Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Optimized deep learning neural network predictive controller for continuous stirred tank reactor[☆]

S.N. Deepa[*], I. Baranilingesan

Department of Electrical and Electronics Engineering, Anna University, Regional Campus: Coimbatore, Coimbatore, 641 046

ARTICLE INFO

Article history:
Received 8 March 2017
Revised 3 July 2017
Accepted 4 July 2017
Available online xxx

Keywords:
Continuous stirred tank reactor
Deep learning neural network controller
Model predictive control
Particle swarm optimization
Gravitational search algorithm
Controller model

ABSTRACT

In this paper, a deep learning neural network model predictive controller (DLNNMPC) is designed to analyse the performance of a non-linear continuous stirred tank reactor (CSTR) that performs parallel and series reactions. The data generated employing the state space model of CSTR is used to train the designed deep learning neural network controller. Deep Learning Neural Network (DLNN) progresses the training with its weights tuned by the proposed hybrid version of evolutionary algorithms – Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA). The developed hybrid PSO – GSA based DLNN model of continuous stirred tank reactor is employed in this paper for model predictive controller design. The effectiveness of the proposed DLNNMPC tuned by hybrid PSO – GSA for CSTR is validated for its performance on comparison with that of other designed Proportional – Integral (PI) and Proportional – Integral – Derivative (PID) controllers as available in early literatures for the same problem under consideration.

1. Introduction

In general, the existing real time processes do not go well with the developed precise mathematical models and hence handling these types of processes are very difficult. Considering the recent growth in science and technology, the complete knowledge of the physical processes are vague and are uncertain due to their operational complexity. Emerging technology leads to the computation of exact mathematical models of the system under consideration and then the developed models are employed to carry out the required predictions. Employing conventional controllers for the real time processes, they were noted to come across certain inexactness and as well there exists no significant observation in comparison with that of the original system models. For the past two decades, certain novel methods of control technique have been developed and these controllers were noted to gain attention from the control designers. These novel methods of control approaches fall under the category of intelligent controllers. Generally, conventional controllers or traditional controllers are employed to control the dynamic behaviour of the systems represented by differential or difference equations and the growing intelligent controllers are employed for systems which do not possess adequate differential or difference equations for their representation. There exist various intelligent control techniques – fuzzy models, neural models and knowledge expert system models in the growing scenario. This paper focuses on developing intelligent neural control models for analysing the performance of the considered continuous stirred tank reactor.

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. V. Suma.

[*] Corresponding author.
E-mail address: deepapsg@gmail.com (S.N. Deepa).

http://dx.doi.org/10.1016/j.compeleceng.2017.07.004
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

For the past few years, different neural controller variants have been employed for control aspect of CSTR process and are as given in this section.

Li and Shi [1] developed an adaptive neural control method using the extreme learning machine (ELM) and are on the basis of mean value theorem and Back-stepping control. The employed control method is then applied to the instance of continuous stirred tank reactor (CSTR) system in the chemical process and the simulation results are presented to verify the effectiveness of the method. Abdullah et al. [2] presented the design of neuro controller NARMA-L2 for composition control in an isothermal Continuous Stirred Tank Reactor (CSTR) by manipulating the input feed composition. The NARMA-L2 controller design is implemented in two stages in which the first stage is system identification to model the process and the second stage is designing the process controller. Li and Li [3] modelled an adaptive control approach based on the approximation property of the neural networks for a class of continuous stirred tank reactor with maintained output constraints and uncertainties. In order to control this class of systems, the systems are decomposed by using the mean value theory and the unknown functions are approximated by using the neural networks, and Barrier Lyapunov function has also been introduced.

Li [4] developed an adaptive control scheme and studied it for a class of continuous stirred tank reactors (CSTR) with unknown functions. Because the nonlinear property and the unknown functions are included in the considered reactor, it leads to a completed task for designing the controller. Shyamalagowri and Rajeswari [5] presented a neural network predictive controller to identify complex nonlinear systems with no complete model information. Closed loop method is preferred because it is sensitive to disturbances and also it is not necessary to identify the transfer function model of an unstable system. The nonlinear process considered here also is a process reactor CSTR.

Li [6] studied the control problem of continuous stirred tank reactors (CSTR). An adaptive controller that uses the neural networks (NNs) is provided to solve the unknown terms. The proposed approach overcomes the effect of the dead zone input. The dead zone input in the systems is compensated by introducing a new Lyapunov form and Young's inequality. Alexandridis et al. [7] presented a comparison between direct and indirect neural control methods based on the radial basis function (RBF) architecture. The performances of the two control schemes are evaluated and compared on a highly nonlinear control problem, namely control of a continuous stirred tank reactor (CSTR) with multiple stable and unstable steady states.

Todorov et al. [8] presents a Takagi-Sugeno type recurrent fuzzy-neural network with a global feedback. To improve the predictions and to minimize the possible model oscillations, a hybrid learning procedure based on Gradient descent and the fast converging Gauss-Newton algorithms, has been designed. The potentials of the obtained predictive controllers are demonstrated by simulation experiments to control a nonlinear Continuous Stirred Tank Reactor. Alexandridis et al. [9] presents a novel control scheme based on approximating the inverse process dynamics with a radial basis function (RBF) neural network model, trained with the fuzzy means algorithm. The proposed approach is applied to the control of a nonlinear Continuous Stirred Tank Reactor (CSTR) exhibiting multiple equilibrium points, including an unstable one. Reddy et al. [10] analysed a neural network based predictive controller to a non-linear continuous stirred tank reactor (CSTR). The performance of present neural network based model predictive controller (NNMPC) has been evaluated through simulations for servo and regulatory problems of CSTR.

Shahriari-Kahkeshi and Askari [11] presented the design of a recurrent neural network trained Shuffled Frog Leaping Algorithm (RNN-SFLA) for identification and tracking control of a non-linear continuous stirred tank reactor (CSTR). The capability and efficiency of the proposed method is illustrated by the temperature control of a nonlinear CSTR. Sadeghi et al. [12] developed a new approach to tune a proportional-integral (PI) controller. To validate the proposed idea, a highly nonlinear plant, named Continuous Stirred Tank Reactor (CSTR) has been controlled by the nonlinear PI controller. A classical linear PI controller and a neuro controller are selected for comparison.

Na et al. [13] presented an adaptive neural control design for nonlinear pure-feedback systems with an input time-delay. Simulation on a continuous stirred tank reactor (CSTR) and practical experiments on a three-tank liquid level process control system are included to verify the reliability and effectiveness. Jafari and Salahshoor [14] developed an adaptive version of growing and pruning RBF neural network to predict the system output and implement Linear Model-Based Predictive Controller (LMPC) and Non-linear Model-based Predictive Controller (NMPC) strategies.

Man and Shao [15] presented a model predictive control strategy based on neural network for a continuous stirred tank reactor (CSTR). A segmentation method was adopted to identify Hammerstein-Wiener model coefficient by least squares support vector machines and then to construct a nonlinear predictive controller which was by a linear optimal component and radial basis function neural networks in series. Alexandridis and Sarimveis [16] presented a new methodology for controlling processes that exhibit multiple steady states. The proposed approach is based on a Model Predictive Control (MPC) framework, where the dynamics of the process are modelled by a Radial Basis Function (RBF) neural network. The proposed methodology has been applied to the control of a non-isothermal Continuous Stirred Tank Reactor (CSTR) that exhibits three steady state points.

Nekoui et al. [17] presented the optimal design of PID controller based on a particle swarm optimization (PSO) approach for continuous stirred tank reactor (CSTR). The mathematical model of experimental system had been approximate near the operating point for the PSO algorithm to adjust PID parameters for the minimum integral of time multiplied by absolute error condition. Ge et al. [18] proposed a new direct inverse model control strategy based on a new improved CMAC (Cerebellar Model Articulation Controller) neural network to control a kind of nonlinear system with strong hysteresis i.e. continuous-stirred tank reactor (CSTR).

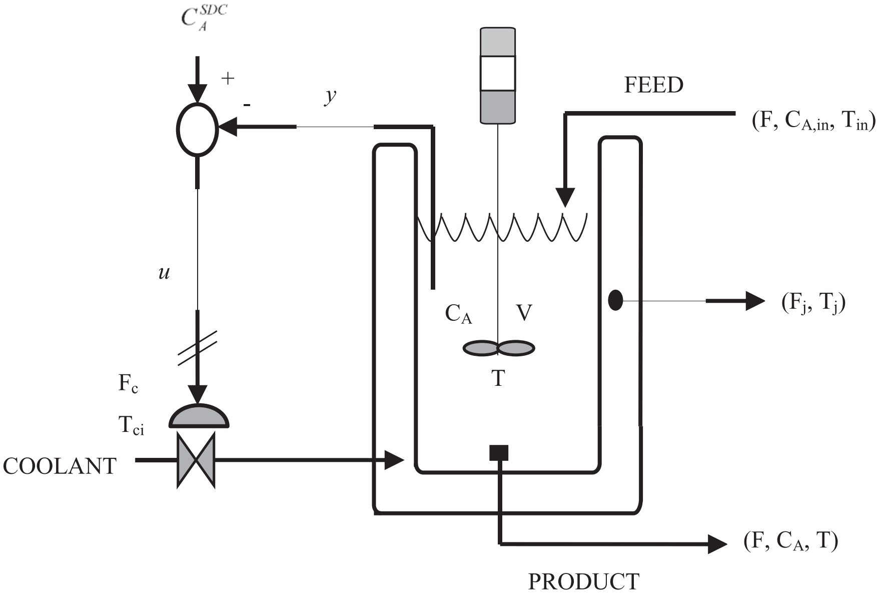

Fig. 1. CSTR Process with cooling jacket.

Vasičkaninova and Bakošova [19] developed a neural network model of a nonlinear plant to predict future plant performance. The controller calculates the control input that will optimize plant performance over a specified future time horizon. The proposed model has been simulated for a CSTR process. Ren and Rad [20] investigates a new adaptive non-linear compensation controller for a class of time-delay non-linear systems with partly known dynamics. The effectiveness of the proposed scheme has been demonstrated through the application to the control of a continuous stirred tank reactor.

Wang and Li-jun [21] proposed an adaptive inverse control based on neural network for the continuous stirred tank reactor (CSTR) system. The drought of prediction is introduced to adaptive inverse control. The predictive model of the CSTR system's output is established by model identifier, which adopts Elman network. Varshney et al. [22] demonstrated that neural networks can be used effectively for control of non-linear dynamical systems. The proposed control scheme is based on the artificial neural network and is applied to an isothermic continuous stirred tank reactor (CSTR). The modified back propagation algorithm has been used to train the neural networks. Neural network based model control scheme has been implemented for both set point and regulatory control action and the comparison have been made for a set of constant momentum term.

Zhai et al. [23] developed an intelligent proportional- integral-derivative (PID) control method by combining a traditional PID controller and a neural network compensator for the nonlinear term. The proposed control method is based on the traditional incremental digital PID controller, and multi-layer neural network is employed to estimate and compensate the nonlinear term, therefore, the proposed control can be easily realized in distributed control system (DCS). Kamalabady and Salahshoor [24] proposed an adaptive Nonlinear Model Predictive Control (NMPC) scheme for control of nonlinear time-varying processes. The proposed single input single output (SISO) and multi input multi output (MIMO) NMPC controllers are evaluated on a highly non-linear time-varying non-isothermal continuous stirred tank reactor (CSTR) benchmark problem. The simulation results demonstrate the potential capabilities of the two developed NMPC controllers to identify and control the CSTR process with superior performance over the conventional PID controller. Considering all these reviews on controller design for CSTR and observing their limitations on time delay compensation, increased steady state error and settling time, this paper focuses on developing a variant of classic neural model for model predictive control design.

The organization of the paper is as follows. Section 2 presents the description of the continuous stirred tank reactor along with the formulation of the mathematical model. Sections 3 and 4 provides the need for optimized controller to design CSTR model and an overview on deep learning neural network controller respectively. The proposed hybrid evolutionary optimization approach; PSO – GSA along with its algorithmic steps is elucidated in Section 5. The modelled DLNN controller for CSTR model is presented in Sections 6 and 7 gives the DLNN model predictive controller design for CSTR. Section 8 details the implementation and simulation results obtained employing the proposed controller model for the design of CSTR. Section 8 gives the conclusion of the developed work in this paper.

2. Process description and mathematical modelling of CSTR

In this paper, the control problem for an ideal continuous stirred tank reactor is considered and is as shown in Fig. 1. CSTR model performs exothermic and irreversible first order reaction, as specified by A → B wherein the stream of fluid

Please cite this article as: S.N. Deepa, I. Baranilingesan, Optimized deep learning neural network predictive controller for continuous stirred tank reactor, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.07.004

Table 1
CSTR steady state operating data.

CSTR process variables	Parametric values under normal operating condition
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/(g K)
Heat exchange	2e5 cal/mol
Reaction rate constant	7.2e10 min^{-1}
Activation energy term	9980 K

gets fed continuously to the reactor. As the CSTR gets mixed perfectly, the output stream possess the concentration and the temperature as the same that of the fluid in the reactor. The reactor is surrounded by the jacket, which as well possess necessary feed and exit output streams. There is a perfect mix within the jacket and it maintains lower temperature than that of the reactor. Energy gets passed within the walls of the reactor into the jacket and tends to remove the heat that gets generated by reaction. Typical examples exist in industries for these types of reactors. These industrial reactors possess complicated kinetics, but the characteristics are noted to be similar.

The various assumptions made in this process includes – there is a perfect mixing, cooling jacket is directly manipulated henceforth the balance of energy around the jacket is not required, shaft work is very minimal or negligible, possess constant parameter values, volume and other physical properties. Fundamentally, the law of kinetics is given by,

$$-r_A = k(T)c_A = k_o e^{\left(-\frac{E}{RT}\right)} c_A \tag{1}$$

With the assumptions made of constant volume, constant heat capacity of the reacting mixture and as well on the assumption that there exists perfect mixing within the reactor, the mass balance equation for A and an overall energy balance for the reactor is as given by,

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

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

Based on the assumption 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,

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

In the above equations, 'c_A' indicates the concentration of A, 'T' refers temperatures in Kelvin (K), 't' is the time in 'min', 'c' are the concentrations in 'mol/L', 'T_j' indicates the jacket temperature in Kelvin (K), 'F' specifies the volumetric flow rate in 'L/min', 'c_p' are the specific heat capacities in 'cal/(g K)', 'V' indicates the reactor volume in litres 'L', 'm_o' is the overall effective mass of cooling and heating system, 'ρ' specifies densities in 'g/L', 'c_w' is the water heat capacity in 'cal/(g K)', 'A_r' is the heat exchange surface in 'cal/mol', 'U' refers to the heat transfer coefficient in 'cal/(min K)', 'P' the power input to the heater in 'kW' and 'T_{CW}' refers to the cooling water temperature in Kelvin 'K'. The steady state operating data of the CSTR is taken and the transfer function derived based on these values is employed in the simulation studies. Table 1 presents the CSTR's steady state operating data [13].

On performing the linearization with respect to the concentration at the time of reaction and temperature, the transfer function model of the CSTR system is obtained as [17],

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

The state space model derived for the considered CSTR system is as given by [17],

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

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

3. Need for optimization based neural model predictive controller for derived CSTR model

It is well observed that even with the growth of technology and scientific progress, numerous physical process possess incomplete knowledge on the systems, they are imprecise and vague as well. Conventional methods employ exact mathematical model of the experimental phenomenon and then these exact models are employed for the prediction purpose. At this juncture, it is to be noted that there exists an indefinable structure and effect, and non-exactness for the original models. Thus, this paper devises an optimized neural network model predictive controller for carrying out the prediction and system identification process.

The applicability of neural network controller is due to its capability of storing the experienced knowledge and the complete data are made available for carrying out the prediction process. This paper focuses on devising the model predictive controller employing the deep learning neural network based on their better learning and parameter initialization capability and as well the existence of extensive number of hidden units. The benefits of DLNN includes – the information gets distributed, possess the ability to learn, permits knowledge indexing i.e., to hold more amount of information and easy accessibility of the stored information.

Since, the weight initialization of the modelled deep learning neural network is carried out in a random manner, this paper also developed a hybrid form of stochastic evolutionary based optimization algorithm – hybrid PSO – GSA. This proposed hybrid PSO – GSA algorithm is employed to tune the weights of the deep learning neural network for all the interlinked connections and the tuned DLNN carries out the model prediction of the CSTR process. The intrinsic non-linearity of the CSTR process makes the control process a challenging one. In this research paper, the design and development of a deep learning neural network model predictive controller is proposed for the isothermal CSTR with the reactions as specified in Section 2.

4. Deep learning neural network controller

The growth of deep learning neural networks is its deep architecture that contains multiple hidden layers and each hidden layer carries out a non-linear transformation between the layers [25]. DLNNs get trained based on two features:

- Pre-training of the deep neural networks employing unsupervised learning techniques like auto encoders layer by layer.
- Fine tuning of the DLNNs employing back propagation neural network.

Basically, auto encoders are employed with respect to the unsupervised learning technique and the input data is the output target of the auto-encoder. An auto encoder consists of two parts – encoder and decoder network. The operation of an encoder network is to transform the input data that is present in the form of a high-dimensional space into codes pertaining to low-dimensional space. The operation of the decoder network is to reconstruct the inputs from the corresponding codes. In encoder neural network, the encoding function is given by 'f_θ'. The encode vector (E^v) is given by,

$$E^v = f_\theta\left(x^v\right) \tag{7}$$

where, 'x^v' is the dataset of the measured signal.

The reconstruction operation is carried out at the decoder neural network and its function is given by 'g_θ'. This reconstruction function maps the data set 'x^v' from the low dimensional space into the high dimensional space. The reconstructed form is given by,

$$\hat{x}^v = g_\theta\left(E^v\right) \tag{8}$$

The ultimate of these encoder and decoder neural networks is to minimize the reconstruction error $E(x,\hat{x})$ for that many numbers of training samples considered. $E(x,\hat{x})$ is specified as a loss function that is used to measure the discrepancy between the encoded and decoded data samples. The key objective of the unsupervised auto encoder is to determine the parameter sets that minimize the reconstruction error 'E'.

$$\delta_{ae}(\theta, \theta') = \frac{1}{N}\sum_{v=1}^{N} E\left(x^v, g_{\theta'}\left(f_\theta\left(x^v\right)\right)\right) \tag{9}$$

The encoding and decoding function of the DLNN will be present along with a non-linearity and are given by,

$$\begin{aligned} f_\theta(x) &= f_{af\_e}(b + W_x) \\ g_{\theta'}(x) &= f_{af\_d}(b + W_x^T) \end{aligned} \tag{10}$$

where, f_{af_e} and f_{af_d} refers the encoder activation function and decoder activation function respectively, 'b' indicates the bias of the network, and W and W^T specifies the weight matrices of the DLNN model. The reconstruction error is given by,

$$E(x, \hat{x}) = \left\| x - \widehat{x} \right\|^2 \tag{11}$$

In order to carry out the pre-training of a DLNN model, the 'N' auto encoders developed in previous module should be stacked. For the given input signal x^v, the input layer along with the first hidden layer of DLNN are considered as the

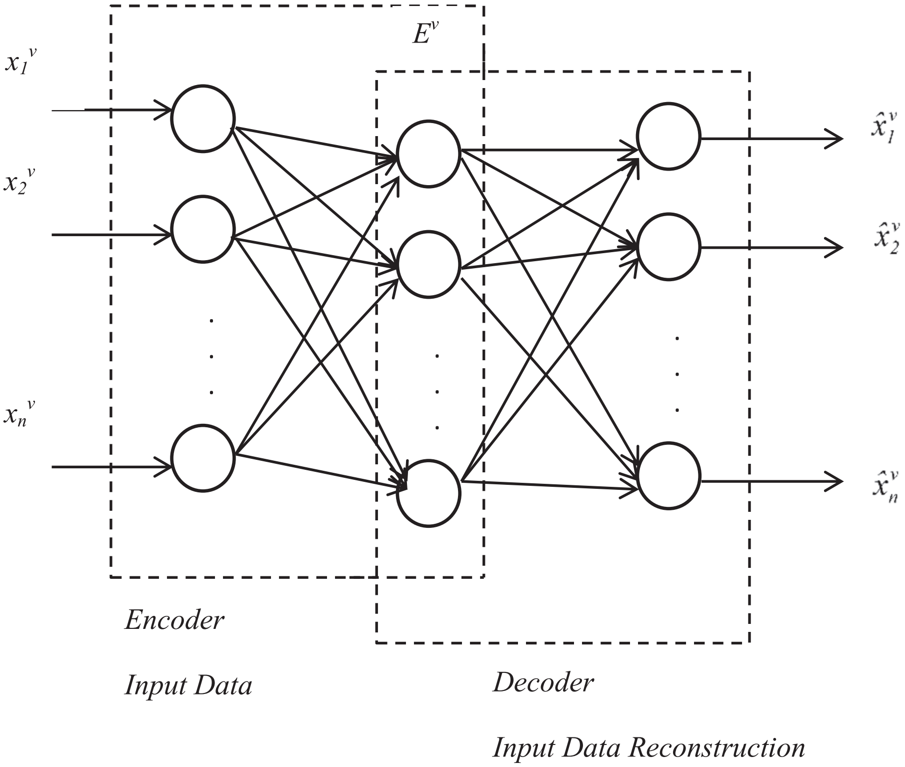

Fig. 2. Architecture model of Deep Learning Neural Network.

encoder neural network of the first auto encoding process. When the first auto encoder is noted to be trained by minimizing the reconstruction error, the first trained parameter set θ_1 of the encoder neural network is employed to initialize the first hidden layer of the DLNN and the first encode vector is obtained by,

$$E_1{}^{v} = f_{\theta_1}\left(x^v\right) \tag{12}$$

Now, the input data becomes the encode vector $E_1{}^{v}$, the first and the second hidden layer of the DLNN are considered as the encoder neural network for the second auto encoder. Subsequently, the second hidden layer of the DLNN gets initialized by that of the second trained auto encoder. This process gets continued upto the *N*th auto encoder that gets trained for initializing the final hidden layer of the DLNN model. The final or the *N*th encode vector in generalized form for the vector x^v is obtained by,

$$E_N{}^{v} = f_{\theta_N}\left(E_{N-1}{}^{v}\right) \tag{13}$$

where, '$\theta_{\rm N}$' denotes the *N*th trained parameter set of the encoder neural network. Thus, in this way, all the DLNN's hidden layer gets pre-trained by means of the N-stacked auto encoders. It is well noted that the process of pre-training avoids local minima and improves generalization aspect of the problem under consideration. Fig. 2 shows the fundamental architecture of the deep learning neural network model.

The above completes the pre-training process of DLNN and the next process is the fine-tuning process in the DLNN model. DLNN models output is calculated from the input signal x^v as,

$$y^v = f_{\theta_{N+1}}\left(E_N^v\right) \tag{14}$$

where, θ_{N+1} denotes the trained parameter set of the output layer. Here, Back Propagation Network (BPN) algorithm is employed for minimizing the error of the output by carrying out the parameter adjustments in DLNN backwards. In case, the output target of x^{v} is t^{v}, then the error criterion is given by,

$$MSE(\Psi) = \frac{1}{N}\sum_{v=1}^{N} E(y^v, t^v) \tag{15}$$

where, $\Psi = \{\theta_1, \theta_2, \theta_3, \ldots, \theta_{N+1}\}$.

The parameter set Ψ is updated by the following equation,

$$\Psi = \Psi - \alpha \frac{(\partial MSE(\Psi))}{\partial \Psi} \tag{16}$$

Where, 'α' is the learning rate of the fine tuning process.

5. Proposed population based hybrid PSO – GSA algorithm

This section presents the proposed hybrid particle swarm optimization and gravitational search algorithm as to be applied for tuning the weight values of deep learning neural networks in both auto encoder and decoder process in this research work. Fundamentally, PSO is a population based algorithm which moves on the search space to find better solutions improving the exploration and exploitation rate. Also, GSA is based on the force and the gravitational mass and acceleration which tend towards finding the better solution. Each of these algorithms individually when performs their process, at some point get stuck with stagnation and the generation loops without finding a necessary solution. Thus, to overcome the lacuna observed during their individual operations, the best features of both the algorithms are combined together to improve their exploration and exploitation rate such that they result in better tuning of weight values of DLNN model in this work.

5.1. Overview of PSO

The development of PSO is based on the simulation of social behaviour that occurs in bird flocking, fish schooling and so on [26]. PSO algorithm generates a swarm of individuals (called particles), where each particle is a candidate solution. Particles go on a simple behaviour: combine the better solutions obtained from neighbouring particles and its own achieved solutions. The position of a particle thus is influenced by the best particle in a neighbourhood, Pr_{bti} as well as the best solution found by the particle itself, becomes the global one Gl_{bt}. Therefore, the position of the particle p_i^{k+1} is adjusted using,

$$p_i^{k+1} = p_i^k + \varphi_i^{k+1}, \tag{17}$$

where, φ_i^{k+1} is the velocity component and it represents the step size. The velocity is computed employing the following equation,

$$\varphi_i^{k+1} = w\varphi_i^k + c_1 r_1 \{\underset{bti}{\mathrm{Pr}} - p_i^k\} + c_2 r_2 \{Gl_{bt} - p_i^k\}, \tag{18}$$

Where, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, r_1, $\mathrm{r}_2 \in \mathrm{U}(0, 1)$ are random numbers.

5.2. Overview of GSA

Newtonian physics theory is the fundamental for GSA algorithm and the collection of masses are its search agents. The most accepted theory is the Newton's law of universal gravitation, as its saying goes "every massive particle in the universe attracts other massive one with a force that is directly proportional to the product of their masses and is inversely proportional to the square of the distance between them" [27],

$$F = G\frac{M_1 M_2}{R^2}, \tag{19}$$

where, M_1 and M_2 are the masses of particles 1 and 2 respectively, R is the distance between the particle masses, G is the gravitational constant, and F is the magnitude of the gravitational force. Newton's second law states that when a force F is applied to a mass, its acceleration a only depends on the force and it's mass M and is given by,

$$a = \frac{F}{M} \tag{20}$$

Each and every mass gets accelerated towards their resultant forces that act on it from the other masses and is as given in Fig. 3. The position of the particle mass, p_i^{k+1} is adjusted using

$$p_i^{k+1} = p_i^k + \varphi_i^{k+1} \tag{21}$$

where, $\varphi_i^{k+1} = \varphi_i^k + a.$

5.2.1. Configuration of GSA

To configure GSA to get hybridized with PSO to tune DLNN model, primarily a mathematical modelling is done by the solution particles with 'N' particles [27],

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

The acceleration of mass at time, 't' and its dimension is given by,

$$a_i^k(t) = \frac{F_i^k(t)}{M_i(t)} \tag{22}$$

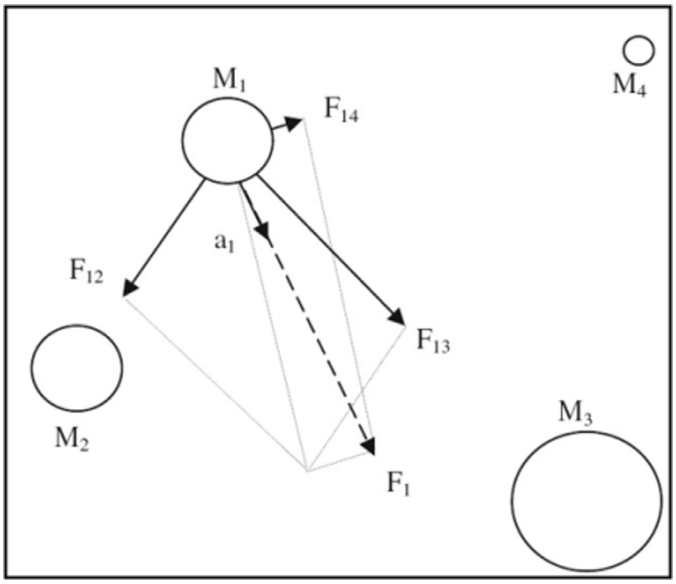

Fig. 3. GSA – A positional diagram on movement of mass.

The total force acting on the particle j is given by,

$$F_i^k(t) = \sum_{j\in 1,\, j\neq i}^{n} rand_j F_{ij}^k(t) \tag{23}$$

Where, $rand_j$ indicates a uniform random number between 0 and 1. At a particular time t, the force is given by,

$$F_{ij}^k(t) = G(t)\frac{M_i(t)M_j(t)}{R_{ij}(t)+\varepsilon}\left(p_j^k(t) - p_i^k(t)\right) \tag{24}$$

Where, $M_i(t)$ and $M_j(t)$ denotes the masses of objects i and j, $G(t)$ is the gravitational constant at time t, ε is a small constant, $R_{ij}(t)$ is the Euclidean distance between i and j objects and is given by,

$$R_{ij}(t) = \left\| d_i(t) - d_j(t) \right\|^2 \tag{25}$$

The gravitational constant is given by,

$$G(t) = G(t_0)\frac{t_0^\alpha}{t}, \alpha > 1, \tag{26}$$

where, $G(t_0)$ is the value of the gravitational constant, α is the user specified constant and t is the maximum number of iterations and t_0 refers to the current iteration number. The mass $M_i(t)$ is computed using the following equation,

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{n} q_j(t)} \tag{27}$$

with, $q_i(t) = \frac{P_i(t) - P_{\max\_i}(t)}{P_{\min\_i}(t) - P_{\max\_i}(t)}$, where, $q_i(t)$ is the strength of mass i at time t, and P_i is the fitness function of the problem under consideration.

5.3. Proposed hybrid PSO – GSA algorithm for DLNN model

As said in the introductory part of Section 5, the fundamental idea of the developed hybrid PSO – GSA is that it combines the social thinking capability of PSO to get global better solutions with that of the local search capability of GSA for exploring towards the better solution. As PSO is an effective meta-heuristic approach that is modelled based on the flocking behaviour of birds and the GSA is based on the gravitational law and the concept of interaction of masses, combining the position and velocity equation of both PSO and GSA, the modified position and velocity equation of hybrid PSO – GSA are given respectively as,

$$p_i^{k+1} = p_i^k + \varphi_i^{k+1} \tag{28}$$

$$\varphi_i^{k+1} = w\varphi_i^k + c_1 r_1 \times \underset{\underbrace{\downarrow}_{GSA\,Coef\,f\,icient}}{a_i^k} + c_2 r_2 \{\underset{\underbrace{\downarrow}_{PSO\,Coef\,f\,icient}}{Gl_{bt}} - p_i^k\} \tag{29}$$

The above formulated equations depict the hybridization process of the particle swarm optimization and gravitational search algorithmic process. The key parameter of GSA 'a_i^k', is incorporated for its local search mechanism and the parameter of PSO 'Gl_{bt}', depicting the social thinking capability is also brought into the velocity update mechanism.

Thus, the proposed hybrid PSO-GSA can be thought as a more adaptive form of population based stochastic algorithm. The algorithmic steps formulated to implement the proposed hybrid PSO – GSA are as follows:

Step 1: Initialize the number of particles, acceleration constants, number of generations and other necessary parameters.
Step 2: Randomly generate the number of particles (or agents).
Step 3: Evaluate the fitness for the generated particles, employing the defined fitness function (In case of DLNN model, the mean square error (MSE) will be the fitness function for the generations to go).
Step 4: When the fitness value is better than the best fitness value that has occurred so far, then set the current value as that of the new present particle value.
Step 5: Based on the evaluated fitness value, make the changes in the acceleration coefficient and obtain its value.
Step 6: Select the particle with the best fitness value as the global best value of all the particles.
Step 7: Then perform the position and velocity updation,
For each particle, calculate particle velocity,

$$\varphi_i^{k+1} = w\varphi_i^k + c_1 r_1 \times a_i^k + c_2 r_2 \{Gl_{bt} - p_i^k\} \tag{30}$$

Update the particle positions,

$$p_i^{k+1} = p_i^k + \varphi_i^{k+1} \tag{31}$$

The position of all the particles gets updated.
Step 8: Stopping condition: The algorithm repeats steps 2 to 7, until it reaches the maximum iterations or till the minimum error criterion is not satisfied. On reaching the specified stopping condition, the algorithm returns the global best particle value and its respective fitness value. (Any one of the stopping condition will only be used, mostly the convergence will be towards achieving the minimum error criterion).

6. Proposed tuning of DLNN model using developed hybrid PSO – GSA algorithm

The developed hybrid PSO – GSA algorithm is employed in this paper to tune the weight values of the encoder and decoder mechanisms of deep learning neural network model. Basically, the weight values of any neural network structures during the start-up of the learning process get randomly initialized. This work focuses on avoiding the random initialization of weights in DLNN, because random initialization leads to delayed convergence of the network and thus the weight values are tuned employing the proposed hybrid version of particle swarm optimization and gravitational search algorithm. The proposed algorithm for hybrid PSO – GSA based DLNN model is as follows:

Step 1: Start the algorithmic process.
Step 2: Obtain the training datasets to feed into the DLNN model and initialize the necessary parameters.
Step 3: Construct DLNN with 'N' hidden layers
Step 4: Perform the training of ith *auto encoder*
Step 5: Initialize ith *hidden layer parameters of DLNN employing the parameters of the auto encoder.*
Step 6: Check whether 'i' is greater than 'N', if 'no' carry out step 4, if yes go to the next step.
Step 7: Calculate the dimensions of the output layer.
Step 8: Fine tune the parameters of DLNN through the BPN algorithm.
Step 9: Invoke proposed hybrid PSO – GSA technique. The initial parameters for this case will be the final weight values computed at the fine tuning of the DLNN model. The particles for this case will be generated based on the seed values of the weights obtained in step 8 of this algorithm.
Step 10: Perform step 3 to step 8 of the algorithm given in Section 5.3.
Step 11: Return the tuned weight values for which the better fitness is attained.
Step 12: Employ these weight values from step 3 to step 8 of this algorithm. With the final fine-tuned DLNN model go to the next step.
Step 13: Return the trained DLNN.
Step 14: Output the solutions achieved.
Step 15: Stop the process on termination condition met. The termination condition is the number of iterations or reaching the minimal mean square error. (Any one of the stopping condition will only be used, mostly the convergence will be towards achieving the minimum error criterion).

Please cite this article as: S.N. Deepa, I. Baranilingesan, Optimized deep learning neural network predictive controller for continuous stirred tank reactor, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.07.004

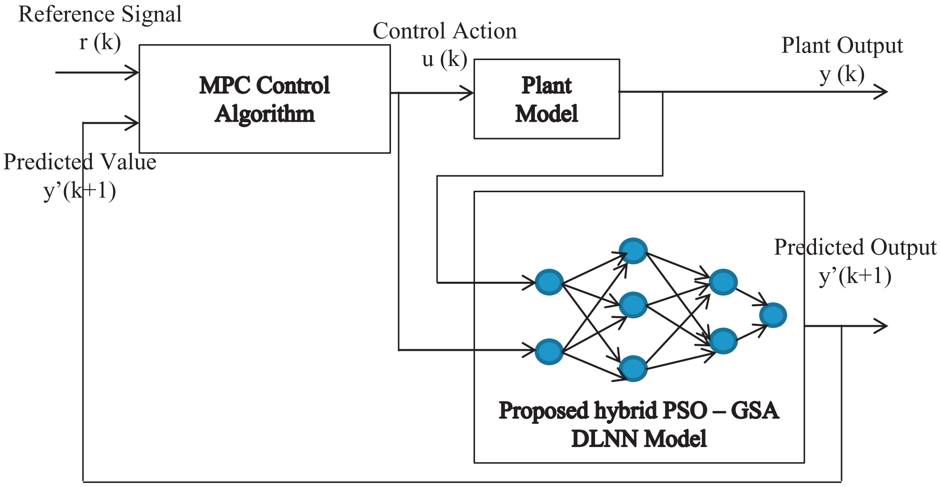

Fig. 4. Block diagram of the proposed DLNN Model Predictive Controller.

The error that is attained at the maximum number of iterations will be noted and if that is not found to be the best minimal error, the process will be repeated with re-initialization weights in deep learning neural networks and the training process will progress for tuned weights based on hybrid PSO – GSA algorithm. In this proposed approach, the pre-training process of DLNN handles non-linear transformations and fine tuning operation determines the discriminative information of the system under consideration. Further, tuning the weight values of DLNN model leads to faster convergence of the network thereby achieving better accuracy and improving the performance of the system. The forthcoming section presents the implementation of the proposed technique for model prediction of CSTR system.

7. Design of proposed hybrid PSO – GSA based DLNN model predictive controller for CSTR

The proposed hybrid PSO – GSA based deep learning neural network predictive controller of nonlinear CSTR is implemented and simulated in MATLAB R2013a (Version 8.1.0.604) environment and executed in a PC with Intel Core2 Duo Processor with 2.27 GHz speed and 2.00GB RAM. The modelled controller computes the output that optimizes plant performance over a given future time horizon. The first and foremost stage of model predictive controller is to carry out the system identification process by determining the neural network plant model. Further, the plant model is employed by the DLNNMPC for predicting the future performance. This section presents the modelling and simulation process of the system identification process, employing the proposed DLNNMPC model.

7.1. DLNN model predictive controller

As it is well known, the model predictive controller performs control actions based on the predicted values for the process. The completely trained neural network perfectly represents the proposed non-linear dynamic model of the system. The implementation of the control algorithms are carried out based on the errors between the predicted value and the reference signal. The proposed DLNN model predictive controller is as shown in Fig. 4.

In Fig. 4, the term '*k*' refers to the sample time. It is to be noted from Fig. 4, that the control action '*u(k)*' to act on the CSTR plant model is based on the predicted values and not based on the real output. The proposed model considers a deep learning neural network model that is trained accurately with minimal mean square error and is satisfactorily ready to represent the plant. The modelled predictive controllers tend to minimize the performance index '*J*' with respect to the error minimization and the control action is given by,

$$e(k+1) = r(k+1) - -y'(k+1) \tag{32}$$

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

$$u(k+1) = u(k) - \xi \frac{\partial J}{\partial u(k)} \tag{33}$$

Where, 'ξ' is the constant between 0 to 1.

7.2. System identification of the CSTR model

During the process of system identification, the first stage of model predictive control is to train the deep learning neural network model, so that it represents the dynamics of the CSTR system. The plant output and the control signal are employed

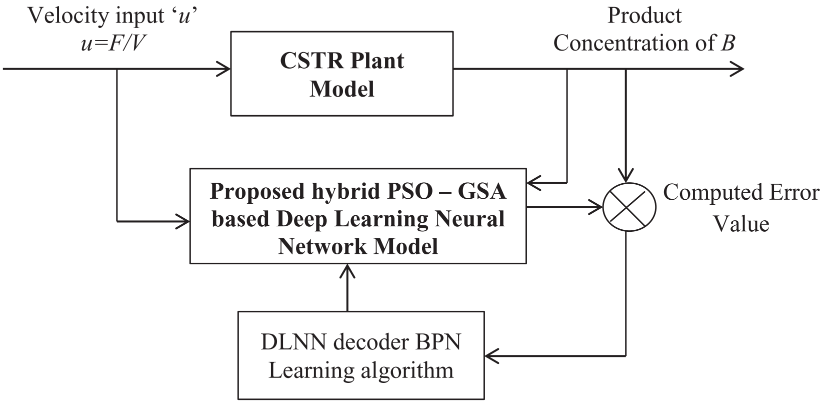
Velocity input ‘u’
u=F/V
CSTR Plant Model
Product Concentration of B
Proposed hybrid PSO – GSA based Deep Learning Neural Network Model
Computed Error Value
DLNN decoder BPN Learning algorithm

Fig. 5. Block diagram of system identification for the CSTR model.

Table 2
Parameters of hybrid PSO – GSA algorithm.

Parameters	GSA	Parameters	PSO
Gravitational constant	100	Particle size	50
α	20	Acceleration constants c_1 and c_2	2.0
ε	$2.2204e^{-16}$	w_{min}	0.2
Number of agents	50	w_{max}	1.0
Number of generations	100	v_i^{max}	1.0
		v_i^{min}	0.01
		Maximum generations	100

Table 3
Parameters for the proposed DLNNMPC model.

Parameters	DLNNMPC model
Weights and bias	Initially randomly initialized between 0 and 1
Unsupervised training (Auto encoder)	Employing encoders and decoders
Activation function	Sigmoidal function
Learning rate	0.2
Maximum iterations	100
Fine tuning method	Back propagation neural network
Learning rule	Gradient descent learning rule

to train the proposed DLNN model. The proposed DLNN model predictive controller employs the previous inputs and previous plant outputs for predicting the future values of the plant output. The proposed hybrid PSO – GSA based DLNN model is trained offline in batch mode, employing the data collected during the plant operation. The block diagram of the proposed system identification process for a CSTR model is as shown in Fig. 5.

8. Numerical simulation and results computed employing proposed DLNNMPC model

The developed hybrid PSO – GSA algorithm based DLNN model predictive controller is employed to perform the system identification process of the CSTR plant model. The optimal parameters chosen for the operation of hybrid PSO – GSA algorithm is tabulated in Table 2. Mean Square Error criterion is employed as the fitness function for tuning the weight values of the deep learning neural network controller employing the proposed hybrid PSO – GSA algorithm. Table 3 gives the parameters of the deep learning neural network controller that predicts the plant output based on the control action and the actual plant output.

The proposed hybrid PSO – GSA enables the DLNN model to obtain its tuned weight values during the training process and the mean square error generated over the 100 generations are as tabulated in Table 4. From Table 4, it is observed that at the end of 100th generation, the mean square error has reduced to a minimal value of 0.0357. Fig. 6 presents the convergence plot of the mean square error with respect to the number of generations of hybrid PSO – GSA for training the weight values of DLNN model with the fitness function as the mean square error criterion.

Based on the number of hidden neurons in the hidden layer, the respective training and testing efficiency noted are tabulated in Table 5. From Table 5, it is noted that when the number of hidden neurons is 5, best training and testing efficiency is achieved. The effectiveness of DLNN models lies in its deep learning process with pre training process at the

Table 4
Fitness function (mean square error) evolved during iterations.

Generations	Mean square error
10	5.9628
20	5.0007
30	4.3392
40	2.9604
50	2.2541
60	1.7658
70	0.9987
80	0.6420
90	0.1762
100	0.0357

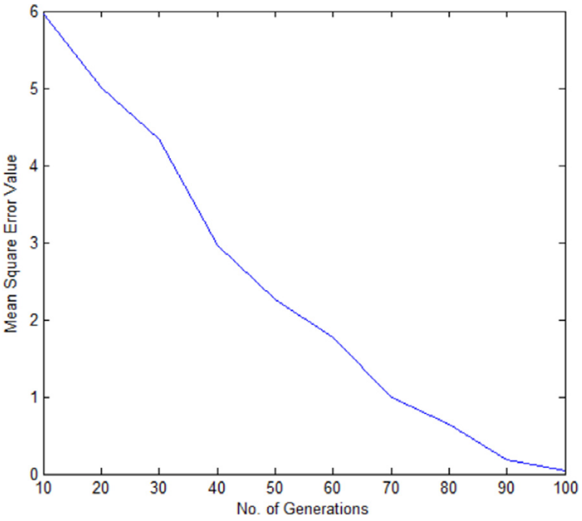

Fig. 6. Convergence plot of mean square error with respect to number of generations.

Table 5
Training and Testing efficiency of proposed PSO – GSA based DLNN model predictive controller model.

Proposed hybrid PSO – GSA based DLNNMPC model				
Number of hidden nodes	Training		Testing	
	Training time (in s)	Training accuracy	Testing time (in s)	Testing accuracy
5	**34**	**97.09%**	**22**	**97.16%**
10	65	96.16%	35	96.24%
15	72	94.03%	67	94.01%
20	109	93.92%	79	91.67%

auto encoders and fine tuning process at the decoder employing BPN algorithms and as well, the tuning of weights is achieved by the proposed hybrid PSO – GSA algorithm.

The performance of the proposed DLNNMPC model with its weights tuned by hybrid PSO – GSA is validated for its results through simulations with respect to set-point tracking and disturbance rejection problem with CSTR. The control problem is defined as: concentration of desired product (B) acts as the controlled variable and the manipulated variable is the velocity input ‘u’ which is given by (F/V). The considered disturbance in this case is the change in feed concentration of the reactant A. The closed loop solutions of DLNN model predictive controller is compared with that of the PID controllers for feed concentration of A. The normal operating parametric values considered include: feed concentration $C_A = 10\,mol/l$, $u = F/V = 0.5708$ l/min, concentration of the desired product = 1.117 mol/l. The parameters of the PID controllers are: $K_p = 5$, $K_i = 50$ and $K_d = 0.5$. From Table 5, it is inferred that as the number of hidden nodes increases the training and testing accuracy decreases, this is due to the fact that the hidden nodes have increased the complexity and for the CSTR application considered it is well noted to have less number of hidden nodes for better accuracy. Increasing the hidden nodes have made the saturation during the training progress and henceforth is the attained results.

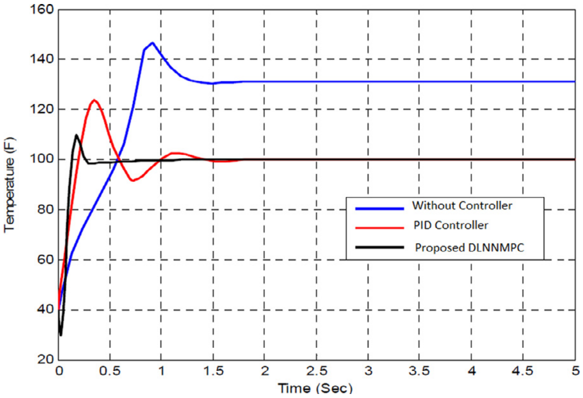

Fig. 7. Temperature response of CSTR model.

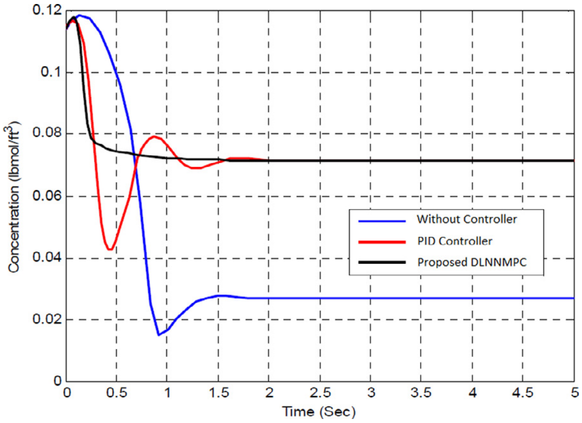

Fig. 8. Concentration control of CSTR model.

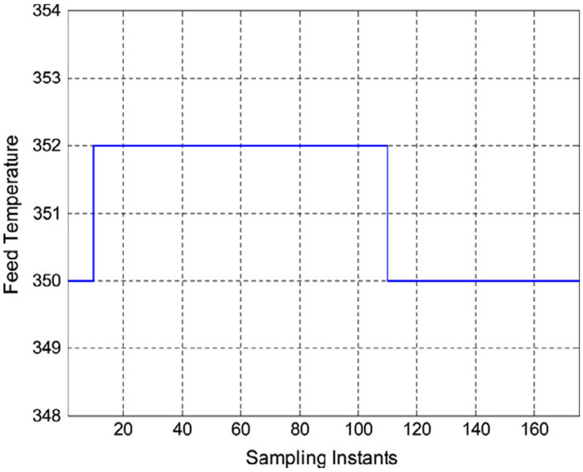

Fig. 9. Feed temperature variation of CSTR model.

The servo problem is analysed by giving a step change in set point of desired concentration from 1.1169 to 1.2003 and the temperature and concentration responses of the proposed DLNNMPC and conventional PID Controller are as shown in Fig. 7 and Fig. 8 respectively. The variation in the feed temperature is as shown in Fig. 9. The developed DLNNMPC is noted to perform better in comparison than that of the PID controller as shown in Figs. 7 and 8 respectively. Table 6 shows the integral square error computed employing the proposed controller and conventional PID controller. The performance of the

Table 6
Integral square error (ISE) for the considered controllers.

Devised controllers	Integral square error
PID controller [23]	0.3361
Neural network MPC [10]	0.31
NN adaptive control model [3]	0.163
Proposed DLNNMPC	1.11e−05

DLNNMPC is faster than PID and as well the error gets minimized as seen in Table 6. The carried out simulation process show that the proposed DLNNMPC tracks the set point faster in comparison with that of the conventional PID controller.

From Table 6, it is inferred that for the considered CSTR system, the proposed hybrid PSO – GSA based deep learning neural network controller model is implemented and its ISE value is noted to be 1.11e−05, which is minimal in comparison with that of the other models considered for validation from the literatures [3,10,23]. Hence, the proposed DLNNMPC is noted to be an effective model predictive controller for the CSTR system.

9. Conclusion

The model predictive controller based on deep learning neural network hybridized with evolutionary algorithms is designed for continuous stirred tank reactor in this paper. The system identification process is carried out employing the deep learning neural network - model predictive model. The proposed hybrid particle swarm optimization– gravitational search algorithm approach is employed for tuning the weight values of the deep learning neural network model. Social behaviour of particle swarm optimization and local capability of gravitational search algorithm is effectively considered and the hybrid particle swarm optimization – gravitational search algorithm has been devised for tuning the deep learning neural network model. The process of hybridization of particle swarm optimization and gravitational search algorithm results in improving the exploration and exploitation rate during the search process so that effective weights are obtained for the initial start-up of DLNN model. The modelled deep learning neural network - model predictive model is employed for predicting the output for the plant model. The results prove that the proposed hybrid particle swarm optimization – gravitational search algorithm based deep learning neural network model achieves better minimal integral square error in comparison with that of earlier other controllers available in the literature proving its effectiveness. For the set proportional – integral – derivative control values and the designed neural net model with the specified number of hidden nodes, better integral square error values were attained for the CSTR model. The developed algorithm with model predictive control resulted in effective control action for the continuous stirred tank reactor system.

References

[1] Li, J, Shi, Q., 2016. “Adaptive control for a class of uncertain pure-feedback nonlinear systems using Backstepping based on extreme learning machine “, CIESC J, Vol. 67, No. 7, pp: 2934–2943.

[2] Abdullah N, Yee TC, Mohamed A, Mustafa MM, Osman MH, Mohamad AB. Control of continuous stirred tank reactor using neural networks. Indian J Sci Technol 2016;9(21):1–24.

[3] Li DJ, Li DP. Adaptive controller design-based neural networks for output constraint continuous stirred tank reactor. Neurocomputing 2015;153:159–63.

[4] Li DJ. Neural network control for a class of continuous stirred tank reactor process with dead-zone input. Neurocomputing 2014;131:453–9.

[5] Shyamalagowri M, Rajeswari R. Neural network predictive controller based nonlinearity identification case study: nonlinear process reactor-CSTR. In: Advanced materials research, 984. Trans Tech Publications; 2014. p. 1326–34.

[6] Li D. Adaptive neural network control for a class of continuous stirred tank reactor systems. Sci Chin Inf Sci 2014;57(10):1–8.

[7] 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), 2014 6th, September. IEEE; 2014. p. 91–6.

[8] Todorov Y, Terzyiska M, Petrov M. Recurrent fuzzy-neural network with fast learning algorithm for predictive control. In: International conference on artificial neural networks, September. Berlin Heidelberg: Springer; 2013. p. 459–66.

[9] Alexandridis A, Stogiannos M, Kyriou A, Sarimveis H. An offset-free neural controller based on a non-extrapolating scheme for approximating the inverse process dynamics. J Process Control 2013;23(07):968–79.

[10] Prabhaker Reddy G, Radhika G, Anil K. Control of continuous stirred tank reactor using artificial neural networks based predictive control. In: Advanced materials research, 550. Trans Tech Publications; 2012. p. 2908–12.

[11] Shahriari-kahkeshi M, Askari J. Nonlinear continuous stirred tank reactor (CSTR) identification and control using recurrent neural network trained shuffled frog leaping algorithm. In: Control, instrumentation and automation (ICCIA), 2011 2nd international conference. IEEE; 2011. p. 485–9.

[12] Sadeghi MS, Bavafa F, Zarif MH, Alavi SMS. New approach to design a nonlinear PI controller for a CSTR system. In: 2nd international conference on control, instrumentation and automation (ICCIA) December. IEEE; 2011. p. 219–24.

[13] Na J, Ren X, Shang C, Guo Y. Adaptive neural network predictive control for nonlinear pure feedback systems with input delay. J Process Control 2012;22(1):194–206.

[14] Jafari MR, Salahshoor K. Adaptive predictive controllers using a growing and pruning RBF neural network. Iran J Chem Chem Eng 2011;30(2).

[15] Hong Man, Cheng SHAO. Neural network predictive control of continuous stirred-tank reactor based on Hammerstein-Wiener model. CIESC J 2011;8:032.

[16] Alexandridis A, Sarimveis H. Control of processes with multiple steady states using MPC and RBF neural networks. European symposium on computer-aided process engineering (ESCAPE-21), Thessaloniki, Greece; 2011.

[17] Nekoui MA, Khameneh MA, Kazemi MH. Optimal design of PID controller for a CSTR system using particle swarm optimization. In: 14th international conference on power electronics and motion control (EPE/PEMC), September. IEEE; 2010. p. T7–63.

[18] Ge Y, Ma S, Luo X. Direct inverse model control based on a new improved CMAC neural network. In: International conference on intelligent computing, August. Berlin Heidelberg: Springer; 2010. p. 25–32.

[19] Vasičkaninová A, Bakošová M. Neural network predictive control of a chemical reactor. Acta Chimica Slovaca 2009;2(2):21–36.

[20] Ren XM, Rad AB. Adaptive non-linear compensation control based on neural networks for non-linear systems with time delay. Int J Syst Sci 2009;40(12):1283–92.
[21] Wang H-N, Li-jun UN. Application of adaptive inverse control based on neural network for CSTR system. Fangzhi Gaoxiao Jichukexue Xuebao 2009;22(2):258–61.
[22] Varshney T, Varshney R, Sheel S. ANN based IMC scheme for CSTR. In: Proceedings of the international conference on advances in computing, communication and control, January. ACM; 2009. p. 543–6.
[23] Zhai LF, Chai TY, JING F. Intelligent PID control of continuous stirred tank reactors. J Syst Simul 2009;20(12):3753–7.
[24] Kamalabady AS, Salahshoor K. New SISO and MISO adaptive nonlinear predictive controllers based on self organizing RBF neural networks. In: ISCCSP 2008. 3rd international symposium on communications, control and signal processing, 2008, March. IEEE; 2008. p. 703–8.
[25] Grzegorczyk K, Kurdziel M, Wójcik PI. Encouraging orthogonality between weight vectors in pretrained deep neural networks. Neurocomputing 2016;202:84–90.
[26] Kennedy J, Eberhart RC. Swarm intelligence. 1st Ed. San Francisco, California: Morgan Kaufman Publishers; 2001.
[27] Rashedi Esmat, Nezamabadi-pour Hossein, Saryazdi Saeid. GSA: a gravitational search algorithm, 179. Elsevier, Information Sciences; 2009. p. 2232–48.

Please cite this article as: S.N. Deepa, I. Baranilingesan, Optimized deep learning neural network predictive controller for continuous stirred tank reactor, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.07.004

S.N. Deepa is an Associate Professor of Electrical Engineering at Anna University, Regional Campus: Coimbatore, Coimbatore, India. She has published 9 books with National/International Publishers, 79 International Journal Papers, 12 National Journal Papers and 53 Papers in National/International Conferences. Her Research areas include linear and non-linear control system design and analysis, soft computing techniques, adaptive and robust control systems and medical image processing.

I. Baranilingesan is a teaching fellow and research scholar of Electrical Engineering at Anna University, Regional Campus: Coimbatore, Coimbatore, India. He works in the area of process control; more into the design of controllers for considered process systems. His research areas include process control, automation, soft computing techniques and distributed control system.

Please cite this article as: S.N. Deepa, I. Baranilingesan, Optimized deep learning neural network predictive controller for continuous stirred tank reactor, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.07.004