Received: August 12, 2023. Revised: September 18, 2023. 659

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

FOPID Neural Network Controller Design for Nonlinear CSTR System

Zaid S. Mohsen[1]* **Mohamed Jasim Mohamed[1]**

[1]*Control and System Engineering Department, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: cse.20.20@grad.uotechnology.edu.iq

Abstract: In this paper, four control structures are provided for continuous ctirred-tank reactor (CSTR) system's fractional/integer order proportional integral derivative neural network controllers. The revised neural network weights and the controller's parameters are optimized using the optimization technique called ant colony optimization (ACO). The proposed controllers' resistance to changes in the initial state, outside disturbances, and parameter modifications is also tested. The fractional order proportional integral derivative neural network controllers provide the best assurance and also enhance the system's robustness to changes in the initial state, external disturbances, and parameter variations, according to the results of MATLAB code.The fractional order proportional integral derivative neural network controller1(FOPIDNNC1) is the best structure among all those with the minimum cost function equal to 0.011588 for the set-point variations, 0.015325 for uncertainty parameter, 0.018274 for disturbances rejection, and the best structure fractional order proportional integral derivative neural network controller3 (FOPIDNNC3) among all those with the minimum cost function equal to 0.008733 for tracking.

Keywords: Continuous stirred-tank reactor (CSTR), PID controller, Neural network (NN), Ant colony optimization (ACO), Fractional-order controller.

1. Introduction

The continuously stirred tank reactor (CSTR) model was selected. Through an irreversible exothermic reaction of first order A, the reactor changes chemical matter A into chemical matter [1]. CSTR, a chemical reactor with characteristics of non-linear dynamics, is frequently utilized in chemical and biological processes as well as water treatment. The CSTR might be a single tank or, more frequently, a group of tanks. It is frequently employed in the pharmaceutical industry ensuring consistent product quality for medium- and large-scale operations, and is particularly applicable to liquid-phase processes [2].

The most crucial component of chemical processes is typically the chemical reactor. They are the nucleus of the process that turns raw resources into finished goods. Chemical reactor design and operation are crucial to the overall performance of industrial activity as well as the protection of the environment. The modes of operation for reactors the flow may be batch or continuous. A batch flow reactor adds the reactants, thoroughly mixes the mixture, and then waits for the reaction to complete before releasing the mixture. In contrast, a continuous flow reactor's feed and discharge are both continuous. The extremely non-linear, complex, and usually unstable chemical reactions in these reactors are controlled to reduce the environmental impact of chemical waste. For instance, the instability could result in an abrupt temperature rise, a decrease in the rate of reaction, a safety breach, and eventually a shorter reactor life [3].

In the suggested technique [4], an adaptive structure for a proportional integral - derivative (PI-D) controller is created utilizing a multi-layer quantum neural network (QNN). Utilizing the particle swarm optimization (PSO) technique, the quantum neural network (QNN) is trained. The training procedure is being improved for accuracy and convergence speed. The simulation findings support the assertion that the suggested control scheme performs noticeably better under various

operating situations when compared to PID and a controller based on a perceptron neural network.

The multi-layer perceptron (MLP) neural network provides the PID controller settings, which are then online modified using the backpropagation approach. Through computer simulation and experimental examination, the adaptive neural network PID (ANNPID) controller is examined and contrasted with the traditional PID controller, the type 1 fuzzy logic controller (type 1 FLC), and the type 2 fuzzy logic controller (type 2 FLC). The findings collected demonstrate that the neural network PID performs better than the other controllers, and its parameters are simple to adjust to account for changes in the system parameters and the impacts of disturbances [5].

A PID with gains that are tuned for stabilizing temperature control in the CSTR plant and is based on particle swarm optimization (PSO). This is done by choosing parameters to get the optimum outcome under various circumstances. To make sure and confirm that the PSO-PID can regulate the plant robustly, two input disturbances are taken into account. The system's reaction is also contrasted with those of traditional techniques like Ziegler-Nichols tuning, genetic algorithms, and classical PSO. It is discovered that the PSO-PID gain patterns, as opposed to iterations, can enhance performance significantly more than the counterparts under consideration [6].

The elman neural network (ENN) controller for concentration control of the CSTR stands out among the most crucial components of chemical industries as a nonlinear system. A benefit for improved convergence and efficient tracking of system output was added by the network's capacity to store the memory of the previous iteration through recurrent linkages. Through the use of performance measures, the effectiveness of the proposed elman NN controller is confirmed. [7].

For regulating CSTR, a hybrid optimum control approach called FOPID control is presented. It combines elite opposition-based learning with a chaotic state of matter search. PID controllers come in several forms, including fractional order PID. With the application of fractional calculus, it achieves rigorous adoption for substantially closed-loop system stability while also being more adaptable and responsive. In comparison to alternative metaheuristic algorithms, the results demonstrate that the CSMSEOBL-adjusted FOPID controller offers superior and optimum performance. The implementation of chaotic state of matter search with elite opposition-based learning (CSMSEOBL) on a non-linear control problem leads to faster convergence, greater exploration and exploitation capabilities, and exhibits promising results in terms of overshoot, settling time, and integral time square error (ITAE) for performance optimization [8].

The issue, depending on how the systems work, is the challenges that numerous chemical companies face when attempting to control chemical reactors like the CSTR system. Due to the process's high degree of complexity, instability, and non-linearity, the state and output variables are extremely sensitive to changes in input, which is what leads to control problems. At different operating points, the dynamic characteristics may also provide a shifting gain indication. These qualities make a process difficult to control and call for the employment of more sophisticated methods [3]. To address this issue, proposed a fractional order PID neural network controller.

In this work, four structures of the fractional order PID neural network (FOPIDNN) controller will be combined in order to increase the controller's robustness and produce a more powerful and adaptable design compared with the conventional PID controller. The traditional tuning including self-tuning, auto-tuning, and Ziegler-Nichols can also be incorrect, oscillatory, time-consuming, and unable to account for the system's nonlinearities or uncertainties. Therefore to get the optimal values for the controller's parameters, the ant colony optimization (ACO) method is used .

The following are the primary contributions of the suggested controllers:

1- For the same task,The creation of four structures for fractional order PID neural network controllers .
2- By altering the beginning condition and non-demonstrable parameter fluctuations, the proposed controllers' robustness is shown.in[9].
3- The results of the proposed FOPIDNN controllers are superior to or converge to the best values achieved from the existing controllers when compared with [4].

This work's remaining sections are organized as follows. Section 2, continuous ctirred-tank reactor (CSTR) dynamical model is explained. Section 3, the proposed ant colony optimization (ACO) is outlined. Section 4, The suggested FOPIDNN controllers are illustrated. Section 5,The simulation's results are presented. Section 6,The final section is introduces the main conclusion.

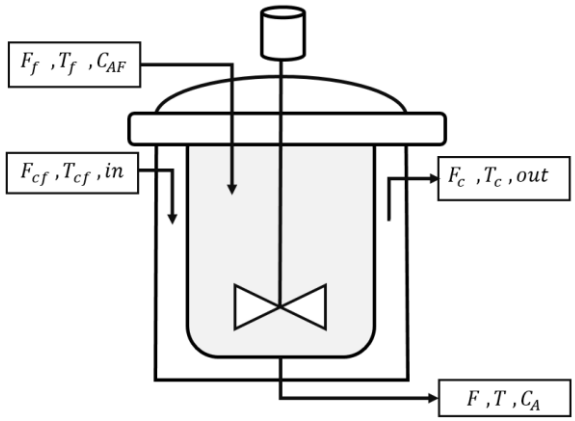

Figure. 1 Structure of CSTR

Table 1. Variables and parameters of the CSTR

symbol	Description
V	Reactor volume
F	Feed rate
k_0	Time constant
R	Perfect gas constant
E	Activation energy
ΔH	Enthalpy of reaction
T_C	Temperature of the double jacket
T	Reaction temperature
T_{in}	Supply temperature
C_{in}	Feeding concentration
c_p	Specific heat capacity at constant pressure
C	Concentration of the component
ρ	Density of solution
U_A	Thermal transfer coefficient

2. Mathematical model of CSTR

The continuous ctirred-tank reactor (CSTR) scheme is shown in Fig. 1. It consists of a cooling jacket-enclosed tank reactor. Before feeding the reactor, the reactants are combined at a specific initial concentration of (C_{A0}) and an initial temperature of (T_0) The reactor is then filled with reactants and catalysts at a constant flow rate (F). To increase the effectiveness of the chemical reaction, the stirrer is utilized to mix the reactants. The reactor experiences an exothermic, irreversible reaction denoted by the formula $A \longrightarrow B$. Additionally, a cooling stream with a beginning temperature of (T_{j0}) and a constant flow rate of (F_j) is employed to pass through the cooling jacket to keep the reactor at the correct operating temperature. Thus, the chemical reaction's head can be absorbed by the cooling stream [10].

Creation of CSTR deterministic dynamic model, and a description of variables and parameters of the CSTR in Table 1.

Step 1: The following assumptions were made for creating a deterministic model of the CSTR[11]:

- CSTR have complete mixing to prevent spatial gradients in the mixture's velocity, temperature, concentration, and other parameters.
- The amount of shaft work generated by the stirring action is little.
- The fact that there is no pressure drop in the CSTR suggests that they operate at constant pressure.
- The CSTR don't experience any pressure drops.
- The volume of the CSTR varies.
- The specific heat capacities Cp and densities of the CSTR are constants.

Step2: The production rate determines as follows: the concentration is given by:

$$C = \frac{n}{V} \tag{1}$$

where V is the volume that the substance is present in, and n is the number of moles contained in the substance's molecules. The rate of production is given by

$$r = \frac{1}{V}\frac{dn}{dt} = \frac{dC}{dt} \equiv \frac{dn}{dt} = V\frac{dC}{dt} \tag{2}$$

The pace of production increases if we have a reactant that needs to be transformed into a product.

$$r = -\frac{dC}{dt} = kC \tag{3}$$

However, the Van't Hoff equation demonstrates the temperature dependence of the reaction rate k, which is represented as

$$\frac{d(ln\,k)}{dT} = \frac{\triangle Hr}{RT^2} \equiv d(ln\,k) = \frac{\Delta Hr}{RT^2}\,dT \tag{4}$$

Eq. (1) is solved for k by integrating both sides concerning T. This results in

$$\int_{k_0}^{k} d(ln\,s) = \int_{T_{mean}}^{T} \frac{\Delta Hr}{Ry^2}\,dy \tag{5}$$

$$[ln\ s]_{k_0}^{k} = -\frac{\Delta Hr}{R}[y^{-1}]_{T_{mean}}^{T} \tag{6}$$

So,

$$ln\,k - ln\,k_0 = -\frac{\Delta Hr}{R}(T^{-1} - T_{mean}\ ^{-1}) \tag{7}$$

$$ln\left(\frac{k}{k_0}\right) = -\frac{\Delta Hr}{R}(T^{-1} - T_{mean}\ ^{-1}), \tag{8}$$

$$\frac{k}{k_0} = e^{-\frac{\Delta H_r}{R}\left(T^{-1}-T_{\text{mean}}-1\right)} \quad (9)$$

$$k = k_0 e^{-\frac{\Delta H_r}{R}\left(\frac{1}{T}-\frac{1}{T_{\text{mean}}}\right)} \quad (10)$$

If we let $\triangle Hr = E$ and $\frac{1}{T_R} = \frac{1}{T} - \frac{1}{T_{\text{mean}}}$, we get

$$k = k_0 e^{-\frac{E}{RT_R}} \quad (11)$$

The production rate for The system studied is CSTR A → B given by :

$$-r_A = kC_A = k_0 \cdot exp\left(\frac{-E}{RT_R}\right) C_A \quad (12)$$

Step 3: The following is the generic equation for a mass balance in any system:

$$(Rate\ Generation) = Vk_0\ exp\left(-\frac{E}{RT}\right) C_A \quad (13)$$

$$(\ Flow\ In\ ) - (\ Flow\ Out) = F(C_{AF} - C_A) \quad (14)$$

$$(\ Rate\ Accumulation\ ) = V\frac{dC_A}{dt} \quad (15)$$

$$V\frac{dC_A}{dt} = F(C_{AF} - C_A) - Vk_0\ exp\left(-\frac{E}{RT}\right) C_A \quad (16)$$

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{AF} - C_A) - k_0\ exp\left(-\frac{E}{RT}\right) C_A \quad (17)$$

Step 4: The following is the generic equation for an energy balance in any system:

$$Rate\ Heat\ Generation = (-\Delta H)Vk_0\ exp\left(-\frac{E}{RT}\right) C_A \quad (18)$$

$$(\ Heat\ In\ ) - (\ Heat\ Out) = \rho C_p F\left(T_f - T\right) \quad (19)$$

$$(\ Rate\ Energy\ Accumulation\ ) = V\rho C_\rho \frac{dT}{dt} \quad (20)$$

$$(\ Heat\ Transfer\ ) = hA(T - T_C) \quad (21)$$

$$\frac{dT}{dt} = \frac{F}{V}(T_F - T) - \frac{hA}{V\rho C_P}(T - T_C) - \frac{(\Delta H)}{\rho C_P} K_0 \exp\left(-\frac{E}{RT}\right) C_A \quad (22)$$

The dynamics of the CSTR process under consideration are expressed in Eqs. (18) and (24) [12].

Step 5: Further derivation of the following straightforward normalized equations to represent the dynamic:

Table 2. The parameters of CSTR

Parameters	Nominal value
Damokhler number (Da)	0.072
activated energy (φ)	20
the heat of reaction (B)	8
heat transfer (δ)	0.3

$$x_1 = \frac{(C_{AF}-C_A)}{C_{AF}}, x_2 = \frac{(T_F-T)}{T_F}, \varphi = \frac{E}{RT}, t' = t\frac{F}{V}$$
$$\delta = \frac{hA}{F\rho C_p}, D_a = \frac{VK_0 e^{-\varphi}}{\text{F}}, B = \frac{(-\Delta H)\varphi C_{AF}}{\rho C_p T_F},$$
$$u = \frac{(T_c-T_F)}{T_F}\ \varphi$$
$$\dot{x}_1 = -x_1 + D_a(1-x_1)exp\left(\frac{x_2}{1+x_2/\varphi}\right) + d \quad (23)$$

$$\dot{x}_2 = -x_2(1-\delta) + D_a\ B(1-x_1)exp\left(\frac{x_2}{1+x_2/\varphi}\right) + \delta\ u \quad (24)$$

where x_1 and x_2 are the dimensions concentration C and temperature of reactor T, respectively; $y = x_1$ and u the outlet temperature of the product and the temperature of cooling water, respectively; d disturbances.

3. Ant colony optimization

To make a neural network system accomplish the required tasks, the network must be trained. In supervised learning, a feedforward neural network is trained using one of the most well-known training techniques, backpropagation. Updating the weights of the network needs the differentiation of the error signal. Such a learning strategy also requires input and output data sets, both of which may be difficult for some systems to acquire. Contrarily, optimization methods have lately been used to train neural networks and have demonstrated promising results in handling a variety of engineering challenges. These algorithms don't need error differentiation like the backpropagation approach does because they are based on minimizing a cost function by adjusting specific design parameters. They thus serve as global optimizers and are less susceptible to falling into local minimums than local search techniques. Because of these benefits, the current study's neural network training uses the ant colony optimization approach [13].

The ant-inspired algorithm was a valid approach to optimization issues that Dorigo and colleagues introduced in the 1990s. When they come to food on their way back, ants release a pheromone. A path with more pheromones, which indicates the quickest way between food and the nest, is finally chosen by other

ants by following the pheromones released by the ants. The more pheromones there are along a path, the more ants have chosen that way to travel along. An ant uses a probabilistic method to choose a path [14].

3.1 The ACO algorithm

1- Set up the variables .
- Define the archive's size (k). .
- The maximum number of evaluations $(MaxIt)$
- Intensification factor (q)
- Deviation-distance ratio (ξ)
- Population size (m)

2- Initialize the archive.
- A uniform distribution is used to create ants at random in the range $[X_{min}\ ,\ X_{max}]$ and save them in the archive.
- Each ant's fitness value is determined and recorded in the archive.
- According on their fitness score, the ants in the archive are categorized as little, medium, and giant.

$$cost\ S_l\ = cost\ function(S_l)$$

3- Determine the weight W_l

$$w_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \quad (25)$$

The weight of the solution increases with decreasing (q), increasing the likelihood that it will be chosen. On the other hand, the odds of being chosen are more equal the more uniform the solution weight is. The parameter is therefore primarily utilized to control both local and global optimization.

4- Calculate the probability P_l.

$$p_l = \frac{w_l}{\sum_{r=1}^{k} w_r} \quad (26)$$

5- Gaussian function sampling.
- The first step: The population's ants choose a Gaussian function $g_l^i(x)$ from the Gaussian kernel function based on probability (P_l), and they then confirm a guidance solution (S_l)
- The second step: The Gaussian function $g_l^i(x)$ samples each location vector of the guidance solution (S_l).

$$g_l^i(x) = \frac{1}{\sigma_l^i\sqrt{2\pi}} e^{-\frac{\left(x-\mu_l^i\right)^2}{2\sigma_l^{i2}}} \quad (27)$$

Where: the mean μ_l^i and standard deviation σ_l^i for each solution, calculated by:

$$\mu_l^i = s_l^i \quad (28)$$

$$\sigma_l^i = \xi \sum_{j=1}^{k} \frac{\left|s_j^i - s_l^i\right|}{k-1} \quad (29)$$

6- Make a fitness value calculation. The freshly created ants are added to the file when the fitness value of the population formed by sampling using a Gaussian distribution is computed.

7- Pheromone update
- The fitness value determines the order of all the ants in the archive.
- The first ants are in the archive (k).
- The last ants (m) re removed.
- Recorded is the best ant

8- Procedure for iteration. Steps 4 through Step 7 will be repeated if the criterion for iteration termination is not satisfied.

4. Fractional order PID neural network

Three components make up the PID controller's feedback control loop proportional, integral, and derivative modes [15]. The proportional control mode normally operates the controller. If the controller gain is set to a very high value, the control loop will start to oscillate and become unstable [16]. On the other hand, if the gain is too low, it will be difficult to react to disturbances or changes in the set point. the integrated control mode continuously boosts or reduces the controller's output. The derivative mode is a PID controller's third control mode. With this form of controller, trial-and-error correction is more challenging [17].

The fractional order controller (FOPID) same as the conventional PID controller, however, derivative order (μ) and integral order (λ) are fractional. Because of the flexible order of fractional calculus, a novel and effective performance extension for FOPID can be offered [18].If $\lambda = \mu = 0$ The generation of a standard proportional P mode .If $\lambda = \mu = 1$The generation of a standard integer-order PID controller. If $\lambda = 0\ , \mu = 1$ The generation of a standard derivative-order PID controller. If $\lambda = 1, \mu = 1$ The generation of a standard integer-order PID controller [19].

Fractional differ integral has several mathematical definitions. These definitions often produce approximations rather than exact results. The most crucial definition in this mathematical list :

Riemann–Liouville (RL) [20]

$$aD_t^{\alpha} f(t) = \frac{1}{r(n-\alpha)} \frac{d^n}{d^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (30)$$

For $(\mathrm{n} - 1 < \alpha < \mathrm{n})$

And $\Gamma(x)$ is the well-known Euler's Gamma function.

$$g(t, x, {}_aD_t^{\alpha 1} x, {}_aD_t^{\alpha 2} x \ldots) = 0, \ \alpha_{\mathrm{k}} \in \mathrm{R}^{+} \quad (31)$$

Definition of caputo (C)

$$\begin{matrix} {}_a^C D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{\mathrm{a}}^t \frac{f^n(\tau) d\tau}{(t-\tau)^{\alpha-n+1}} \\ n - 1 < \alpha < n \end{matrix} \quad (32)$$

Defining grunwald-letnikov (GL) [21].

$${}_a^{GL} D_t^{\alpha} f(t) = lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (33)$$

The FOPID controller's differential equation is described by [18]:

$$U(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) \quad (34)$$

Given that they are capable of learning and providing an effective function approximation, neural networks have been employed in modelling [22]. A variety of nonlinear systems have been controlled using them more recently. There are two different classification algorithms for control systems built using neural networks. Training the network to carry out the necessary activities is required for both designs [23].

To fine-tune the control parameters of the first technique, a neural network is employed. Due to the network connections' random initialization weights, one characteristic of classic neural networks is that learning takes a long period. This characteristic prevents the algorithm from achieving a quick response from the control system and prevents it from performing stationary.

There are several advantages to using a neural network in conjunction with a FOPID controller:

1. Improved control performance: Neural networks can learn complex nonlinear relationships between the process variables and the control output, which can lead to better control performance compared to FOPID controllers.
2. Robustness: Neural networks can be trained to handle disturbances, uncertainties, and nonlinearities in the controlled system, which can improve the robustness of the control system.
3. Adaptability: Neural networks can adapt to changes in the controlled system over time, which can improve the overall performance of the control system.
4. Reduced tuning effort: FOPID controllers require manual tuning of the controller parameters, which can be time-consuming and require expert knowledge. A neural network can automate this tuning process, reducing the tuning effort and improving the overall control performance.
5. Scalability: Neural networks can be used in control systems with multiple inputs and outputs [24]. The control of CSTR system is proposed to be controlled by four structures of FOPIDNN controllers.

4.1 Fractional order proportional integral derivative neural network controller1 (FOPIDNNC1)

In practical control systems, it is occasionally challenging or even impossible to achieve flawless tracking of the reference signal due to a number of factors including system nonlinearities, disturbances, and sensor noise. It is feasible to decrease these effects and boost tracking performance by utilizing reference compensation.

- **Layer1 :**

$$I_i(n) = [e(t), e(t-1), e(t-2)] \quad (35)$$

- **Layer2 :**

$$H_j(n) = f_j\left(\sum_i W_{ji} \cdot I_i(n) + b_j\right) \quad (36)$$

- **Layer3 :**

$$O_k(n) = \sum_j V_{kj} \cdot H_j(n) + b_k \quad \mathrm{n} \quad (37)$$

- **The control signal becomes:**

$$u_{pid} = k_P e(t) + k_i \int \ \mathrm{e}(t) dt + k_d \dot{e}(t) \quad (38)$$

$$u = u_{pid} + k_p O_1 + k_d O_2 + k_i O_3 \quad (39)$$

Where, $I_i, H_j, O_k, W_{ji}, V_{kj}, f_j, b_j, b_k$: layer1, layer2, layer3, the weight between layers (1&2), the weight between layers (2&3), the hyperbolic tangent activation function, and the bias weight of the layer(2&3), respectively[25]. A FOPIDNNC1 is shown in Fig. 2.

4.2 Fractional order proportional integral derivative neural network controller2 (FOPIDNNC2)

The advantage of the dynamic-feedback neural network is that it successfully cuts down on the network's input dimension and, consequently, training time.

- **Layer1 :**

$$I_i(n) = e(t) \quad (40)$$

- **Layer2 :**

$$P(n) = net_1(n) = k_p(e(n)) \quad (41)$$

$$I(n) = net_2(n) = k_i \cdot st \cdot e(n) + I(n-1) \quad (42)$$

$$D(n) = net_3(n) = k_d\big(e(n) - e(n-1)\big)/st \quad (43)$$

- **Layer3 :**

$$H_j(n) = f_j\left(\sum_i W_{ji} \cdot net_i(n) + H_j(n-1) + h_k(n-1)\right) \quad (44)$$

$$f_j(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (45)$$

- **Layer4 :**

$$h_k(n) = g_k\left(\sum_j V_{kj} \cdot H_j(n) + h_k(n-1)\right) \quad (46)$$

$$g_k(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (47)$$

- **Layer5 :**

$$u(n) = O_l(n) = \sum_k Z_{lk}\, h_k(n) \quad (48)$$

Where, I_i , $FOPID, H_j$ and h_k, O_l , W_{ji} , V_{kj} , Z_{lk}, f_j , g_k , and st:layer1, layer2, layer3, layer4, layer5, the weight between layer(2&3), the weight between layer(3&4), the weight between layer(4&5), and the sigmoid activation function, step size, respectively[26]. A FOPIDNNC2 is shown in Fig. 3.

4.3 Fractional order proportional integral derivative neural network controller3 (FOPIDNNC3)

The Elman neural network comprises many context nodes, it thus exhibits certain dynamic properties. a new, updated model-based dynamic system identification scheme is shown along with an improved form of the Elman network model. The input layer, hidden layer, context layer, and output layer make up the majority of the Elman neural network's four layers of structure.

- **Layer1 :**

$$I_i(n) = e(t) \quad (49)$$

- **Layer2 :**

$$P(n) = net_1(n) = k_p(e(n)) \quad (50)$$

$$I(n) = net_2(n) = k_i \cdot st \cdot e(n) + I(n-1) \quad (51)$$

$$D(n) = net_3(n) = k_d\big(e(n) - e(n-1)\big)/st \quad (52)$$

- **Layer3 :**

$$H_j(n) = f_j\left(\sum_i W_{ji} \cdot net_i(n) + \sum_c Q_{jc}\, C_c(n)\right) \quad (53)$$

$$f_j(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (54)$$

- **Layer4 :**

$$h_k(n) = g_k\left(\sum_j V_{kj} \cdot H_j(n) + \sum_c R_{kc}\, C_c(n)\right) \quad (55)$$

$$g_k(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (56)$$

- **Layer5 :**

$$u(n) = O_l(n) = \sum_k Z_{lk} \cdot h_k(n) \quad (57)$$

- **Layer6 :**

$$C_c(n) = H_j(n-1) + \alpha C_c(n-1) \quad (58)$$

Where, I_i , $FOPID, H_j$, h_k , O_l , C_c, W_{ji}, V_{kj}, Z_{lk}, Q_{jc}, R_{kc} , f_j , g_k , α , and st: layer1, layer2, layer3, layer4, layer5, layer6, the weight between layers (2&3), the weight between layers (3&4), the weight between layers (4&5), the weight between layer(3&6), the weight between layer(4&6), the sigmoid activation function, the feedback gain to the self-connection of layer6,and step size , respectively [27]. FOPIDNNC4 is shown in Fig. 6.

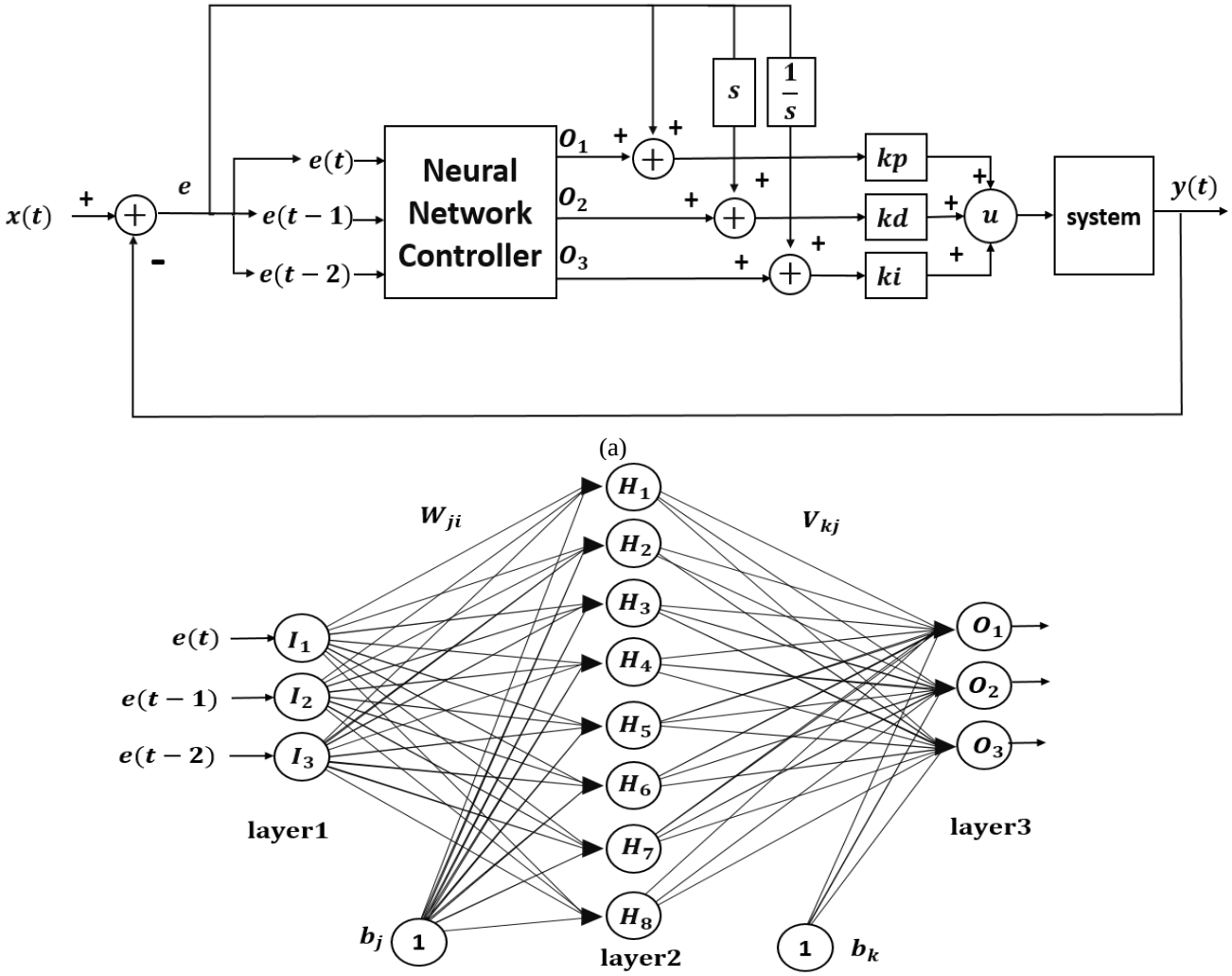

Figure. 2: (a) FOPIDNNC1, (b) neural network controller

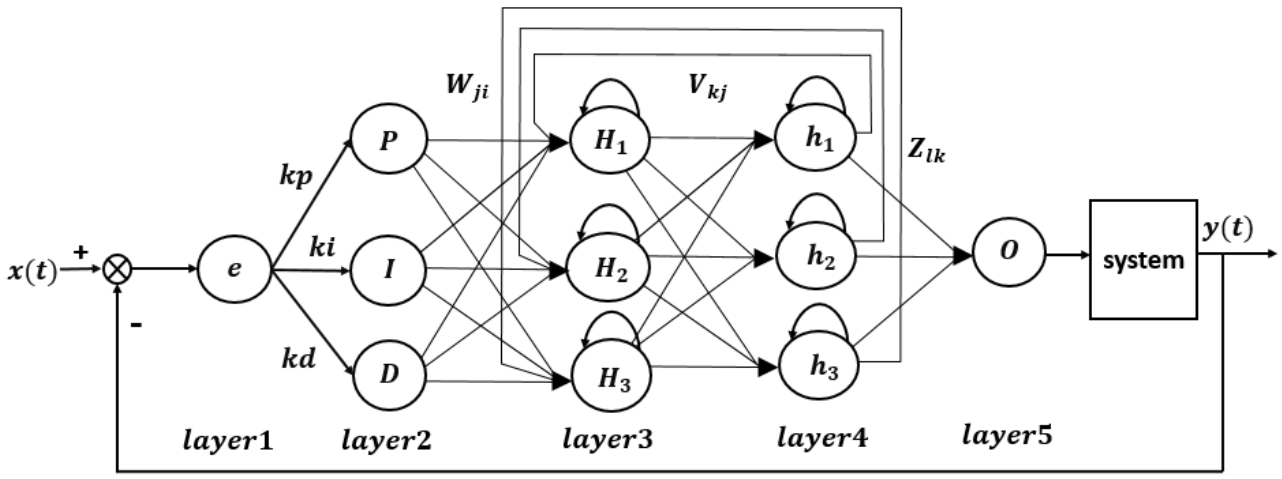

Figure. 3 FOPIDNNC2

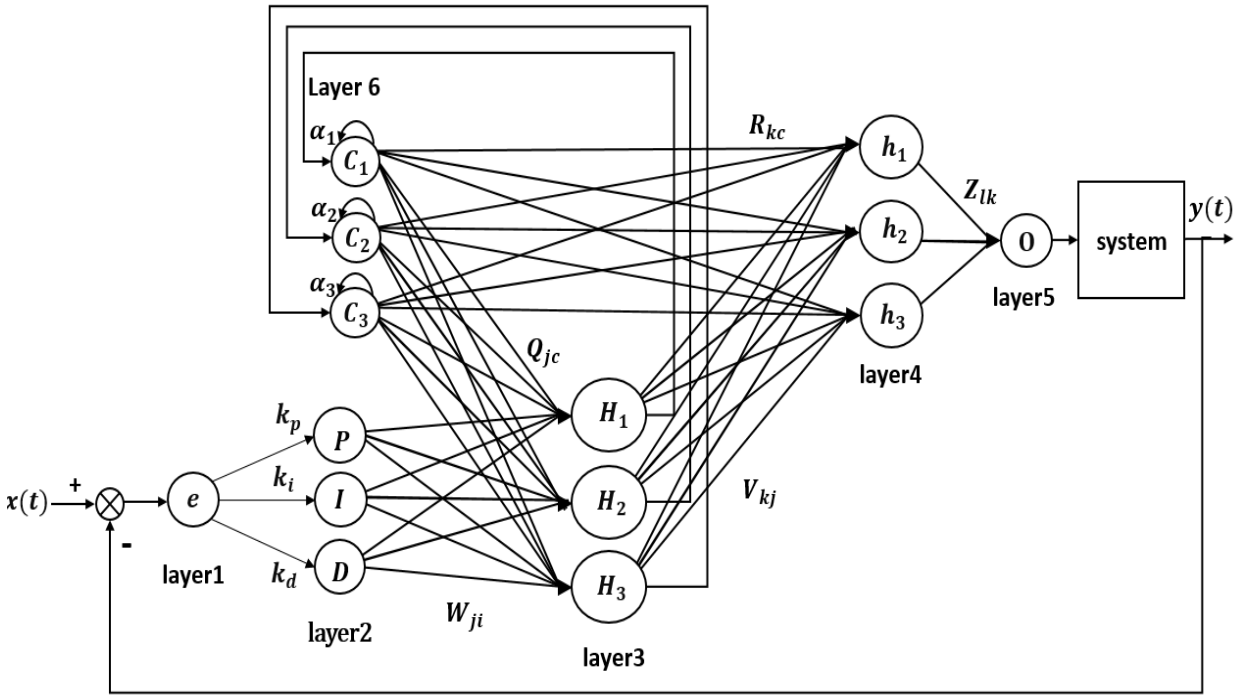

Figure. 4 FOPIDNNC3

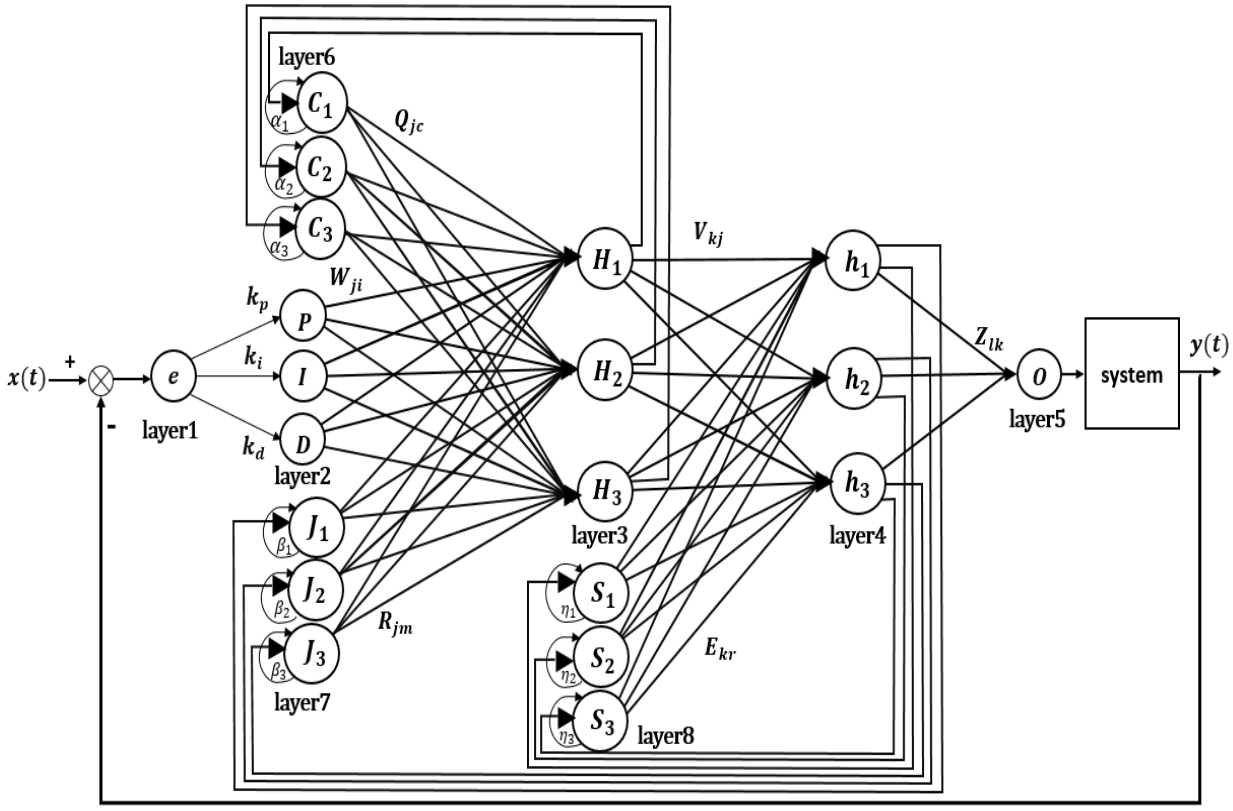

Figure. 5 FOPIDNNC4

4.4 Fractional order proportional integral derivative neural network controller4 (FOPIDNNC4)

The Jordan-Elman recurrent neural network (RNN) is including a feedback connection from the network's output back into the hidden layer. This feedback connection speeds up learning by enabling the network to handle sequential data more skillfully and to keep a short-term recall of prior inputs

- **Layer1 :**

$$I_i(n) = e(t) \quad (59)$$

- **Layer2 :**

$$P(n) = net_1(n) = k_p(e(n)) \quad (60)$$

$$I(n) = net_2(n) = k_i \cdot st \cdot e(n) + I(n-1) \quad (61)$$

$$D(n) = net_3(n) = k_d\big(e(n) - e(n-1)\big)/st \quad (62)$$

- **Layer3 :**

$$H(n) = f_j\left(\sum_i W_{ji} \cdot net_i(n) + \sum_c Q_{jc}\, C_c(n) + \sum_m R_{jm}\, J_m(n)\right) \quad (63)$$

$$f_j(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (64)$$

- **Layer4 :**

$$h_k(n) = g_k\left(\sum_j V_{kj} \cdot H_j(n) + \sum_r E_{kr}\, S_r(n)\right) \quad (65)$$

$$g_k(x) = \frac{6}{1+e^{(-x)}} - 3 \quad (66)$$

- **Layer5 :**

$$u(n) = O_l(n) = \sum_k Z_{lk} \cdot h_k(n) \quad (67)$$

- **Layer6** :

$$C_c(n) = H_j(n-1) + \alpha C_c^H(n-1) \quad (68)$$

- **Layer7** :

$$J_m(n) = h_k(n-1) + \beta J_m(n-1) \quad (69)$$

- **Layer8** :

$$S_r(n) = h_k(n-1) + \eta S_r(n-1) \quad (70)$$

Where, $I_i, FOPID, H_j, h_k, O_l, C_c, J_m, S_r, W_{ji}, V_{kj}, Z_{lk}, Q_{jc}, R_{kc}, E_{kr}, f_j, g_k, \alpha, \beta, \eta, and\ st$: layer1, layer2, layer3, layer4, layer5, layer6, layer7, layer8, the weight between layers (2&3), the weight between layer(3&4), the weight between layer(4&5), the weight between layer(3&6), the weight between layer(3&7), the weight between layer(4&8), the sigmoid activation function, the feedback gain to the self-connection of layer (6&7&8), and step size respectively [28]. A FOPPIDNNC4 is shown in Fig. 5.

Table 3. Weight of the cost function

weights	value
r_1	0.999999
r_2	0.000001

Table 4. Parameters of ACOR optimization

parameters	value
Population size (m)	80
the size of the archive (k).	100
Intensification Factor (q)	0.5
Deviation-Distance Ratio (ξ)	1

Table 5. Parameters of controllers

parameters	value
Number of parameters	FOPIDNNC1=64
	FOPIDNNC2=26
	FOPIDNNC3=38
	FOPIDNNC4=62
kp, kd, ki	$[-250,250]$
γ	$[-2,0]$
μ	$[0,2]$
The weight of the neural network	$[-1,1]$

Table 6. Comparing the values of the cost function

Controller	Cost function
FOPIDNNC1	0.011588
FOPIDNNC2	0.013524
FOPIDNNC3	0.012883
FOPIDNNC4	0.013542

5. Simulation and result

The proposed controllers are simulated using MATLAB code , and step size $= 0.001$ The commands of the proposed FOPIDNN controller can be modified to meet design specifications and give the user a range of control limit options. The test also makes use of the cost function (J) performance index.

$$J = \int_0^t (r_1 \times (e)^2 + r_2 \times (u)^2)dt \quad (71)$$

The weights of the cost function and the evolutionary algorithmic parameters are displayed in Table 3. In terms of the starting point, the parameters of ant colony optimization (ACO) are in Table 4. all algorithms have the same parameters and the controllers parameter in Table 5.

The CSTR system's initial condition $x_1 = 0.144$ and output desired $x_{d1} = 0.445$ and simulation time is set to $10\ sec$. In Figs. 6 and 7 display the simulation results for controllers. the

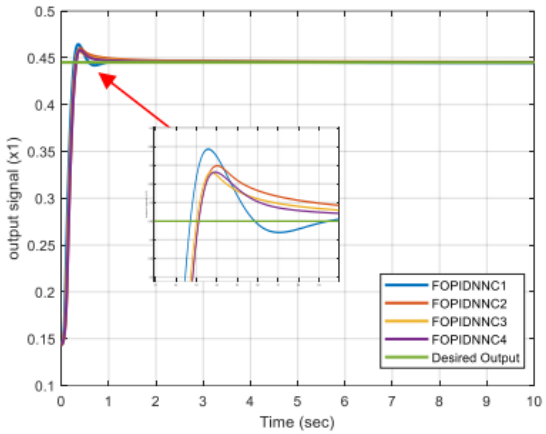

Figure. 6 Time response for set-point of CSTR

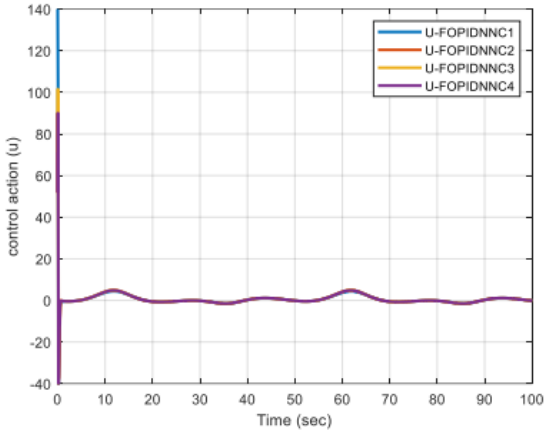

Figure. 9 Control action for Trajectory tracking of CSTR

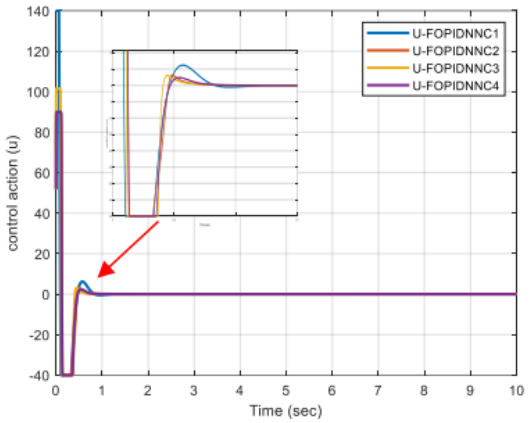

Figure. 7 Control action for set-point of CSTR

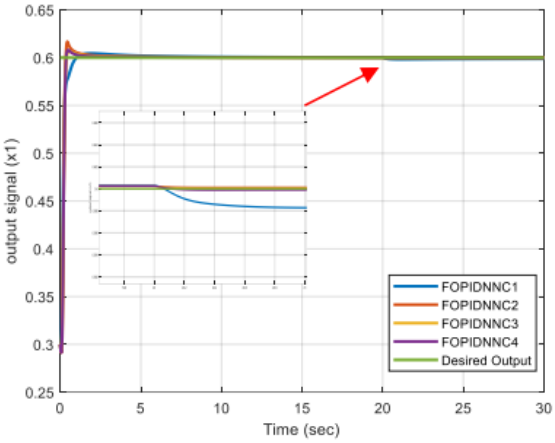

Figure. 10 Time response for uncertainty of CSTR

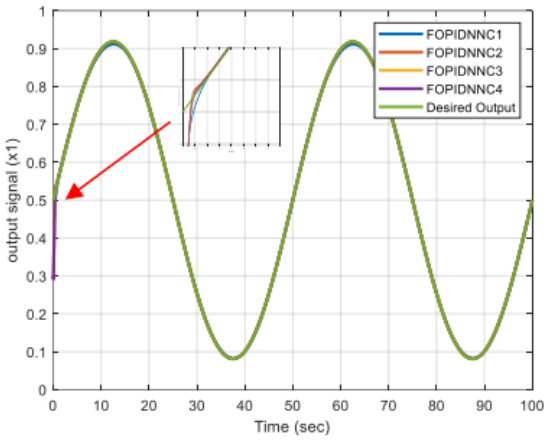

Figure. 8 Time response for trajectory tracking of CSTR

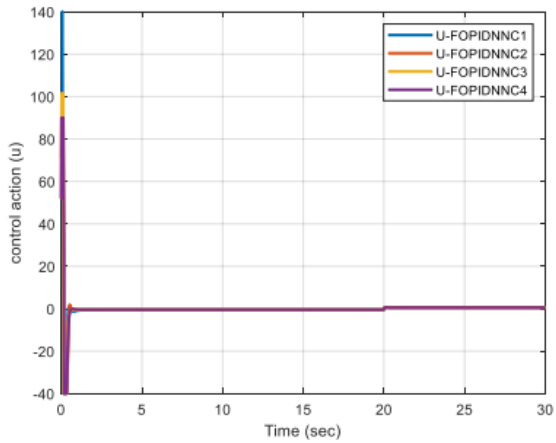

Figure. 11 Control action for uncertainty of CSTR

Table 7. Comparing the values of the cost function

Controller	Cost function
FOPIDNNC1	0.008961
FOPIDNNC2	0.009127
FOPIDNNC3	0.008733
FOPIDNNC4	0.009177

Table 8. Comparing the values of the cost function

Controller	Cost function
FOPIDNNC1	0.015325
FOPIDNNC2	0.018230
FOPIDNNC3	0.017071
FOPIDNNC4	0.018267

Table 9. Comparing the values of the cost function

Controller	Cost function
FOPIDNNC1	0.018274
FOPIDNNC2	0.018965
FOPIDNNC3	0.018516
FOPIDNNC4	0.018946

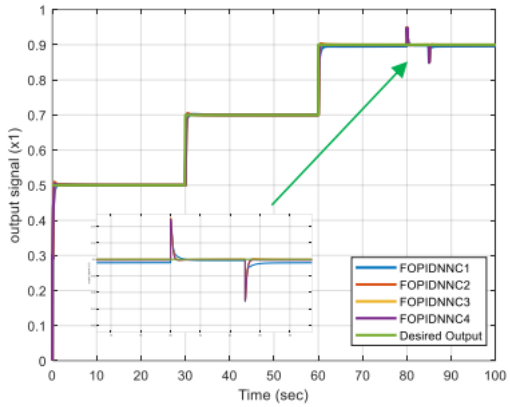

Figure. 12 Time response for disturbance of CSTR

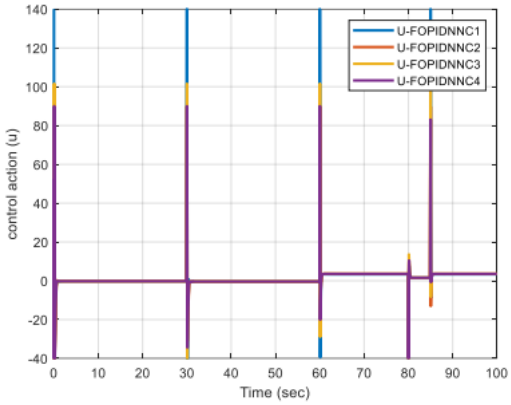

Figure. 13 Control action for disturbance of CSTR

corresponding cost function (J) of these controllers as in Table 6.

Previous findings demonstrate that the reaction of the set-point variations for (x_1) is with minimum overshoot and a quick settling time , and has less control signal (u) is necessary in the event of FOPIDNNC4. in the event of FOPIDNNC1, maximum overshoot and maximum settling time, and it has the minimum cost function (J) .

5.1 Robustness analysis of the controllers

A. Trajectory tracking

The reference signal in this test is sinusoidal $R = \frac{2\pi}{15}\sin\left(\frac{\pi}{25}t\right) + 0.5$ and has the following initial conditions: $x_1 = 0.3, x_2 = 0.3$, and the simulation time is set to 100 sec. In Table 7, the estimated cost function. Figs. 8 and 9 display the outcomes of all controllers' simulations.

The previous case shows that the response of the trajectory tracking for (x_1) is with minumum overshoot and and less cost function (J) in FOPIDNNC3.with minumum settling time , and with the least amount of control signal (u) required in the case of the FOPIDNNC4.

B. Uncertainty parameter

The reference signal is R=0.6. Following this, the simulation time is set to 30s, the parameter δ is changed from 0.3 to 0.4 at t=20s, and the initial conditions are $x_1 = 0.3, x_2 = 0.3$. The estimated cost function is shown in Table 8. The results of all controllers are shown in Figs. 10 and 11.

The precedent case demonstrates that the uncertainty response for (x_1) is with minumum overshoot in the event of and less cost function (J) in FOPIDNNC1.with maximum overshoot the event of FOPIDNNC2. the least amount of control signal (u) required in the case of the FOPIDNNC4

C. Disturbance

The initial conditions for this test are $x_1 = 0.3, x_2 = 0.3$,. the output has a variety of adjusting points,$R = 0.5\ at\ 0 \leq t < 30,\ R = 0.7\ at\ 30 \leq t \leq 60,\ R = 0.9\ at\ t \geq 60$, and disturbance d = 0.05 at $80 \leq t < 85$. In Table 9, the cost function. Figs. 12 and 13 show all controllers' simulations.

The test case shows that the disturbance response for (x_1) is with minimum overshoot and minimum settling time , and the least amount of control signal (u)required in the case of FOPIDNNC4. maximum overshoot and less cost function (J) in FOPIDNNC1.

5.2 Discussion and comparison

Best performance and lower the cost function was attained by the proposed case study for the CSTR system in the set-point task, trajectory tracking, disturbance rejection, and parameter uncertainty. comparison with other controllers in [4] ,such as multi-layer quantum neural network and adaptive PI-D controller with modified particle swarm optimization (QNN-MPSO). modified PID controller for particle swarm optimization (PID-MPSO). modified particle swarm optimization using a perceptron neural network (PNN-MPSO). The corresponding cost function (J) of these controllers is illustrated in Table 10.

6. Conclusion

Four FOPIDNN controller structures were suggested for a CSTR control in this paper. The neural network weights and controller parameters are tuned using the metaheuristic ACO. The robustness of these controllers has also been investigated in

Table 10. Comparing the values of the cost function

a. Set-point variations	
Controller	Cost function
FOPIDNNC1 in case study.	0.011588
FOPIDNNC2 in case study.	0.013524
FOPIDNNC3 in case study.	0.012883
FOPIDNNC4 in case study.	0.013542
PID - MPSO in [4].	0.020707
PNN-MPSO in [4].	0.018729
QNN-MPSO in [4].	0.018340
b. Trajectory tracking	
Controller	Cost function
FOPIDNNC1 in case study.	0.008961
FOPIDNNC2 in case study.	0.009127
FOPIDNNC3 in case study.	0.008733
FOPIDNNC4 in case study.	0.009177
PID - MPSO in [4].	0.034912
PNN-MPSO in [4].	0.034601
QNN-MPSO in [4].	0.032665
c. Uncertainty arameter	
Controller	Cost function
FOPIDNNC1 in case study.	0.015325
FOPIDNNC2 in case study.	0.018230
FOPIDNNC3 in case study.	0.017071
FOPIDNNC4 in case study.	0.018267
PID - MPSO in [4].	0.029390
PNN-MPSO in [4].	0.024629
QNN-MPSO in [4].	0.021605
d. Disturbance	
Controller	Cost function
FOPIDNNC1 in case study.	0.018274
FOPIDNNC2 in case study.	0.018965
FOPIDNNC3 in case study.	0.018516
FOPIDNNC4 in case study.	0.018946
PID - MPSO in [4].	0.074844
PNN-MPSO in [4].	0.067072
QNN-MPSO in [4].	0.064811

terms of model uncertainty, disturbance rejection, and initial conditions. The findings demonstrate that the FOPIDNN controllers, of which the FOPIDNNC1 is the best, have a decent ability to rapidly minimize the variance between real and desired routes without chattering in control signals.

Conflicts of interest

The authors declare that they have no conflicts of interest. This research was conducted independently and without funding from any external sources.

Author contributions

Conceptualization: Zaid S. Mohsen and Mohamed Jasim Mohamed; methodology: Zaid S. Mohsen, Mohamed Jasim Mohamed; software: Mohamed Jasim Mohamed; validation: Zaid S. Mohsen, Mohamed Jasim Mohamed; formal analysis: Zaid S. Mohsen; investigation: Zaid S. Mohsen and Mohamed Jasim Mohamed; resources: Zaid S. Mohsen and Mohamed Jasim Mohamed; data curation: Zaid S. Mohsen; writing-original draft preparation: Zaid S. Mohsen, Mohamed Jasim Mohamed; writing-review and editing: Zaid S. Mohsen and Mohamed Jasim Mohamed; visualization: Zaid S. Mohsen; supervision: Mohamed Jasim Mohamed; project administration: Mohamed Jasim Mohamed and Zaid S. Mohsen ; funding acquisition, Not funded.

Acknowledgments

The authors acknowledge the department of control and systems engineering at the University of Technology in Iraq.

References

[1] J. E. R. Castellanos and V. H. G. Palacio, “A Tuning Proposal for Fuzzy Gain Scheduling Controllers for Industrial Continuous Processes”, In: *Proc. of 2018 IEEE 2nd Colomb. Conf. Robot. Autom. CCRA 2018*, pp. 9–14, 2018.

[2] A. D. Kakule and P. Kerkar, “Implementation Of Temperature Regulation and Controllers”, In: *Proc. of 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*”, pp. 758–764, 2018.

[3] M. B. K. Belarbi, “Fuzzy modelling and model reference neural adaptive control of the concentration in a chemical reactor (CSTR)”, *AI Soc*, Vol. 0, No. 0, pp. 0, 2018.

[4] E. Salahshour, M. Malekzadeh, F. Gordillo, and J. Ghasemi, “Quantum neural network-based intelligent controller design for CSTR using modified particle swarm optimization algorithm”, *Trans. Inst. Meas. Control*, Vol. 41, No. 2, pp. 392–404, 2019.

[5] B. Mohamed, K. Kara, A. Oussama, and L. Hadjili, “Adaptive Neural Network PID Controller”, In: *Proc. of 2019 IEEE Int. Conf. Environ. Electr. Eng. 2019 IEEE Ind. Commer. Power Syst. Eur. EEEIC/I CPS Eur. 2019*, 2019.

[6] P. Deulkar and S. Hanwate, “Analysis of PSO-PID controller for CSTR temperature control”, In: *Proc. of 2020 IEEE 1st Int. Conf. Smart Technol. Power, Energy Control. STPEC 2020*, 2020.

[7] I. Baranilingesan, “Optimization algorithm-based Elman neural network controller for continuous stirred tank reactor process model”, *Curr. Sci.*, Vol. 120, No. 8, pp. 1324–1333, 2021.

[8] N. Khanduja and B. Bhushan, “Optimal design

of FOPID Controller for the control of CSTR by using a novel hybrid metaheuristic algorithm", *Sadhana - Acad. Proc. Eng. Sci.*, Vol. 46, No. 2, 2021.

[9] W. D. Chang, "Nonlinear CSTR control system design using an artificial bee colony algorithm", *Simul. Model. Pract. Theory*, Vol. 31, pp. 1–9, 2013.

[10] W. Zhou, H. Liu, H. He, J. Yi, and T. Li, "Neuro-Optimal Tracking Control for Continuous Stirred Tank Reactor With Input Constraints", *IEEE Trans. Ind. Informatics*, Vol. 15, No. 8, pp. 4516–4524, 2019.

[11] C. S. Engineering, "Mathematical modelling and analysis of deformable perturbed continuously stirred tank reactors (cstrs)", *Computational and Communication Science Engineering*, 2022.

[12] N. Zerari and M. Chemachema, "Robust adaptive neural network prescribed performance control for uncertain CSTR system with input nonlinearities and external disturbance", *Neural Comput. Appl.*, Vol. 32, No. 14, pp. 10541–10554, 2020.

[13] O. A. R. A. Wahhab and A. S. A. Araji, "Path Planning and Control Strategy Design for Mobile Robot Based on Hybrid Swarm Optimization Algorithm", *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 3, pp. 565–579, 2021, doi: 10.22266/ijies2021.0630.48.

[14] Z. Kayhomayoon, F. Babaeian, S. G. Milan, N. A. Azar, and R. Berndtsson, "A Combination of Metaheuristic Optimization Algorithms and Machine Learning Methods Improves the Prediction of Groundwater Level", *Water (Switzerland)*, Vol. 14, No. 5, pp. 1–25, 2022.

[15] I. H. Ibrahim and H. I. Ali, "Quantitative PID Controller Design using Black Hole Optimization for Ball and Beam System", *Iraqi J. Comput. Commun. Control Syst. Eng.*, Vol. 21, No. 3, pp. 65–75, 2021.

[16] J. Tavoosi, "A new general type-2 fuzzy predictive scheme for PID tuning", *Appl. Sci.*, Vol. 11, No. 21, pp. 1–15, 2021.

[17] M. J. Mohamed, "Design a Fuzzy PID Controller for Trajectory Tracking of Mobile Robot", *Engineering and Technology Journal*, Vol. 36, No. 1, 2018.

[18] M. J. Mohamed, "Comparison Between PID and FOPID Controllers Based on Particle Swarm Optimization", In: *Proc. of The Second Engineering Conference of Control, Computers and Mechatronics Engineering (ECCCM2, 2014)*, No. February, 2014.

[19] H. I. Abdulameer and M. J. Mohamed, "Fractional Order Fuzzy PID Controller Design for 2-Link Rigid Robot Manipulator", *Int. J. Intell. Eng. Syst.*, Vol. 15, No. 3, pp. 103–117, 2022, doi: 10.22266/ijies2022.0630.10.

[20] Y. Jiang and B. Zhang, "Comparative Study of Riemann-Liouville and Caputo Derivative Definitions in Time-Domain Analysis of Fractional-Order Capacitor", *IEEE Trans. Circuits Syst. II Express Briefs*, Vol. 67, No. 10, pp. 2184–2188, 2020.

[21] X. Zhou, C. Zhao, and Y. Huang, "A Deep Learning Optimizer Based on Grünwald–Letnikov Fractional Order Definition", *Mathematics*, Vol. 11, No. 2, 2023.

[22]S. Automatisés, T. S. Mahmood, and O. F. Lutfy, "A Wavelet Neural Network-Based NARMA-L2 Feedforward Controller Using Genetic Algorithms to Control Nonlinear Systems A Wavelet Neural Network-Based NARMA-L2 Feedforward Controller Using Genetic Algorithms to Control Nonlinear Systems", *Journal Européen des Systèmes Automatisés* , No. October, 2022.

[23] A. J. Humaidi and A. S. M. A. Obaidi, "PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot", *International Review of Applied Sciences and Engineering* , 2023.

[24] M. Mohamed and M. Hamza, "Design PID Neural Network Controller for Trajectory Tracking of Differential Drive Mobile Robot Based on PSO", *Eng. Technol. J.*, Vol. 37, No. 12A, pp. 574–583, 2019.

[25] S. Jung and S. S. Kim, "Control experiment of a wheel-driven mobile inverted pendulum using neural network", *IEEE Trans. Control Syst. Technol.*, Vol. 16, No. 2, pp. 297–303, 2008.

[26] J. Szkoa, K. Pancerz, and J. Warcho, "Recurrent neural networks in computer-based clinical decision support for laryngopathies: An experimental study", *Comput. Intell. Neurosci.*, Vol. 2011.

[27] A. Thammano and P. Ruxpakawong, "Nonlinear dynamic system identification using recurrent neural network with multi-segment piecewise-linear connection weight", *Memetic Comput.*, Vol. 2, No. 4, pp. 273–282, 2010.

[28] G. D. Sen, G. O. Gunel, and M. Guzelkaya, "Extended Kalman Filter Based Modified Elman-Jordan Neural Network for Control and Identification of Nonlinear Systems", In: *Proc. - 2020 Innov. Intell. Syst. Appl. Conf. ASYU 2020*, No. 1, 2020.