A PSO optimized novel PID Neural Network model for Temperature Control of Jacketed CSTR: Design, Simulation, and a Comparative Study

Snigdha Chaturvedi (snigdha0chaturvedi@gmail.com)
Delhi Technological University https://orcid.org/0000-0002-7729-7063
Narendra Kumar
Delhi Technological University
Rajesh Kumar
Delhi Technological University

Research Article

Keywords: Backpropagation, Neural network, PID controller, CSTR, Particle swarm optimization

Posted Date: June 6th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2851632/v1

Version of Record: A version of this preprint was published at Soft Computing on September 4th, 2023. See the published version at https://doi.org/10.1007/s00500-023-09138-0.

Title page

A PSO optimized novel PID Neural Network model for Temperature Control of Jacketed CSTR: Design, Simulation, and a Comparative Study

Snigdha Chaturvedi [a,] *, Narendra Kumar [b], Rajesh Kumar [c]

[a] Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

[b]Delhi Technological University, Shahbad, Daulatpur, Main Bawana Road, Delhi-110042, India

[c]National Institute of Technology, Kurukshetra-136119, India

Email address: snigdha_2k17phdee07@dtu.ac.in, narendrakumar@dtu.ac.in, rajeshmahindru23@nitkkr.ac.in

Corresponding Author details:

Snigdha Chaturvedi

Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

Email address: snigdha_2k17phdee07@dtu.ac.in

A PSO optimized novel PID Neural Network model for Temperature Control of Jacketed CSTR: Design, Simulation, and a Comparative Study

Abstract

This paper proposes a Particle Swarm Optimization (PSO) tuned novel Proportional Integral Derivative (PID) like neural network (PID-NN). The structure of proposed PID-NN is very simple having only 3 neurons in the hidden layer and a single output neuron. The proportional, integral, and derivative gains of the PID controller are represented by the three weights in the neural network's output layer, respectively. The suggested approach uses the PSO method to optimize the output layer weights, which correspond to PID gains. The non-linear Continuous Stirred Tank Reactor (CSTR) plant, one of the most popular chemical industry processes, is utilized to test the suggested approach. A jacketed CSTR's temperature is controlled via a Particle Swarm Optimization tuned PID like neural network (PSO-NN-PID) controller. In terms of time domain specifications, the performance of the PSO-based NN-PID controller, the back propagation-based NN-PID controller (BP-NN-PID), and the conventionally tuned PID controller are compared. Mean square error is the objective function used in PSO-NN-PID and BP-NN-PID to optimize PID settings. The results show that the overshoot decreases from 44.13% in case of Zeigler- Nichols tuned PID controller to 26.33% in case of BPNN-PID controller, and further reduces to 23.13% in case of proposed PSO-based NN-PID controller. The decrease in rise time is observed from 0.2727 seconds in case of BPNN tuned PID to 0.1283 seconds in case of proposed PSO-NN-PID controller.

Keywords: Backpropagation, Neural network, PID controller, CSTR, Particle swarm optimization

1. Introduction

Most of the linear control problems have been solved because of intensive study on linear control systems during the past few decades. However, most systems in the real world including those that govern vehicles, biological systems, aircrafts, and robots are inherently nonlinear in nature. In recent years, nonlinear control issues have received a lot of attention. Certain laws, like superposition and homogeneity, are observed only in linear systems. A nonlinear system does not operate uniformly according to a common law. A nonlinear system displays various behavior in several operating regions[1]. Therefore, it becomes difficult to develop an effective controller for non-linear system. In industrial applications, the PID controller has perhaps been the most successful controller. Despite the development of numerous nonlinear system control strategies, such as artificial neural networks, fuzzy logic control, adaptive control, and model predictive control. 90% of industrial controllers are PID controllers. The controller's ongoing success serves as evidence that in engineering, simplicity and ease of use are most important aspect. The crucial step while implementing a PID controller is the effective tuning of three gains, K_p proportional gain, K_i integral gain and K_d derivative gain[2]. A study found that 30% of installed controllers employ manual PID tuning and that 65% of them are improperly tuned. Numerous adjustments are performed to account for the nonlinear effects when employing a PID controller on a nonlinear system. In past few decades several intelligent optimization techniques have been used to tune PID gains. A lot of research has been done on controlling linear systems with PID controllers and in most cases, we attain desirable results[3]. But the problem arises when PID controller is applied to a nonlinear system. Many researchers have addressed this problem and proposed various methods of handling the nonlinear effects. Some of the methods used in past are linearizing the plant around a stable operating point, gain scheduling, anti-reset windup etc. [4]. Following are the four primary divisions of classical PID tuning methods: The first method involves analytically calculating the PID parameters from algebraic equations between the plant and the performance objectives. Analytical tuning methods include lambda tuning and internal model control. These may produce a straightforward formula that can be used for online tuning, but the model must be accurate, and the objective must take the form of an analytical formula. Among the methods used in the second category are Zeigler Nichols[5] tuning, Cohen Coon tuning[5], and other heuristic PID tuning strategies. Once more, these can be employed as a formula or rule-based system for online use, frequently with trade-off design goals.

Because of the above-mentioned challenges while applying PID controller, hybrid form of PID parameter adjustments is becoming more popular. The advantages of simple PID and soft computing techniques can be combined to tune PID parameters effectively. This method can improve robustness of the system. With the development of AI techniques in few decades, application of these methods to the PID parameter tuning has proved to be efficient than conventional methods. To address this problem in past few decades, a lot of researchers have applied different optimization algorithms to tune PID parameters. The ANN is used as an intelligent controller for modelling and control of nonlinear system. Fuzzy logic has also been used as controller for tuning PID parameters. The fuzzy controller uses error and change in error to obtain PID gain values. However, fuzzy logic requires expert knowledge to determine appropriate membership functions[6]. Therefore, NN can solve the problem because they have high computational speed and have been used in past for determining PID parameters efficiently. Metaheuristic algorithms have also been used to address complex nonlinear problems. With the research of several algorithms like GA, PSO, ACO, CSA etc. They suffer from the drawback of right selection of parameters and slow computational speed. These algorithms mostly do not give desired results because it becomes difficult to choose several parameters correctly. So, there is a need to develop hybrid method combining NN and PSO. PSO being an algorithm widely used for lot of complex problems has proved its efficiency[7]–[11]. Neural network can be utilized for tuning PID gains. A Back propagation neural network can be modified as a PID like structure and provide with the advantages of both BP neural network and PID control. This technique has been utilized and being applied to several nonlinear problems[6], [12]–[14].

In this study a novel hybrid approach is proposed utilizing Particle swarm optimization for weight initialization and optimization for PID like neural network structure. The above proposed technique is compared with the BP-NN-PID, and conventionally tuned Zeigler Nichols tuned PID controller. The proposed controllers are applied on temperature control of a Jacketed CSTR which is a nonlinear benchmark control problem.

The structure of the paper is as follows: section 1 contains the introduction, literature review, research gaps identified and contributions of the paper, section 2 describes the problem statement, section 3 describes the mathematical modelling of the CSTR temperature control, section 4 discusses the proposed PSO based NN -PID structure and algorithm, section 5 describes the classical Back propagation tuned NN-PID controller, section 6 describes the simulation results and finally conclusion is given in section 7.

Literature Review

In past a lot of researchers have done a substantial amount of research in PID controller tuning. Earlier classical tuning methods like Zeigler Nichols[5], Cohen coon and IMC type tuning[15] was mostly used for PID tuning. In [16] authors, have reviewed the various issues and problems while tuning of PID controller. In [2] authors, have discussed various problems in tuning PID controller when multi objectives are to be achieved. The intelligent methods were proposed to tune PID controller. In [3], authors proposed an extended PID controller while controlling nonlinear systems with uncertainty. In [17], authors have presented an review of various conventional and intelligent tuning methods and compared their performance on an level control system. In [18] authors have proposed tuning method for an AVR system using chaotic optimization method and compared the performance in terms of rise time, settling time and overshoot. In [8], authors have applied PSO and TLBO algorithm to tune PID controller for concentration and temperature control of CSTR. In [7], authors have used cuckoo search algorithm to tune PID parameters of quadrotor. The simulation results proved that this method is more efficient than classical tuned methods. In [19], authors have used fuzzy logic to tune PID parameters applied to control airway pressure of respiratory system. The fuzzy logic based PID controller proved better as compared conventionally tuned methods in terms of rise time, overshoot, settling time etc. In [20], authors have applied PSO to tune PID gains of an artificial ventilation system. The results proved that PSO tuned controller gave better response as compared to conventional methods. In [21], authors proposed an improved teaching learning based optimization algorithm by adding concept of number of teachers, teaching factor adaptability etc. The improved teaching learning-based optimization algorithm was tested on benchmark functions. The results showed its superiority. In [22], authors have compared hybrid genetic algorithm type fractional order PID controller, Fmincon based pattern search based PID controller and hybrid MPC controller. Hybrid genetic algorithm based FOPID gives better response in terms of performance indices. In [23], authors have proposed gravitational search based fuzzy logic control for controlling position of nonlinear ball and beam system. The results showed that the results were better as compared to conventional controller. In [24] proposed a new algorithm to optimize PID parameters by combining swarm algorithms and learning processes. The proposed algorithm was better as compared to GA, PSO and NN tuned controllers. In [25], authors described the methods to develop fuzzy controllers for tuning PID gains. In [26], authors proposed a multi-objective fuzzy based fractional order PID controller tuned by evolutionary algorithms to control a hydraulic turbine governing system. In [27], authors applied an arithmetic optimization algorithm called empirical identification algorithm to find best PID values. In [28], authors have implemented hybrid optimization algorithms combining PSO with GWO and GSA to tune PID parameters. The proposed controllers were implemented to three bench mark nonlinear problems. The hybrid optimization techniques proved to be faster and less convergence error as compared to earlier used evolutionary algorithms. In [29], authors have proposed a new intelligent controller design based on sequential quadratic programming to derive the transfer function.

Research Gaps

PID tuning is a challenging process when applied to a nonlinear system, according to the literature review. To adjust PID controllers, several researchers have used a variety of techniques. Numerous AI techniques, including neural networks, fuzzy logic, and bio-inspired algorithms, are used to fine-tune PID controllers. The AI-based tuning techniques have been shown to be superior to the traditional techniques. The neural network based intelligent methods have the advantage of high computational speed whereas, the evolutionary algorithms can be easily applied to any non-linear system and gives accurate results. So, there is a need to combine the advantages of both the techniques and develop hybrid controllers.

Contributions of the paper

The main contributions of the paper are summarized as below:

1. A novel PSO tuned PID like neural network is proposed. The structure of PID like neural network is kept simple having only three neurons in the hidden layer. Therefore, the implementation of the controller is very easy.
2. The proposed controller is compared with the BP based NN-PID controller and ZN tuned PID controller.
3. The controller developed is tested for robustness under the disturbance application and is found to perform better than BP based NN-PID controller and ZN tuned PID controller.

2. Problem Statement

The structure of the proposed control scheme is shown in Fig.1. In the Figure 1, e(k) is the error between actual output and desired reference input. The optimized PID like neural network is connected in cascade with the non-linear jacketed CSTR plant. The neural network weights are adjusted with PSO algorithm to obtain P, I and D parameters.

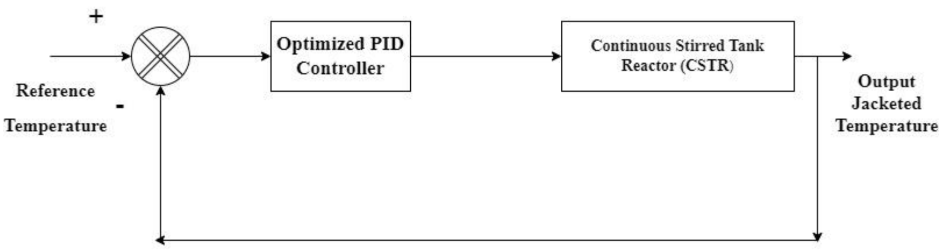

Figure 1 Structure of the proposed control scheme

The cost function is calculated after each iteration. The objective of the controller is to minimize the cost function:

$$E_c = \frac{1}{2}\sum_{k=1}^{T}(r(k) - y_{act}(k))^2 \quad (1)$$

Where, $r(k)$ is the desired reference input and $y_{act}(k)$ is the actual output.

3. Mathematical Modelling of a Jacketed CSTR

Continuous Stirred tank Reactor (CSTR) is reactor mostly used in chemical industries. It is generally used in processes where continuous flow of product is required to reach a product. Chemical reactors have a lot of effect on the output due to heat, so it is important to control heat in a chemical reactor. In a jacketed CSTR temperature control can be done by controlling the temperature of jacket around the reactor. The jacket temperature can be controlled by a coolant flowing in it. It operates with the following assumptions:

1. The reactor volume is kept constant V_R.
2. It is operated under steady state conditions with perfect mixing

In the CSTR considered we consider a simple reaction $A \xrightarrow{yields} B$. The concentration of the feed flown initially of product A is donated as C_{AR0}, the initial temperature of the product A is T_{R0} and it is assumed that there is a constant flow rate q_R. Irreversible reactions take place inside the reactor. The final products generated have a concentration C_{AR} and temperature T_R. The heat produced in the exothermic reaction is controlled by flowing a coolant in the jacketed layer of the reactor. The coolant temperature is T_{CRO} and its flow rate is q_{CR}.

The material balance of jacketed CSTR can be given as:

$$\frac{dV_o\rho_o}{dt} = q_{in}\rho_{in} - q_{out}\rho_{out} \quad (2)$$

Where, V_o is the reactor volume, ρ_o is the density of reactor, ρ_{in} is the density of feed flow input, q_{in} and q_{out} is the flow rates of input and output feed. In this case as per the assumptions $\rho_{in} = \rho_{out} = \rho_o$. The flow rates are also equal.

The balance equation for component A is given as,

$$V_o \frac{dC_{AR}}{dt} = q_R(C_{AR0} - C_{AR})V_R - V_o r_A \tag{3}$$

Where, C_{AR} is the concentration of component A in the reactor, r_A is the rate of reaction and is given as,

$$r_A = K_{R0} \exp\left(\frac{-E_R}{RT_R}\right) C_{AR}$$

Where, K_{R0} is the frequency factor, E_R is the activation energy and T_R is the reactor temperature

The energy balance equations for the reactor considering above given assumptions and neglecting kinetic and potential energy are given as,

$$V_R \rho_R C_P \frac{dT_R}{dt} = q_R \rho_R C_P (T_R - T) - (-\Delta H) V_R r_A + \rho_{CR} C_{PC} q_{CR} \left[1 - \exp\left(\frac{-hA}{\rho_{CR} C_{PC} q_{CR}}\right)\right] (T_R - T) \tag{4}$$

In Equation 3 RHS represents heat accumulation in the reactor, in LHS the first term represents energy in and out due to flow of component A, the second term represents the heat due to reaction in the reactor and the third term represents heat transferred to the jacket. Where, T_R is the reactor temperature and T is the jacket temperature.

The above equations can be presented in the state variable form choosing C_{AR} and T_R as state variables,

$$\frac{dC_{AR}}{dt} = \frac{q_R}{V_R}(C_{AR0} - C_{AR}) - K_{R0} \exp\left(\frac{-E_R}{RT_R}\right) C_{AR} \tag{5}$$

$$\frac{dT_R}{dt} = \frac{q_R}{V_R}((T_{RO} - T_R) - \left(\frac{-\Delta H}{\rho_R C_P}\right) K_{R0} \exp\left(\frac{-E_R}{RT_R}\right) C_{AR} + \left(\frac{\rho_{CR} C_{PC}}{\rho_R V_R C_P}\right) q_{CR} \left[1 - \exp\left(\frac{-hA}{\rho_{CR} C_{PC} q_{CR}}\right)\right] (T_{RO} - T_R) \tag{6}$$

The steady state solutions can be obtained by putting the two equations 5 & 6 to zero. For this study the parameters of jacketed CSTR chosen are shown in Table 1.

Table 1 Parameters of Jacketed CSTR

Parameters	**Values**
C_{AR} Concentration of input feed component A	**0.0882 mol/l**
Reactor temperature T_R **Flow rate of the coolant q_{CR}**	**441.2 K** **100 l/min**
Feed flow rate q_R	**100 l/min**
Input Feed temperature T_O	**350 K**
Jacket temperature T	**350 K**
Volume of reactor V_R	**100 l**

Coefficient of heat transfer hA	**7×** $\mathbf{10^5}$ $\boldsymbol{cal/(minK)}$
Reaction rate constant $\boldsymbol{K_R}$	**7.2×**$\mathbf{10^{10}}$ **min-1**
Activation Energy $\boldsymbol{\frac{E_R}{R}}$	**1×**$\mathbf{10^4}$ **K**
Heat produced in Reaction $\boldsymbol{(-\Delta H\ )}$	**-2×** $\mathbf{10^5}$ $\boldsymbol{cal/mol}$
$\boldsymbol{C_H, C_{HC}}$ **Specific heat**	**1 Cal (g/K)**
Densities, $\boldsymbol{\rho_{R,}\rho_{CR}}$	$\mathbf{1 \times 10^3}\boldsymbol{g/l}$

3. Structure of NN Based PID Controller

The structure of BPNN Based PID Controller consists of input layer, hidden layer, and output layer. There is a single node in the input layer, three nodes in the hidden layer and one node in the output layer. The three nodes of the hidden layer correspond to three PID parameter gains, proportional Kp, integral gain Ki and derivative gain Kd. The activation functions used are all linear in nature. To produce the integral action the integral nodea_1 is taken as feedback by using delaying effect. To produce the derivative effect the derivative node a_3 is given activation feedback. The proportional node a_2 is the general node without any feedback. The BPNN based PID structure is shown in Figure 2,

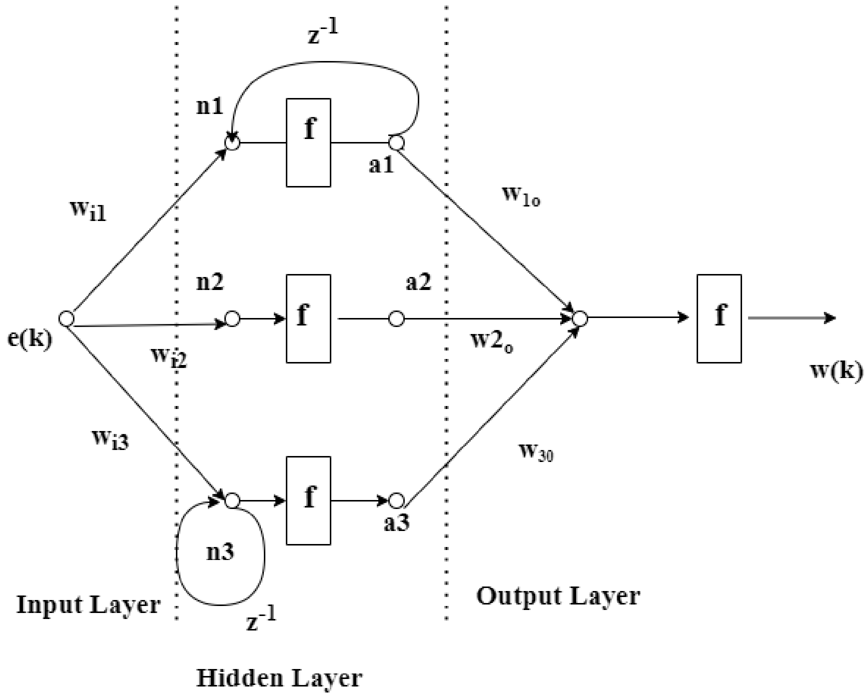

Figure 2 Structure of BPNN Based PID controller

The outputs of the hidden layer a_1, a_2 and a_3 at sampling times k may be given as,

$$a_1(k) = e(k)w_{i1}(k) + a_1(k-1) \tag{7}$$

$a_2(k) = e(k)w_{i2}(k)$ (8)

$a_3(k) = e(k)w_{i3}(k) - w_{i3}(k-1)e(k-1)$ (9)

The output of the neural network-based controller is,

$u(k) = \sum_{j=1}^{3} w_{jo}(k)a_j(k) = w_{1o}(k)a_1(k) + w_{2o}(k)a_2(k) + w_{30}(k)a_3(k)$ (10)

In the first node z^{-1} is added as delay. So, $a(k-1) = a(k)z^{-1}$

Then, $e(k-1) = z^{-1}e(k)$ (11)

The output of the first hidden layer is

$a_1(k) = \frac{w_{i1}(k)e(k)}{1-z^{-1}}$ (12)

This equation represents an integral relationship. Therefore, this node generates a gain proportional to the integral of the error. The derivative action was produced by the derivative node using activation feedback.

$a_3(k) = w_{i3}(k)e(k)[1 - z^{-1}]$ (13)

This equation represents differential mode of operation. It is evident from the control output u(k) is that the neural network-based controller produces output like a PID controller. The second node of the neural network produces action like proportional control, the first node produces output like an integral action and third node produces output like a derivative control. A conventional BP based NN is complex having several layers, but a PID NN proposed here is simple structure where the output layers produce the sum or controller output. In this way by training the weights of PID based NN we can easily tune the PID parameters to obtain desired results.

4. PID Like Neural Network tuned by PSO

In this paper, a novel method is proposed to tune PID NN using PSO. PSO is an evolutionary algorithm which is inspired by bird flocking or fish schooling. A random population of particles is initialized in a search space. The particles have certain velocity and position. The particles fly in search space to the global best position after certain iterations. After each iteration, the particles position and velocity are adjusted in accordance with the value of fitness function.

4.1 ***Mathematical Formulation***

PSO is used to tune weights of PID like NN proposed in the study. The position of particles represents the weights in each iteration. The dimensionality of PSO algorithm chosen is equal to the total number of weights i.e., six. There are three weights used connecting input layer to hidden layer and three weights connecting hidden layer to output layer. Let the position vector of particles with S search space be defined as,

$x_p = [x_{pi1}, x_{pi2}, \dots \dots \dots, x_{piS]}^T$ (14)

And the velocity vector denoted as,

$v_p = [v_{pi1}, v_{pi2}, \dots \dots \dots \dots .. v_{iS}]^T$ (15)

The optimization is done by PSO and not by classical Back propagation algorithm. The objective function used for tuning NN based PID controller weights is Mean square error. Each particle moves in accordance to minimize the fitness function. The local best position of the particles is given by,

$$P_{abestij} = [P_{abesti1}, P_{abesti2} \dots\dots\dots\dots\dots .. P_{absetiS}]^T \quad (16)$$

The global best position is denoted by,

$$G_{obestij} = [G_{obesti1}, G_{obesti2} \dots\dots\dots\dots\dots .. G_{obsetiS}]^T \quad (17)$$

The particles change their position by updating the synaptic weights of the neural network to reduce the value of fitness function. The process of updating velocity after each iteration k is given by the following equation,

$$v_{ij}(k+1) = w_o v_{ij}(k) + c_1 r_1 \left[P_{abestij}(k) - x_{ij}(k)\right] + c_2 r_2 [G_{obestij}(k) - x_{ij}(k)] \quad (18)$$

Where, w_o is the weight of the algorithm, c_1 and c_2 are social factors and r_1 and r_2 are random numbers in interval [0,1].

The position of particles at $(k+1)^{th}$ iteration is,

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k) \quad (19)$$

After the updating the velocity and position, the values of P_{abesti} and G_{obesti} vectors also change accordingly. The two vectors are updated by the following law,

$$P_{abesti}(k+1) = \begin{cases} P_{abesti}(k) \ if \ f(x_i(k) \geq f(P_{abest}(k) \\ x_i(k) \qquad\qquad\qquad\qquad else \end{cases} \quad (20)$$

$$G_{obesti}(k+1) = \min [P_{abesti}(k+1) \quad (21)$$

The flowchart of the PSO Algorithm is shown in Figure 3.

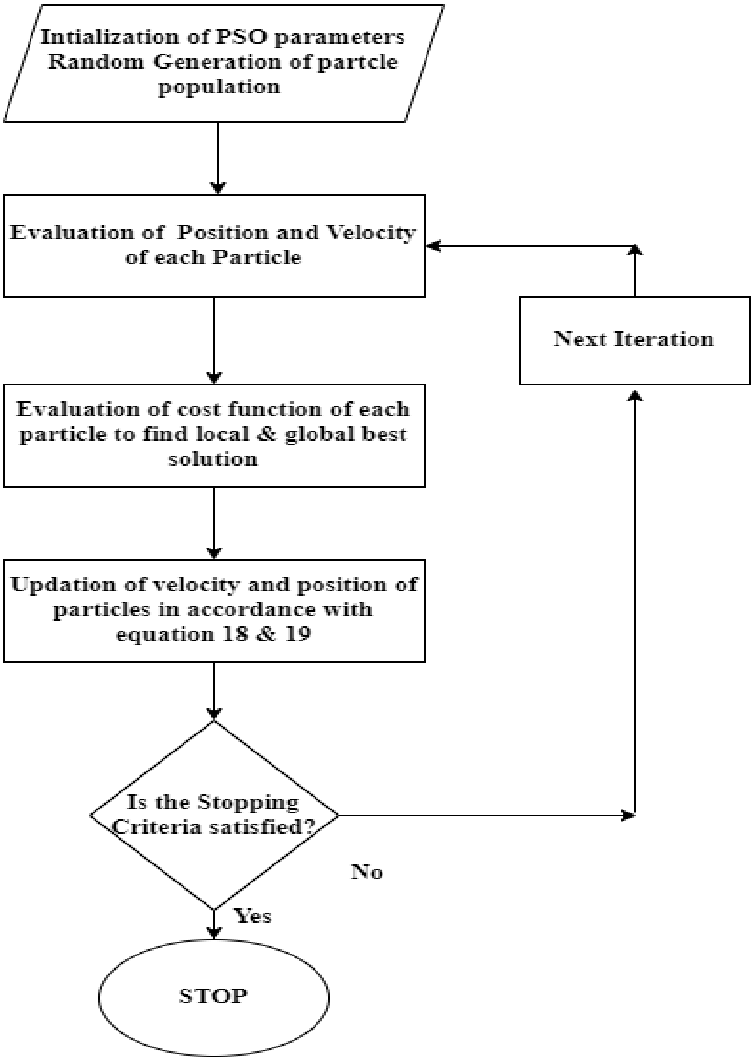

Figure 3 Flowchart of PSO

Despite of the development of several metaheuristic algorithms, PSO remains one of the most widely used algorithm to solve various non-linear problems. The main advantages of PSO are its simple structure, ease of implementation, fast computational efficiency, and robustness to the selection of parameters. The block diagram explaining the function of PSO-NN-PID is shown in Figure 4.

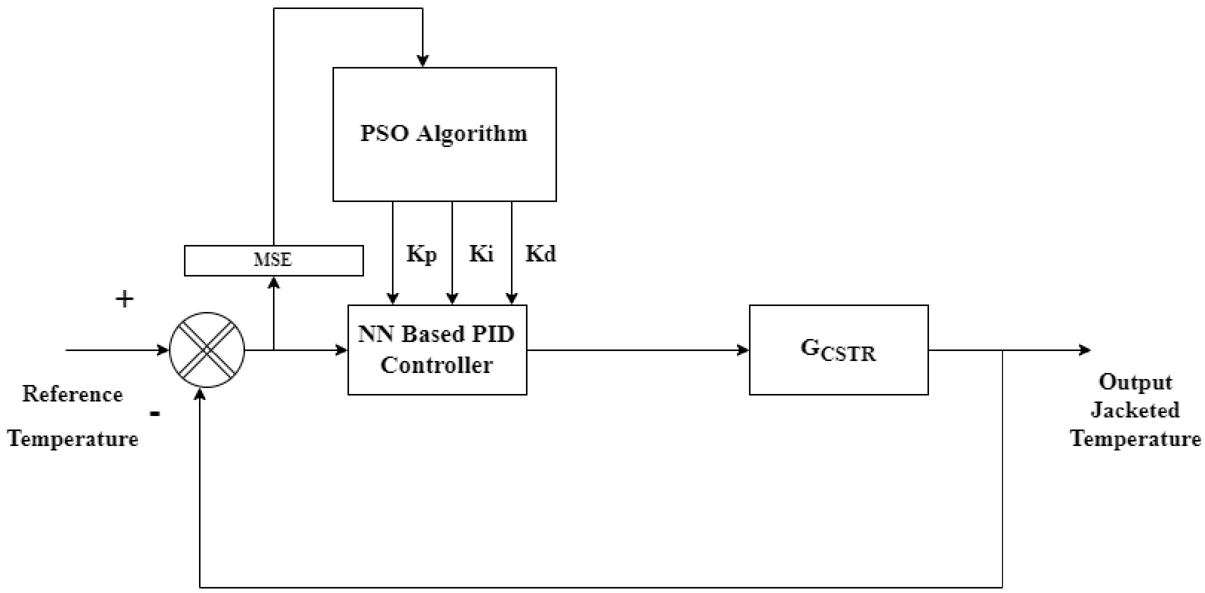

Figure 4 Block Diagram of PSO based novel NN-PID controller

It can be seen in the Figure 2 that the NN like PID controller parameters are tuned by PSO. The objective function used is mean square error (MSE). The PSO tuned NN based PID controller then gives the control signal to the Jacketed CSTR for temperature control. The algorithm of proposed PSO tuned NN-PID Controller is described below:

4.2 Algorithm of the proposed PSO tuned NN-PID Controller

1. Define PID like neural network having one neuron in the input layer, three neurons in the hidden layer and one neuron in the output layer.
2. Initialize random weights within the range [-1,1].
3. PSO is initialized with dimension size 3, population size 25 and maximum number of iterations as 50.
4. The fitness function is chosen as Mean square error.
5. For all particles do
6. Calculation of the control law $u(k)$ by initial weights.
7. Calculation of output $y_i(k)$
8. Evaluate current position and velocity in the search space
9. Calculate the value of objective function for the current iteration
10. If current position gives best objective function, then,
11. $P_{abesti} = w_i^k$
12. Else If the current objective function is the best overall objective function, then,
13. $G_{abesti} = w_i^k$
14. Endif
15. Move the particles in the search space
16. Update the position and velocity of the particles,
17. Do until stopping criteria is met
18. END

5. Classical Back propagation algorithm for tuning NN PID controller

The objective of the controller is to minimize the error between the actual output and the desired reference input. The output error can be written mathematically as,

$$error_i(k) = r_i(k) - yout_i(k) \tag{22}$$

Where, I = 1,2,….S and $yout_i(k)$ are the output variables measured and $r_i(k)$ is the reference input

The fitness function MSE is given as,

$$MSE(k) = \sum_{i=1}^{S} MSE_i(k) \tag{23}$$

Now, we state the rules for updating weights of neural network controller

$$\frac{\partial MSE(k)}{\partial w_{ij}(k-1)} = \sum_{i=1}^{S}\left[\frac{\partial MSE_i(k)}{\partial yout_i(k)}.\frac{\partial yout_i(k)}{\partial u(k-1)}\right]\frac{\partial u(k-1)}{\partial w_{ij}(k-1)} \tag{24}$$

$$\frac{\partial yout_i(k)}{\partial u(k-1)} \approx \frac{\Delta yout_i(k)}{\Delta u(k-1)} \approx \frac{yout_i(k)-yout_i(k-1)}{u(k-1)-u(k-2)} \tag{25}$$

We use sign function to find the result. The final rule of updating weights is,

$$\Delta w_{ij}(k-1) = -\alpha_{rij}\frac{\partial MSE(k)}{\partial w_{ij}(k-1)} = \alpha_{rij}\sum_{i=1}^{S} error_i(k).sign\frac{\Delta yout_i(k)}{\Delta u(k-1)}.error_i(k-1) \tag{26}$$

Where, j=1,2,3

Weights of output layer are,

$$w_{oi} = 1, i = 1,2,3 \tag{27}$$

The function of back propagation based NN PID controller is shown in Figure 6.

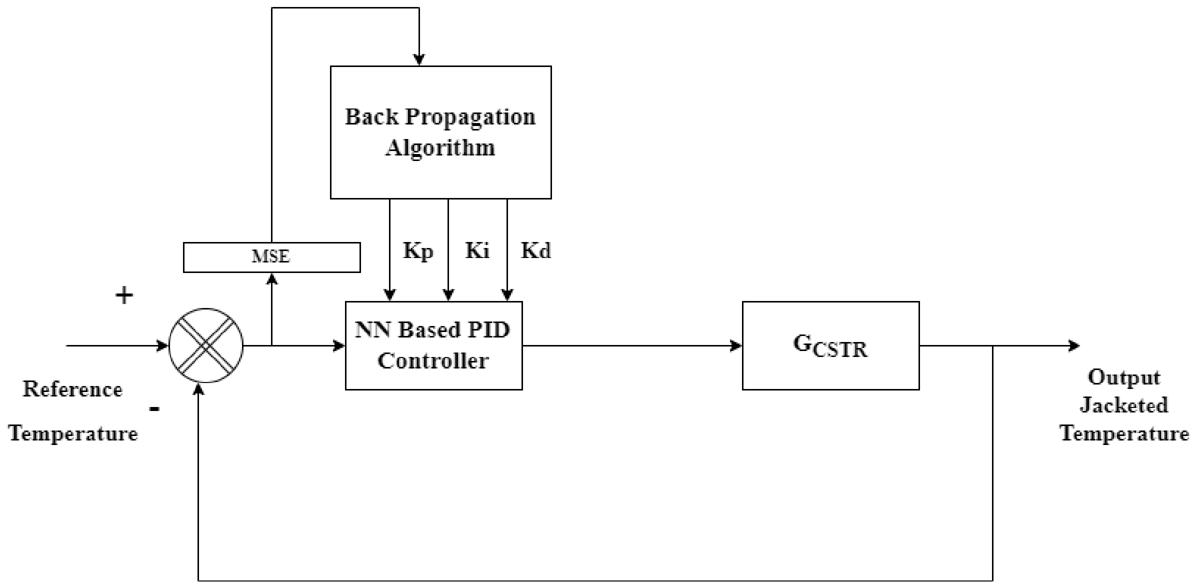

Figure 2 Block Diagram of BP-NN based PID controller

The BP based NN like PID controller shown in the Figure 3 gives the optimized PID parameters. The NN based PID controller gives the control signal to control the temperature of jacketed CSTR.

6. Simulation Results

The proposed hybrid PSO-NN technique is implemented to tune PID parameters for temperature control of jacketed CSTR and compared with BPNN-PID and ZN-PID tuning method. The parameters chosen while applying PSO algorithm are presented in Table 2

Table 2 PSO Parameters

S.no.	Parameters	Values
1.	$C_1 \& C_2$ Acceleration Constants	2
2.	w_{max}	0.9
3.	w_{min}	0.4
4.	Maximum no. of Iterations	50

The comparative step responses for various controllers for temperature control of jacketed CSTR is given in Figure 7.

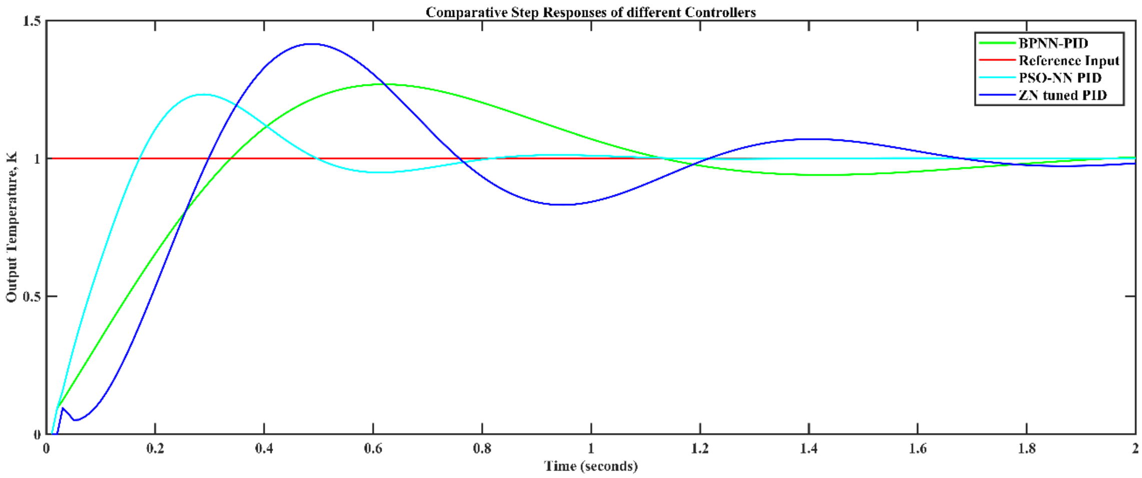

Figure 3 Comparative Step responses of Various Controllers

The performance indices explaining the step responses quantitatively is presented in Table 3.

Table 3 Comparative Step performance indices of various Controllers

S.no.	**Method used**	**Rise time (in secs)**	**Overshoot %**	**Peak time (in secs)**	**Settling time (in secs)**
1.	Zeigler Nichols	0.1810	44.08	0.4900	1.6731
2.	BPNN-PID	0.2727	26.33	0.6200	1.8194
3.	PSO-NN-PID	**0.1283**	**23.13**	**0.2900**	**0.7480**

It is evident from Figure 4 and Table 3 that rise time, tr is lowest 0.1283 in case of PSO-NN-PID tuned controller as compared to BPNN-PID and ZN tuned PID. The maximum overshoot has also reduced from 44.08% in ZN tuned PID to 26.33% in BPNN-PID and has further reduced to 23.13% in PSO-NN-PID. The settling time is also lowest 0.7480 in case of PSO-NN-PID controller. The values of different optimized PID gains Kp, Ki and Kd is given in Table 4.

Table 4 K_p, K_i and K_d parameters of various Controllers

S.no.	**Method Used**	K_p	K_i	K_d
1.	ZN Tuned	4.0795	0.8703	12.019
2.	PSO-NN-PID	**9.9000**	**2.0272**	**9.4000**
3.	BPNN-PID	4.3300	0.3100	6.1100

The proposed controller is also tested for disturbance rejection. The comparative step responses of different controllers in case of disturbance application from time=1 second to time=2 seconds is presented in Figure 8. The performance indices of the responses presented in Figure 5 are quantitatively compared in Table 5.

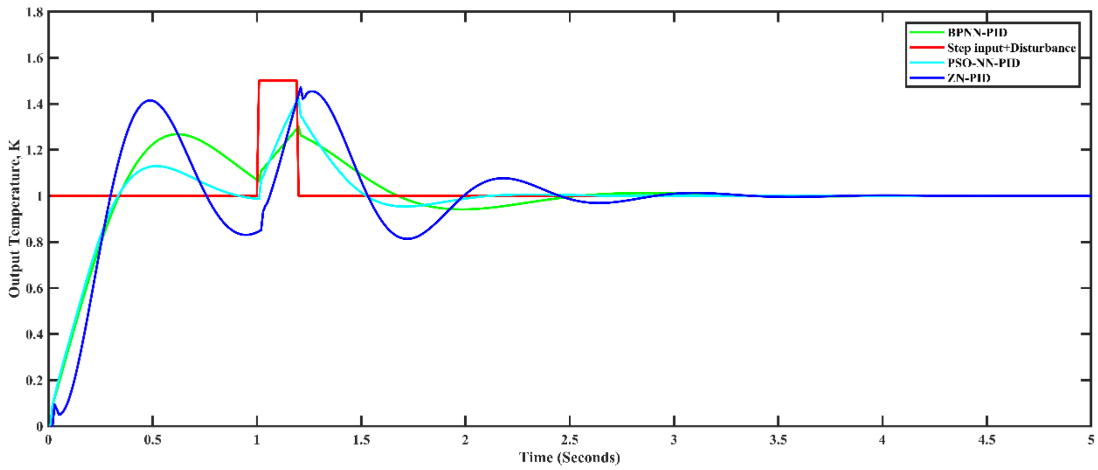

Figure 4 Comparative Step responses of various controllers under disturbance application

Table 5 Comparative Step performance indices of various controllers under disturbance application

S.no.	Method used	Rise time (in secs)	Overshoot %	Peak time (in secs)	Settling time (in secs)
1.	Zeigler Nichols	0.1839	47.03	1.21	2.77
2.	BPNN-PID	0.2713	30.00	1.21	2.36
3.	PSO-NN-PID	**0.1289**	50.00	**1.20**	**1.88**

The Table 5 shows that the PSO-NN-PID has the best rise time of 0.1289 seconds as compared to 0.2713 seconds in BPNN-PID and 0.1839 seconds in ZN-PID. There is slight increase in maximum overshoot in case of PSO-NN-PID controller but if we compare the settling time, it shows that the controller efficiently reduces the effect of disturbance quickly. Settling time is lowest 1.88 seconds in case of PSO-NN-PID controller. The values of controller gains are given in Table 6.

Table 6 K_P, K_i and K_d parameters of various Controllers under disturbance application

S.no.	Method Used	K_P	K_i	K_d
1.	ZN Tuned	4.0795	0.8703	12.019
2.	PSO-NN-PID	9.9000	1.9983	9.4000
3.	BPNN-PID	4.2145	0.2909	6.2350

The comparative analysis of the performance indices is shown in the Figure 9.

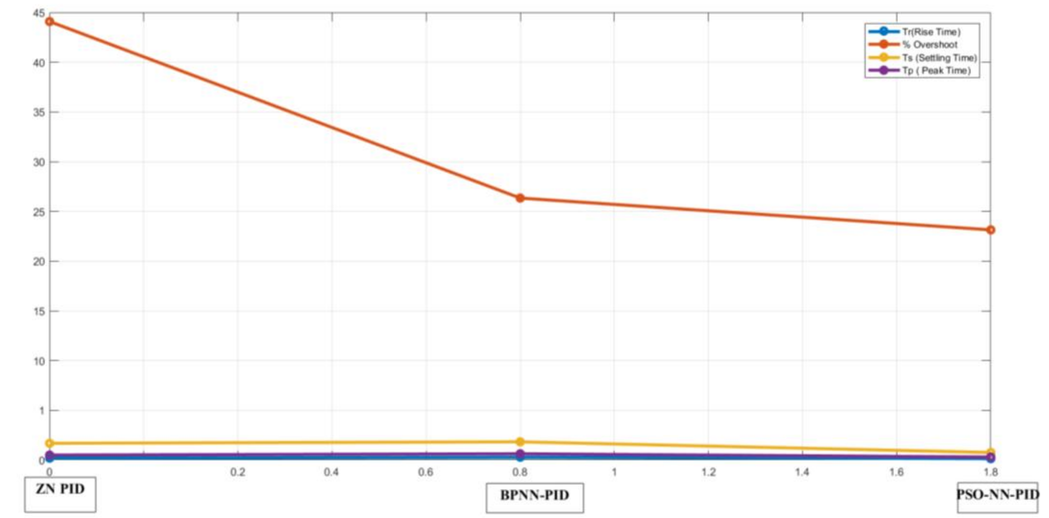

Figure 5 Comparative graph of rise time, overshoot, peak time and settling time for various controllers

The control signal u(t) and error signal e(t) are shown in Figure 10 & 11.

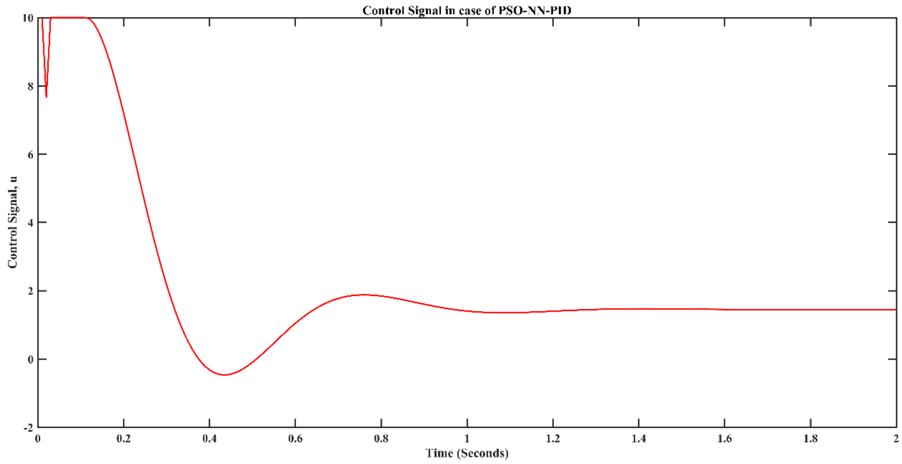

Figure 6 Control Signal in case of PSO-NN-PID controller

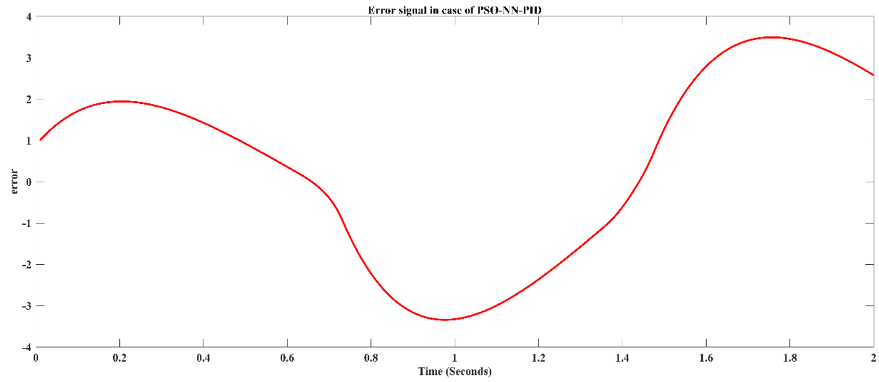

Figure 7 Error Signal in case of PSO-NN-PID controller

The graph showing variation in Kp, Ki and Kd with respect to time is shown in Figure 12.

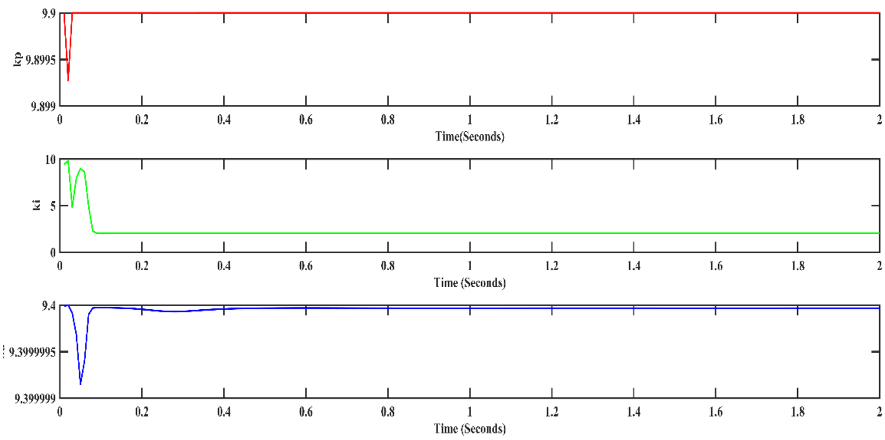

Figure 8 Variation or Kp, Ki and Kd values with time

The comparative bar graph of MSE in the case of three controllers, ZN-PID, BPNN-PID and PSO-NN-PID is shown in Figure 13.

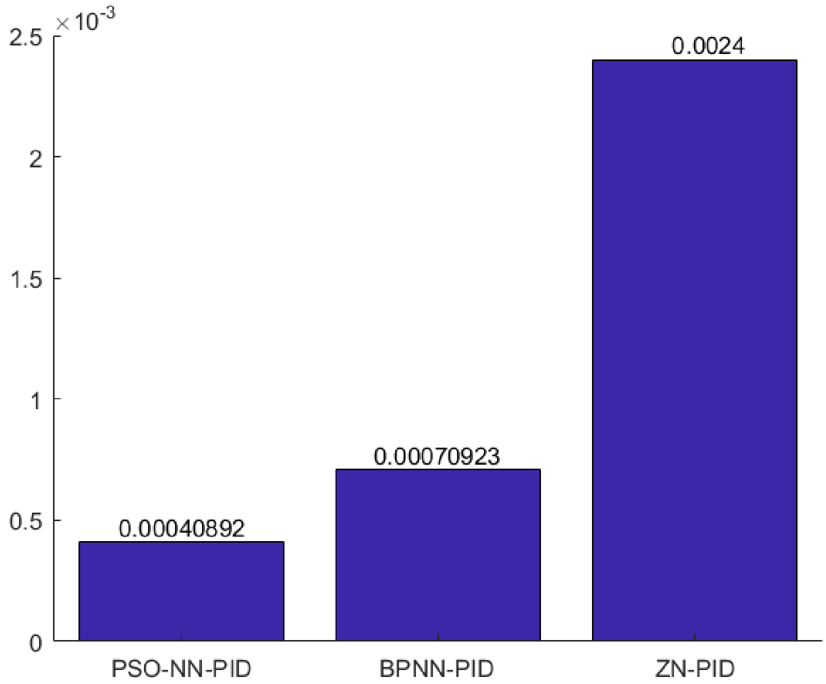

Figure 9 Comparative bar graph of MSE values for various controllers

6.1 Discussion

A comparison of simulation results of the three controllers are presented in the above section. ZN- tuned PID, BP tuned PID-NN and PSO tuned PID-NN are compared on a non-linear jacketed CSTR plant. From the results it is evident that in case of PSO tuned PID-NN controller there is a decrease of 33% in rise time and 47.7% in overshoot. To test the robustness of the proposed scheme a disturbance signal of amplitude 1.5 was applied. In case of PSO tuned PID-NN controller there is slight increase in overshoot but it can be observed that the controller was able to subside the disturbance quickly in 1.88 seconds as compared to ZN tuned PID taking 2.77 seconds and BP tuned PID-NN taking 2.36 seconds. Figure 6 shows a comparative graph of rise time, peak time, overshoot, peak time and settling time. It can be seen from the graph that there is a reduction in rise time, overshoot and settling time in case of PSO-PID-NN. Figure 8 & 9 represent the variation of control signal and error signal with time in case of PSO-PID-NN. Figure 10 represents the variation of $K_p, K_i \; and \; K_d$ with respect to time in case of PSO-PID-NN. Figure 11 shows a comparative bar graph of different controllers. The value of cost function MSE is least in case of PSO-PID-NN and largest in case of ZN tuned PID.

7. Conclusion

The study proposes a novel NN based PID controller based on PSO tuning method. The two techniques used combine the advantage of high computational speed, in case of neural network and accuracy in case of PSO. The NN is modified as a PID like structure where the hidden layer has three neurons and the output layer has three weights which are proportional to P, I and D gains respectively. The weights are optimized using PSO algorithm. The proposed technique is compared with Back propagation algorithm tuned NN-PID controller. The simulation results show that overshoot reduces to 23.13% to

44.08% in case of PSO-NN-PID. The rise time also reduces from 0.18 seconds to 0.1283 seconds. The controller developed was also tested for robustness under the application of disturbance and is found to be better than BP based NN-PID controller and conventionally tuned PID controller.

8. Statements and Declarations

Compliance with Ethical Standards

Ethical Approval

Not Applicable

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflict of interest

The authors certify that there is no conflict of interest with any individual/organization for the present work.

Informed Consent

Not Applicable

Authorship Contributions

All authors have contributed significantly in the study. The study conception and design were performed by Snigdha Chaturvedi and Narendra Kumar. Analysis and interpretation of results were done by Rajesh Kumar. Draft manuscript preparation was done by Snigdha Chaturvedi, Narendra Kumar and Rajesh Kumar. All authors reviewed the results and approved the final version of the manuscript

9. References

[1] J. Kang, W. Meng, A. Abraham, and H. Liu, “An adaptive PID neural network for complex nonlinear system control,” *Neurocomputing*, vol. 135, pp. 79–85, 2014, doi: 10.1016/j.neucom.2013.03.065.

[2] K. H. Ang, G. Chong, S. Member, and Y. Li, “PID Control System Analysis , Design , and Technology,” vol. 13, no. 4, pp. 559–576, 2005.

[3] C. Zhao and L. Guo, “Control of Nonlinear Uncertain Systems by Extended PID,” vol. 9286, no. c, pp. 1–8, 2020, doi: 10.1109/TAC.2020.3030876.

[4] “1-0-1-k-j-astrom-pid-controllers-theory-design-and-tuning-2ed.pdf.” .

[5] N. Method, “Ziegler - Nichols Method : Ziegler - Nichols Tuning Chart :,” vol. 25, no. 10, pp. 1–2, 2015.

[6] R. Kumar, S. Srivastava, and J. R. P. Gupta, “Artificial Neural Network based PID controller for online control of dynamical systems,” *1st IEEE Int. Conf. Power Electron. Intell. Control Energy Syst. ICPEICES 2016*, pp. 1–6, 2017, doi: 10.1109/ICPEICES.2016.7853092.

[7] S. E. I. Hasseni, L. Abdou, and H. E. Glida, “Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms,” *Evol. Intell.*, vol. 14, no. 1, pp. 61–73, 2021, doi: 10.1007/s12065-019-00312-8.

[8] V. Srivastava and S. Srivastava, “Control of continuous stirred tank reactor (CSTR) using nature inspired algorithms,” *Journal of Information and Optimization Sciences*, vol. 40, no. 2. pp. 329–338, 2019, doi: 10.1080/02522667.2019.1578092.

[9] S. Chatterjee and V. Mukherjee, “PID controller for automatic voltage regulator using teaching-learning based optimization technique,” *Int. J. Electr. Power Energy Syst.*, vol. 77, pp. 418–429, 2016, doi: 10.1016/j.ijepes.2015.11.010.

[10] A. W. Lins and R. Krishnakumar, “Tuning of PID controller for a PV-fed BLDC motor using PSO and TLBO algorithm,” *Applied Nanoscience (Switzerland)*. 2022, doi: 10.1007/s13204-021-02272-x.

[11] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, “African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems,” *Comput. Ind. Eng.*, vol. 158, no. January, p. 107408, 2021, doi: 10.1016/j.cie.2021.107408.

[12] R. K. Mandava and P. R. Vundavilli, “An adaptive PID control algorithm for the two-legged robot walking on a slope,” *Neural Computing and Applications*, vol. 32, no. 8. pp. 3407–3421, 2020, doi: 10.1007/s00521-019-04326-2.

[13] G. Q. Zeng, X. Q. Xie, M. R. Chen, and J. Weng, “Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems,” *Swarm Evol. Comput.*, vol. 44, pp. 320–334, 2019, doi: 10.1016/j.swevo.2018.04.008.

[14] T. Varshney and S. Sheel, “Improved NN-PID control of MIMO systems with PSO-based initialisation of weights,” *Int. J. Autom. Control*, vol. 8, no. 2, pp. 158–172, 2014, doi: 10.1504/IJAAC.2014.063362.

[15] S. Bennett, “Development of the PID Controller,” *IEEE Control Syst.*, vol. 13, no. 6, pp. 58–62, 1993, doi: 10.1109/37.248006.

[16] O. Ayokunle, K. Akingbade, and F. Dahunsi, “Annual Reviews in Control The dilemma of PID tuning ☆,” *Annu. Rev. Control*, no. May, 2021, doi: 10.1016/j.arcontrol.2021.05.002.

[17] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, and M. F. Silva, “Comparison of PID Controller Tuning Methods : Analytical / Classical Techniques versus Optimization Algorithms,” pp. 533–538, 2017.

[18] L. dos Santos Coelho, “Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach,” *Chaos, Solitons and Fractals*, vol. 39, no. 4, pp. 1504–1514, 2009, doi: 10.1016/j.chaos.2007.06.018.

[19] I. M. Mehedi, H. S. M. Shah, U. M. Al-Saggaf, R. Mansouri, and M. Bettayeb, “Fuzzy PID Control for Respiratory Systems,” *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/7118711.

[20] A. T. Yimchunger, D. Acharya, and D. K. Das, “Particle Swarm Optimization based PID-Controller Design for Volume Control of Artificial Ventilation System,” *2020 IEEE Calcutta Conf. CALCON 2020 - Proc.*, vol. C, pp. 278–282, 2020, doi: 10.1109/CALCON49167.2020.9106480.

[21] R. V. Rao and V. Patel, “An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems,” *Sci. Iran.*, vol. 20, no. 3, pp. 710–720, 2013, doi: 10.1016/j.scient.2012.12.005.

[22] S. Sakthiya Ram, C. Kumar, A. Ramesh Kumar, and T. Rajesh, “Hybrid optimization techniques based automatic artificial respiration system for corona patient,” *Automatika*, vol. 63, no. 2. pp. 226–243, 2022, doi: 10.1080/00051144.2022.2031538.

[23] M. J. Mahmoodabadi and N. Danesh, “Gravitational search algorithm-based fuzzy control for a nonlinear ball and beam system,” *J. Control Decis.*, vol. 5, no. 3, pp. 229–240, 2018, doi: 10.1080/23307706.2017.1403864.

[24] J. Pongfai, X. Su, H. Zhang, and W. Assawinchaichote, “A novel optimal PID controller

autotuning design based on the SLP algorithm," *Expert Syst.*, vol. 37, no. 2, pp. 1–15, 2020, doi: 10.1111/exsy.12489.

[25] H. B. Kazemian, "Developments of fuzzy PID controllers," *Expert Syst.*, vol. 22, no. 5, pp. 254–264, 2005, doi: 10.1111/j.1468-0394.2005.00316.x.

[26] T. Piraisoodi, W. I. Maria Siluvairaj, and M. A. K. Kappuva, "Multi-objective robust fuzzy fractional order proportional–integral–derivative controller design for nonlinear hydraulic turbine governing system using evolutionary computation techniques," *Expert Syst.*, vol. 36, no. 2, pp. 1–15, 2019, doi: 10.1111/exsy.12366.

[27] M. Issa, "Enhanced Arithmetic Optimization Algorithm for Parameter Estimation of PID Controller," *Arab. J. Sci. Eng.*, 2022, doi: 10.1007/s13369-022-07136-2.

[28] Srivastava, V., Srivastava, S., Chaudhary, G., & Blanco Valencia, "Performance improvement and Lyapunov stability analysis of nonlinear systems using hybrid optimization techniques," *Expert Syst.*, 2022.

[29] C. N. Huang and A. Chung, "An Intelligent Design for a PID Controller for Nonlinear Systems," *Asian J. Control*, vol. 18, no. 2, pp. 447–455, 2016, doi: 10.1002/asjc.1074.