An Intelligent Neural Network Controller for Non-Linear CSTR Process Control

Baranilingesan, I.*+
Department of Electrical and Electronics Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamilnadu, INDIA

Deepa, S.N.
Department of Electrical Engineering, National Institute of Technology Arunachal Pradesh, Jote, Papum Pare District, Arunachal Pradesh, INDIA

Yagambal Jayalaskshmi, N.
Department of Electrical and Electronics Engineering, Dr. Mahalingam College Engineering and Technology, Coimbatore, Tamilnadu, INDIA

ABSTRACT: *The non-linear process control is the most important problem of statement in chemical industries. Stirred tanks are frequently used as industrial reactors, where a chemical component of a flow stream resides in the tank for a period of time before proceeding to other steps in a chemical process. In this research article, a computational intelligence-based controller is introduced for CSTR concentration control model. The proposed RBFNN model is optimally tuned by two-hybrid optimization strategies based on combining the best features of Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), and their variants such as Deterministic Particle Swarm Optimization Algorithm, and Differential Gravitational Search algorithm (DPSO-DGSA). An experiment is conducted to examine the effectiveness of the proposed controller methodology to compare it to other state-of-the-art approaches.*

KEYWORDS: *CSTR; RBFNN; Hybrid DPSO-DGSA; concentration control; Optimization.*

INTRODUCTION

The CSTR is a complicated and non-linear chemical process system, the non-linearity imposes a huge challenge on system identification and process controller design. The CSTR continuously mixes the reactant in the reactor, and the processed output is taken out through the outlet. The concentration of the product is to be maintained at a desired level which is affected by the temperature of the reactor. The outer Jacket of the reactor is circulated with water to remove the heat generated during exothermic reactions. But this process is highly non-linear in nature so the concentration control is a highly challenging task. In the proposed study, an intelligent neural network controller is developed using hybrid optimization techniques to handle the non-linearity of the system. The temperature of the reactor is

* *To whom correspondence should be addressed.*
+ *E-mail: barani.ling@gmail.com*
1021-9986/2023/3/977-988 12/$/6.02

continuously monitored and controlled such that the concentration is maintained at the desired set point. To address this, issue numerous researches have been carried out for the past few decades based on computational intelligence strategies. The performance of the classic controllers is greatly influenced by the model parameters that can be improved by adopting the optimization algorithms. The test results on employing Elephant Herding Optimization (EHO) over the CSTR system have reported improved system performances, *Elhosseini*, *Sehiemy*, *Rashwan*, and *Gao* (2019). The Fuzzy-based strategy has been introduced to handle the uncertainty of CSTR system, *Apale* and *Patil* (2019). The hybridization of the heuristic and meta-heuristic algorithms was employed to tune the PID controller parameters such as Particle Swarm Optimization (PSO), Gravitational Search Algorithm(GSA), Artificial Bee Colony Optimization (ABC), Genetic Algorithm (GA) improved the system performance considerably, *Goud* and *Swarnkar* (2019a, 2019b). Srivastava and Srivastava (2019) have compared the performance of a conventional PID controller that is optimally framed by PSO and Teaching-Learning-Based Optimization (TLBO) algorithms. *Salahshour*, *Malekzadeh*, *Gordillo*, and *Ghasemi* (2019) presented a PI Controller with an adaptive mechanism based on multi-level network quantization (MLQNN) optimally framed by PSO algorithm, the optimally tuned model demonstrated improved performance as compared to the perceptron network. The fitness index-based system identification for the CSTR plant is developed with online and offline estimation methods. *Simorgh*, *Razminia*, and *Shiryaev* (2020). The Dynamic optimization (DO) problem on CSTR is solved by the CVP strategy using the (non-dominated sorting genetic algorithm) NSGA II algorithm. *Shirude* and *Padhiyar* (2019). The online optimization scheme for filtering nonlinearity in a multivariable system is proposed and estimated to be outperformed by the (Model Predictive Controller) MPC controller *Ma* (2020). *Shoga, Thelkar, Bharatiraja, et al*. (2019) have presented an online auto-tuning controller-based recursive least square algorithm with improved system performance as compared to (Intrinsic Mode Controller) IMC and MPC controller. The Fractional order PID controller optimally framed by GA has shown an improved system than to conventional PID controller *Jaiswal, Kumar, Seepana,* and *Babu* (2020). By using GA to optimize the controller settings, the performance of a traditional Multi-resolution PID controller is improved. To optimize the PID controller design, a hybrid combination of GA-PSO is given. *Utkarsh*, *Kantha*, and *Jatoth* (2016).

The Neural network models showed an incredible response to system dynamics and uncertainty, in introducing a neural work-based adaptive backstepping controller. *Alshammari*, *Mahyuddin*, and *Jerbi* (2020). Radial Basis Function Neural Network (RBFNN) controller is widely used in CSTR controller design, and a special quasi-LPV model is developed based on its parameter variation rate. *Zhou*, *Peng*, *Zhang*, and *Zeng* (2019). A RBF-MPC model is developed based on linear and non-linear optimization strategies. *Li*, *Jiang*, and *Han* (2019). A robust controller is designed based on the radial basis technique with 2-stage scheduling quasi min-max by *Zhou*, *Peng*, *Zeng*, and *Tian* (2019). The Nonlinear Autoregressive Moving Average (NARMA-L2) controller designed by *Abdullah*, *Yee*, *Mohamed*, *et al*. (2016) on composition control of isothermal CSTR by manipulating the composition of the input parameters. In order to predict the overall reticle of the system the RBFNN is used. *Bagheri et al*. (2015) introduced the modeling of sequencing batch reactor based on *MLPANN* and *RBANN*. *Li* (2014) proposed a neural network-based adaptive controller, by presenting a new Lyapunov approach and the dead zone input effect. *Alexandridis*, *Stogiannos*, *Loukidis* (2014) proposed a control algorithm for approximating the inverse dynamics based on RBFNN, the training of the neural network carried out by fuzzy means algorithm. For the process that possesses multiple steady states, *Alexandridis* and *Sarimveis* (2011) presented a control scheme for a nonlinear CSTR process based on a model predictive controller and modeled by the RBFNN network.

The following inferences are made from the above literature study:

1. The performance of classic controllers is improved by feeding optimal controller parameters based on swarm intelligence and evolutionary-based optimization algorithms.

2. On comparing with conventional controllers the neural network controllers can handle system uncertainty and disturbance rejection with better adaptability.

3. Among numerous neural network controllers, the Radial basis function model is found unique in problem identification and system dynamics approximation.

The RBFNN is a type of multilayered feed-forward network with its advantages such as better generalization ability, pattern classification, function approximation, recognition and faster learning abilities due to locally tuned parameters. In general, the learning parameters and hyperparameters of the network is initialized randomly and continuously updated over the course of iteration. The random initialization of the model parameters may result in poor performance of the controller; a hybrid optimization approach is needed to deal with this problem by combining the best features of (PSO) and (GSA) and their variants such as a combination of Deterministic PSO (DPSO) and Differential GSA (DGSA) are proposed in this article.

MATHEMATICAL MODELLING OF THE CSTR SYSTEM

The proposed system performs an irreversible first-order exothermic reaction as represented in Fig. 1. The reactor is continuously fed with the fluid and the system reaction is represented as A → B. In order to ensure that the product is well combined, a continual stirring mechanism is employed. There must be no difference in temperature and concentration between the reactor's output stream and the reactor's internal fluid. Fluid is continuously circulated through reactor in order to accompany the procedure. To ensure that the product concentration in the reactor is unaffected by temperature changes, the reaction wastes energy that is then transmitted to the reactor's walls and eliminated by the circulating fluid in the jacket.

The following assumptions were made in this process,

- The perfect mixing of the fluids is expected
- The energy balancing around the jacket need not be considered as the cooling jacket is controlled directly.
- The shaft work assumed to be minimal or even negligible.
- Assumed to have constant parameter values and constant volume.

The following transfer function was derived based on the steady-state operating conditions of CSTR and used in MATLAB, *Nekoui et al.* (2010)

$$G(s) = -\frac{1.3083}{(13.5102s+1)(6.2417s+1)}e^{-4-8961s} \tag{1}$$

For the considered CSTR system the state space model derived and is as given by,

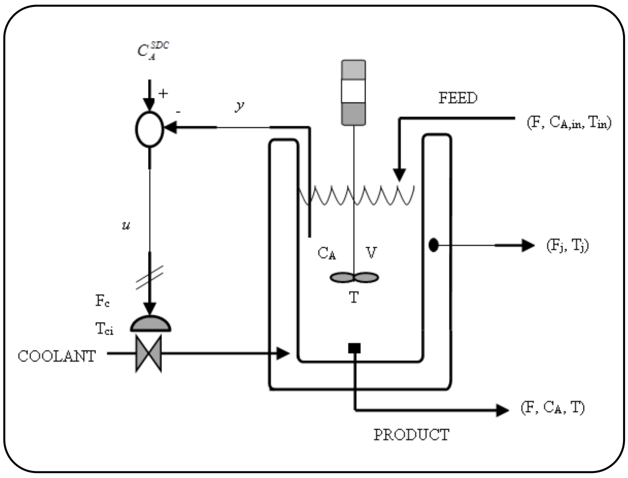

Fig. 1: CSTR Process with cooling jacket.

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-0.1239 & -0.00017\\ 7.4454 & -0.05894\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0.003151\\ -0.81985\end{bmatrix}u$$
$$y = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{2}$$

Controlling the reactor temperature (T) is the primary goal of the proposed system, and this is accomplished through the use of a variable flow rate coolant supply.

THE OUTLINE OF RADIAL BASIS FUNCTION NEURAL NETWORK (RBFNN)

Broomhead and *Lowe* (1988) proposed the RBFN model; later, *Moody* and *Darken* (1989) presented a multiphase RBFN approach with normalized activation functions. It is able to more accurately approximate any continuous function (*Park* and *Sandberg* (1991); *Liao, Fang,* and *Nuttle* (2003)). Fig. 2, $X \in \Re^m$ is the input vector, Ω is the radial basis function $\phi(X, Z_i)$, $Z_i \in \Re^m$ is the i^{th} node hidden unit. The set of functions possessed by the neurons in the hidden layers is the primary foundation for the network's functionality.

The architecture of the network is described by a three-layer structure, the input layer, the hidden layer, and the output layer. The RBFN is meant for its linear optimization because of its single hidden layer framework. However, it becomes non-linear when the size of the basis increases or by increasing the number of hidden layers. Fig. 2 shows the architecture of the RBFNN model, where the bell-shaped curves presented in the hidden layer represent the radial basis function in future space, and also in the

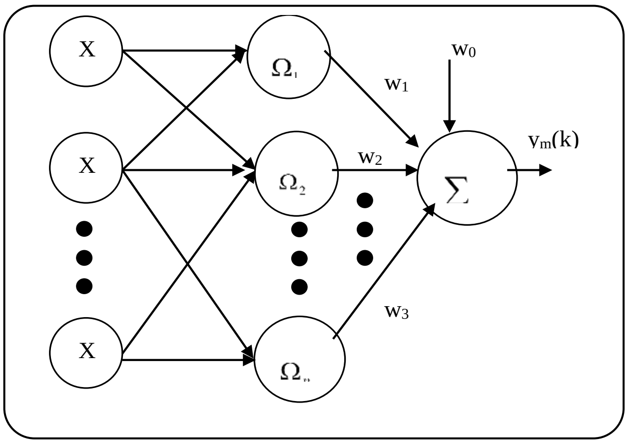

Fig. 2: Architecture of RBFNN.

RBFNN model, there are no weight vectors assigned between the input to hidden nodes. The input signal is passed from the input layer to the output layer through the hidden layer, where the hidden neuron processes the input signal through the Gaussian activation function and the corresponding output is generated. For the given input vector of X, the entire feature space is partitioned by Gaussian functional nodes, where the output signal generated corresponding to the distance between the center of the neuron and the input vector. The Gaussian function is the basis function granted for common choice because it replenishes local features without requiring many changes to the previous mapping and has global mapping generalization capability, *Strumiłło et al.* (2003). In the RBFNN framework, the center of the input neuron is represented by weight vectors, which are precomputed in such a way that, the entire space is filled with the receptive field of input neurons, the input vectors lining in close Euclidean space is represented as receptive field, the corresponding training algorithm is presented as follows:

The learning algorithm of RBFNN model is aimed update the model parameters for each iteration such that the mean square error of the training data is reduced. Initially the weight and basis coefficient of RBFNN model are randomly chosen between 0 to 1. The employed activation function is Gaussian function with gradient descent learning rule of learning rate is 0.2 The RBFNN training algorithm is presented below:

Step 1: Initialization

Step 2: Until attaining the stopping condition follow the steps from 3 to 8.

Step 3: For every input, set follow the steps 4 to 7.

Step 4: Each node of the input layer is assigned with the input vector $X=[x_1,x_2,...,x_n]^T$, where 'n' is the number of the input neurons and transmits to hidden neurons in the hidden layer.

Step 5: A sufficient number of centers are identified for the basis function such that a sufficient number of sampling is made with input vectors.

$C_j=[c_{j1},c_{j2},.c_{ji}..,c_{jn}]^T$, j= 1,2,…,m and i= 1,2,…,n

Where, C_j is the center vector at a node j of the network?

The radial width of the network,

$B=[b_1,b_2,...,b_m]^T$

Step 6: For the hidden layer with a radial vector $h=[h_1,h_2,...,h_m]^T$, the output is calculated by applying the activation function, 'm' is the number of hidden neurons.

The Gaussian function applied at node follows the mathematical function,

$$h_j=\exp\left[-\frac{\left\|x-c_j\right\|^2}{2b_j^2}\right] \quad (3)$$

$$y_m(k)=W^Th=w_1h_1+w_2h_2+...+w_mh_m \quad (4)$$

Where, W is the weight vector of the network.

Step 7: Calculate the error of the network,

$$E(k)=\frac{1}{2}(y(k)-y_m(k))^2 \quad (5)$$

Step 8: The weight is adjusted based on gradient decent algorithm.

DISCUSSION ON PROPOSED HYBRID EFFECTIVE OPTIMIZATION ALGORITHMS FOR THE TRAINING OF RADIAL BASIS NEURAL NETWORK

During the process of training the weight vector of neural network is randomly initialized that affects the convergence speed of the model. So, in this section evolutionary-based hybrid optimization algorithms are developed to optimize the parameters of Radial Basis Neural Networks (RBFNN). The network is fed with optimal weight values updated by the proposed algorithms for each iteration until a stopping criterion of minimum ISE or a maximum number of iterations is reached.

In this study, the PSO and GSA has been chosen over other optimization strategies because of their simple algorithmic framework. The local stagnation issues faced by individual algorithms are addressed by the process of hybridization.

PSO: A quick overview

PSO is social inspired swarm intelligence algorithm that achieves better solution based on the local and global behaviour of the particles, and they tend to achieve solutions individually before combining the better solutions obtained by neighbouring individuals. As a result of the neighborhood particles influencing an individual particle's position in a population, the individually best solution of a particle in the population becomes the global best one Cl_{bt}.

The position of the particle p_i^{k+1} is adjusted by,

$$p_i^{k+1} = p_i^k + \phi_i^{k+1} \tag{6}$$

Where,

ϕ_i^{k+1} - The velocity component signifies the step size, and the velocity equation may be used to compute it:

$$\phi_i^{k+1} = w\phi_i^k + c_1 r_1\{Pr_{bti} - p_i^k\} + c_2 r_2\{Gl_{bt} - p_i^k\}\ , \tag{7}$$

Where, *w*- the weight of inertia,

c_1 and c_2 coefficients of acceleration,

$r_1, r_2 \in U\ (0,1)$ are arbitrarily chosen numbers.

An introduction to DPSO

The conventional PSO algorithm has the main advantage is its simple structure and ease of implementation, but has its limitations due to its random factors. The simple deterministic PSO system proposed by *Clerc* and *Kennedy* (2002), that suffers from the computing complexity of eigen value estimation. The novel deterministic particle swarm optimization adopted in the proposed study is realized based on the novel deterministic procedure described in *Jin'no* (2009).

The canonical form of DPSO,

$$\begin{vmatrix} \phi_i^{k+1} \\ y_i^{k+1} \end{vmatrix} = \begin{vmatrix} \delta & -\omega \\ \omega & \delta \end{vmatrix} \begin{vmatrix} \phi_i^k \\ y_i^k \end{vmatrix} \tag{8}$$

$$y_i^k \equiv p_i^k - z_i^k \tag{9}$$

$$z_i^k = \frac{c_1 Pr_{bti} + c_2 Gl_{bt}}{\psi}\ ,\ \psi = c_1 + c_2 \tag{10}$$

Where z_i^k is the desired fixed point, $\delta \in \Re$ and $\omega \in \Re$

If the velocity component becomes less than 1.00×10^{-8}, then it is multiplied by 5.12×10^8.

Discussion about the GSA

The Newtonian theory of gravitation is fundamental for gravitational search algorithms, and the search agents are simply a collection of masses, *Saryazdi*, *Rashedi*, and *Nezamabadi-Pour* (2009). Based on Newtonian theory, the law of gravitation is represented by the equation,

$$F = G\frac{M_1 M_2}{R^2} \tag{11}$$

Where,

M_1 and M_2 -particle masses, *R*- distance between the particle masses, *G* -gravitational constant, *F*- magnitude of the gravitational force, According to Newton's second law of gravitation the relation between applied force F, on a mass M and its corresponding acceleration given by,

$$a = \frac{F}{M} \tag{12}$$

On inspiring how the individual mass gets accelerated based on their resultant force act on them as shown in Fig. 3, the position and the velocity of the particles in the GSA algorithm is adjusted,

$$p_i^{k+1} = p_i^k + \phi_i^{k+1} \tag{13}$$

$$\phi_i^{k+1} = \phi_i^k + a\ . \tag{14}$$

Configuration of GSA

For the mathematical modeling of *N*' considered particles,

$$p_i^k = p_g = [p_1, p_2, p_3, \cdots, p_j, ..., p_n]$$

The mass acceleration in time-distance is given by,

$$a_i^k(t) = \frac{F_i^k(t)}{M_i(t)} \tag{15}$$

The total force exerted on particle j is given by,

$$F_i^k(t) = \sum_{j\in 1, j\neq i}^{n} rand_j F_{ij}^k(t) \tag{16}$$

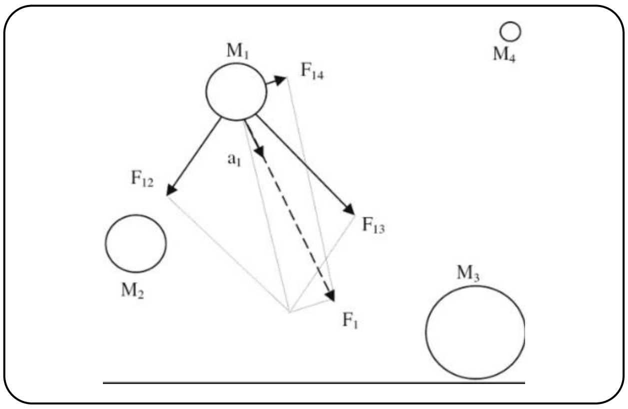

Fig. 3: Gravitational search algorithm movement of mass.

Where, $rand_j$ - a uniform random number between 0 and 1

$$F_{ij}^{k}(t) = G(t)\frac{M_i(t)M_j(t)}{R_{ij}(t)+\varepsilon}\left(p_j^k(t)-p_i^k(t)\right) \tag{17}$$

Where, $M_i(t)$ and $M_j(t)$ -the masses of objects i and j, $G(t)$ - gravitational constant at time t

ε -small constant, $R_{ij}(t)$ - the Euclidean distance between i and j objects and is given by,

$$R_{ij}(t) = \left\| d_i(t) - d_j(t) \right\|^2 \tag{18}$$

Overview of DGSA

GSA's performance is mainly dependent on its ability to conduct exploration and exploitation, with the global variant exhibiting exploration capability, which is intended for faster convergence as all agents are drawn to the best mass agent, and the local variant of GSA exhibiting exploitation capability, Neighbouring entities are informed of which agent has the best position.

DGSA is a proposed algorithm that combines both the exploration and exploitation properties of GSA. A differential parameter is incorporated that influences the balance between the global and local search directions, Rizwana and Deepa (2017). The velocity update for global variant of p_i be gv_i^{k+1} and for local variant the velocity update of p_i be lv_i^{k+1}.

$$g\phi_i^{k+1} = rand_i \times g\phi_i^k + a_i^k \tag{19}$$

$$l\phi_i^{k+1} = rand_i \times l\phi_i^k + a_i^k \tag{20}$$

The following are possible DGSA velocity and position update expressions:

$$\phi_i^{k+1} = \gamma g\phi_i^{k+1} + (1-\gamma) l\phi_i^{k++1} \tag{21}$$

$$p_i^{k+1} = p_i^k + \phi_i^{k+1} \tag{22}$$

The differential parameter $\gamma \in [0,1]$, for $\gamma = 1$ stands for standard global GSA and $\gamma = 0$ symbolises local GSA. To attain balance between global and local search ability the differential parameter is varied between 0 and 1, thereby it possesses the capacity to enhance the algorithm's exploitation and exploration abilities.

The equation (17) can be written as,

$$F_i^k(t) = \sum_{\substack{j\in pbest \\ j\neq i}} rand_j \times F_{ij}^k(t) \tag{23}$$

PROPOSED HYBRID PSO-GSA OPTIMIZATION ALGORITHM

For enhanced solution discovery, the suggested hybrid PSO-GSA combines the PSO's social thinking capacity with the GSA's local search capabilities. As such, the hybrid PSO-GSA position and velocity equations and the modeled heuristic hybrid learning process incorporates the same.

$$p_i^{k+1} = p_i^k + \phi_i^{k+1} \tag{24}$$

$$\phi_i^{k+1} = w\phi_i^k + \underset{\text{GSA Coefficient}}{\underset{\downarrow}{c_1 r_1 \times a_i^k}} + \underset{\text{PSO Coefficient}}{\underset{\downarrow}{c_2 r_2 \{Gl_{bt} - p_i^k\}}} \tag{25}$$

Where, 'a_i^k' it is a significant GSA parameter that is used in its local search mechanism and 'Gl_{bt}' expresses the social thinking ability of PSO, where both are brought together to carry out the hybrid search mechanism, the flowchart of the algorithm is presented in Fig. 4.

PROPOSED HYBRID DPSO – DGSA OPTIMIZATION ALGORITHM

The hybrid DPSO-DGSA algorithm combines the characteristics of deterministic particle swarm optimization with the better achievement of exploration and exploitation capability of DGSA as an efficient heuristic learning algorithm for the RBFNN controller. The modified velocity and position expression for hybrid DPSO – DGSA can be expressed as,

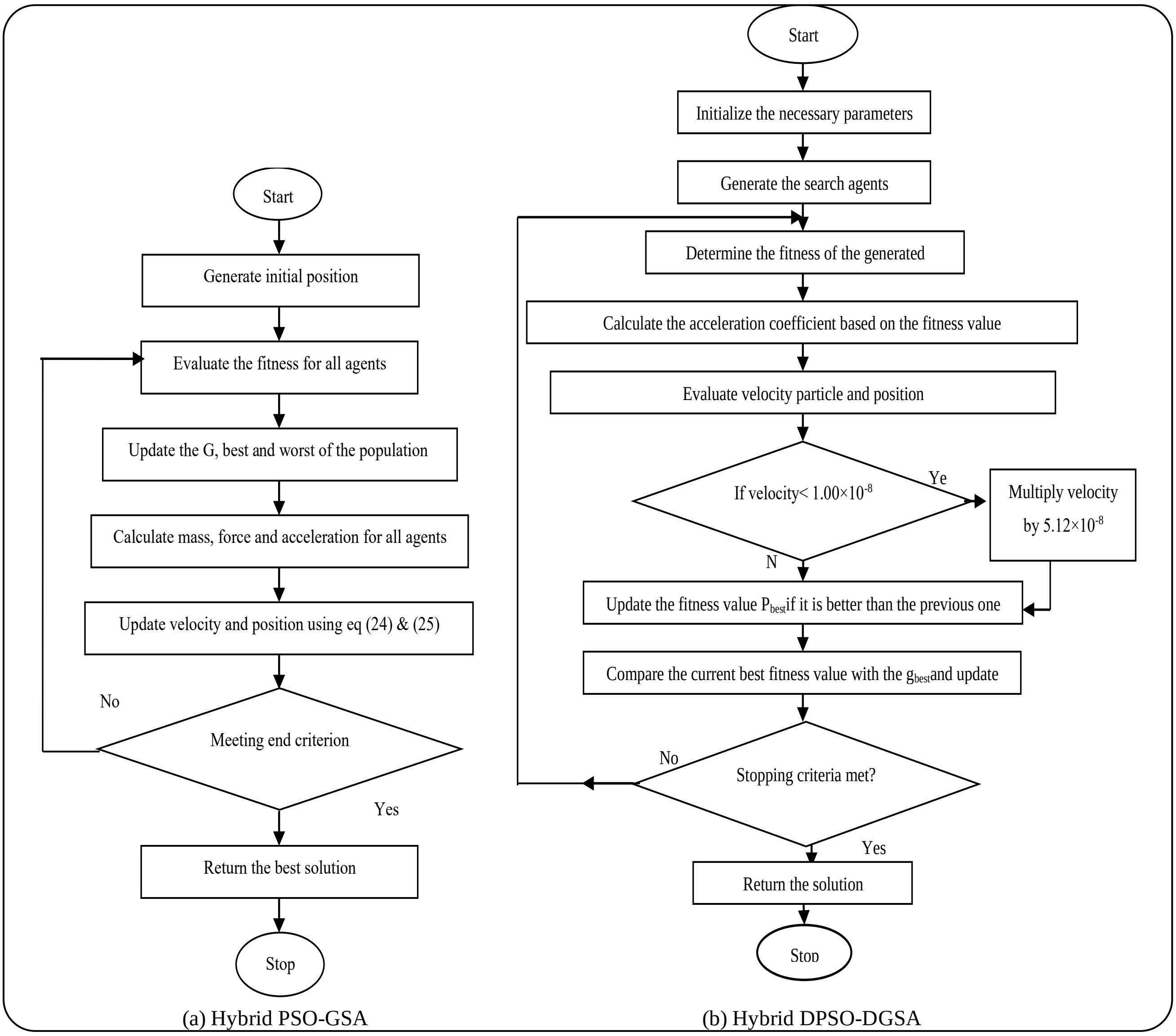

Fig. 4: Flow chart of the hybrid optimization algorithms.

$$\phi_i^{k+1} = \delta\phi_i^k - \omega p_i^k + a_i^k z_i^k \omega \quad (26)$$

$$p_i^{k+1} = z_i^{k+1} + \omega\phi_i^k + \delta(p_i^k - z_i^k) \quad (27)$$

The expression above depicts the hybridization of DPSO and DGSA. The flow chart of the proposed DPSO-DGSA optimization algorithm is presented in Fig. 4.

For all the proposed algorithms, the stopping criteria are 100 iterations and the fitness function framed is to reduce the integral square error of the system. When the stopping criteria are met, the algorithms return the best global solution based on their fitness function evaluation.

PROPOSED TUNING OF RBFNN WITH TRAINING ALGORITHMS

In the MATLAB R2020a environment, the proposed evolutionary optimization algorithm-based RBFNN controller model of a nonlinear CSTR is implemented and simulated. The modelled controller optimises and fine-tunes the system's performance with greater precision. Fig. 5 depicts the typical structure of a CSTR system employing an evolutionary algorithm-based modified RBFNN controller, where 'k' sample time. The control action is focused on controller's predicted values rather than the actual values of the CSTR plant's output, the input –output parameters are segregated into set of training

Table 1: Parameters of the proposed algorithms.

Parameters	GSA	Parameters	PSO	Parameters	DPSO	Parameters	DGSA
Number of search agents	50	Particle Size	50	Population size	50	Number of search agents	50
Gravitational Constant	100	Acceleration constants c_1 and c_2	2.0	ω	0.22	Gravitational Constant	100
A	20	w_{min}	0.2	δ	0.80	γ	0.5
E	$2.2204e^{-16}$	w_{max}	1.0	Number of iterations	500	Number of iterations	500
Number of iterations	500	v_i^{max}	1.0				
		v_i^{min}	0.01				
		Maximum Generations	500				

and testing data. The entire data is segregated into 70% of data is employed for training and remaining 30 % is employed for testing. During the process of training 5-fold cross validation is employed during training process. The entire training data is segregated into 5 sets, while the four parts are employed for training the remaining one part is employed for testing. So that at the end of the training process all the set would have been trained and tested for at least once.

In terms of error minimization, the modelled RBFNN controller tends to minimise the performance index 'J'.

$$e(k+1) = r(k+1) - y'(k+1) \tag{28}$$

$$J = \int e(k)^2 \tag{29}$$

The control signal 'u(k)' is updated function is represented as,

$$u(k+1) = u(k) - \xi \frac{\partial J}{\partial u(k)} \tag{30}$$

The system identification process is the first stage of the control strategy, and the trained RBFNN model represents the system dynamics. Predicted plant outputs are used to manage the plant, and the network uses this information to forecast future outputs based on the plant's prior output history.

The following is a summary of the model's training procedures:

Step 1: Start the algorithm

Step 2: Feed the data inputs into the RBFNN model.

Step 3: Follow the steps from 3 to 8.

Step 4: Call the optimization algorithm and initialize the parameters.

Step 5: For the better fitness attained return the tuned weights values

Step 6: Employ these weight values over the RBFNN model

Step 7: Return the process

Step 8: Results of the solutions found.

Step 9: Until attaining the stopping condition repeat the steps from 4 to 9.

In the proposed training algorithm based on an effective evolutionary learning scheme, the network pre-trains itself with the conventional procedure which leads to the attainment of faster convergence with better system performance. Table 1 displays the parameter settings for the optimization procedures under consideration. These methods of learning are applied to a radial basis neural network model as well, and their performance is shown in the next portions of this paper.

DISCUSSIONS ON THE PROPOSED RBFNN CONTROLLER'S PERFORMANCE

Feed concentration CA = 10 mol/l, F/V = 0.5708 s-1, and product concentration = 1.117 mol/l are all standard operating conditions. It is necessary to conduct a comparative analysis of the suggested evolutionary algorithms-based RBFNN controller model's closed-loop performance characteristics to verify the controller's performance.

The servo response of the considered system is analyzed using the proposed evolutionary-based RBFNN controllers with a step change in the desired concentration set point from 0.823 to 0.1. Figs. 6 and 7 show the set point tracking of controller responses as a function of temperature and concentration. The performance of individual algorithms validated by minimization

Table 2: Training and Testing Efficiency of the proposed RBFNN controller.

Number of Hidden Neurons	PSO-RBFNN		GSA-RBFNN		DPSO-RBFNN		DGSA-RBFNN		PSO-GSA-RBFNN		DPSO-DGSA-RBFNN	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
2	70.9	70.5	72.4	71.3	72.6	73.0	74.5	74.2	75.4	73.4	77.8	73.4
3	76.3	74.3	77.3	76.5	78.4	76.1	78.9	76.4	79.7	79.4	80.4	79.4
4	79.3	78.3	80.9	77.5	81.2	78.4	81.9	77.5	82.4	81.9	85.3	82.4
5	86.2	86.3	87.9	88.2	90.2	90.8	93.9	94.0	95.1	95.7	97.4	97.9
6	86.8	84.4	88.0	85.3	92.3	85.9	94.3	88.9	95.3	89.6	98.0	93.9
7	86.2	78.4	87.1	84.8	92.7	85.2	93.2	87.5	94.3	89.2	97.4	91.2

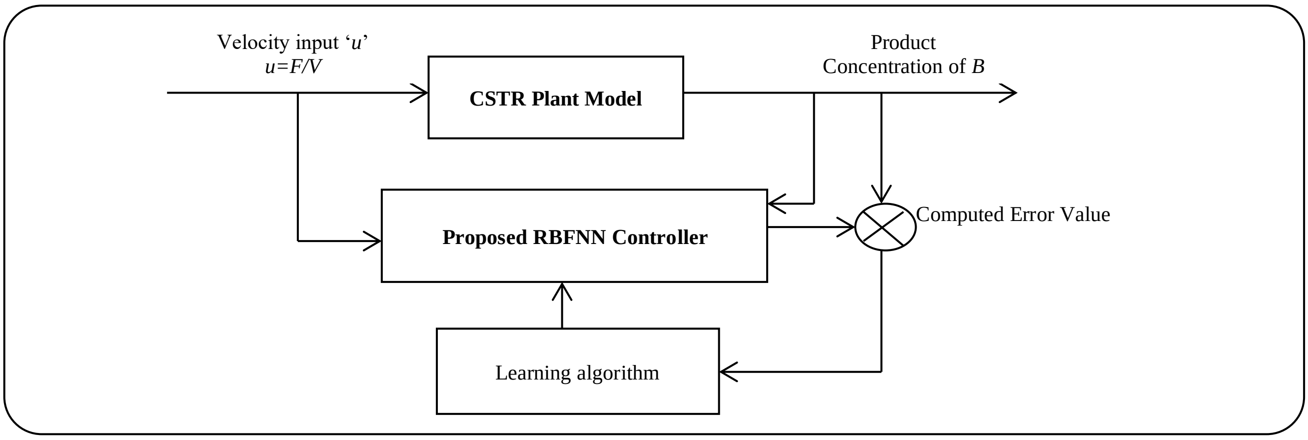

Fig. 5: The CSTR model's system identification block diagram.

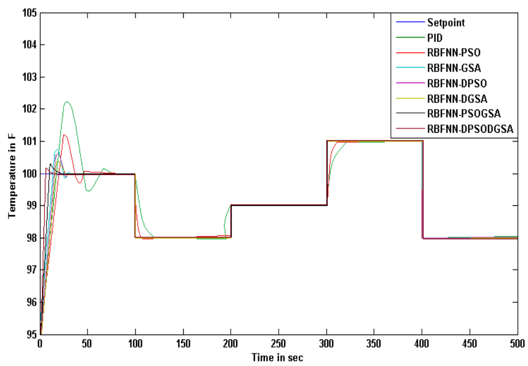

Fig. 6: Set point tracking of temperature.

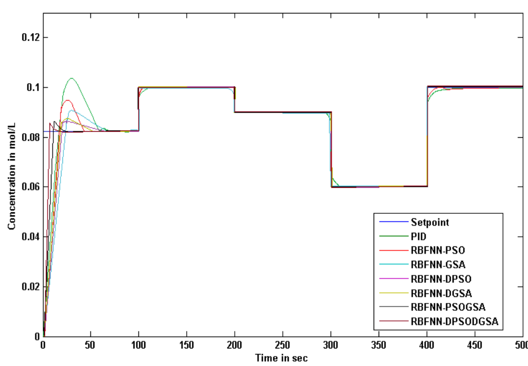

Fig. 7: Set point tracking of concentration.

of integral square error for a fixed number of iterations. Based on the trial and error method, the number of hidden neurons is fixed, and the corresponding training and testing efficiency noted are mentioned in Table 2. From the table, it is observed that increasing the number of hidden neurons improves training efficiency but overfitting occurs, so testing efficiency is greatly reduced, and decreasing the number of hidden neurons affects training efficiency greatly. Table 3 shows the obtained ISE values for the proposed controller models, the closed loop response of the proposed controller is presented in Fig. 8, the corresponding concentration response is presented in Fig. 9. The convergence plot is depicted in Fig. 10, that shows the speed of convergence of proposed models and clearly demonstrates the effectiveness of the proposed hybrid DPSO-DGSA optimization algorithm.

Table 3: Comparison study of the proposed controller.

Devised Controllers	Integral Square Error	Settling time in sec
Classic PID	0.3361	0.7922
Zhou et al. [16]	6.43 e-02	0.8927
Li et al. [17]	7.32 e-03	0.992
Zhou et al. [18]	9.82 e-03	0.7853
Proposed PSO-RBFNN	7.45 e-03	0.6832
Proposed GSA-RBFNN	6.32 e-03	0.7792
Proposed DPSO-RBFNN	5.20 e-03	0.6538
Proposed DGSA-RBFNN	4.14 e-03	0.5268
Proposed PSOGSA-RBFNN	7.87 e-04	0.4892
Proposed DPSODGSA-RBFNN	1.05 e-04	0.1317

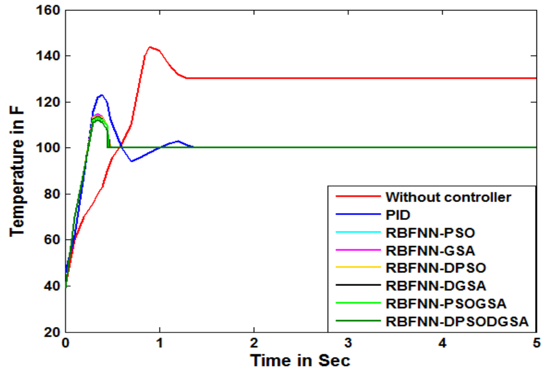

Fig. 8: Temperature response of CSTR model.

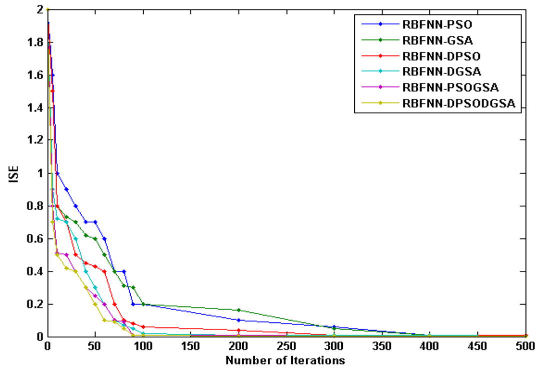

Fig. 10: Convergence Graph.

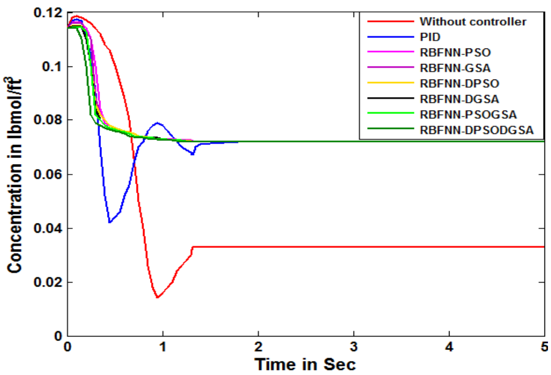

Fig. 9: Close loop response of concentration.

Simulated responses indicate that a hybrid DPSO-DGSA RBFNN controller with superior performance and lower integral square error and faster convergence is more likely due to the algorithm's more deterministic and differential training properties.

According to Table 3, the proposed hybrid DPSO-DGSA based radial basis neural network controller model outperforms the other models considered for validation in this paper and existing model proposed in the literature for the considered CSTR system, and the ISE value is 1.05 e-04, in compared to the other models examined for validation, this is minor. Based on the above analysis, the proposed hybrid DPSO-DGSA algorithm optimised RBFNN controller has shown better training efficiency with improved convergence.

CONCLUSIONS

For CSTR concentration control, a radial basis function neural network controller is discussed in this paper. Introducing evolutionary based algorithms such as PSO, GSA, DPSO, DGSA, and effective hybrid PSO – GSA, hybrid DPSO – DGSA for the training of the radial basis neural network improves the performance of the controller. The proposed effective controller is designed in stages. First, training algorithms are proposed to tune the proposed controller parameters, and then an RBFNN neural network is framed based on predicting the system's future output based on past performance history. Based on a comparison of the proposed controller algorithms, such as PSO – RBFNN, GSA – RBFNN, DPSO – RBFNN, DGSA-RBFNN, hybrid PSO-GSA based RBFNN, hybrid DPSO – DGSA based RBFNN model, and the existing models in the literature in recent times, the hybrid DPSO – DGSA based RBFNN controller achieves satisfactory response for control of CSTR plant.

Received : Jun. 22, 2022 ; Accepted : Oct. 31, 2022

REFERENCES

[1] Abdullah N., Yee T.C., Mohamed A., Mustafa M.M., Osman M.H., Mohamad A.B., Control of Continuous Stirred Tank Reactor using Neural Networks, *Indian Journal of Science and Technology*, **9(21)** (2016).

[2] Alexandridis A., Sarimveis H., "Control of Processes with Multiple Steady States Using MPC and RBF Neural Networks", *In European Symposium on Computer-Aided Process Engineering (ESCAPE-21)*. Thessaloniki, Greece (2011).

[3] Alexandridis A., Stogiannos M., Loukidis A., Ninos K., Zervas E., Sarimveis H., "Direct Versus Indirect Neural Control Based on Radial Basis Function Networks", *In Computer Science and Electronic Engineering Conference (CEEC)*: 91-96 (2014).

[4] Alshammari O., Mahyuddin M.N., Jerbi H., A Neural Network-Based Adaptive Backstepping Control Law with Covariance Resetting for Asymptotic Output Tracking of a CSTR Plant, *IEEE Access*, **8**: 29755-29766 (2020).

[5] Apale T.D., Patil A.B., Optimization Study of Fuzzy Parametric Uncertain System, *International Journal of Artificial Intelligence*, **8(1)**: 14-25 (2019).

[6] Bagheri M., Mirbagheri S.A., Ehteshami M., Bagheri Z., Modeling of a Sequencing Batch Reactor Treating Municipal Wastewater Using Multi-Layer Perceptron and Radial Basis Function Artificial Neural Networks, *Process Safety and Environmental Protection*, **93**:111-123 (2015).

[7] Broomhead D.S., Lowe D., Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks, *Royal Signals and Radar Establishment Malvern*, United Kingdom (1988).

[8] Deepa S.N, Rizwana J., Minimization of Losses and FACTS Installation Cost Using Proposed Differential Gravitational Search Algorithm Optimization Technique, *Journal of Vibration and Control*, **23(2)**: 235-251 (2017).

[9] Elhosseini M.A., El Sehiemy R.A., Rashwan Y.I., Gao X.Z., On the Performance Improvement of Elephant Herding Optimization Algorithm, *Knowledge-Based Systems*, **166**: 58-70 (2019).

[10] Goud H., Swarnkar P., Investigations on Metaheuristic Algorithm for Designing Adaptive PID Controller for Continuous Stirred Tank Reactor, *MAPAN*, **34(1)**: 113-119 (2019a).

[11] Goud H., Swarnkar P., Analysis and Simulation of the Continuous Stirred Tank Reactor System Using Genetic Algorithm, *InHarmony Search and Nature Inspired Optimization Algorithms*, Singapore: Springer (2019b).

[12] Jaiswal S., Kumar C.S., Seepana M.M., Babu G.U.B., Forthcoming. Design of Fractional Order PID Controller Using Genetic Algorithm Optimization Technique for Nonlinear System, *Chemical Product and Process Modeling* (2020).

[13] Jin'no K., A Novel Deterministic Particle Swarm Optimization System, *Journal of Signal Processing* **13(6)**: 507-513 (2009).

[14] Kantha A.S., Utkarsh A., Jatoth R.K., Hybrid genetic Algorithm-Swarm Intelligence-Based Tuning of Temperature Controller for Continuously Stirred Tank Reactor, *International Journal of Modelling, Identification and Control*, **25(3)**: 239-248 (2016).

[15] Li D., Adaptive neural Network Control for a Class of Continuous Stirred Tank Reactor Systems, *Science China Information Sciences*, **57(10)**: 1-8 (2014).

[16] Li D.J., Li D.P., Adaptive Controller Design-Based Neural Networks for Output Constraint Continuous Stirred Tank Reactor, *Neurocomputing*, **153**: 159-163 (2015).

[17] Li S., Jiang P., Han K., "RBF Neural Network based Model Predictive Control Algorithm and its Application to a CSTR Process", *2019 Chinese Control Conference (CCC), IEEE* : 2948-2952 (2019).

[18] Liao Y., Fang S.C., Nuttle H.L., Relaxed Conditions for Radial-Basis Function Networks to be Universal Approximators, *Neural Networks*, **16(7)**:1019-1028 (2003).

[19] Clerc M., Kennedy J., "The Particle Swarm – Explosion, Stability, and Convergence in a Multidimensional Complex Space", *IEEE Transactions on Evolutionary Computing*, **6(1)**: 58-73 (2002).

[20] Ma T., Filtering Adaptive Tracking Controller for Multivariable Nonlinear Systems Subject to Constraints Using Online Optimization Method, *Automatica*, **113**:108689 (2020).

[21] Moody J., Darken C.J., Fast Learning in Networks of Locally-Tuned Processing Units, *Neural computation* **1(2)**: 281-294 (1989).

[22] Park J., Sandberg I.W., Universal Approximation Using Radial-Basis-Function Networks, *Neural Computation*, **3(2)**: 246-257 (1991).

[23] Rashedi E., Nezamabadi-Pour H., Saryazdi S., GSA: A Gravitational Search Algorithm, *Information Sciences*, **179(13)**: 2232-2248 (2009).

[24] Salahshour E., Malekzadeh M., Gordillo F., Ghasemi J., Quantum Neural Network-Based Intelligent Controller Design for CSTR Using Modified Particle Swarm Optimization Algorithm, *Transactions of the Institute of Measurement and Control*, **41(2)**: 392-404 (2019).

[25] Shirude S., Padhiyar N., Optimal Grade Transition of a Non-Isothermal Continuous Reactor with Multi-Objective Dynamic Optimization Approach, *Chemical Engineering Research and Design*, **147**: 63-72 (2019).

[26] Shoga T., Thelkar A.R., Bharatiraja C., Mitiku S., Adedayo Y., Self-Tuning Regulator Based Cascade Control for Temperature of Exothermic Stirred Tank Reactor, *FME Transactions*, **47(1)**: 202-211 (2019).

[27] Simorgh A., Razminia A., Shiryaev V.I., System Identification and Control Design of a Nonlinear Continuously Stirred Tank Reactor, *Mathematics and Computers in Simulation*, **173**: 16-31 (2020).

[28] Srivastava V., Srivastava S., Control of Continuous Stirred Tank Reactor (CSTR) Using Nature Inspired Algorithms, *Journal of Information and Optimization Sciences*, **40(2)**: 329-338 (2019).

[29] Strumiłło P., Kamiński W., "Radial Basis Function Neural Networks: Theory and Applications", In *Neural Networks and Soft Computing*:107-119 (2003).

[30] Zhou F., Peng H., Zhang G., Zeng X., A Robust Controller Design Method Based on Parameter Variation Rate of RBF-ARX Model, *IEEE Access* 7: 160284-94 (2019).

[31] Zhou F., Peng H., Zeng X., Tian X., RBF-ARX Model-Based Two-Stage Scheduling RPC for Dynamic Systems with Bounded Disturbance, *Neural Computing and Applications*, **31(8)**: 4185-200 (2019).

[32] Nekoui M.A., Khameneh M.A., Kazemi M.H., September. "Optimal Design of PID Controller for a CSTR System Using Particle Swarm Optimization", In *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010* (pp. T7-63). IEEE (2010).