

Optimization algorithm-based Elman neural network controller for continuous stirred tank reactor process model

I. Baranilingesan

Department of Electrical and Electronics Engineering, KPR Institute of Engineering and Technology, Coimbatore 641 407, India

A continuous stirred tank reactor (CSTR) is a stand-out nonlinear system among the most essential units of chemical industries. In this article, an Elman neural network is designed to analyse the characteristics of nonlinear behaviour of the CSTR system. The data generated employing the state-space model of CSTR are used to train the designed Elman neural network controller and the controller parameters are optimally tuned by the proposed hybrid swarm intelligence-based optimization algorithm. Two different hybridizations have been developed, including DPSO, DGSA and hybrid DPSO–DGSA and successfully employed in controller tuning. The significance of the proposed controller is validated by a comparative analysis made with conventional methods and the performance is experimentally demonstrated using MATLAB software.

Keywords: Continuous stirred tank reactor, neural network controller, nonlinear behaviour, optimization algorithm, process control.

A continuous stirred tank reactor (CSTR) is one of the important branches of studies in chemical processes industries. Consequently, there are several ongoing studies in the area of chemical and control engineering to improve the performance of the CSTR system. Even though various techniques have been proposed to control the parameters of the CSTR, due to its high nonlinearity the conventional methodologies do not produce better controller performance. The CSTR has noteworthy heat effects; so it is essential to add and remove heat from the reactor based on the temperature difference existing between the jacketed fluid and reactor fluid. In addition to its nonlinearity, the effect of use, affect the performance of the reactor. Research is being carried out to design an effective controller in order to enhance the performance of the CSTR system. In this article, we present the design of an intelligent neural network controller.

Baruah and Dewan¹ proposed a model using the proportional integral derivative (PID) controller and an evolutionary algorithm for controlling the temperature of the

CSTR model. Employing optimization techniques such as particle swarm optimization (PSO) and genetic algorithm (GA) are employed to find the optimal values of the controller parameters. Babushanmugham *et al.*² developed a controller for the CSTR process model to maintain and control the temperature in the reactor using a GA-based PI controller. The proposed approaches based on performance indices of the controller are much better than the internal model controller (IMC) and Ziegler–Nichols (ZN) methods. Salahshour *et al.*³ proposed a multi-layer quantum neural network (QNN) combined with the PSO algorithm for controlling the CSTR system. An adaptive structure was designed with the help of multilayered QNN for the PID controller design. To improve the accuracy and convergence speed of the training process, few alterations have been made in the movement of each particle. On comparing with the performance of the conventional PID and perceptron neural network, the optimized controller design shows improved performance. Saini *et al.*⁴ designed a controller for nonlinear dynamical behaviour of the CSTR. A comparative study between the PID controllers and fuzzy logic controllers has been made to control the temperature of the CSTR. The activity of the CSTR has been tested for various operating points to control the temperature. So and Jin⁵ developed a new methodology for nonlinear variable gain PID controller using a set of fuzzy rules. To minimize the integral absolute error (IAE) and the weighted control input, the parameters were tuned with the help of a GA. Suvire *et al.*⁶ presented a linear algebraic based methodology for the CSTR model in order to track the output variables. The Monte Carlo randomized method was employed to tune the controller parameter and test the system characteristics. This method has proven to track the desired performance with reduced error than that of other conventional controllers. Bingi *et al.*⁷ proposed the fuzzy gain scheduling (FGS) mechanism for the CSTR process to tune the set-point weighted PID (SWPID) controller. The simulation results show that FGS–SWPID method achieves better set-point tracking and disturbance rejection compared to the FGS–PID method.

Mungale *et al.*⁸ have highlighted the importance of self-tuning regulator (STR) and its application for the chemical mixing process. This method is used to maintain

e-mail: barani.ling@gmail.com

constant temperature in the CSTR process by updating the process parameters and the controlled parameters. Feng *et al.*⁹ have derived a method for accurate modelling and prevention of outermost disturbances in the CSTR model. This method helped achieve the best solution for CSTR plant with the help of model-free adaptive sliding mode control methodology. Based on the various controller techniques presented in the study made by Tripathi *et al.*¹⁰, the combined GA-PID controller outperformed the conventional PID and Fuzzy logic controllers. To model the multi input multi output (MIMO) system a linearization strategy is employed¹¹. The evolutionary algorithms help tune the model parameters to optimal value, so that the system tracks the desired set point effectively. Luning *et al.*¹² designed a disturbance observer-based sliding mode control for the CSTR. For the corresponding mathematical modelling process, stability is analysed using the Lyapunov method. Taieb *et al.*¹³ introduced a methodology for the MIMO fuzzy optimal model predictive control (FOMPC) using the adaptive particle swarm optimization (APSO) algorithm. The FOMPC-APSO provides a better solution by obtaining the optimal weights of the objective function. Kantha *et al.*¹⁴ presented the hybrid form of swarm intelligence technique for tuning the PID controller. The PSO and GA are used to tune the parameters of the PID controller. The results depict that the performance of the PSO based-PID controller performance is better than that of GA. Analysis of the control scheme for nonlinear control CSTR system by Darius and Sivagamasundari¹⁵ was based on accurate and reliable controlling for the process variable. The model predictive controller can perform better set point tracking than that of PI controller. Supanna *et al.*¹⁶ designed a robust PID controller using the Kharitonov theorem for control of chemical CSTR. The stability boundary was plotted for the four Kharitonov vortex polynomials in the K_p and K_i parameters. An intersecting point was chosen identified in the plane from which the stabilizing

parameters of the PID controller were identified. Saini *et al.*¹⁷ presented various controllers like PID, NARMA-L2 and LQG for controlling the temperature of the reactor. The step and Gaussian disturbance have been applied to CSTR to check the execution of various controllers. The temperature control of CSTR is best accomplished by linear quadratic Gaussian (LQG). Sharma *et al.*¹⁸ studied the process faults occurring suddenly in the CSTR method. The fault occurrences in CSTR system is effectively diagnosed by employing neural network models, the change in flow rate and the activist speed due the fault condition is modelled by outrageous learning machine-based Hammerstein-Wiener model, Han *et al.*¹⁹. A neural network model with gain scheduling strategy is implemented to estimate the non-linear behaviour of SISO (single input single output) and MIMO CSTR system^{20,21}. The recurrent neural network (RNN) is implemented with MPC (model predictive controller) to optimize the pharmaceutical assembling. Sughan and Poongodi²² compared the performance of the PID, NARMA and Fuzzy PID for temperature control of CSTR plant, the NARMA strategy outperformed the conventional PID controller strategy. Sana and Rafik²³ employed state feedback based parallel distributed compensation (PDC) strategy for CSTR system to attain better stability.

Considering all these reviews on controller design for CSTR and their limitations with regard to time-delay compensation, increased steady-state error and settling time, this article focuses on developing a variant of a classic neural model for the control of the CSTR system by minimizing integral square error.

CSTR system modelling

Figure 1 shows the regular CSTR model. It is well known that the system undergoes an exothermic reaction, wherein fluid flows continuously and is fed into the reactor. Fundamentally it is represented by $A \rightarrow B$. Considering the exothermic reaction, it produces heat that reduces the reaction rate. By initiating a coolant flow rate, temperature of the reactor can be altered to produce controlled concentration of the product. Several assumptions are made in the process, which include perfect mixing in the reactor and jacket, maintaining constant volume in the reactor and jacket and finally the constant parameter values. This system can be illustrated by nonlinear concurrent equations which efficiently combine the laws of chemical reaction and thermodynamics.

Basically, the law of kinetics is given by

$$-r_A = k(T)c_A = k_0 e^{-(E/RT)} c_A, \quad (1)$$

where r_A is the reaction rate, K the rate constant, T the absolute temperature, c_A the feed concentration, E the

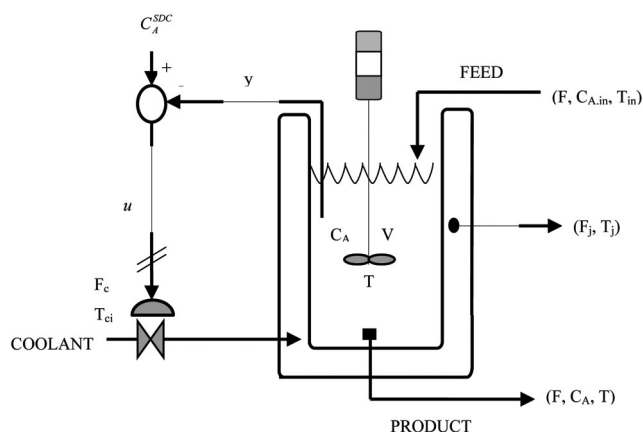


Figure 1. Continuous stirred tank reactor (CSTR) process with cooling jacket.

activation energy, R the Gas constant and k_0 is the Frequency factor.

The model is framed based on assumptions such as constant volume, constant heat capacity of reacting mixture and perfect mixing within the reactor. The mass balance equation is written as

$$\frac{dc_A}{dt} = \frac{F}{V}(c_{A,in} - c_A) - k(T)c_A, \quad (2)$$

$$\frac{dT}{dt} = \frac{F}{V}(T_{in} - T) - \frac{h_r}{\rho c_p} k(T)c_A - \frac{UA_r}{V\rho c_p}(T - T_j). \quad (3)$$

According to the assumptions of uniform temperature of the jacket fluid within the circulation tubes and constant water heat capacity, the energy balance equation for the jacket is given by

$$\frac{dT_j}{dt} = F_{cw} \frac{\rho_w}{m_o}(T_{cw} - T_j) + \frac{P}{c_w m_o} + \frac{UA_r}{c_w m_o}(T - T_j). \quad (4)$$

For the equations from (1) to (4), T is temperatures, t the time, c the concentration, T_j the jacket temperature, F the volumetric flow rate, c_p denotes the specific heat capacity, V the reactor volume, m_o the overall effective mass of cooling and heating system, ρ specifies density, c_w the water heat capacity, A_r the heat exchange surface, U the heat transfer coefficient, P the power input to the heater and T_{cw} is the cooling water temperature. The steady-state operating data of the CSTR model are used to derive the transfer function which is utilized in the simulation studies. Table 1 presents the steady-state operating data of CSTR.

On performing linearization with respect to concentration at the time of reaction and temperature, the transfer function model of the CSTR system is obtained as

$$G(s) = -\frac{1.3083}{(13.5102s+1)(6.2417s+1)} e^{-4-8961s}. \quad (5)$$

The state space model derived for the CSTR system is as given by

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -0.1239 & -0.00017 \\ 7.4454 & -0.05894 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.003151 \\ -0.81985 \end{bmatrix} u, \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \end{aligned} \quad (6)$$

where x_1 and x_2 are the state variables represented by the concentration (C_A) and temperature (T) of the reactor.

Need for an optimization-based neural network for the derived CSTR model

Irrespective of numerous techniques available to address the considered problem statement, the need of optimal design of CSTR system is always an open field of research because of its high non-linearity. So, in this paper hybrid optimization based neural network model is developed and its performance is evaluated.

Overview of the Elman neural network

The Elman neural network was introduced in 1990. Lin and Hong²⁴ presented the architecture of recurrent link incorporated into a hidden layer of RNN. The RNNs are special types of neural networks because of their internal memory storage ability which makes them precise for future output prediction. The only difference between feed-forward and recurrent neural networks is the input data. In the former the input data are in one direction, whereas in the latter the data make a loop and the RNN is based on utilizing the previous history of data to produce the control signal. It consists of four layers, namely input layer, hidden, recurrent link and output layer. In this network, the output of the hidden layer feedback is with one step delay and it forms the recurrent layer which stores complete information of the hidden layer as a memory element. In this network, a set of context units are introduced to perform recurrent operation, whose activation values are feedback from the hidden units (set of neurons in hidden layer).

Proposed Elman neural network controller

The neural network configuration assumes a major role in performing control action and for parameter estimation of control systems under consideration. Before considering the inputs to be provided for the network model, it is

Table 1. Continuous stirred tank reactor (CSTR) steady-state operating data

CSTR process variables	Parametric values under normal operating conditions
Concentration (c)	0.08235 mol/l
Reactor temperature (T)	441.81 K
Volumetric flow rate (F)	100 l/min
Reactor volume (V)	100 l
Cooling water temperature (T_{cw})	350 K
Heat transfer coefficient (U)	7e5 cal/(min K)
Specific density (ρ)	1000 g/l
Specific heat capacity (c_p)	1 cal/(gK)
Heat exchange	2e5 cal/mol
Reaction rate constant	7.2e10 min ⁻¹
Activation energy term	9980 K

necessary to perform scaling. The datasets are scaled in the range [0, 1] after which the initial parameters for training the network are assigned and the learning rule operation is carried out. Figure 2 shows the proposed Elman architecture.

The data presented for computation pass through each layer independently and finally, the output arrives at the output layer (Figure 2). The input is multiplied with a weight value and fed to the hidden layer, where it is processed through a hyperbolic sigmoid activation function. The network is learned based on the current activation function on the computed values and with previous history of control output. The value of $X(K)$ obtained from the hidden layer is passed to the output layer where it is processed using the purelin activation function. During the training process of the neural network, the previous iteration value is utilized in Elman model.

$X_1, X_2, X_3, \dots, X_n$ are data inputs, $Y_1, Y_2, Y_3 \dots Y_n$ are control output, $W = (W_{11}, W_{12}, W_{13} \dots W_{1n}, W_{21}, W_{22}, W_{23} \dots W_{2n}, W_{31}, W_{32}, W_{33} \dots W_{3n})$ are weighted interconnection between input and the hidden layers, $V = (V_{11}, V_{12}, V_{13}, \dots, V_{1n}, V_{21}, V_{22}, V_{23}, \dots, V_{2n}, \dots, V_{n1}, V_{n2}, V_{n3} \dots V_{nn})$ is the weighted interconnection between the hidden and the output layer (Figure 2).

$W_c = (W_{c11}, W_{c12}, \dots, W_{c1n}, W_{c21}, W_{c22} \dots W_{c2n}, W_{c31}, W_{c32}, \dots, W_{c3n})$ is the weighted interconnection of recurrent link layer and input vector and $X_c(K) = X(K - 1)$ is the input entering recurrent link layer.

Proposed algorithm for the design of the controller

The control action employing Elman neural network model was carried out as follows. The iterative intensification of the network continued till the consi-

dered stopping criterion with respect to minimal mean square error was attained.

Step 1: Initialization

In this step the network parameters like learning rate, initial weights and activation functions of the Elman model were initialized.

Step 2: Data scaling

The input was scaled by applying the min-max technique.

$$\phi'_i = \left(\frac{\phi_i - \phi_{\min}}{\phi_{\max} - \phi_{\min}} \right) (\phi'_{\max} - \phi'_{\min}) + \phi'_{\min}, \tag{7}$$

where ϕ_i, ϕ_{\min} and ϕ_{\max} are the actual data, minimum and maximum input data and ϕ'_{\min} and ϕ'_{\max} are the minimum and maximum desired target value respectively.

Step 3: Designing the controller

The Elman neural network has been framed with desired number of input, output and hidden layers, here the activation function employed is hyperbolic sigmoid function.

Step 4: Training and performance evaluation of the network

The model is trained with set of input data, the network gets trained with the past iteration experience and the iterations are stopped on attaining minimal mean square error, which is the set stopping criterion.

Proposed population-based hybrid optimization algorithm for the CSTR model

To overcome the issues of basic optimization techniques like local optimal point trapping, and trade-off between exploration and exploitation capabilities, a hybrid version of variant algorithms is presented to enhance the performance of the controller. From recent studies, it is evident that hybrid optimization techniques provide more efficient behaviour with higher flexibility. Generally, PSO is a population-based algorithm that explores the search space to find better solutions improving the exploration and exploitation rate. On combining these two algorithms, the limitations of each algorithms on utilizing separately such as local stagnation issue is addressed. The characteristics of DPSO and DGSA is combined as hybrid algorithm to perform efficient tuning operation of the proposed Elman model.

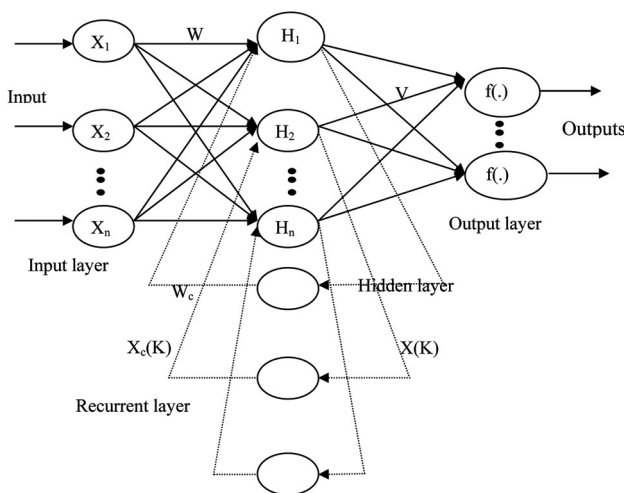


Figure 2. Elman neural network controller.

Deterministic particle swarm optimization

The conventional particle swarm optimization algorithm has random parameters, which have a direct impact on the performance of the algorithm. However, these values need not be optimal as they are usually considered by heuristic assumptions. These stochastic factors introduce dynamics into the system. A modified form of particle swarm optimization known as deterministic particle swarm optimization (DPSO) algorithm is employed here. In DPSO stochastic factors are eliminated.

The canonical form of DPSO is

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

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

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

where z_i^k is the desired fixed point, $\delta \in \mathfrak{R}$ and $\omega \in \mathfrak{R}$.

If the velocity component becomes less than 1.00×10^{-8} , then it is multiplied by 5.12×10^{-8} , the proposed deterministic PSO is termed as re-acceleration PSO.

Pseudo-code of the DPSO algorithm

```

Start
Initialize the population Do{
for i=1 to population size
{
calculate fitness value
If current_pBest>pBest
then pBset=current_pBest
else pBest = pBest}
gBest=particle with best pBest among population
for i=1 to population size
{
update velocity and position of the particles
if particle_velocity < 1.00 × 10-8
then particle_velocity= particle_velocity*5.12 × 10-8
}
}repeat until stopping criteria is achieved.
Stop
    
```

Differential gravitational search algorithm

Considering the gravitational search algorithm (GSA), the major issue is its low exploitation capability with enhanced exploration ability as the final stages of iterations are reached. This results in local optimal point trapping. So for good performance of the proposed algorithm,

it is necessary to have a better trade-off between the exploration and exploitation abilities. For this, a modified form of GSA has been employed, viz. differential gravitational search algorithm (DGSA), which utilizes both local and global variants^{25,26}.

In the DGSA, the balance between the local and global search directions are influenced by differential parameter. The velocity update for global variant of p_i be $g\phi_i^{k+1}$ and for local variant the velocity update of p_i be $l\phi_i^{k+1}$.

$$g\phi_i^{k+1} = \text{rand}_i \times g\phi_i^k + a_i^k, \tag{11}$$

$$l\phi_i^{k+1} = \text{rand}_i \times l\phi_i^k + a_i^k. \tag{12}$$

The velocity and position update expressions for DGSA are given by

$$\phi_i^{k+1} = \mu g\phi_i^{k+1} + (1-\mu)l\phi_i^{k+1}, \tag{13}$$

$$p_i^{k+1} = p_i^k + \phi_i^{k+1}, \tag{14}$$

The differential parameter $\mu \in [0, 1]$; for $\mu = 1$ refers to a standard global GSA and $\mu = 0$ refers to standard local GSA.

The exploration and exploitation properties of the standard GSA are combined by the differential parameter μ , with linear increment of values between 0 and 1. Thus it can improve the exploitation and exploration capabilities of the algorithm.

The total force acting on the particle is given by

$$F_i^k(t) = \sum_{\substack{j \in pbest \\ j \neq i}} \text{rand}_j \times F_{ij}^k(t). \tag{15}$$

Pseudo-code of the DGSA algorithm

```

Start
Initialize the search agents Do{
for i=1 to population size
{
calculate fitness value
}
Update gravitational constant G(t), mass M(t),
acceleration of mass a(t), the total force F(t) using
traditional GSA.
for i=1 to population size
{
update agents position and velocity using (13),(14)
}
}repeat until stopping criteria is achieved
Stop
    
```

Table 2. Parameters of the developed optimization algorithms

Parameters	Deterministic particle swarm optimization (DPSO)	Parameters	Differential gravitational search algorithm (DGSA)
Population size	50	Number of search agents	50
ω	0.22	Gravitational constant	100
δ	0.80	Differential factor	0.5
Number of iterations	100	Number of iterations	100

Proposed hybrid DPSO–DGSA optimization algorithm

The proposed hybrid DPSO–DGSA algorithm combines the characteristics of DPSO with better achievement in the exploration and exploitation capabilities of DGSA. The easy realization between the eigenvalues and parameters of DPSO makes the system easy to control and the stochastic factors are not present explicitly, which makes the system deterministic. To attain optimal solution the reacceleration factor has been introduced, with the characteristics of DPSO along with better exploration and exploitation capabilities of DSGA, combining the best features and developing an efficient heuristic algorithm for designing a controller (Table 2).

The modified velocity and position expressions for hybrid DPSO–DGSA are given by

$$\varphi_i^{k+1} = \delta\varphi_i^k - \omega p_i^k + a_i^k z_i^k \omega, \quad (16)$$

$$p_i^{k+1} = z_i^{k+1} + \omega\varphi_i^k + \delta(p_i^k - z_i^k). \quad (17)$$

The stopping criteria for the proposed algorithm is attainment of minimum mean square error or attainment of maximum iterations. On attaining the given stopping criterion, the algorithms return the best global solution corresponding to their fitness function evaluation and halt the process.

Pseudo-code of the hybrid DPSO–DGSA technique

```

Start
Initialize the population Do{
for i=1 to population size
{
calculate fitness value
If current_pBest>pBest
then pBest=current_pBest
else pBest = pBest
update G(t),M(t),a(t),F(t) with considered differential
factor
}
gBest=particle with best pBest among population
for i=1 to population size
{
update velocity and position of the particles using
(16)&(17)

```

```

if particle_velocity < 1.00 × 10-8.
then particle_velocity= particle_velocity*5.12 × 10-8.
}
} repeat until stopping criteria is achieved.
Stop

```

Proposed Elman neural network controller model

The Elman NN controller is designed with the proposed optimization model to perform control action for the CSTR system. In this model, the Elman NN performs the controller action and the weights of the Elman network get tuned with the adopted training strategies in order to achieve minimum square error and faster convergence. Generally, the initial weights between all the layers, including the recurrent layer in the Elman neural network model are randomly initialized; but the random initialization results in immature convergence of the neural network model. Consequently, this work focuses on applying the hybrid DPSO–DGSA approach to tune the optimal weights of the Elman neural model in order to achieve faster convergence.

The proposed optimally designed Elman Neural Network Controller is presented as follows:

- Step 1: Start the algorithmic process.
- Step 2: Initialize the necessary parameters like learning rate, initial weights and activation function of the Elman neural network model.
- Step 3: Perform a scaling process for the collected data using the min–max scaling method.
- Step 4: Assign parameters for implementation of the Elman neural controller.
- Step 5: Perform the training process of Elman using gradient descent learning rule.
- Step 6: Invoke the selected training algorithm; the initial parameters for the developed optimization algorithms are the final weights computed in the previous step. The initial parameters for this method of the training process will be the final weights determined while fine-tuning of Elman model.
- Step 7: Employ the DPSO algorithm. Employ the DGSA algorithm. Employ the hybrid DPSO–DGSA algorithm.
- Step 8: Return the tuned weight values for which the best fit is obtained.

Step 9: Utilize these optimal weights into steps 3–7 of this algorithm. With the fine-tuned Elman model go to the next step.

Step 10: On completing the training process, test the model with the testing data.

Step 11: Output the solutions computed.

Step 12: The process is continued until attainment of stopping criteria, here the iterative process is stopped on reaching minimum mean square error or maximum number of iterations.

Methodology for hidden neuron fixation in the controller

Next we present the methodology adopted to fix the optimal number of hidden neurons. The number of neurons of the network greatly influences the error convergence; thereby the stability of the network is improved and ultimately the overfitting and underfitting problem due to inappropriate selection of neurons is avoided. To meet the desired requirements of the network such as error precision and accuracy, an insufficient number of neurons cannot be used, which will result in underfitting problems.

Similarly, overfitting results in overtraining of the network. So it is necessary to fix the number of hidden neurons of the network model. The objective of this study is to present the optimal number of hidden neurons with estimated N_h , such that minimal error criteria are reached. Numerous methods have been proposed to predict the required number of neurons²⁷.

The objective functions set to train the proposed neural network controller are

$$MSE = \sum_{i=1}^N \frac{(Y'_i - Y_i)^2}{N}, \tag{18}$$

where Y_i is the predicted output and Y'_i is the actual output.

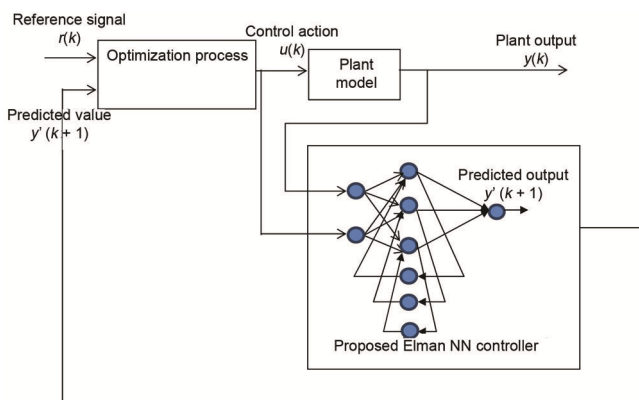


Figure 3. Block diagram of the proposed optimized Elman neural network controller model.

With respect to the concentration control of the CSTR model, the presented inputs to the neural network are temperature (the manipulating variable) and concentration (the predicted output of the plant). The two input neurons are presented with the scaled inputs and the output of the plant is attained. The optimal number of hidden neurons are fixed as 11, the number of input neurons is 2, the output neuron is 1. So, the proposed Elman architecture possess 2-11-1 neurons in input, hidden and output layer respectively.

Applicability of the proposed optimized controller for CSTR plant model

The proposed optimized Elman NN controller of the CSTR system was implemented and simulated in MATLAB R2013a (Version 8.1.0.604) environment. The modelled controller optimizes the system and fine-tunes its performance with better accuracy. Figure 3 shows a typical structure of the CSTR system employing optimization-based Elman. In the figure, k refers to the sample time. It is to be noted that the control action $u(k)$ is to act on the CSTR plant model based on the predicted values and not based on the real output. The proposed model considers the optimal Elman NN model that is trained accurately with minimal mean square error and is satisfactorily ready to represent them.

The modelled controllers tend to minimize the performance index J with respect to the error minimization principle. The index function is given by

$$e(k+1) = r(k+1) - y'(k+1), \tag{19}$$

$$J = \int e(k)^2. \tag{20}$$

The control action $u(k)$ is updated based on the reduction of the gradient values of the performance index

$$u(k+1) = u(k) - \xi \frac{\partial J}{\partial u(k)}, \tag{21}$$

where ξ is a constant between 0 and 1.

Simulation results and comparative analysis of the proposed optimized controller for CSTR model

The designed neural network controller based on the proposed optimization algorithm is employed for optimizing the performance of control activities in the CSTR plant model. The problem statement is to control the concentration of the plant (B) through manipulating the temperature of the jacketed CSTR; the control signal u is

represented by (F/V) . The considered disturbance in this scenario is the change in feed concentration of reactant A .

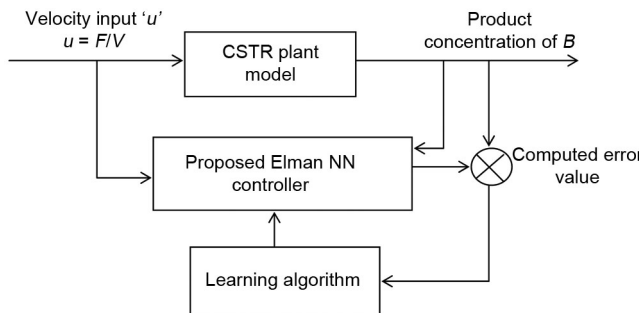


Figure 4. Block diagram of system identification using the controller for the CSTR model.

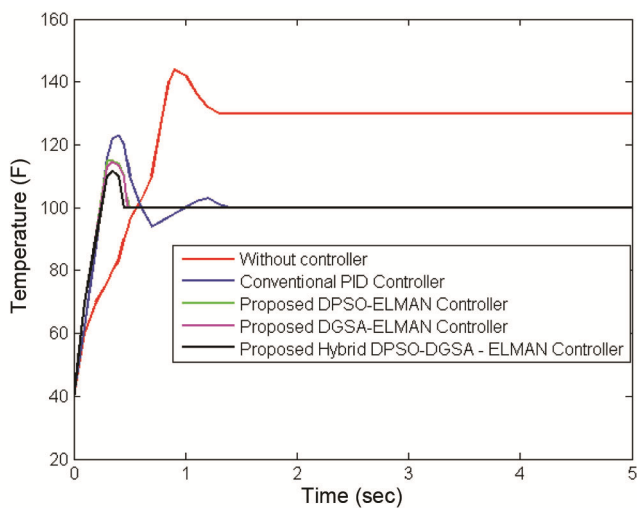


Figure 5. Temperature response of the CSTR model using the proposed controller techniques.

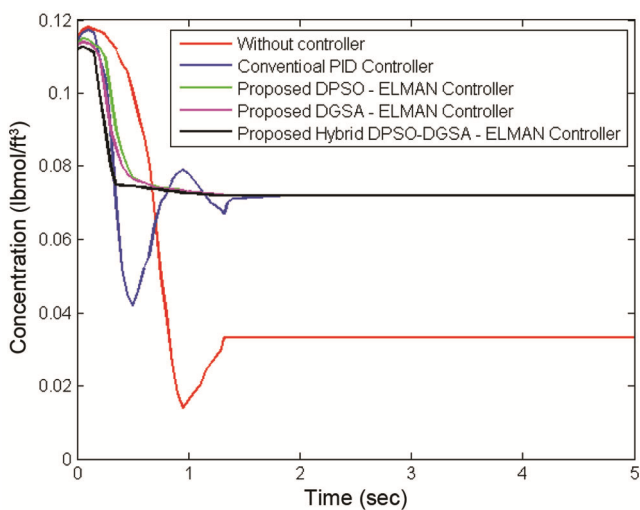


Figure 6. Concentration control of the CSTR system using the proposed controller techniques.

The normal operating parametric values considered include: feed concentration $C_A = 10$ mol/l, $u = F/V = 0.5708$ l/min and concentration of the desired product = 1.117 mol/l. The parameters of the PID controller are: $K_p = 10$, $K_i = 12.04$ and $K_d = 1.76$. The closed-loop performance of the proposed controller is evaluated through time-domain analysis of the designed controller employing the proposed optimized algorithm (Figure 4).

The performance of the model is tested with set point tracking, here the desired concentration level tracked with step change of 1.1169 to 1.2003. The corresponding temperature and concentration plot is presented in Figures 5 and 6. The control strategy of hybrid DPSO–DGSA-based Elman NN controller is found to have improved performance criteria due to better exploration and exploitation mechanisms for searching the optimal weights to design an effective Elman neural controller.

From Figures 5 and 6 it is inferred that the integral square error is greatly reduced for the proposed optimized controllers employed for concentration control of the

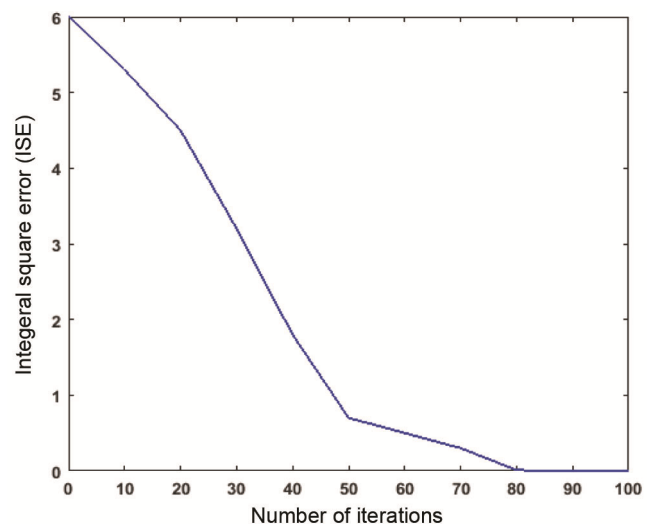


Figure 7. Error convergence graph for the proposed hybrid DPSO–DGSA optimized Elman controller.

Table 3. Error convergence table for the proposed hybrid DPSO–DGSA optimized Elman controller

No of generations	Integral square error
10	5.5081
20	4.6081
30	3.3257
40	1.9274
50	0.8112
60	0.4209
70	0.1932
80	0.0702
90	0.0017
100	0.0022

Table 4. Performance metric of the considered controllers

Devised controllers	Peak overshoot (lbmol/ft ³)	Settling time (sec)	Integral square error
PID controller (Ziegler–Nichols method)	0.1173	1.85	0.9361
LQR-based BFA ²⁸	0.1165	1.70	0.759
Fuzzy-optimized PSO ²⁹	0.1145	1.65	0.01568
Proposed DPSO–Elman model	0.1153	1.31	3.48 e-03
Proposed DGSA–Elman model	0.1140	1.30	3.31 e-03
Proposed hybrid DPSO–DGSA–Elman model	0.1136	1.24	2.20 e-03

CSTR process. When compared with various other conventional optimal procedures employed for tuning the Elman controller, the performance of the proposed hybrid DPSO–DGSA is improved with better metrics.

Table 3 and Figure 7 show the integral square error computed employing the proposed optimized Elman NN controller. The performance of the controller in comparison with that of the developed optimized PID controllers is found to be better in all the metrics considered (Table 4).

Hence, the proposed hybrid DPSO–DGSA-based Elman NN model is found to be a better controller for the CSTR system, according to the controller performances as shown in Figures 5 and 6, i.e. better set-point tracking is achieved with minimal ISE. The estimated performance metric displayed in Table 4 proves that the proposed optimized DPSO–DGSA-based Elman controller model achieves satisfactory characteristics in terms of peak overshoot, settling time and ISE than other methods.

Conclusion

This article provides details on designing the optimized Elman NN controller for concentration control of the CSTR. The ability of the network in retaining the memory of the previous iteration through a recurrent links added an advantage for better convergence and effective tracking of the system output. The study has been carried out in two steps. Initially, Elman NN has been designed and trained with a conventional learning algorithm, with the values obtained in the initial iteration of the learning process as the initial set values for optimal tuning algorithms. The effectiveness of the proposed Elman NN controller is validated through the considered performance metrics. Based on a comparative analysis made with existing controllers, the proposed hybrid DPSO–DGSA-based Elman NN controller model demonstrates better performance in respect of the objective function value of the considered CSTR plant model.

- Baruah, S. and Dewan, L., A comparative study of PID based temperature control of CSTR using genetic algorithm and particle swarm optimization. In International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Dehradun, India, 2017, pp. 1–6; doi:10.1109/ICETCCT.2017.8280312.

- Babushanmugham, S., Srinivasan, D. S. and Giriraj Kumar, S. M., Temperature control of a non-linear process using genetic algorithm. *Int. J. Curr. Eng. Technol.*, 2017, 7(2), 585–589.
- Salahshour, E., Malekzadeh, M., Gordillo, F. and Ghasemi, J., Quantum neural network-based intelligent controller design for CSTR using modified particle swarm optimization algorithm. *Trans. Inst. Meas. Control*, 2019, 41(2), 392–404.
- Saini, P., Kumar, R., Sharma, P. and Rajput, N., Design and comparative analysis of controllers for continuous stirred tank reactor (CSTR). In Proceeding of International Conference on Intelligent Communication, Control and Devices, Springer, Singapore, 2017, pp. 351–365.
- So, G. B. and Jin, G. G., Fuzzy-based nonlinear PID controller and its application to CSTR. *Kor. J. Chem. Eng.*, 2018, 35(4), 819–825.
- Suvire, R. B., Serrano, M. E., Vega, J. R., Ortiz, O. A. and Scaglia, G. J. E., Nonlinear Controller for Trajectory Tracking of a Continuous Stirred Tank Reactor. *CEAI*, 2017, vol. 19, No. 4, pp. 112–121.
- Bingi, K., Ibrahim, R., Karsiti, M. N. and Hassan, S. M., Fuzzy gain scheduled set-point weighted PID controller for unstable CSTR systems. In IEEE International Conference on Signal and Image Processing Applications (ICSIPA), IEEE, 2017, pp. 289–293.
- Mungale, N. R., Upadhyay, A. and Pandian, B. J., Self-tuning regulator for an interacting CSTR process. In IOP Conference Series: Materials Science and Engineering, IOP Publishing, 2017, vol. 263, No. 5, p. 052012.
- Feng, J., Cao, L., Ma, L., Zhao, D. and Spurgeon, S. K., Model-free adaptive sliding mode control for continuous stirred tank reactor. In 37th Chinese Control Conference (CCC), IEEE, 2018, pp. 3035–3040.
- Tripathi, S., Kaur, G. G. and Singla, S. G., *Design of Intelligent Controller for Isothermal Continuous Stirred Tank Reactor System*, 2017; http://tudr.thapar.edu:8080/jspui/bitstream/10266/4555/1/final_report_tripathi_done.pdf
- Kavyarasan, K. and Bagyaveereswaran, V., Tuning of PID controller using optimization techniques for a MIMO process. *IOP Conf. Ser.: Mater. Sci. Eng.*, 2017, 052019.
- Luning, M., Ynag Xiao, Zhao Dong-ya and Spurgeon, S., Disturbance observer based sliding mode control for a continuous stirred tank reactor (CSTR). 36th Chinese Control Conference (CCC), 2017, pp. 3748–3753.
- Taieb, A., Soltani, M. and Chaari, A., Parameter optimization of MIMO fuzzy optimal model predictive control by APSO. *Complexity*, 2017, 11; <https://doi.org/10.1155/2017/5813192>
- Kantha, A. S., Utkarsh, A. and Jatoh, R. K., Hybrid genetic algorithm-swarm intelligence-based tuning of temperature controller for continuously stirred tank reactor. *Int. J. Model., Identif. Control*, 25(3), 239–248.
- Darius, E. M. and Sivagamasundari, D. S., Design and implementation of controllers for a CSTR process. *Int. J. Emerg. Technol. Comput. Sci. Electron.*, 2016, 23(1), 175–183.
- Kumar, S. S. and Shreesha, C., Design of robust PID controller for a CSTR plant with interval parametric uncertainty using Kharitonov

- theorem. In Computation of Power, Energy Information and Communication (ICPEIC), International Conference on IEEE, 2016, pp. 430–433.
17. Saini, P., Gaba, S. K., Rajput, N. and Aggarwal, A., Design and analysis of controllers for continuous stirred tank reactor (CSTR). In 3rd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2016, pp. 2778–2784.
 18. Sharma, R. S., Dewan, L. and Chatterji, S., Neural network-based fault diagnosis in CSTR. *Leonardo Electron. J. Pract. Technol.*, 2018, **32**, 193–214.
 19. Han, Z., Cheng, B., Wang, C. and Yang, W., Identification of CSTR using extreme learning machine based Hammerstein–Wiener model. In Control Science and Systems Engineering (ICCSSE). 3rd IEEE International Conference on IEEE, 2017, pp. 733–736.
 20. Tronci, S. and Baratti, R., A gain-scheduling PI control based on neural networks. *Complexity*, 2017, **8**; <https://doi.org/10.1155/2017/9241254>.
 21. Wong, W. C., Chee, E., Li, J. and Wang, X., Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing. *Mathematics*, 2018, **6**(11), 242.
 22. Sudhanan, R. M. and Poongodi, P., Comparative analysis of various control strategies for a nonlinear CSTR system. *Int. J. Nonlinear Sci. Numer. Simul.*, 2017, **18**(3–4), 269–276.
 23. Bzioui, S. and Channa, R., Observer based tracking control for the isothermal CSTR. In International Conference on Control, Automation and Information Sciences (ICCAIS), IEEE, 2018, pp. 102–106.
 24. Lin, W. M. and Hong, C. M., A new Elman neural network-based control algorithm for adjustable-pitch variable-speed wind-energy conversion systems. *IEEE Trans. Power Electron.*, 2010, **26**(2), 473–481.
 25. Kennedy, J. and Eberhart, R. C., A discrete binary version of the particle swarm algorithm. Systems, Man, Cybernetics, IEEE International Conference on Computational Cybernetics and Simulation, 1997, vol. 5, pp. 4104–4108.
 26. Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S., GSA: a gravitational search algorithm. *Inf. Sci.*, 2009, **179**(13), 2232–2248.
 27. Gnana Sheela, K. and Deepa, S. N., Review on methods to fix number of hidden neurons in neural networks. *Math. Prob. Eng.*, 2013, **11**; <https://doi.org/10.1155/2013/425740>
 28. Sundari, S. and Nachiappan, A., Design of optimal linear quadratic regulator for the stabilization of continuous stirred tank reactor (CSTR) process. *Int. J. Pure Appl. Math.*, 2018, **118**(24), 1–19.
 29. Campos, J., Jaramillo, S., Morales, L., Camacho, O. and Chavez, D., PD+ I fuzzy controller optimized by PSO applied to a variable dead time process. In IEEE Third Ecuador Technical Chapters Meeting (ETCM), IEEE, 2018, pp. 1–6.

Received 21 May 2020; accepted 22 December 2020

doi: 10.18520/cs/v120/i8/1324-1333