fractal and fractional

Article

Neural Fractional Order PID Controllers Design for 2-Link Rigid Robot Manipulator

Mohamed Jasim Mohamed [1], Bashra Kadhim Oleiwi [1], Layla H. Abood [1], Ahmad Taher Azar [2,3,4,*] and Ibrahim A. Hameed [5,*]

[1] Department of Control and Systems Engineering, University of Technology, Baghdad 19006, Iraq; 60098@uotechnology.edu.iq (M.J.M.); bushra.k.oleiwi@uotechnology.edu.iq (B.K.O.); 60066@uotechnology.edu.iq (L.H.A.)

[2] College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

[3] Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh 11586, Saudi Arabia

[4] Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt

[5] Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Larsgardsvegen 2, 6009 Alesund, Norway

[*] Correspondence: aazar@psu.edu.sa or ahmad.azar@fci.bu.edu.eg or ahmad_t_azar@ieee.org (A.T.A.); ibib@ntnu.no (I.A.H.)

Abstract: The robotic manipulator is considered one of the complex systems that include multi-input, multi-output, non-linearity, and highly coupled. The uncertainty in the parameters and external disturbances have a negative influence on the performance of the system. Therefore, the controllers that will be designed for these systems must be able to deal with these complexities and difficulties. The Proportional, Integral, and Derivative (PID) controller is known to be simple and well robust, while the neural network has a solid ability to map complex functions. In this paper, we propose six control structures by combining the benefits of PID controller with integer and fractional order and the benefits of neural networks to produce hybrid controllers for a 2-Link Rigid Robot Manipulator (2-LRRM) handling with the problem of trajectory tracking. The Gorilla Forces Troops Optimization algorithm (GTO) was used to tune the parameters of the proposed controller schemes to minimize the Integral of Time Square Error (ITSE). In addition, the robustness of the performance of the suggested control systems is tested by altering the initial position, external disturbances and parameters and carried out using MATLAB. The best performance of the proposed controllers was the Neural Network Fractional Order Proportional Integral Derivative Controller (NNFOPID).

Keywords: neural fractional order PID controllers; self-tuning PID controller; neural network; PID and FOPID controllers; gorilla troops optimizer (GTO); robotic manipulator; 2-link rigid robot manipulator

check for updates

Citation: Mohamed, M.J.; Oleiwi, B.K.; Abood, L.H.; Azar, A.T.; Hameed, I.A. Neural Fractional Order PID Controllers Design for 2-Link Rigid Robot Manipulator. *Fractal Fract.* **2023**, *7*, 693. https://doi.org/10.3390/fractalfract7090693

Academic Editor: Yongguang Yu

Received: 17 July 2023
Revised: 7 August 2023
Accepted: 21 August 2023
Published: 18 September 2023

1. Introduction

The reduction of human efforts became the matter behind all inventions in technology. The computer invention led to various boundaries of thought, giving various choices that gave optimistic solutions in different fields. One such important solution in getting rid of the intervention done by humans in different activities was the robot invention [1]. Robotics is the science responsible for its simulation, design, and control. Today robots are available for different uses in life. It has become an important tool to people in many industrial and daily life missions. Different applications appeared, which include medical, industrial, educational, and military use [2]. Any system during its work may face too many disturbances, even robots, various uncertainties, changes in its parameters, and nonlinearity problems. Due to all these reasons, classical controller like proportional-integral-derivative (PID) is not a good choice for achieving simultaneous effective response for tracking and monitoring the desired command. To fix all these drawbacks of classical control techniques, nonlinear control systems have been adopted [3]. Different researchers have proposed different controllers for a 2-LRRM. In [4], a simple design of RBFNN is adopted to estimate the lumped uncertainties. It is unlike others designed based on the existing barrier Lyapunov function. The control scheme is proposed based on the assumption of using the case when the violation of constraint is happened and is taken into consideration. A controller design for the robot manipulator is utilized to solve the constraints of output error and control input saturation by removing the feasibility condition that appeared in most other controllers. The control scheme must consider the assumption that due to the saturation of control input, the violation constraint may happen. In [5], a neural network was adopted as a neuro controller used for enhancing the performance of tracking for the robot manipulator combined with a sliding mode controller design. Depending on the non-model-based configuration, the selection of the nonlinear gain value must be large enough to save system stability. In [6], a modification in neural network structure is suggested as a novel tuning method used to calculate the gains of the used controller. Also, a new modulation method is presented to enhance the exploration step of the NNA without using initial values. The proposed algorithm is used to select the proportional integral derivative (PID) controller gain values. Also, an important contribution is achieved by adopting a new fitness function to ensure the improvement in the settling time and the overshoot of all arms output at the same time. In [7], a combination of some controllers used with a fuzzy logic technique such as PD, PID, and calculated torque control was suggested. A comparison with classical controllers is done. The fuzzy controller reflects good results in performance and accuracy in enhancing system tracking ability. The main points of this study were the system robustness and stability. Also, the values of the gain used are calculated without using any tuning method, but if we see in [8], a fuzzy-neural-network PID (FNN-PID) structure is proposed to tune the PID controller gains efficiently. In [9], an efficient type-2 adaptive method has been presented to ensure the best tracking ability of a 3-DOF manipulator robot as an industrial robot in a situation with faults. For fuzzy type-2 values online updating, a Lyapunov stability concept is used. In [10], a Fractional adaptive PID (FPID) controller was presented. The stability of a closed-loop FPID control system can be guaranteed under external disturbance by using the Lyapunov approach with a supervisory controller. It is found that when error signals in the learning stage area are bounded, the trajectory of the robot converges to the desired one asymptotically. In [11], a conventional PID controller and a robust nonlinear sliding mode control are introduced to control the motion of the end effector at specific positions for pick and place tasks. A comparative study is done, and the results show that Sliding mode control surpasses PID control with minimum tracking error. In [12], an optimal PID control law is obtained from the proposed (ABC) algorithm and applied to the robotic system. The designed controller optimizes the trajectory of the robot's end effector for a time-variant input and makes the robot robust in the presence of external disturbance. The results of the suggested control method reflect the ability to deliver an error with a small value even if there are payload changes or actuator defects. In [13], a fractional-order fuzzy-based PID controller was presented with the Ant lion optimization algorithm for controlling buck converter. Using fractional calculus with fuzzy logic improves the system response stability and robustness. Then a comparison with two controllers (PID-FOPID) tuned with the PSO algorithm emphasizes the best-proposed controller efficiency than other controllers, while in [14], A sigmoid PID (SPID) controller with a nonlinear sine cosine optimization method is suggested for controlling the terminal voltage in the automatic voltage regulator, this type of controller offer best and stable behavior even when a robustness analysis is done. In [15], an adaptive PID was adopted to control a MAV quadrotor model based on a second-order sliding mode for tuning variables. Then, to reduce the chattering issue in SMC, a fuzzy compensator is suggested, and its stability is tested based on the Lyapunov theorem. In [16], the author presents a PID and fuzzy logic controller for improving the trajectory tracking of Mecanum-wheeled mobile robots. The geometric modeling method is used to infer the kinematic error variables. In [17], a data-driven neuroendocrine-PID is adopted with the benefits of using an adaptive safe experimentation dynamics algorithm to enhance a MIMO system's behavior by minimizing the error value and the fitness function used, which reflects its accuracy and stability. Previous studies showed that all or most of the studies did not solve the signal chattering in the control signal and dealt with the proposed controllers individually. In this paper, six controller structures are designed to successfully solve the signal chattering in the control signal and control a nonlinear, coupled, multi-input, multi-output 2-LRRM system. Moreover, a hybrid controller was proposed between the PID and neural network to achieve the requirements, and it was shown that it is the best among the proposed controllers. Different control schemes are proposed to solve the 2-link rigid robot manipulator control, and the GTO algorithm is used to calculate the best values of the controller's gains. The obtained results have been presented, and a comparison between all suggested controllers has been done.

The paper contributions are explained in points below:

1. Six controller structures are suggested by combining the proportional, integral, and derivate operations and neural networks.
2. Suggest a new objective function to make the tuning process produces a controller that has a minimum chattering in the control signal.
3. Applying a strong competition between the proposed controllers, especially for robustness, among the proposed controllers that integrate the specifications of the PID controller and neural networks.

The rest study sections are listed here. In Section 2, the system modeling is presented. In Section 3, the proposed controller structures are explained. In Section 4, the GTO algorithm is explained. The results and conclusion sections are presented in Sections 5 and 6.

2. Dynamic Model of 2-LRRM

As shown in Figure 1, the 2-LRRM comprises two links with lengths l_1 and l_2 , and their centre of masses is m_1 and m_2 respectively, which are determined at the distal ends of the links, to produce the control torque at joints A and B, DC motors will be utilized to predict the links angular positions θ_1 and θ_2 and velocities $\dot{\theta}_1$ and $\dot{\theta}_2$, the encoders will be applied [18]. The dynamic equation of motion in robotics is utilized for setting up the equations of the controller. Actuators are used for the generation of the torque values that are necessary for the dynamic motion of the manipulator. The correlation between input torques and components configurations' time change rates of the manipulator is concerned with deriving calculations of the manipulator motion due to the moments, and forces applied to it. As a consequence, the manipulator robot dynamic modelling contains the determination of functions that determine forces applying on structures, locations of the joint, velocities, and accelerations [19].

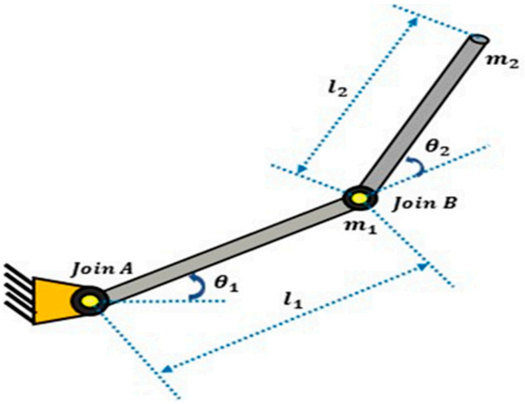

Figure 1. The 2-LRRM structure.

The x-position and y-position equations for m_1 are expressed by:

$$x_1 = l_1 \cos(\theta_1) \tag{1}$$

$$y_1 = l_1 \sin(\theta_1) \tag{2}$$

Likewise, the x position and y position equations of m_2 are defined by

$$x_2 = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{3}$$

$$y_2 = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{4}$$

The kinetic energy is written as:

$$KE = \frac{1}{2} m_1 \left(\dot{x}_1^2 + \dot{y}_1^2 \right) + \frac{1}{2} m_2 \left(\dot{x}_2^2 + \dot{y}_2^2 \right) \tag{5}$$

And the potential energy can be given as:

$$PE = m_1 g l_1 \sin(\theta_1) + m_2 g (l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)) \tag{6}$$

Next, by Lagrange Dynamic, the Lagrangian form can be expressed as:

$$L = KE - PE \tag{7}$$

The Euler-Lagrange Equation can be defined by:

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_i} \right] - \frac{\partial L}{\partial \theta_i} = F\theta_i \tag{8}$$

where, $F\theta_i$ represents the torque on *i*'th link. Finally, according to the Lagrangian equation, the dynamics of the manipulator can be written as the two coupled nonlinear differential equations [20]:

$$\begin{aligned} \tau_1 = & \left[(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(\theta_2) \right] \ddot{\theta}_1 + \left[m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) \right] \ddot{\theta}_2 \\ & - m_2 l_1 l_2 \left(2 \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2 \right) \sin(\theta_2) + (m_1 + m_2) g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_1 + \theta_2) \end{aligned} \tag{9}$$

$$\tau_2 = \left[m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) \right)] \ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2) + m_2 g l_2 \cos(\theta_1 + \theta_2) \tag{10}$$

The dynamics of this manipulator are represented in standard form

$$M(\theta) \ddot{\theta} + V\left(\theta, \dot{\theta}\right) + g(\theta) = \tau \tag{11}$$

With $V\left(\theta, \dot{\theta}\right)$ represents the Coriolis/centripetal vector, $M(\theta)$ represents the inertia matrix, and $g(\theta)$ equals the gravity vector. As can be noted that $M(\theta)$ is symmetric.

$$M = \begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix} \tag{12}$$

$$M_{11} = (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(\theta_2) \tag{13}$$

$$M_{12} = m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) \tag{14}$$

$$M_{12}M_{21}\&\ \ M_{22}\ m_2 l_2^2 \tag{15}$$

V represents the Coriolis and centrifugal matrix, which can be expressed by

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{16}$$

$$V_1 = -m_2 l_1 l_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2 \right) \sin(\theta_2) \tag{17}$$

$$V_2 = m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2) \tag{18}$$

The gravity vector $g = [\ g_{12}\ g_{21}\ ]^T$ is given by:

$$g_{12} = (m_1 + m_2) g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_1 + \theta_2) \tag{19}$$

$$g_{21} = \ m_2 g l_2 \cos(\theta_1 + \theta_2) \tag{20}$$

The nominal values of the parameters for the model are shown in Table 1 [19].

Table 1. The nominal parameters of 2-LRRM.

Parameters	Nominal Value
m_1	0.1 kg
m_2	0.1 kg
l_1	0.8 m
l_2	0.4 m
g	9.81 m/s^2

3. Artificial Gorilla Troops Optimizer (GTO)

An optimization algorithm provides a smart solution to fix different problems because, in real life, the intelligence in the collective behavior of natural organisms has been studied and converted into simple and smart equations [21–23]. In this study, an artificial GTO is suggested for tuning the controller's parameters. Gorilla animals arrange their lives as troops, and many adult females with their children with adult males with silver-coloured hair. They call him silverback gorilla. Each male and female try to go away from their place for another place. Males want to leave their places and be within another group by drawing females' attention to a new troop, while others are still not leaving the same place from their birth and being in the same troop. When the leader dies, the populations available try to control others or handle the silverback to lead them. At this point, the chosen leader is considered the suitable choice, and all other males obey him and leave away from the worst choice. Two phases will describe this algorithm as indicated below [24,25].

3.1. Exploration Phase

In this phase, each one will be considered a competitor to the best decision from the silverback gorilla. Three decisions can be taken in this phase and are appeared in Equation (21), the first decision will increase the algorithm exploration by leaving to an undefined or unknown position, and the other one will make balance behaviour by leaving to a new group while the final choice will raise GTO capability for finding many positions regarded as a known ones.

$$\mathrm{GX}(t+1)=\begin{cases}(\mathrm{UL}-\mathrm{LL})r_1+\mathrm{LL},\ \text{if rand } <\mathrm{P}\\ (r_2-a)\mathrm{X_r}(t)+\mathrm{L}\times\mathrm{H},\ \text{if rand } \geq 0.5\\ \mathrm{X}(i)-\mathrm{L}(\mathrm{L}(\mathrm{X}(t)-\mathrm{GX_r}(t))+r_3(\mathrm{X}(t)-\mathrm{GX_r}(t))),\ \text{if rand } <0.5\end{cases} \tag{21}$$

GX(t + 1) is a nominee in the next iteration. X_r (t) parameter is used for anyone in the troop nominees taken randomly from them and GX_r (t) will be the position of that random one. UL and LL are the upper and lower variables levels, r_1, r_2 and r_3 are assumed to be random items [0] to [1]. The variables a, L and H is calculated below:

$$\mathrm{a}=\mathrm{C}\times(1-\mathrm{It}/\mathrm{MaxIt}) \tag{22}$$

$$\mathrm{C}=\cos(2\times r_4)+1 \tag{23}$$

$$\mathrm{L}=\mathrm{c}\times 1 \tag{24}$$

$$\mathrm{H}=\mathrm{Z}\times\mathrm{X}(t) \tag{25}$$

$$\mathrm{Z}=[-a\ ,a], \tag{26}$$

3.2. Exploitation Phase

Two decisions will be taken in this phase. One decision is to obey the silverback gorilla, the leader, while the other is to be a competitor with the adult ones. Each decision is selected by comparing with the value calculated in Equation (26), if a ≥ W, the first decision is taken, while if a < W second decision will be taken. W is a predetermined parameter given before starting.

3.2.1. Following the Adult Silverback Leader

Gorilla: At this point, the silverback will be the leader, and others will follow his decisions to search their food places. It is taken if a ≥ w and shown clearly in the equation below:

$$\mathrm{GX}(t+1)=\mathrm{L}\times\mathrm{M}\times(\ \mathrm{X}(t)-\mathrm{X_{silverback}}\ )+\mathrm{X}(t) \tag{27}$$

$X_{\mathrm{silverback}}$ is the leader's place, and X(t) is the gorilla location; the variables L and M are chosen.

3.2.2. Competition for Adult Females

In this case, the decision will take as young males have adulthood age; the struggle is severe when they attack each other to initiate new families with adult females. Decision now is taken when a < w as can be described clearly in the equation below:

$$\mathrm{GX}(i)=\mathrm{X_{silverback}}\ -(\mathrm{X_{silverback}}\ \times\mathrm{R}-\ \mathrm{X}(t)\times\mathrm{R})\times\mathrm{A}, \tag{28}$$

$$\mathrm{R}=2\times r_5-1, \tag{29}$$

$$\mathrm{A}=\beta\times\mathrm{E}, \tag{30}$$

$$\mathrm{E}=\begin{cases}\mathrm{N}_1=\text{rand}\geq 0.5\\ \mathrm{N}_2=\text{rand}<0.5\end{cases} \tag{31}$$

where "R" explain the impact force, which is found by solving Equation (23), r_5 is a value taken randomly between [0, 1] as struggle happens, Equation (26) was adopted to find the violence value during struggles, and A is the vector used for utilizing to solve the violence level by using Equation (24). Several parameters will be preset previously, like a, β, and E, which are regarded as an effect of violence on choice levels. The fitness function of all GX (t) is found, as the fitness function of GX (t) < X (t), the GX (t) will use the X (t) as an optimal solution, the best decision taken from the troop is taken as a new silverback leader. Figure 2 below explains all algorithm steps.

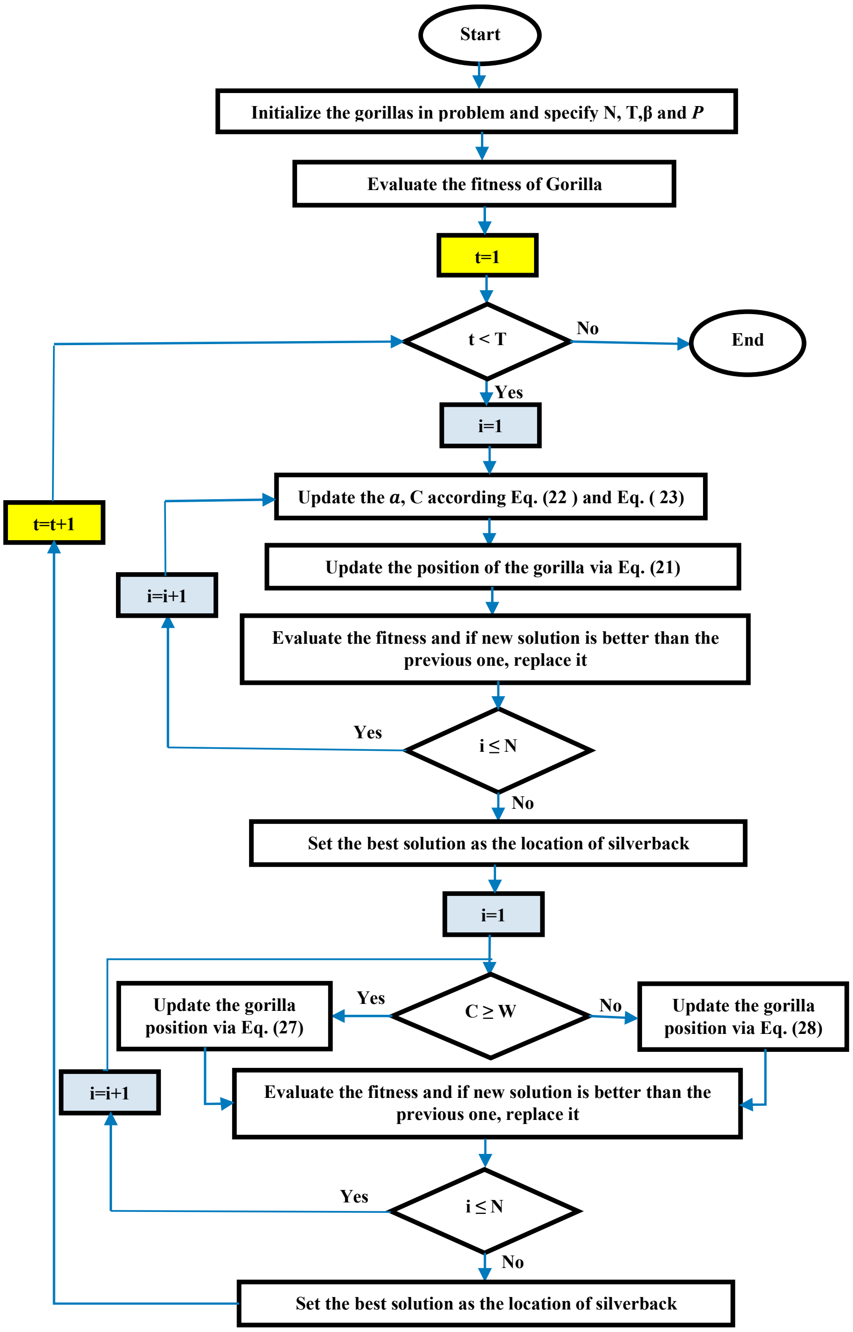
Start
Initialize the gorillas in problem and specify N, T,β and *P*
Evaluate the fitness of Gorilla
t=1
t < T
No
End
Yes
i=1
Update the *a*, C according Eq. (22) and Eq. (23)
t=t+1
Update the position of the gorilla via Eq. (21)
i=i+1
Evaluate the fitness and if new solution is better than the previous one, replace it
Yes
i ≤ N
No
Set the best solution as the location of silverback
i=1
Update the gorilla position via Eq. (27)
Yes
C ≥ W
No
Update the gorilla position via Eq. (28)
i=i+1
Evaluate the fitness and if new solution is better than the previous one, replace it
Yes
i ≤ N
No
Set the best solution as the location of silverback

Figure 2. GTO algorithm steps [21].

4. The Structures of the Proposed Controllers

In this section, we introduce details information about the structures of the proposed hybrid neural controllers. Here we consider the problem of controlling one variable or the system is of type a Single Input a Single Output (SISO) system. Therefore, the following control structures represent part of the controller unit in a Multi-Input Multi-Output (MIMO) system.

4.1. Conventional PID Controller (Con-PID)

PID controller is the major widely proposed prevalent controller. It consists of three actions: proportional action, integral action, and derivative action. The proportional action reduces the rise time of the response, while the Integral action improves steady-state error, and the derivative action reduces the overshoot and improves the stability margin [26,27]. Since the derivative term is sensitive to measurement noise, a filter is added to reduce the effect of measurement noise. The derivative term without a filter may amplify the noise signal in the controller output and causes loss of control. A typical PID controller transfer function with the filter is indicated in Equation (32).

$$G(s)_{PID} = K_p + K_i \frac{1}{\mathrm{s}} + K_d \mathrm{s} \frac{N}{s+N} \tag{32}$$

The three gains K_p , K_i and K_d determine the effect of these proportional, integral, and derivative actions, respectively, and N is the filter's corner frequency. Therefore, the gains K_p , K_i, K_d, and N parameters are needed to be tuned to realize a certain objective. The conventional PID controller structure with filter is shown in Figure 3, and the block diagram of the feedback control system with PID or FOPID controller is shown in Figure 4.

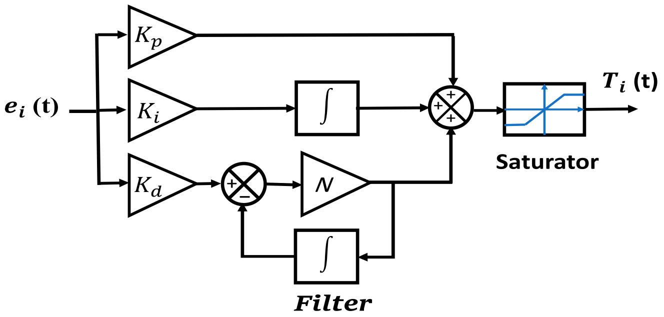

Figure 3. Conventional PID controller structure with filter.

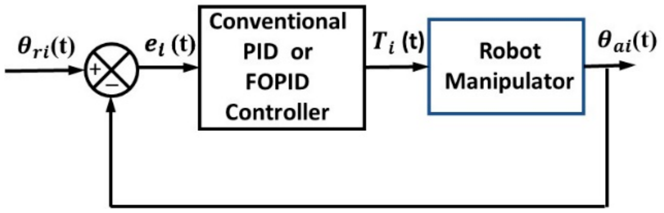

Figure 4. The block diagram of the control system with Conventional. PID or FOPID controller structures.

4.2. Conventional Fractional Order PID Controller (Con-FOPID)

This controller type is regarded as an enhanced scheme of a conventional PID controller because the values of order in the derivative and integral parts are fractional numbers instead of integer numbers [28,29]. These two fractional parameters are µ and λ; accordingly, due to these parameters and the N parameter of the filter, the controller needs to tune six parameters instead of four in case of using the filter. The additional tuned parameters will save robustness in the system's behavior and improve the performance. The FOPID controller structure is shown in Figure 5, and the transfer function of the FOPID controller is shown in Equation (33).

$$G(s)_{FOPID} = K_p + K_i \frac{1}{S^{\lambda}} + K_d s^{\mu} \frac{N}{s+N} \quad (33)$$

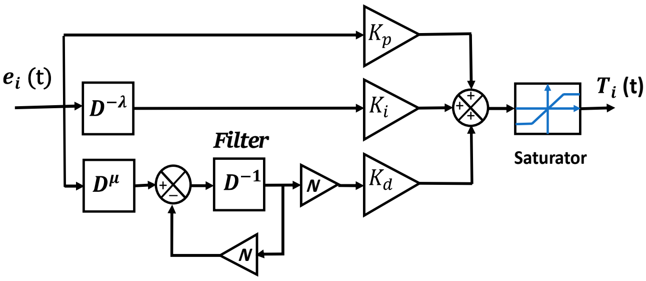

Figure 5. Con-FOPID controller structure.

The tuned parameters of the FOPID controller will be $(K_p,\ K_i, K_d,\ \lambda,\ \mu,\ N)$ [30]. If $\lambda = \mu = 1$, a conventional PID is obtained, while if one is 1 and the other is 0, it will be a conventional PI controller (if $\lambda = 1$, $\mu = 0$) or conventional PD controller (if $\lambda = 0$, $\mu = 1$). There are numerous methods to implement fractional differentiation and integration [31–33]. Oustaloup's recursive approximation filter is widely used to approximate the fractional order transfer function, which is used in this work [34]. The feedback control system with the FOPID controller is shown in Figure 5 [35].

4.3. Self-Tuning Neural Network PID Controller (STNN-PID)

The self-tuning neural network PID controller consists of two main parts: the first part is a feed-forward NN unit used for auto-tuning the parameters $(K_p,\ K_i\ ,\ K_d)$ of the PID controller, and the second part is the three operations (proportional, integral, and derivative) of the conventional PID controller applied to the error between the desired and the actual control variable to produce the control signal. The conventional PID controller receives the values of its three parameters from the neural network at each sampling time. Neural network training is occurred once at the design step according to a performance index then the neural network model still the same during the operation. The neural network model architecture will be fixed before training [36,37].

The structure of the neural network part of the controller is illustrated in Figure 6. The input layer consists of three input neurons as follows; θ_{ri} is the required angular position of *i*th link, θ_{ai} is the actual angular position of the *i*th link, and e_i is the error between the required and actual angular position of the *i*th link. The hidden layer consists of ten neurons, each hidden neuron connected with all input neurons via a weight value for each connection [38]. The activation function H(Σ) is applied to the sum of these three connections for each hidden neuron. The activation function is depicted in Equation (34).

$$H(\Sigma) = \left(2 - \Sigma^2\right) \cdot e^{-\Sigma^2} \quad (34)$$

The output layer consists of three neurons connected to all ten hidden neurons via a weight value for each connection. The output of each output neuron is the sum of all its connections with hidden neurons. The output of each output neuron represents one parameter value of the PID controller and is fed to the conventional PID controller. It is necessary to mention here that the corner frequency parameter N of the filter is tuned offline like the neural network parameters and is not auto-tuning parameter. The equation of the conventional PID controller is shown in Equations (32) and (35)–(38). The feedback control system with a self-tuning PID controller is shown in Figure 7.

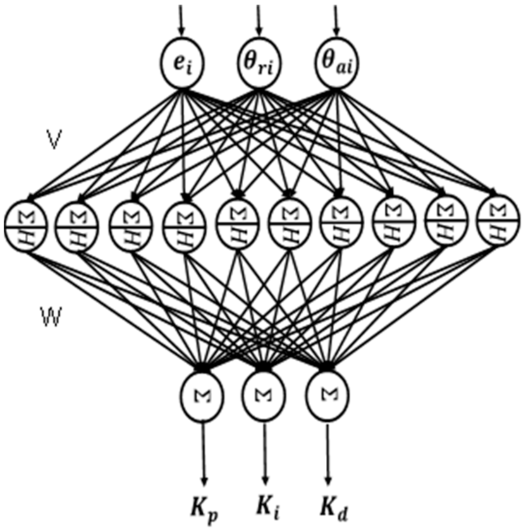

Figure 6. Self-tuning neural network PID controller.

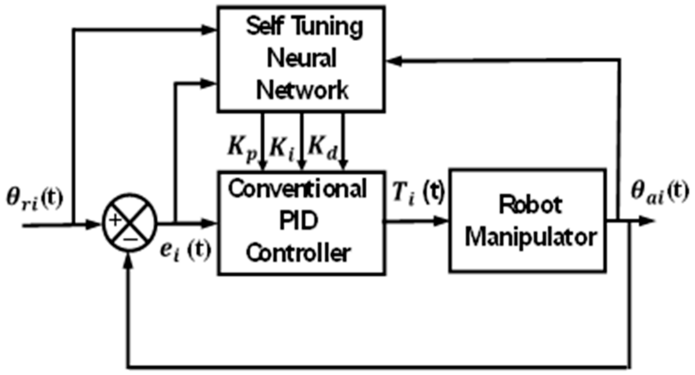

Figure 7. Feedback control system with self-tuning neural network PID controller.

4.4. Self-Tuning Neural Network FOPID Controller (STNN-FOPID)

The self-tuning neural network FOPID controller also consists of two main parts. The first part is a feed-forward neural network with the following features; the input layer, the hidden layer, the connections between neurons, and the type of activation function are the same as those used in the STNN-PID controller. The only difference from the neural network used in STNN-PID is the output layer. This layer consists of five neurons instead of three because of the need for five parameters $(K_p,\ K_i, K_d,\ \lambda,\ \mu)$ to tune the FOPID controller. In other words, the additional neurons in the output layer or additional auto-tuning parameters are needed to tune the fractional order parameters λ and μ of integration and differentiation, respectively while the parameter N of the filter is tuned offline with NN parameters. The structure of the STNN-FOPID controller is shown in Figure 8.

The second part is the three operations (proportional, fractional integral, and fractional derivative) of the FOPID controller applied to the error between the desired and the actual control variable to produce the control signal. The FOPID controller receives five parameters from the neural network at each sampling time. The equation of the FOPID controller is illustrated in Equations (33) and (35)–(39). The block diagram of the feedback control system with a self-tuning neural network STNN-FOPID controller is shown in Figure 9.

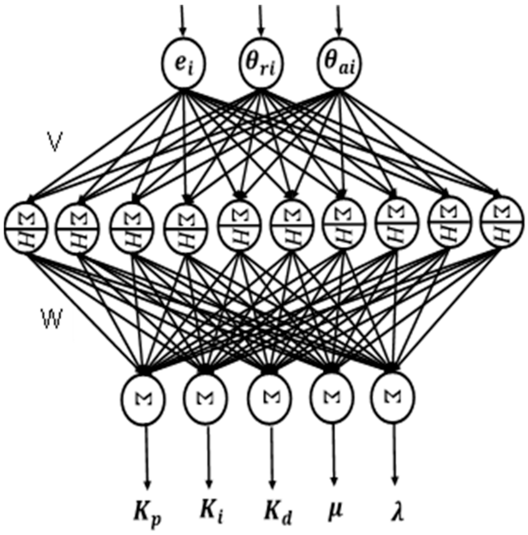

Figure 8. Self-tuning neural network FOPID controller.

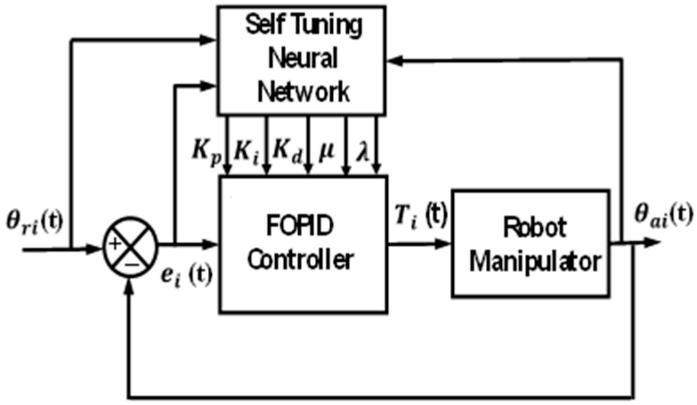

Figure 9. Feedback control system with self-tuning neural network FOPID controller.

4.5. Neural Network PID Controller (NN-PID)

The structure of this proposed hybrid controller NN-PID is shown in Figure 10. where in conventional PID controller λ = 1 and μ = 1. The structure depicts that the input layer consists of one neuron. This neuron represents the error *e*(*k*) between the desired and actual output of the controlled variable. The first hidden layer consists of three neurons. The input-output functions of these neurons are chosen according to the PID control law, while the input to these nodes is the error *e*(*k*) multiplied by the weight of connection for each neuron. This selection makes those neurons represent the role of proportional, integral, and derivative (PID) operations on its input and the weights of connections represent the gains (K_p, K_i , K_d) as shown in Equations (35)–(39).

$$P(k) = K_p \times e(k) \tag{35}$$

$$sum(k) = sum(k-1) + h \times e(k) \tag{36}$$

where, $sum(k) = 0$.

$$I(k) = K_i \times sum(k) \tag{37}$$

$$f(s) = \frac{N}{s+N}\, e(s) \qquad \text{the filter of derivative} \tag{38}$$

$$D(t) = K_d \times \frac{d}{dt} f(t)\ or\ D(k) = K_d \times (f(k) - f(k-1)\ )/h \tag{39}$$

where, Sum is the accumulator of integral operation. $P(k), I(k)$, and $D(k)$ are the outputs of the first hidden layer neurons, which are proportional, integral, and derivative operations of the error, h is the step size of the simulation.

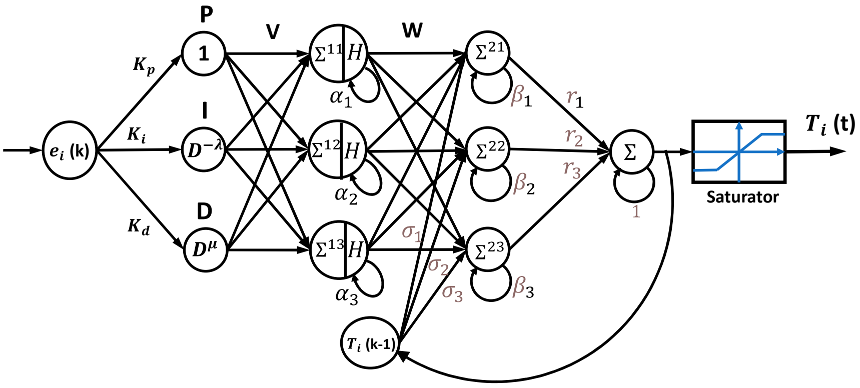

Figure 10. The structure of the NNPID controller.

The second hidden layer of the hybrid neural network controller consists of three neurons. Each neuron receives all outputs of the first hidden layer via different connection weights. The sum of all inputs for each neuron is submitted to the activation function $H(\Sigma)$. The output of each neuron represents the result of its activation function, plus its previous output as illustrated in Equations (40)–(41). The applied activation function is shown in Equation (42).

$$\begin{bmatrix} \sum(k)^{11} \\ \sum(k)^{12} \\ \sum(k)^{13} \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \\ v31 & v32 & v33 \end{bmatrix} \begin{bmatrix} P(k) \\ I(k) \\ D(k) \end{bmatrix} \tag{40}$$

$$\begin{bmatrix} O_1(k) \\ O_2(k) \\ O_3(k) \end{bmatrix} = \begin{bmatrix} H\left(\sum(k)^{11}\right) \\ H\left(\sum(k)^{12}\right) \\ H\ \left(\sum(k)^{13}\right) \end{bmatrix} + \begin{bmatrix} \alpha_1 \cdot O_1(k-1) \\ \alpha_2 \cdot O_2(k-1) \\ \alpha_3 \cdot O_3(k-1) \end{bmatrix} \tag{41}$$

$$H(\Sigma) = \frac{4}{(1 + e^{-\Sigma})} - 2 \tag{42}$$

where, $\sum(\mathrm{k})^{1i}$ is the sum of input connections to the ith neuron of the second layer and $O_i(k)$, the output of the ith neuron of the second layer. $v$ij and α_i are the weight parameters.

The third hidden layer consists of three neurons. Each one receives all neuron's outputs of the second hidden layer via weight for each connection plus its previous output via connection weight plus the previous output of the output layer neuron via the connection weight. Equations (43) and (44) depict the output of the third hidden layer.

$$\begin{bmatrix} \sum(k)^{21} \\ \sum(k)^{22} \\ \sum(k)^{23} \end{bmatrix} = \begin{bmatrix} w11 & w12 & w13 \\ w21 & w22 & w23 \\ w31 & w32 & w33 \end{bmatrix} \begin{bmatrix} O_1(k) \\ O_2(k) \\ O_3(k) \end{bmatrix} \tag{43}$$

$$\begin{bmatrix} S_1(k) \\ S_2(k) \\ S_3(k) \end{bmatrix} = \begin{bmatrix} \sum(k)^{21} \\ \sum(k)^{22} \\ \sum(k)^{23} \end{bmatrix} + \begin{bmatrix} \beta_1 \cdot S_1(k-1) \\ \beta_2 \cdot S_2(k-1) \\ \beta_3 \cdot S_3(k-1) \end{bmatrix} + \begin{bmatrix} \sigma_1 \cdot T(k-1) \\ \sigma_2 \cdot T(k-1) \\ \sigma_3 \cdot T(k-1) \end{bmatrix} \tag{44}$$

where, $\sum(\mathrm{k})^{2i}$ is the sum of input connections to the ith third hidden layer neuron $\mathrm{S}_i(\mathrm{k})$ is the output of this neuron, $w$ij, β_i , σ_i and are weight parameters. Finally, the output layer consists of one neuron, the inputs to this neuron are all outputs of the third hidden layer via weight values and its previous output, as shown in Equation (45).

$$\mathrm{T(k) = T(k-1) + r_1 \cdot S_1(k) + r_2 \cdot S_2(k) + r_3 \cdot S_3(k)} \tag{45}$$

The result of the output layer neuron represents the control signal applied to the system. The structure of the feedback control system with a neural network PID controller is shown in Figure 11.

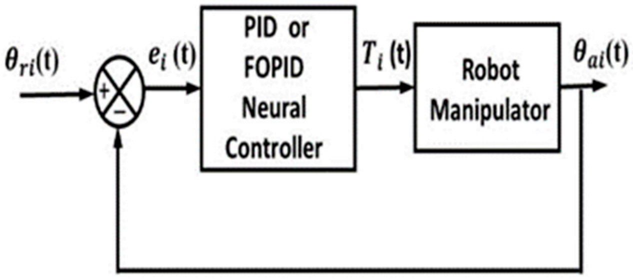

Figure 11. Block diagram of the feedback control system with neural network PID or FOPID controller.

4.6. Neural Network FOPID controller (NN-FOPID)

This hybrid neural network controller has the same number of inputs, hidden, and output layers and the same number of neurons in each layer as the structure of the NN-PID controller. Moreover, the same activation function H(Σ) is also used. The only differences between NN-PID and NN-FOPID structures are the implementation of integration and differentiation operations. The NN-FOPID structure enables the integration and differentiation actions to be performed in fractional order, while the NN-PID performs these operations in integer order only. The three actions of the FOPID controller are shown in Equations (46)–(49). The feedback control system with the FOPID controller is shown in Figure 11.

$$P(t) = K_p \times e(t) \tag{46}$$

$$I(t) = K_i \ \times D^{-\lambda} \, e_{\theta i}(t) \tag{47}$$

$$f(s) = \frac{N}{s+N}\, e(s) \qquad \text{the filter of derivative} \tag{48}$$

$$D(t) = K_d \ \times D^{\mu} \, f(t) \tag{49}$$

5. Simulation Results

The 2-LRRM model with the proposed controllers and test tracks was performed using MATLAB programming. The performance and robustness of all proposed controllers in trajectory tracking are presented in this section. For each link, the trajectory to be tracked is computed so that the manipulator will be forced to follow it. The results are then used to evaluate the proposed controller based on the performance index value of each controller. In this work, the assessment of all proposed controllers' performance is based on the sum of the ITSE for both link trajectories [36]. The optimal controller is based on the lowest value of ITSE, written by Equation (50).

$$\mathrm{ITSE} = \int \Big(\mathrm{t} \times \ \mathrm{e}_1^2(\mathrm{t}) + \mathrm{t} \times \mathrm{e}_2^2(\mathrm{t})\Big)\mathrm{dt} \tag{50}$$

where $e_1(t)$ and $e_2(t)$ are the difference between the desired trajectory and the actual trajectory of the link1 and link2. The desired trajectories θ_{r1} and θ_{r2} of the link1 and link2 can be expressed by Equations (51) and (52), respectively, as follows:

$$\theta_{r1} = \begin{Bmatrix} 0.75\,t^2 - 0.25\,t^3 & (0 < t < 2) \\ -1.53\,t - 1.125t^2 + 0.125t^3 & (2 < t < 4) \end{Bmatrix} \tag{51}$$

$$\theta_{r2} = \begin{Bmatrix} 1.5\,t^2 - 0.5\,t^3 & (0 < t < 2) \\ 12 - 12\,t + 4.5\,t^2 - 0.5\,t^3 & (2 < t < 4) \end{Bmatrix} \tag{52}$$

Before showing the results of all proposed controllers, we noted from many experiments that tuning the proposed controllers according to minimize the ITSE in Equation (50) led, in most cases, chattering in control signals. To make the optimization algorithm exclude the solution with the high chattering in the control signal, we propose a new fitness function for the optimization algorithm to make the chattering under our control.

The classical objective function was presented in Equation (53);

$$\text{(min) J = ITSE} \tag{53}$$

where (min) means minimizing the integral time square error.

The new proposed objective is presented in Equation (54)

$$\text{(min) J = ITSE + Uc} \times \rho \tag{54}$$

where, Uc is the number of times that the slope of the control signal changes its sign. If the system has more than one control signal, the number of slop sign changes for all control signals are collected. ρ is a small positive number chosen as 10^{-8} or 10^{-7}.

The execution time in the simulation part took 4 sec., and the step size of the simulation took 1 ms. The tuning experiment used two initial positions (0.1745, 0.1745) and ($-$0.1745, $-$0.1745) for θ_1 and θ_2 respectively at the same time to assess the solution to increase the amount of learning. The nominal parameters of the 2-LRRM model are considered in the design stage of all proposed controllers.

The GTO is used for tuning the parameters of all proposed controllers regarding the tracking error between the 2-LRRM actual and desired trajectories. The proposed GTO setting is the following; the size of the population = 80, team size = 10, and iterations = 1000. In the last iteration, the optimal solution took as the result of GTO. The number of tuning parameters and their range in the search space used by the GTO algorithm to find the optimal set of parameters is shown in Table 2.

The performance index (ITSE) value and the number of sign changes in all control signals when using the two initial positions above for all suggested controllers are shown in Table 3. The result generally indicates that the ITSE values for the proposed controllers with fractional order integral and fractional order derivative actions are better than controllers with integer order integral and integer order derivative actions. This is because the fractional order of these operations increases the flexibility of the controller by increasing its tuning parameters. The result shows that the proposed hybrid neural network controller NN-FOPID structure has the best value of the $ITSE = 0.748071 \times 10^{-4}$ compared with other proposed controllers.

Now, to illustrate more details about the behavior of the proposed controllers, the initial position (0.1745, 0.1745) is used to execute all proposed, designed controllers on the nominal system. The obtained results are shown in Table 4.

Table 2. The number of design parameters and their search range for all proposed controllers.

Controller	**Total Number of Controller Parameters**	**Range of PID Gains K_p, K_i, K_d**	**Corner Frequency of Derivative Filter N**	**Range of Fractional Parameters**	**All Other Parameters Range**
Con-PID	8	−150 to 150	10 to 100	$\mu \equiv 1$ $\lambda \equiv 1$	--------
Con-FOPID	12	−150 to 150	10 to 100	$\mu \equiv 0$ to 2 $\lambda \equiv 0$ to 2	--------
STNN-PID	122	−150 to 150	10 to 100	$\mu \equiv 1$ $\lambda \equiv 1$	V ≡ −5 to 5 W≡ −1 to 1
STNN-FOPID	162	−150 to 150	10 to 100	$\mu \equiv 0$ to 2 $\lambda \equiv 0$ to 2	V ≡ −5 to 5 W≡ −1 to 1
NNPID	66	−150 to 150	------	$\mu \equiv 1$ $\lambda \equiv 1$	−1 to 1
NNFOPID	70	−150 to 150	------	$\mu \equiv 0$ to 2 $\lambda \equiv 0$ to 2	−1 to 1

Table 3. The ITSE of the Proposed Controllers for Nominal Plant.

Controller	**ITSE**	**No. of Slop Sign Change in All Control Signals**
Con-PID	3.729543×10^{-4}	93
Con-FOPID	2.227023×10^{-4}	47
STNN-PID	3.075515×10^{-4}	91
STNN-FOPID	3.883774×10^{-4}	45
NNPID	0.954084×10^{-4}	85
NNFOPID	0.748071×10^{-4}	94

Table 4. The performance of the proposed controllers when the initial position (0.1745, 0.1745) is used.

Controller Type	**Link No.**	**Rise Time**	**Over Shoot %**	**Settling Time**	**ITSE $\times 10^{-4}$**
Con-PID	L1	0.070	6.6	0.684	1.47752
	L2	0.012	5.95	0.188	0.64646
Con-FOPID	L1	0.074	3.27	0.584	1.30261
	L2	0.054	1.40	0.131	0.10037
STNN-PID	L1	0.069	6.14	0.594	1.12337
	L2	0.012	7.41	0.394	0.83790
STNN-FOPID	L1	0.081	4.40	6.430	1.25247
	L2	0.026	1.05	0.166	1.00217
NN-PID	L1	0.081	1.60	0.134	0.34509
	L2	0.042	2.84	0.103	0.07345
NN-FOPID	L1	0.076	1.80	0.123	0.31060
	L2	0.043	0.47	0.043	0.03249

Table 4 shows that the NN-FOPID controller has a good rise time, minimum overshoot, and minimum settling time, as well as minimum ITSE for link1 and link2. Conversely, the Con-PID controller has the maximum settling time and maximum overshoot and worse rise time and ITSE. In addition, the control actions T_1 and T_2 for the NN-FOPID controller has the least control efforts as well as the trajectory followed by the 2-LRRM end effector for the NN-FOPID controller is the closest to the required path. The tracking of the trajectories of θ_1 and θ_2, the T_1 and T_2 as well as the trajectory tracked by the 2-LRRM end-effector are demonstrated in Figure 12.

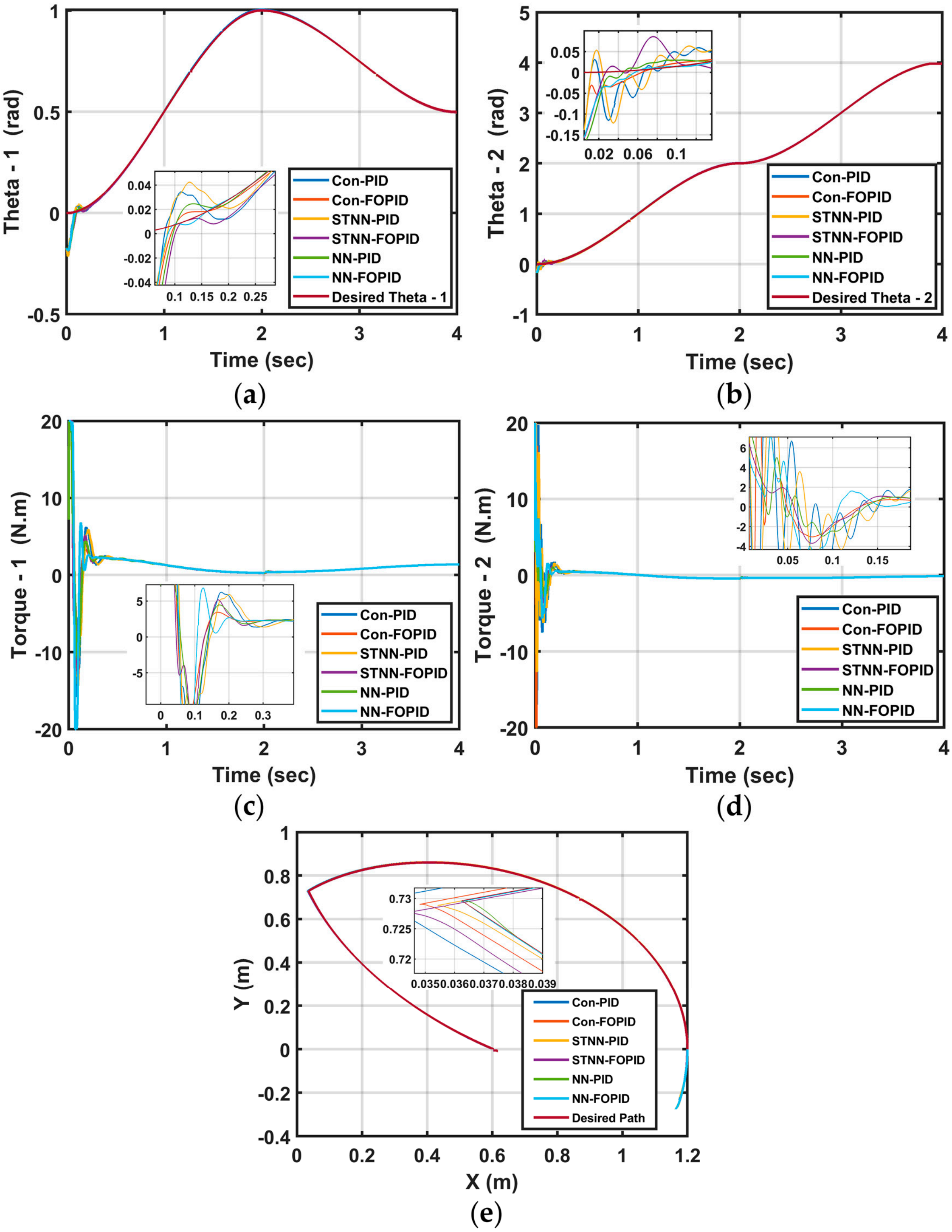

Figure 12. (**a**) Desired and actual θ_1, (**b**) Desired and actual θ_2, (**c**) Torque T_1, (**d**) Torque T_2, (**e**) Desired and actual paths of end-effector.

5.1. Robustness Tests

Now, in the following tests, we will check the robustness of all proposed controllers without retuning the gains of controllers to prove the capability and robustness of each controller.

5.1.1. Change Initial Position

To check the robustness of all proposed controllers, another primary position [0.15, 0.15] for [θ_1, θ_2] took for testing the capability of all the proposed controllers to follow the 2-LRRM on the required trajectory.

The obtained results of the performance index values for all suggested controllers are shown in Table 5. The tracking of the trajectories of θ_1 and θ_2, by the end-effector of the 2-LRRM by altering the initial position for all suggested controllers are presented in Figure 13.

Despite changing the initial positions, the NN-FOPID controller still performs better than the rest, because there is an almost smaller overshoot in the trajectory response for θ_1 and θ_2 and the settling time is almost small. While the STNN-FOPID has the worst response because of the θ_1 and θ_2 responses have large overshoots and long settling times. Moreover, the trajectory followed by the 2-LRRM end effector for the NN-FOPID controller remains the closest to the required trajectory.

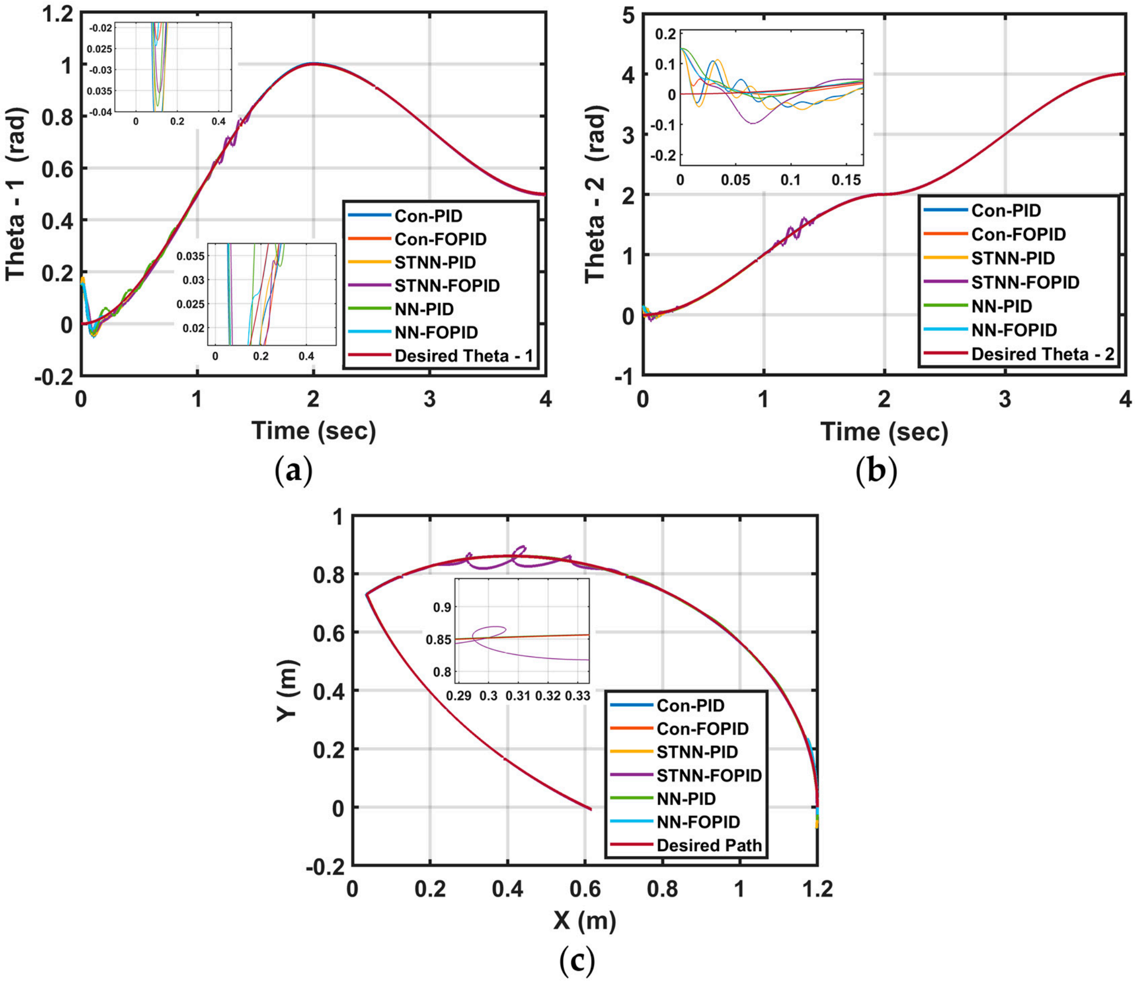

Figure 13. Desired and actual trajectories (**a**) for θ_1 , (**b**) for θ_2 , (**c**) Desired and actual paths of end-effector based on initial positions (0.15, 0.15).

Table 5. The ITSE of the proposed controllers based on initial position (0.15, 0.15).

Controller	ITSE
Con-PID	1.82669×10^{-4}
Con-FOPID	1.17614×10^{-4}
STNN-PID	1.61995×10^{-4}
STNN-FOPID	27.3328×10^{-4}
NN-PID	1.05251×10^{-4}
NN-FOPID	0.24644×10^{-4}

5.1.2. Disturbance Addition

The robustness test of the suggested controllers was performed by using the disturbance term [sin (50t), sin (50t)] to the control signal [T_1 , T_2] and selecting the initial position as [0, 0] to [θ_1, θ_2]. The obtained results of ITSE values for all suggested controllers are demonstrated in Table 6. The tracking of the trajectories of θ_1 and θ_2 and the trajectory followed by the 2-LRRM end-effector based on adding disturbance of sin50t N-m in both links are illustrated in Figure 14. As observed, the NN-FOPID hybrid neural controller gives the smallest ITSE, and smallest overshoot. All trajectories are closest to the desired trajectories and are free of oscillations compared with other control structures. The STNN-FOPID loses control and stability, while STNN-PID gives the worst performance compared with other controllers. Therefore, we conclude that the NN-FOPID is better than others for disturbance rejection.

Table 6. The ITSE of the proposed controllers based on disturbances sin (50t) for each link and initial position (0.0, 0.0).

Controller	ITSE
Con-PID	5.54533×10^{-4}
Con-FOPID	1.43023×10^{-4}
STNN-PID	191.4245×10^{-4}
STNN-FOPID	Unstable
NN-PID	2.1375×10^{-4}
NN-FOPID	0.092827×10^{-4}

5.1.3. Parameters Variations

The parameter variations also examined the performance of all proposed controllers by increasing the masses of both links by 5% and selecting the initial position as [0, 0] to [θ_1, θ_2]. The values of ITSE for all proposed controllers are presented in Table 7. The tracking trajectories of θ_1 and θ_2 by the 2-LRRM end-effector for all controllers is displayed in Figure 15.

From the figures below in Figure 15, it is clear that the trajectory tracking for θ_1 and θ_2 of the NN-FOPID controller are closest to the required θ_{r1} and θ_{r2} compared with other controller structures. As a result, the trajectory followed by the 2-LRRM end-effector with parameter variations for the NN-FOPID controller is very close to the desired trajectory. Therefore, the NN-FOPID controller outperforms all other proposed controllers when the values of system parameters vary. The STNN-FOPID controller also fails in this test and loses control and stability.

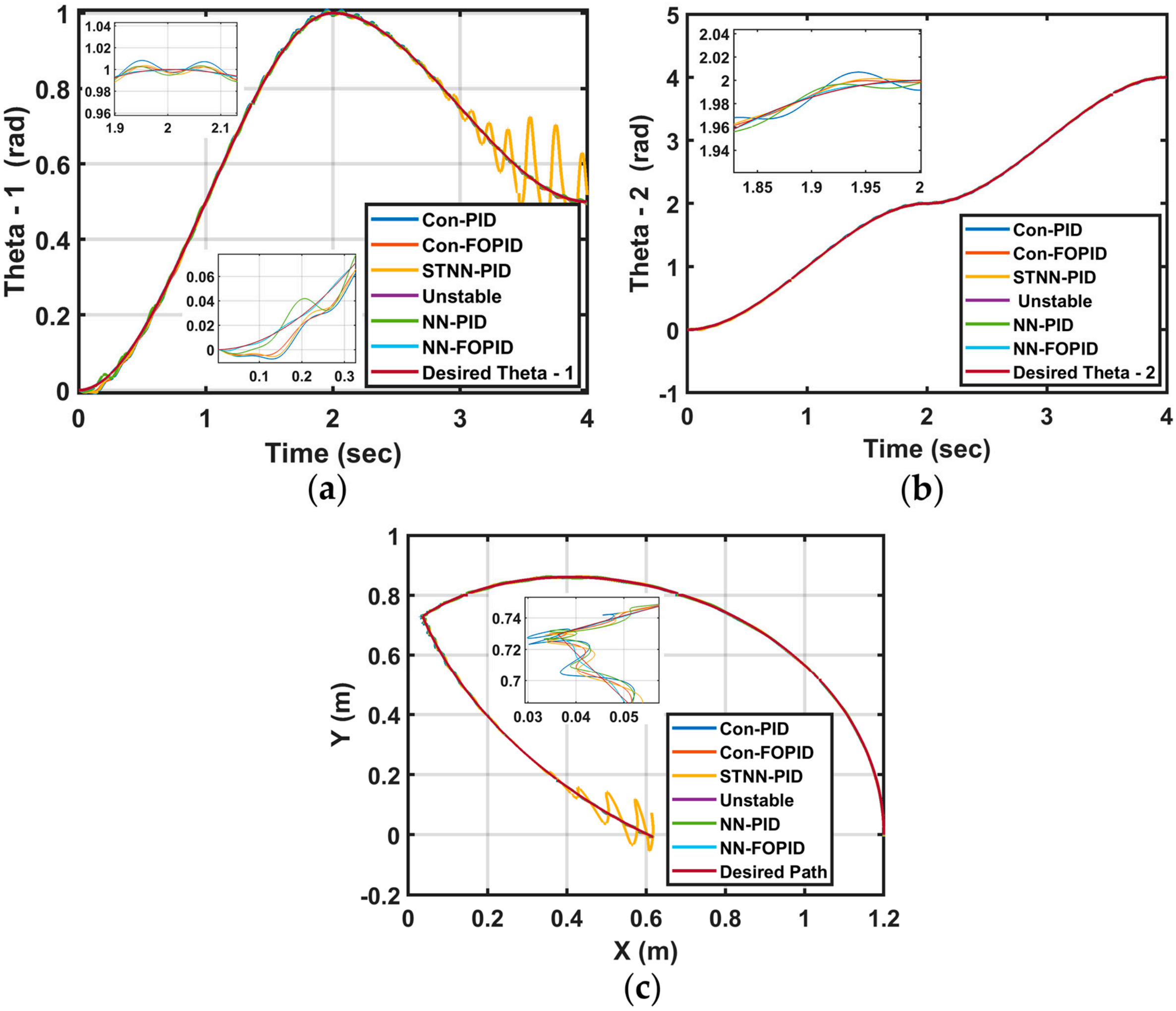

Figure 14. Desired and actual trajectories (**a**) for θ_1, (**b**) for θ_2, (**c**) Desired and actual paths of end-effector with disturbance term [sin (50t), sin (50t)] and the initial position (0, 0).

Table 7. The ITSE of the proposed controllers when increasing both masses of two links by 5% & Initial position (0.0, 0.0).

Controller	ITSE
Con-PID	1.183509×10^{-4}
Con-FOPID	0.691371×10^{-4}
STNN-PID	0.743505×10^{-4}
STNN-FOPID	Unstable
NN-PID	0.196180×10^{-4}
NN-FOPID	0.005068×10^{-4}

5.1.4. All Previous Tests Together

The effect of adding disturbance and parameter variation and changing the initial positions together on the proposed controllers is illustrated by the results in Table 8.

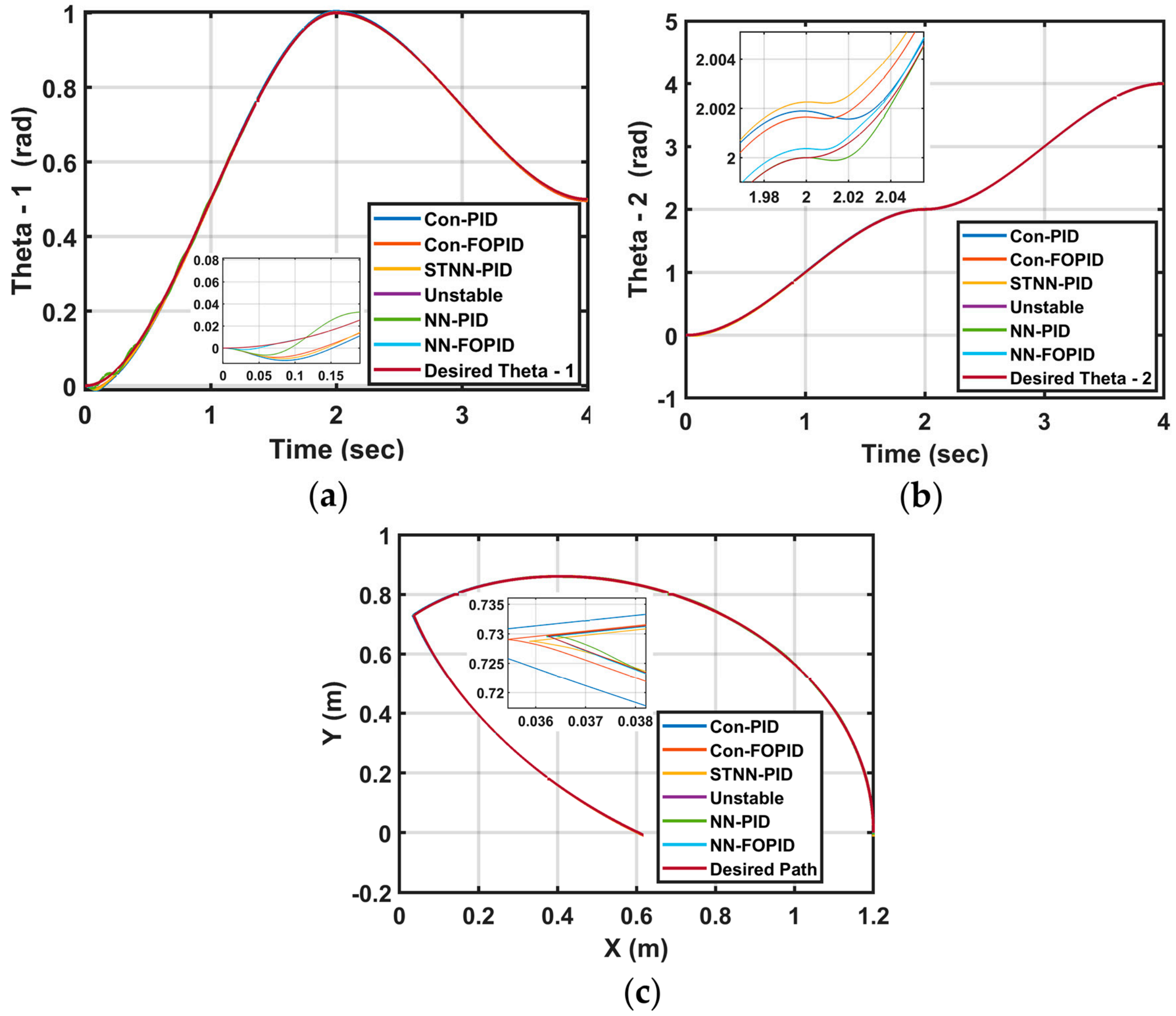

Figure 15. Desired and actual trajectories (**a**) for θ_1 , (**b**) for θ_2, and (**c**) Desired and actual paths of end-effector for 5% increasing in both masses and initial position (0, 0).

Table 8. The ITSE of the proposed controllers based on initial position (0.15, 0.15), adding disturbances sin (50t) for both links torques and increasing 5% in masses of both links.

Controller	ITSE
Con-PID	6.54278×10^{-4}
Con-FOPID	2.09812×10^{-4}
STNN-PID	Unstable
STNN-FOPID	Unstable
NN-PID	4.26462×10^{-4}
NN-FOPID	0.447529×10^{-4}

Figure 16 shows the trajectory tracking of theta1 and theta2 and the path tracked by the end-effector of the 2-LRRM and the T_1 and T_2 for disturbance, parameter variation as well as changing the initial positions for all controllers.

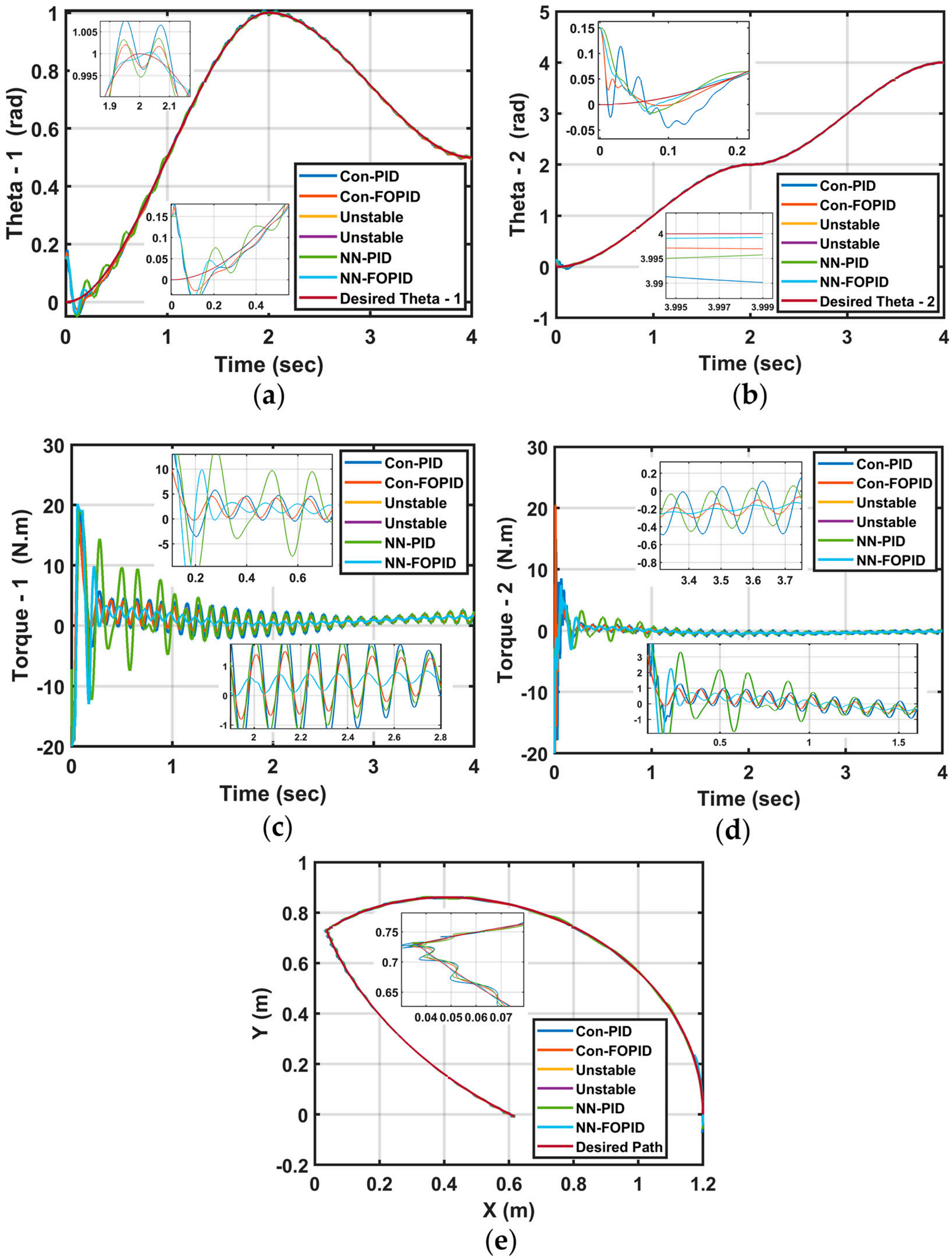

Figure 16. Desired and actual (**a**) for θ_1, (**b**) for θ_2, (**c**) performance of all controllers for Torque T_1 (**d**) performance of all controllers for Torque T_2 , and (**e**) Desired and actual paths of end-effector when using initial position (0.15, 0.15), adding disturbance sin (50t) for both links torques and increasing 5% in masses of both links.

From the results presented, it can be deduced that the fractional order PID controllers generally outperform the integer order PID controllers for parameter variation, disturbance addition, and initial position change and, therefore, more robust. The best controller among all proposed controllers is NN-FOPID because it passed all tests successfully and gives minimum ITSE in all tests and the best performance.

6. Conclusions

In this paper, six controllers are suggested for a 2-LRRM for trajectory tracking problems. These controllers combine the advantages of the PID controller and the neural network. Different control strategies of these combinations and other structures were proposed to strengthen the control capability. The fractional order of integration and differentiation used in the FOPID controller gives another dimension toward strengthening the capability of the hybrid controller. In addition, using a smart metaheuristic GTO algorithm for tuning controllers' parameters integrates the design components of the controller. Two controllers, NN-PID and NN-FOPID, are superior to others in their responses for the nominal system model because they reflect their stability and fast-tracking facility to the desired response path. Then different tests were done to check system capability and robustness in facing the following effects. The first test suggested a change in the initial position of the two links, the second test was an addition of disturbance to the torque of each link, and the third test changed in values of the system parameters where the masses of the two links are changed. The fourth test is taken the previous changes together synchronic. The results show that the performance index ITSE of the NN-FOPID controller for all tests from first to fourth are 0.24644×10^{-4}, 0.092827×10^{-4}, 0.005068×10^{-4}, 0.447529×10^{-4}, respectively. These values are the smallest values of ITSE obtained from testing all proposed controllers and indicate at the same time the smallest overshoot and settling time in system response. The NN-FOPID controller is more robust and capable than other controller structures. The future work will suggest using other optimization techniques instead of using the GTO algorithm, such as the Snake Optimization Algorithm (SOA), Squarril Search Algorithm (SSA), and Differential Search Algorithm (DSA) for tuning the gains of the proposed controllers. The best-proposed controller can be implemented practically using all necessary hardware components, such as a real robot manipulator and sensors, such as a camera.

Author Contributions: Conceptualization, M.J.M. and B.K.O.; Methodology, M.J.M., B.K.O., L.H.A., A.T.A. and I.A.H.; Software, M.J.M., B.K.O. and L.H.A.; Validation, A.T.A. and I.A.H.; Formal analysis, M.J.M., B.K.O., L.H.A., A.T.A. and I.A.H.; Investigation, A.T.A. and I.A.H.; Resources, M.J.M., B.K.O., L.H.A., A.T.A. and I.A.H.; Writing – original draft, M.J.M., B.K.O. and L.H.A.; Writing—review & editing, M.J.M., B.K.O., L.H.A., A.T.A. and I.A.H.; Visualization, L.H.A. and A.T.A.; Funding acquisition, I.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Norwegian University of Science and Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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. Special acknowledgement to Automated Systems & 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.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Raj, R.; Kos, A. A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives. *Appl. Sci.* **2022**, *12*, 6951. [CrossRef]
2. Chen, Y.; Zhao, T.; Dian, S.; Zeng, X.; Wang, H. Balance Adjustment of Power-Line Inspection Robot Using General Type-2 Fractional Order Fuzzy PID Controller. *Symmetry* **2020**, *12*, 479. [CrossRef]
3. Yang, C.; Huang, D.; He, W.; Cheng, L. Neural control of robot Manipulators with trajectory tracking constraints and input saturation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4231–4242. [CrossRef] [PubMed]
4. Rishag, H.T.; Hameed, S.M. Improvement the DFIG active power with variable speed wind using particle swarm optimization. *Diyala J. Eng. Sci.* **2016**, *9*, 12–22. [CrossRef]

5. Elsisi, M.; Mahmud, K.; Lehtonen, M.; Darwish, M.M.F. An Improved Neural Network Algorithm to Efficiently Track Various Trajectories of Robot Manipulator Arms. *IEEE Access* **2021**, *9*, 11911–11920. [CrossRef]
6. Manjeet, M.; Sathans, S. Fuzzy based Control of Two Link Robotic Manipulator and Comparative Analysis. In Proceedings of the 2013 International Conference on Communication Systems and Network Technologies, Washington, DC, USA, 6–8 April 2013; IEEE: Piscataway Township, NJ, USA, 2013; pp. 562–567.
7. Dachang, Z.; Baolin, D.; Puchen, Z.; Shouyan, C. Constant Force PID Control for Robotic Manipulator Based on Fuzzy Neural Network Algorithm. *Complexity* **2020**, *2020*, 3491845. [CrossRef]
8. Rahali, H.; Zeghlache, S.; Benyettou, L. Fault-Tolerant Control of Robot Manipulators Based on Adaptive Fuzzy Type-2 Backstepping in Attendance of Payload Variation. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 312–325. [CrossRef]
9. Mohan, V.; Chhabra, H.; Rani, A.; Singh, V. An expert 2DOF fractional order fuzzy PID controller for nonlinear systems. *Neural Comput. Appl.* **2018**, *31*, 4253–4270. [CrossRef]
10. Bingi, K.; Rajanarayan Prusty, B.; Pal Singh, A. A Review on Fractional-Order Modelling and Control of Robotic Manipulators. *Fractal Fract.* **2023**, *7*, 77. [CrossRef]
11. Sareena, A.; Rikesh, P. Application of PID Controller and Nonlinear Sliding Mode Control on Two Link Robotic Manipulator. *Int. J. Eng. Res. Technol.* **2019**, *8*, 504–508.
12. Elkhateeb, N.; Badr, R.I. Novel PID Tracking Controller for 2DOF Robotic Manipulator System based on Artificial Bee Colony Algorithm. *Electr. Control. Commun. Eng.* **2017**, *13*, 55–62. [CrossRef]
13. Ghamari, S.M.; Narm, H.G.; Mollaee, H. Fractional-order fuzzy PID controller design on buck converter with antlion optimization algorithm. *IET Control. Theory Appl.* **2022**, *16*, 340–352. [CrossRef]
14. Suid, M.H.; Ahmad, M.A. Optimal tuning of sigmoid PID controller using Nonlinear Sine Cosine Algorithm for the Automatic Voltage Regulator system. *ISA Trans.* **2022**, *128*, 265–286. [CrossRef]
15. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z. Position and Attitude Tracking of MAV Quadrotor Using SMC-Based Adaptive PID Controller. *Drones* **2022**, *6*, 263. [CrossRef]
16. Cao, G.; Zhao, X.; Ye, C.; Yu, S.; Li, B.; Jiang, C. Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot. *J. Mech. Sci. Technol.* **2022**, *36*, 2019–2029. [CrossRef]
17. Ghazali, M.R.B.; Ahmad, M.A.B.; Raja Ismail, R.M.T.B. Adaptive safe experimentation dynamics for data-driven neuroendocrine-PID control of MIMO systems. *IETE J. Res.* **2022**, *68*, 1611–1624. [CrossRef]
18. Bandyopadhyay, B.; Kamal, S. *Stabilization and Control of Fractional Order Systems: A Sliding Mode Approach*; Springer International Publishing: Cham, Switzerland, 2015; Volume 317. [CrossRef]
19. Almasri, E.; Uyguroglu, M.K. Modeling and Trajectory Planning Optimization for the Symmetrical Multiwheeled Omnidirectional Mobile Robot. *Symmetry* **2021**, *13*, 1033. [CrossRef]
20. Jothi, G.; Inbarani, H.H.; Azar, A.T.; Devi, K.R. Rough set theory with Jaya optimization for acute lymphoblastic leukemia classification. *Neural Comput. Appl.* **2019**, *31*, 5175–5194. [CrossRef]
21. Inbarani, H.H.; Banu, P.K.N.; Azar, A.T. Feature selection using swarm-based relative reduct technique for fetal heart rate. *Neural Comput. Appl.* **2014**, *25*, 793–806. [CrossRef]
22. Meghni, B.; Dib, D.; Azar, A.T.; Saadoun, A. Effective Supervisory Controller to Extend Optimal Energy Management in Hybrid Wind Turbine under Energy and Reliability Constraints. *Int. J. Dyn. Control.* **2018**, *6*, 369–383. [CrossRef]
23. Abdollahzadeh, B.; Soleimanian Gharehchopogh, F.; Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [CrossRef]
24. Ginidi, A.; Ghoneim, S.M.; Elsayed, A.; El-Sehiemy, R.; Shaheen, A.; El-Fergany, A. Gorilla Troops Optimizer for Electrically Based Single and Double-Diode Models of Solar Photovoltaic Systems. *Sustainability* **2021**, *13*, 9459. [CrossRef]
25. Vázquez-Aveledo, S.; Romero, R.J.; Montiel-González, M.; Cerezo, J. Control Strategy Based on Artificial Intelligence for a Double-Stage Absorption Heat Transformer. *Processes* **2023**, *11*, 1632. [CrossRef]
26. Mosaad, A.M.; Attia, M.A.; Abdelaziz, A.Y. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. *Ain Shams Eng. J.* **2019**, *10*, 755–767. [CrossRef]
27. Abdulameer, H.I.; Mohamed, M.J. Fractional Order Fuzzy PID Controller Design for 2-Link Rigid Robot Manipulator. *Int. J. Intell. Eng. Syst.* **2022**, *15*, 103–117.
28. Delavari, H.; Ghaderi, R.; Ranjbar, N.A.; HosseinNia, S.H.; Momani, S. Adaptive Fractional PID Controller for Robot Manipulator. *arXiv* **2010**, arXiv:1206.20272010.
29. Meghni, B.; Dib, D.; Azar, A.T.; Ghoudelbourk, S.; Saadoun, A. Robust Adaptive Supervisory Fractional order Controller For optimal Energy Management in Wind Turbine with Battery Storage. *Stud. Comput. Intell.* **2017**, *688*, 165–202.
30. Micev, M.; Calasan, M.; Oliva, D. Fractional Order PID Controller Design for an AVR System Using Chaotic Yellow Saddle Goatfish Algorithm. *Mathematics* **2020**, *8*, 1182. [CrossRef]
31. Alzabut, J.; Tyagi, S.; Abbas, S. Discrete Fractional? Order BAM Neural Networks with Leakage Delay: Existence and Stability Results. *Asian J. Control.* **2022**, *22*, 143–155. [CrossRef]
32. Thaiprayoon, C.; Sudsutad, W.; Alzabut, J.; Etemad, S.; Rezapour, S. On the qualitative analysis of the fractional boundary value problem describing thermostat control model via ψ-Hilfer fractional operator. *Adv. Differ. Equ.* **2021**, *2021*, 201. [CrossRef]
33. Alzabut, J.; Abdeljawad, T. On existence of a globally attractive periodic solution of impulsive delay logarithmic population model. *Appl. Math. Comput.* **2008**, *198*, 463–469. [CrossRef]

34. Wiora, J.; Wiora, A. Influence of Methods Approximating Fractional-Order Differentiation on the Output Signal Illustrated by Three Variants of Oustaloup Filter. *Symmetry* **2020**, *12*, 1898. [CrossRef]
35. Patel, R.; Kumar, V. Multilayer Neuro PID Controller based on back propagation algorithm. *Procedia Comput. Sci.* **2015**, *54*, 207–214. [CrossRef]
36. Chertovskikh, P.A.; Seredkin, A.V.; Gobyzov, O.A.; Styuf, A.S.; Pashkevich, M.G.; Tokarev, M.P. An Adaptive PID Controller with an Online Auto-Tuning by a Pretrained Neural Network. *J. Phys. Conf. Ser.* **2019**, *1359*, 012090. [CrossRef]
37. Roy, R.; Tu, Y.-P.; Sheu, L.-J.; Chieng, W.-H.; Tang, L.-C.; Ismail, H. Path Planning and Motion Control of Indoor Mobile Robot under Exploration-Based SLAM (e-SLAM). *Sensors* **2023**, *23*, 3606. [CrossRef] [PubMed]
38. Ali, O.A.M.; El-Zoghby, H.M.; Ghany, A.G.M.A. Maximizing the Generated Power from Hybrid Wind-Solar System Based on Fuzzy Self Tuning Single Neuron PID Controller. In Proceedings of the 2018 Twentieth International Middle East Power Systems Conference (MEPCON), Cairo, Egypt, 18–20 December 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 748–753. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.