local search, or circling the population's best solution, $Ld_{best}$, to update solutions in the leader group. For the group, two distinct approaches are employed as expressed in equations 129 and 130 [55,56]:

$$Ld_m^{new} = Ld_{best} + [FG_2 \cdot r_5 \cdot cos(2.\pi \cdot r').(Ld_{best} - Ld_m)] \ if \ r_6 < 0.5 \tag{129}$$

$$Ld_m^{new} = -Ld_{best} + [FG_2 \cdot r_7 \cdot cos(2.\pi \cdot r').(Ld_{best} - Ld_m)] \ if \ r_6 \geq 0.5 \tag{130}$$

where the random numbers $r_5$, $r_6$, and $r_7$ range from 1 to 0. $FG_2$ is an iteration function that can be stated as indicated in equation (131):

$$FG_2 = 2 - \left(\frac{1}{G^{max}}\right)G \tag{131}$$

In conclusion, Fig. 6 summarizes how COOA can be applied to a generic problem [57].
The steps to implement the proposed method are explained as follows.

1 Choose the control structure you want to applied, also count the number of parameters required to tune in this controller and the range space for each parameter.
2 Choose the optimization algorithm utilized to adjust the required parameters of the controller and choose the population size, number of iterations, and other parameters required for setting the optimization algorithm.
3 determine the objective function as well as the necessary constraint imposed on the system so that to formulate a correct fitness function for the optimization algorithm.
4 Choose the population of random solutions using the determined search space in step 1.
5 Evaluate each solution in population and determine its fitness value
6 Update the present population to produce new population and increase iterations by one.

## 6. Simulation results

This section addresses the performance of the suggested controllers for 3-LRRM trajectory tracking response and robustness. MATLAB is utilized to simulate the suggested controllers design regarding the 3-LRRM tracking issue and the test path. The simulation step size is 1 msec, and the time of the simulation is assumed to be 10 s. For every link, the torque constraints were set to [−200,200] N-m. In addition, the fraction operator design makes utilizing Oustaloup's approximation of the eleventh order (N = 5), which has a frequency range of [0.001, 1000] rad/s. The manipulator model follows a test trajectory tracking of every link after it has been analyzed, and the ITSE objective function has been taken into consideration. This helps to ensure that errors are minimized and that the tracking of the desired path is completed more quickly. The COOA was used to adjust the suggested controllers' parameters based on the tracking error for the 3-LRRM between the actual and reference trajectory. when two initial conditions of the positions (−0.15, −0.85, −1.15) and (0.15,-0.55,-0.85) rad for theta1, theta2, and theta3, are used in training process. The following is the COOA setting: The maximum number of iterations equals 500, and the population size is 100. The performance of suggested controllers is assessed using the ITSE computation. The controller considered most optimal is the one with the lowest ITSE. The following formula can be used to determine the ITSE, as demonstrated in equation (132):

$$\min J = \int t \times e(t)^2 \, dt \tag{132}$$

An essential strength of neural networks is their ability to bend themselves in order to capture complex underlying data structures. In the design of neural network controllers, this ability leads to the production of the most complex control signals of high frequency (i.e. chattering phenomenon). In fact, a chattering signal cannot be applied practically. Therefore, in order to overcome this problem, the new objective function is modified as indicated equation (133);

**Table 2**
Total number of the suggested controllers' design parameters along with their range of search.

| Controller | Total Number of Controller Parameters | PID Gains' Range $K_p$, $K_i$, $K_d$ | All other Parameters Range |
|---|---|---|---|
| NN-PIPD | 120 | −150 to 150 | −1 to 1 |
| NN + PID | 111 | −150 to 150<br>$N$ from 10 to 100 | −1 to 1 |
| ELNN-PID | 144 | −150 to 150 | −1 to 1 |

**Table 3**
The ITSE and the sign change of all control signals for the proposed Controllers when Nominal Plant is used with two initial positions (−0.15, −0.85,-1.15) and (0.15,-0.55,-0.85).

| Controller Type | ITSE | **Number of slope sign changes for all Control Signals of the 3-links** |
|---|---|---|
| NN-PIPD | 2.09072 × 10-5 | 187 |
| NN + PID | 4.69431 × 10-5 | 57 |
| ELNN-PID | 8.23779 × 10-5 | 71 |

**Table 4**
Details Features of Nominal System Trajectory with initial position (x1 = 0.15, x2 = −0.55, x3 = −0.85).

| Type of Controller | Link No | Rise Time | Settling Time | Over Shoot % | ITSE × 10-5 |
|---|---|---|---|---|---|
| NN-PIPD | L1 | 0.070 | 0.118 | 0.223 | 0.37839 |
| | L2 | 0.029 | 0.088 | 15.17 | 0.29108 |
| | L3 | 0.029 | 0.101 | 16.30 | 0.30737 |
| NN + PID | L1 | 0.042 | 0.858 | 32.58 | 1.66783 |
| | L2 | 0.028 | 0.730 | 40.29 | 1.26461 |
| | L3 | 0.025 | 0.268 | 17.85 | 0.21392 |
| ELNN-PID | L1 | 0.047 | 0.101 | 14.92 | 0.76218 |
| | L2 | 0.190 | 0.190 | 0.031 | 1.08184 |
| | L3 | 0.270 | 0.270 | 0.027 | 1.89864 |

$$\min J = \int t \times e(t)^2 \, dt + Co \times \rho \tag{133}$$

where $Co$ denotes number of slope sign alters of the control responses, and ρ indicates a small value, selected to be $10^{-8}$.

Table 2 displays the adjusting gains number and search space range that COOA uses to identify the best parameters set. Table 3 lists all of the proposed controllers' tuned gains, and Table 4 lists each one's rising time, maximum overshoot, settling time and ITSE. The performance of each proposed controller is shown in Fig. 7 along with the end-effector x-y plots, controller outputs, and trajectory tracking curves. Fig. 7(a–c) demonstrates the performance of the reference and actual values for theta1, theta2, and theta3. While Fig. 7(d–f) illustrates the output of the controller (torque1), torque2, and torque3, and Fig. 7 (g) displays the reference and actual trajectories.

According to the results and the comparison among the proposed controllers, the NN + PIPD controller produced smoother response with faster convergence to the desired path and had lowest values of ITSE. With the lowest ITSE value and the smallest settling time, the NN + PIPD controller superior to other suggested controllers; in contrast, the ELNN-PID controller performs the worst.

### *6.1. Robustness tests*

To demonstrate the effectiveness and robustness of each controller, the experiments that without adjusting the gains of the controllers will be perform.

#### *6.1.1. Initial condition test*

By altering the starting positions of theta1, theta2, and theta3 to (0.2, −0.5, and −0.8) rad for the test of trajectory following, robustness of the suggested controllers is examined. The associated ITSE are provided in Table 5. Fig. 8 indicates the performance of the reference and actual values. Fig. 8(a–c) displays the following of trajectory for theta1, theta2, and theta3, respectively, as well as the trajectory followed by the 3-LRRM end-effector with each controller's initial position changed. Whilst Fig. 8 (d) demonstrates trajectories-based beginning position (0.2, −0.5, −0.8) rad.

It is observed that for varying starting positions of theta1, theta2, and theta3, respectively, the NN-PIPD controller performs better than all other suggested controllers and has the lowest ITSE value. In addition, the proposed NN-PIPD controller was able to track the desired trajectory with the minimum settling time.

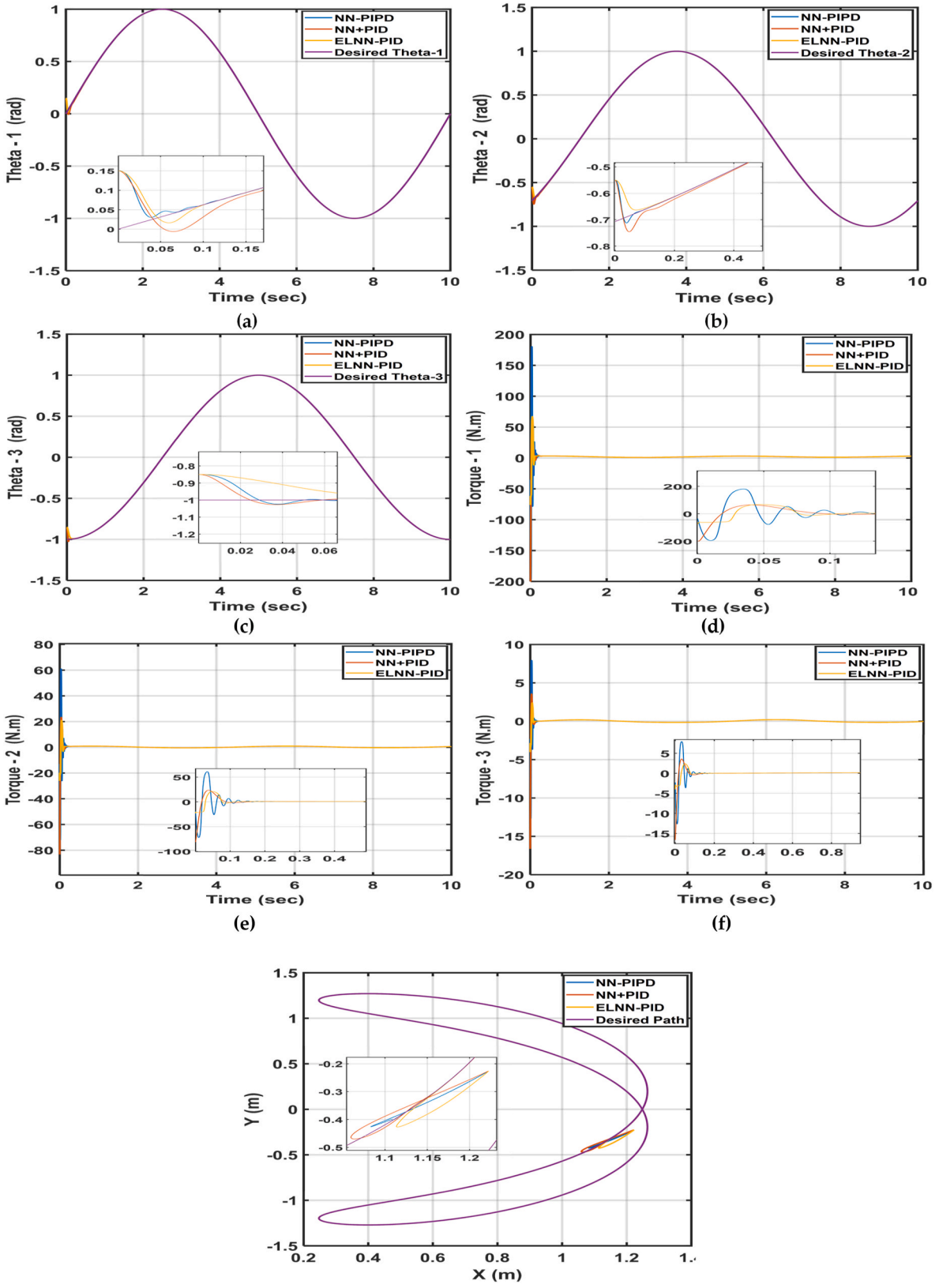


**Fig. 7.** The performance of the reference and actual values (a) Theta1, (b) Theta2, (c) Theta3, and the output of the controller (d) (torque1), (e) torque2, (f) torque3, and (g) reference and actual trajectories.

**Table 5**
The suggested controllers' ITSEs with their starting initial positions (0.2, −0.5, −0.8).

| Controller | ITSE |
|---|---|
| NN-PIPD | 2.3113 × 10-5 |
| NN + PID | 76.6074 × 10-5 |
| ELNN-PID | 8.38089 × 10-5 |

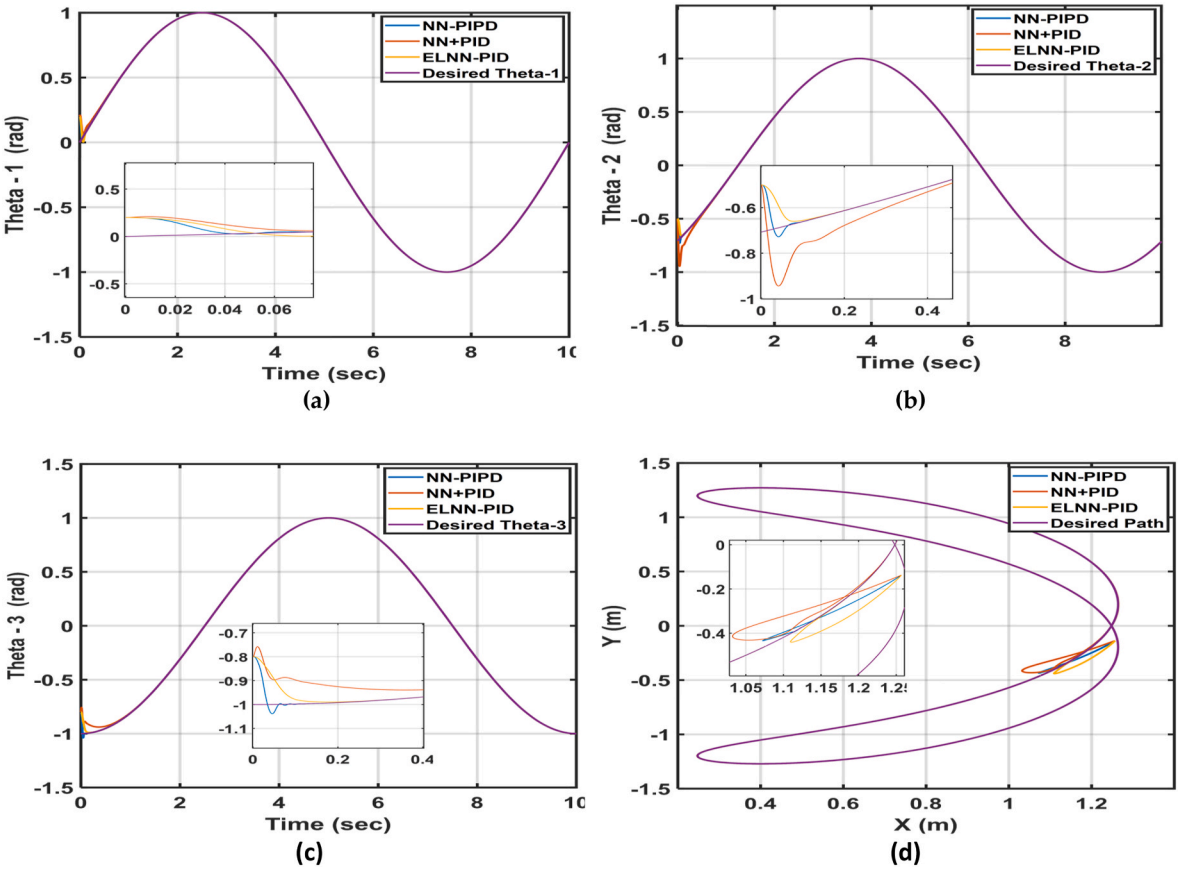


**Fig. 8.** The performance of the reference and actual values (a) Theta1, (b) Theta2, (c) Theta3, and (d) trajectories based beginning position (0.2, −0.5,-0.8) rad.

**Table 6**
The ITSE performance of the suggested controllers when adding disturbances sin (100t) to all control signals in period 2–6 s and initial condition is (0, −0.7, −1).

| Controller | ITSE |
|---|---|
| NN-PIPD | 0.086695 |
| NN + PID | 0.059097 |
| ELNN-PID | 0.464902 |

### *6.1.2. Disturbance adding test*

By involving the disturbance value equals to [sin (100t)] N-m in for (2 s ≤ $t$ ≤ 6 s) to the controller signal into 3-links together and setting the initial position (0,-0.7,-1) rad without altering the parameters of the proposed controllers, the disturbance rejection for each proposed controller has also been investigated. Table 6 displays the result that was obtained. Fig. 9 displays the performance of the reference and actual values. Fig. 9(a–b) indicates the path following of theta1, theta2, and theta3, respectively, as well as the trajectory

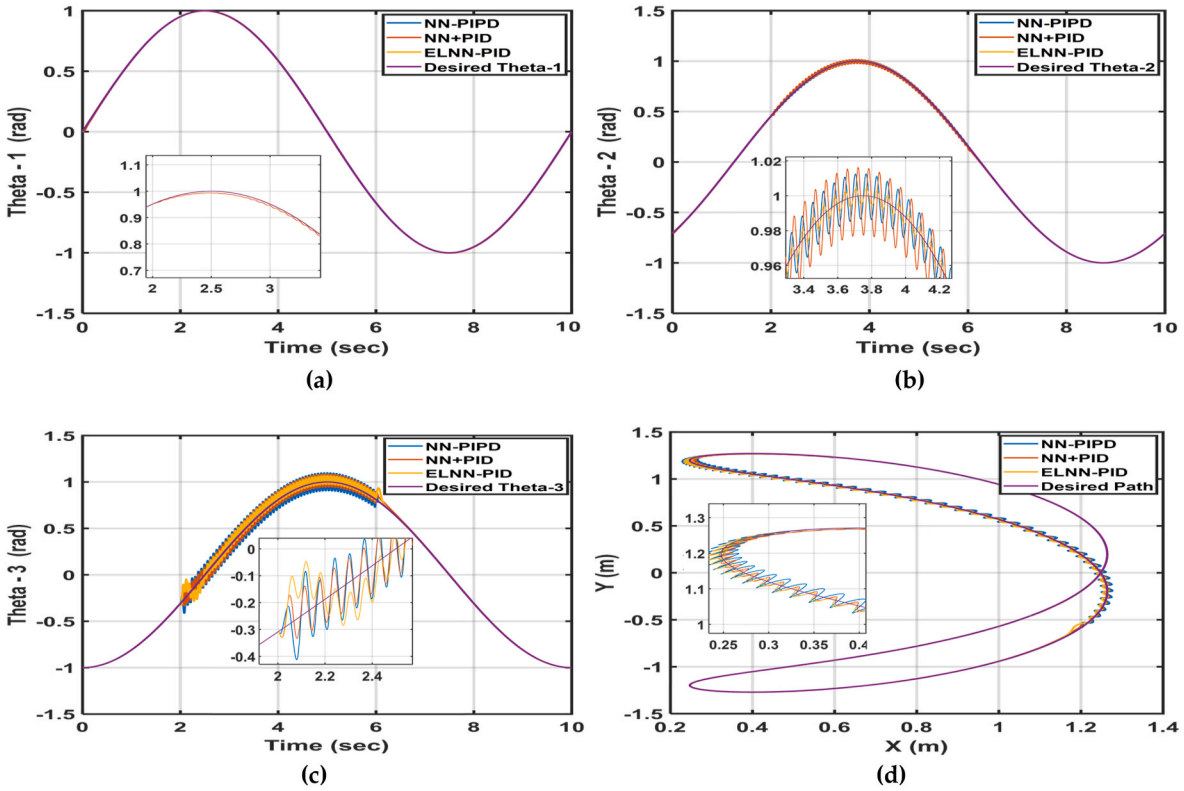


**Fig. 9.** The performance of the reference and actual values (a) Theta1, (b) Theta2, (c) Theta3, and (d) trajectories using disturbance [sin (100T)] N-M to all links using starting position (0, −0.7,-1) rad.

**Table 7**
Initial position (0.0, −0.7, −1) and the ITSE of the suggested controllers for a 10 % increase in the third link's mass.

| Controller | ITSE |
|---|---|
| NN-PIPD | 0.06574 × 10-5 |
| NN + PID | 323.861 × 10-5 |
| ELNN-PID | 1.95897 × 10-5 |

followed by 3-LRRM end-effector utilizing disturbance of [sin (100t)] N-m in all links. Meanwhile Fig. 9(d) demonstrates the trajectories using disturbance [sin (100T)] N-M to all links using starting position (0, −0.7,-1) rad.

Comparative results explicitly demonstrate that NN + PID is the optimal controller and that it performs better for disturbance rejection than the other suggested controllers. While the proposed NN + PIPD controller represents the second one in terms of performance and there is no significant difference between it and NN + PID. But ELNN + PID controller was the worst in terms of performance.

### *6.1.3. Parameter change test*

Picking and placing objects of varying masses for a variety of applications is the main purpose of the robot manipulator with its end-effector. The values variation of the suggested controllers is investigated by adding the masses of link3 by 10 % and making sure the controller values stay the same. Table 7 presents the obtained ITSE, and Fig. 10 the performance of the reference and actual values. Fig. 10(a–c) displays the trajectory following of theta1, theta2, and theta3, receptively for each controller based on mass changes. While, Fig. 10(d) represents trajectories based 10 % adding in mass of link3 with beginning position (0, −0.7, −1) rad.

The findings indicate that the suggested controllers for parameter variation have a higher ITSE than the NN-PIPD controllers. When compared to other proposed controllers, the NN-PIPD controller performs the best due to its lowest ITSE value, quick trajectory tracking, and improved theta1, theta2, and theta3 response. NN + PID is the worst controller because its ITSE is the highest.

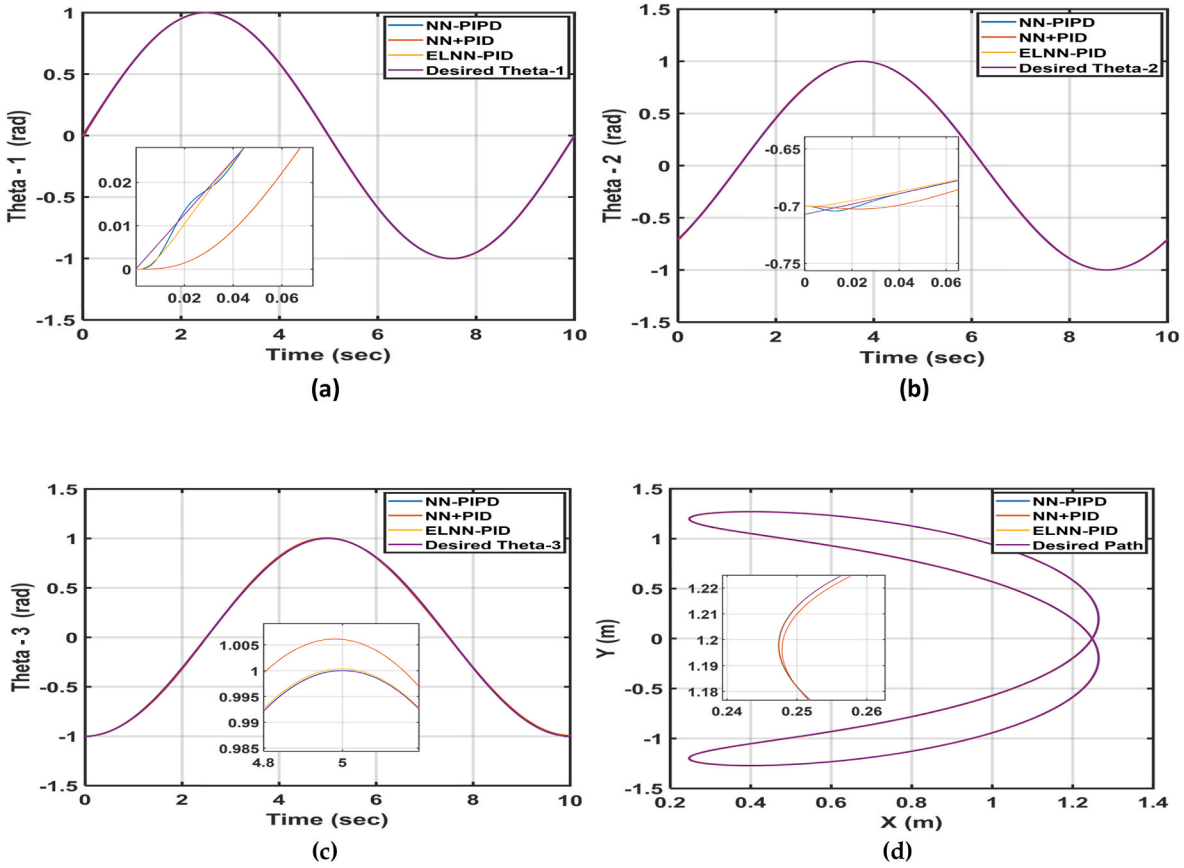


**Fig. 10.** The performance of the reference and actual values (a) Theta1, (b) Theta2, (c) Theta3, and (d) Trajectories based 10 % adding in mass of link3 with beginning position (0, −0.7, −1) rad.

**Table 8**
**The** reference and actual trajectories using starting position (0.2, −0.5, −0.8) with disturbance [sin (100t), sin (100t), sin (100t)] and 10 % increasing in mass of the third link.

| Controller | ITSE |
|---|---|
| NN-PIPD | 0.001777 |
| NN + PID | 0.057480 |
| ELNN-PID | 0.034756 |

### *6.1.4. All test case*

The effects of varying the parameters, adding disturbance, and adjusting the initial positions are combined to demonstrate the effectiveness of the suggested controllers. Table 8 lists the obtained ITSE. The trajectory following of theta1, theta2, and theta3 by the 3-LRRM's end-effector are displayed in Fig. 11(a–c), respectively. Meanwhile, Fig. 11(d–f) indicates the output of the controller (torque1), (torque2), (torque3), respectively and Fig. 11 (g) represents the reference and actual trajectories using the beginning position (0.2, −0.5, −0.8) rad with disturbance [sin (100T)] N-M for all links as well as 10 % adding in mass of link3.

Despite initial position changes, disturbance additions, and parameter variations, the NN-PIPD controller's ITSE remains the lowest of all the suggested controllers. In contrast, the ELNN-PID controller represents the worst controller with largest ITSE and the maximum settling time in its response as can be seen in the responses from theta1, theta2, and theta3. As a result of the comparison analysis, the optimal controller was considered to be an NN-PIPD controller for trajectory following, disturbance rejection, and parameter variation with a minimum ITSE of 0.001777. The results proved that despite the changes in parameters and condition, the proposed NN-PIPD controller is sufficient for the robot's requirements in terms of accuracy and response.

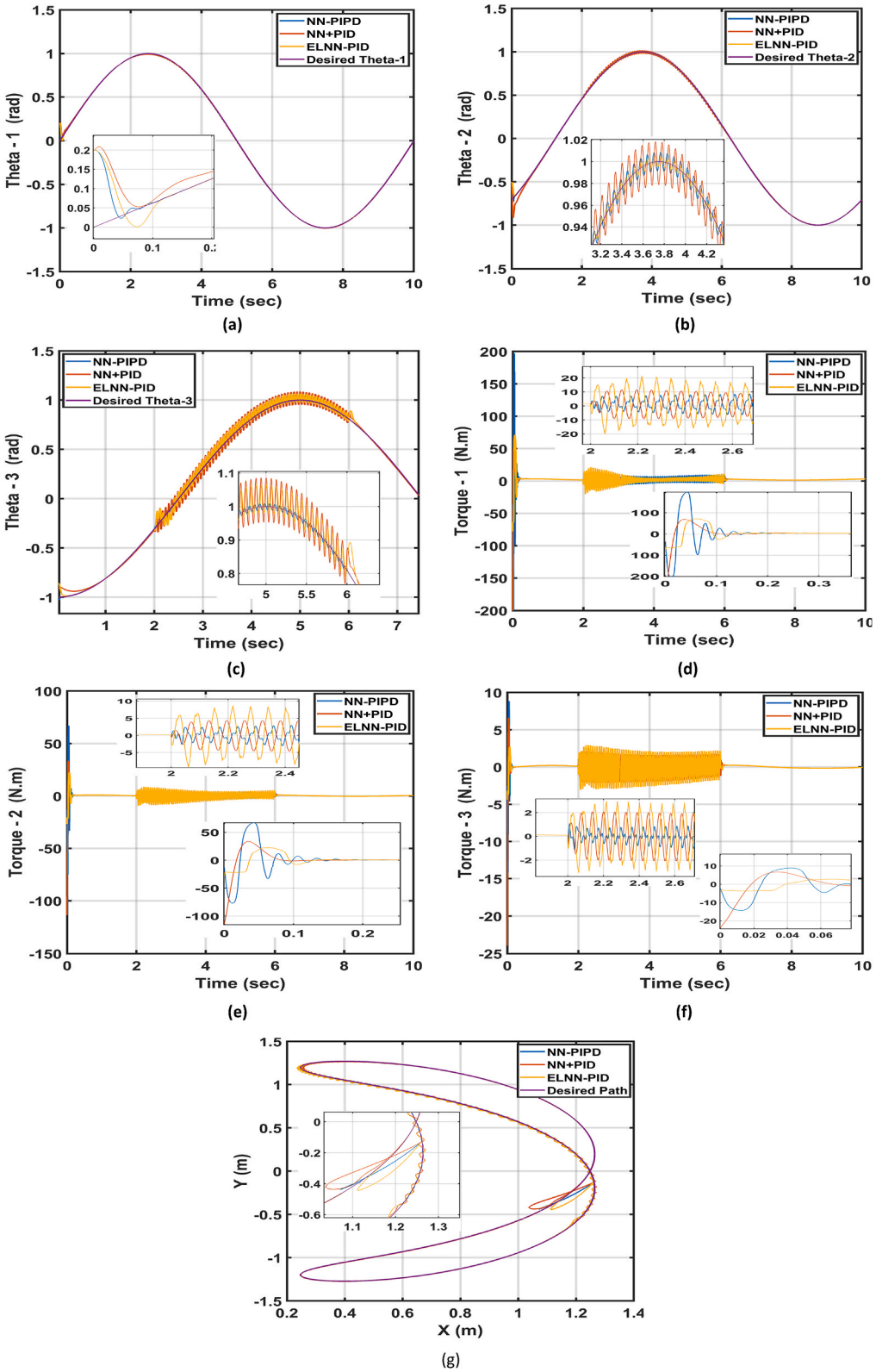


**Fig. 11.** The performance of the reference and actual values (a) Theta1, (b) Theta2, (c) Theta3, and the output of the controller (d) torque1, (e) torque2, (f) torque3, and (g) reference and actual trajectories using the beginning position (0.2, −0.5, −0.8) rad with disturbance [sin (100T)] N-M for all links as well as 10 % adding in mass of link3.

## 7. Conclusions

The main goal of this study is to tackle the trajectory tracking issue of the 3-LRRMbased on designing three control structures using a combination Neural Network (NN) with Proportional, Integral and Derivative (PID) actions named NN-PIPD controller, NN + PID controller NN + PID controller and ELNN-PID controller. The parameters of the proposed controllers are adjusted utilizing the COOA in order to reduce the ITSE. A new objective function for tuning process to produce a controller with minimum value of the chattering in the control signal is proposed. The performance of the proposed controllers is evaluated in terms of disturbance rejection, fluctuating initial conditions, reference trajectory tracking, and model uncertainty. Simulation results show that the proposed NN-PIPD controller performed better than other proposed controllers in terms of tracking performance, stability, and robustness. As a result of the comparison analysis, the optimal controller was considered to be an NN-PIPD controller for trajectory tracking, disturbance rejection, and parameter variation with a minimum ITSE of 0.001777. In future work, the same idea can also be implemented using other hybrid control structures and optimization strategies. The proposed controllers can also be implemented practically utilizing a real robotic manipulator equipped with all necessary hardware components and sensors. The analysis and discussion in this work can be extended to another robot manipulator, such as the SCARA or PUMA 560 robot.

## Data availability

The data that support the findings of this study are available from the corresponding author.

## CRediT authorship contribution statement

**Mohamed Jasim Mohamed:** Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Formal analysis, Conceptualization. **Bashra Kadhim Oleiwi:** Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Formal analysis, Conceptualization. **Ahmad Taher Azar:** Writing – review & editing, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Ibrahim A. Hameed:** Writing – review & editing, Visualization, Validation, Resources, Methodology, Funding acquisition, Formal analysis.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was funded by the Norwegian University of Science and Technology.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors would like to acknowledge the support of the Norwegian University of Science and Technology for paying the Article Processing Charges (APC) of this publication. The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for their support. This research is supported by Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. In addition, the authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.

## References

[1] Y.J. Huang, Variable structure control for a two-link robot arm, Electr. Eng. 85 (4) (2003) 195–204.
[2] V.T. Yen, W.Y. Nan, P.V. Cuong, Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators, Neural Comput. Appl. 31 (11) (2018) 6945–6958.
[3] M.R. Hassan, S.A. Al-Samarraie, Optimal feedback control for HVAC systems: an integral sliding mode control approach based on barrier function, Mathematical Modelling of Engineering Problems 10 (3) (2023) 1053–1062, https://doi.org/10.18280/mmep.100341.
[4] A.A. Kareem, B.K. Oleiwi, M.J. Mohamed, Planning the optimal 3D quadcopter trajectory using a delivery system-based hybrid algorithm, International Journal of Intelligent Engineering and Systems 16 (2) (2023) 427–439, https://doi.org/10.22266/ijies2023.0430.34.
[5] M. Bhave, L. Dewan, S. Janardhanan, A novel third order sliding mode controller for the orientation and position of planar three link rigid robotic manipulator, in: International Conference on Instrumentation Control and Automation, 2013.
[6] M.J. Mohamed, B.K. Oleiwi, L.H. Abood, A.T. Azar, I.A. Hameed, Neural fractional order PID controllers design for 2-link rigid robot manipulator, Fractal and Fractional 7 (9) (2023) 693, https://doi.org/10.3390/fractalfract7090693, pp. 2–24.
[7] K. Jitendra, K. Vineet, K. Rana, Design of robust fractional order fuzzy sliding mode PID controller for two-link robotic manipulator system, J. Intell. Fuzzy Syst. 35 (2018) 1–15, https://doi.org/10.3233/JIFS-169813.
[8] H.I. Abdulameer, M.J. Mohamed, Fractional order fuzzy PID controller design for 2-link rigid robot manipulator, International Journal of Intelligent Engineering and Systems 15 (3) (2022) 103–117, https://doi.org/10.22266/ijies2022.0630.10.
[9] H. Tlijani, A. Jouila, K. Nouri, Wavelet neural network sliding mode control of two rigid joint robot manipulator, Adv. Mech. Eng. 14 (8) (2022), https://doi.org/10.1177/16878132221119886.

[10] M. Vijay, et al., An expert 2DOF fractional order fuzzy PID controller for nonlinear systems, Neural Comput. Appl. 31 (5) (2019) 4253–4270, https://doi.org/10.1007/s00521-017-3330-z.

[11] F. Jamshidi, M. Vaghefi, WOA-based interval type II fuzzy fractional-order controller design for a two-link robot arm, J. Electr. Comput. Eng. Innovat. 7 (1) (2019) 69–82, https://doi.org/10.22061/JECEI.2019.5783.256.69.

[12] M. Vijay, et al., Robust self-tuning fractional order PID controller dedicated to non-linear dynamic system, J. Intell. Fuzzy Syst. 34 (3) (2018) 1467–1478, https://doi.org/10.3233/JIFS-169442.

[13] R. Fareh, M. Bettayeb, M. Rahman, Control of serial link manipulator using a fractional order controller, International Review of Automatic Control (IREACO) 11 (1) (2018) 29–35, https://doi.org/10.15866/ireaco.v11i1.13275.

[14] R. Sharma, K.P.S. Rana, V. Kumar, Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator, Expert Syst. Appl. 41 (9) (2014) 4274–4289, https://doi.org/10.1016/j.eswa.2013.12.030.

[15] Mohammad A. Faraj, Abdulsalam Mohammed Abbood, Fractional order PID controller tuned by bat algorithm for robot trajectory control, Indonesian Journal of Electrical Engineering and Computer Science 21 (1) (2021) 74–83, https://doi.org/10.11591/ijeecs.v21.i1.pp74-83. ISSN: 2502-4752.

[16] A.M. Abdul-Alsada, K.M.H. Raheem, M.M.S. Altufaili, A fuzzy logic controller for a three links robotic manipulator, AIP Conference Proceedings, AIP Publishing LLC 2386 (1) (2022), https://doi.org/10.1063/5.0066871.

[17] S. Xu, C. Zhang, A. Mohammadzadeh, Type-3 fuzzy control of robotic manipulators, Symmetry 15 (2) (2023) 483, https://doi.org/10.3390/sym15020483, pp. 2–18.

[18] J. Kern, D. Marrero, C. Urrea, Fuzzy control strategies development for a 3-DoF robotic manipulator in trajectory tracking, Processes 11 (2023) 3267, https://doi.org/10.3390/pr11123267.

[19] J. Kumar, V. Kumar, K.P.S. Rana, Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system, Neural Comput. Appl. 32 (11) (2020) 7235–7257, https://doi.org/10.1007/s00521-019-04215-8.

[20] H. Cao, Design of a fuzzy fractional order adaptive impedance controller with integer order approximation for stable robotic contact force tracking in uncertain environment, Acta Mech. Automatica 16 (1) (2022) 16, https://doi.org/10.2478/ama-2022-0003.

[21] U. Kabir, M. Hamza, G.S. Shehu, Performance analysis of PID, PD and fuzzy controllers for position control of 3-Dof robot manipulator, Zaria Journal of Electrical Engineering Technology, Department of Electrical Engineering, Ahmadu Bello University 8 (1) (2019) 18–25. ArXiv, abs/1910.12076.

[22] V. Kumar, K.P.S. Rana, D. Kler, Efficient control of a 3-link planar rigid manipulator using self-regulated fractional-order fuzzy PID controller, Applied Soft Computing Journal 82 (1) (2019) 105531, https://doi.org/10.1016/j.asoc.2019.105531.

[23] Y. Deng, Fractional-order fuzzy adaptive controller design for uncertain robotic manipulators, Int. J. Adv. Rob. Syst. 16 (2) (2019), https://doi.org/10.1177/1729881419840223.

[24] M.I. Azeez, A.M.M. Abdelhaleem, S. Elnaggar, et al., Optimization of PID trajectory tracking controller for a 3-DOF robotic manipulator using enhanced artificial bee colony algorithm, Sci. Rep. 13 (11164) (2023), https://doi.org/10.1038/s41598-023-37895-3.

[25] Y. Oktarina, F. Septiarini, T. Dewi, P. Risma, M. Nawawi, Fuzzy-PID controller design of 4 dOF industrial arm robot manipulator, Comput. Eng. Appl. J. 8 (2) (2019) 123–136, https://doi.org/10.18495/comengapp.v8i2.300.

[26] M.K. Jangid, S. Kumar, J. Singh, L. Abualigah, Simulation and parametric optimization of controller for three-link robotic Manipulator in casting, Research Square (2023), https://doi.org/10.21203/rs.3.rs-2564440/v1.

[27] H.R. Patel, V.A. Shah, Shadowed type-2 fuzzy sets in dynamic parameter adaption in cuckoo search and flower pollination algorithms for optimal design of fuzzy fault-tolerant controllers, Math. Comput. Appl. 27 (6) (2022) 2–32, https://doi.org/10.3390/mca27060089, 89.

[28] H.R. Patel, V.A. Shah, A metaheuristic approach for interval type-2 fuzzy fractional order fault-tolerant controller for a class of uncertain nonlinear system, Automatika 63 (4) (2022) 656–675, https://doi.org/10.1080/00051144.2022.2061818.

[29] H.R. Patel, V.A. Shah, Type-2 fuzzy logic applications designed for active parameter adaptation in metaheuristic algorithm for fuzzy fault-tolerant controller, International Journal of Intelligent Computing and Cybernetics 16 (2) (2023) 198–222, https://doi.org/10.1108/IJICC-01-2022-0011.

[30] H.R. Patel, Fuzzy-based metaheuristic algorithm for optimization of fuzzy controller: fault-tolerant control application, International Journal of Intelligent Computing and Cybernetics 15 (4) (2022) 599–624, https://doi.org/10.1108/IJICC-09-2021-0204.

[31] H.R. Patel, V.A. Shah, Stable fuzzy controllers via LMI approach for non-linear systems described by type-2 T–S fuzzy mode, International Journal of Intelligent Computing and Cybernetics 14 (3) (2021) 509–531, https://doi.org/10.1108/IJICC-02-2021-0024 (23).

[32] H.R. Patel, V.A. Shah, Application of metaheuristic algorithms in interval type-2 fractional order fuzzy TID controller for nonlinear level control process under actuator and system component faults, International Journal of Intelligent Computing and Cybernetics 14 (1) (2021) 33–53, https://doi.org/10.1108/IJICC-08-2020-0104.

[33] H. R. Patel, S. K. Raval, V. A. Shah, A novel design of optimal intelligent fuzzy TID controller employing GA for nonlinear level control problem subject to actuator and system component fault, International Journal of Intelligent Computing and Cybernetics, 14 (1). DOI: https://doi.org/10.1108/IJICC-11-2020-0174.

[34] H. Patel, V. Shah, V, An optimized intelligent fuzzy fractional order TID controller for uncertain level control process with actuator and system component uncertainty, in: Book: Fuzzy Information Processing (2020), 2022, pp. 183–195, https://doi.org/10.1007/978-3-030-81561-5_16.

[35] H.R. Patel, V.A. Shah, Comparative analysis between two fuzzy variants of harmonic search algorithm: fuzzy fault tolerant control application, IFAC-PapersOnLine 55 (7) (2022) 507–512, https://doi.org/10.1016/j.ifacol.2022.07.494.

[36] H.R. Patel, V.A. Shah, General type-2 fuzzy logic systems using shadowed sets: a new paradigm towards fault-tolerant control, in: Australian & New Zealand Control Conference (ANZCC), Gold Coast, Australia, 2021, pp. 116–121, https://doi.org/10.1109/ANZCC53563.2021.9628361.

[37] H.R. Patel, A.V.A. Shah, Fuzzy logic-based metaheuristic algorithm for optimization of type-1 fuzzy controller: fault-tolerant control for nonlinear system with actuator Fault, IFAC-PapersOnLine 55 (1) (2022) 715–721, https://doi.org/10.1016/j.ifacol.2022.04.117. ISSN 2405-8963.

[38] R. Sejal, H. Patel, S. Vipul, R. Umesh, K. Paresh, Fault-tolerant Control Using Optimized Neurons in Feed-Forward Backpropagation Neural Network for MIMO Uncertain System: A Metaheuristic Approach, 2023, pp. 597–609, https://doi.org/10.1007/978-3-031-39774-5_66.

[39] H.R. Patel, V.A. Shah, Decentralized stable and robust fault-tolerant PI plus fuzzy control of MIMO systems: a quadruple tank case study, Int. J. Smart Sens. Intell. Syst. 12 (1) (2019) 1–20, https://doi.org/10.21307/ijssis-2019-004.

[40] H.R. Patel, Metaheuristic optimization algorithm for optimal design of type-2 fuzzy controller, Int. J. Appl. Evol. Comput. 13 (1) (2022) 1–15, https://doi.org/10.4018/IJAEC.315637.

[41] H. Patel, Optimal intelligent fuzzy TID controller for an uncertain level process with actuator and system faults: population-based metaheuristic approach, Franklin Open 4 (2023) 100038, https://doi.org/10.1016/j.fraope.2023.100038.

[42] H.R. Patel, V.A. Shah, Simulation and comparison between fuzzy harmonic search and differential evolution algorithm: type-2 fuzzy approach, IFAC-PapersOnLine 55 (66) (2022) 412–417, https://doi.org/10.1016/j.ifacol.2022.09.059. ISSN 2405-8963.

[43] H. He, P. Shi, Y. Zhao, Adaptive connected hierarchical optimization algorithm for minimum energy spacecraft attitude maneuver path planning, Astrodyn 7 (2023) 197–209, https://doi.org/10.1007/s42064-022-0149-x.

[44] Q. Jiaming, R. Guangtao, B. Wang, L. Jian, M. Wanyu, Z. Peng, N.-A. David, Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls, IEEE Rob. Autom. Lett. (2023) 1–8, https://doi.org/10.1109/LRA.2023.3346758.

[45] G. Ran, Z. Shu, H. Lam, J. Liu, C. Li, Dissipative tracking control of nonlinear markov jump systems with incomplete transition probabilities: a multiple-event-triggered approach, IEEE Trans. Fuzzy Syst. 31 (7) (2023) 2389–2400, https://doi.org/10.1109/TFUZZ.2022.3225672.

[46] G. Ran, Z. Shu, H.-K. Lam, J. Liu, C. Li, Dissipative tracking control of nonlinear Markov jump systems with incomplete transition probabilities: a multiple-event-triggered approach, IEEE Trans. Fuzzy Syst. 31 (7) (2023) 2389–2400, https://doi.org/10.1109/TFUZZ.2022.3225672.

[47] Sanjay Yadav, Dipendra Sah, Nikita Sharma, Prajjwol Uprety, Shacheendra Kishor Labh, Design and control of thre- link robotic manipulator, in: Proceedings Conference: Recent Trends in Science, Technology and Innovation (RTSTI), 2023.

[48] F.L. Lewis, D.M. Dawson, C.T. Abdallah, Robot Manipulator Control Theory and Practice, CRC Press, 2004, pp. 1–632.

[49] S. Kucuk, Z. Bingul, Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: Theory, Modelling, and Control, 2006.
[50] F. Piltan, M. Yarmahmoudi, M. Mirzaie, S. Emamzadeh, Z. Hivand, Design novel fuzzy robust feedback linearization control with application to robot manipulator, Int. J. Intell. Syst. Appl. 5 (5) (2013) 1–10, https://doi.org/10.5815/ijisa.2013.05.01.
[51] J. Kern, D. Marrero, C. Urrea, Fuzzy control strategies development for a 3-DoF robotic Manipulator in trajectory tracking, Processes 11 (12) (2023) 3267, https://doi.org/10.3390/pr11123267, pp. 1–30.
[52] L.C. Kien, T.T.B. Nga, T.M. Phan, T.T. Nguyen, Coot optimization algorithm for optimal placement of photovoltaic generators in distribution systems considering variation of load and solar radiation, Hindawi Mathematical Problems in Engineering 2022 (4) (2022) 1–17, https://doi.org/10.1155/2022/2206570.
[53] H.K. Rushdi, F.M. Al-Naima, Coot optimization algorithm for parameter estimation of photovoltaic model, MEST Journal 10 (2) (2022) 177–185, https://doi.org/10.12709/mest.10.10.02.16.
[54] M.S. Abed, O.F. Lutfy, Q.F. Al-Doori, Online optimization application on path planning in unknown environments, J. Eur. Systèmes Automatisés 55 (1) (2022) 61–69, https://doi.org/10.18280/jesa.550106.
[55] A. Kharidege, D. Jianbiao, Y. Zhang, Performance study of PID and fuzzy Controllers for position control of 6 DOF arm Manipulator with various defuzzification strategies, Engineering, Computer Science 11 (2016) 3–8. https://api.semanticscholar.org/CorpusID:114932759.
[56] M.S. Abed, O.F. Lutfy, Q.F. Al-Doori, Online path planning of mobile robots based on African vulture's optimization algorithm in unknown environments, J. Eur. Systèmes Automatisés 55 (3) (2022) 405–412, https://doi.org/10.18280/jesa.550313.
[57] D. Baolin, Z. Puchen, C. Shouyan, Constant force PID control for robotic manipulator based on fuzzy neural network algorithm, Complexity 2020 (2020), https://doi.org/10.1155/2020/3491845.