

Research Article

Identification of a Typical CSTR Using Optimal Focused Time Lagged Recurrent Neural Network Model with Gamma Memory Filter

S. N. Naikwad¹ and S. V. Dudul²

¹ Department of Electrical Engineering, College of Engineering and Technology, Bahulgaon, Akola-444 104, India

² Department of Applied Electronics, Faculty of Engineering and Technology, Sant Gadgebaba Amravati University, Amravati-444 602, India

Correspondence should be addressed to S. N. Naikwad, snaikwad@rediffmail.com

Received 16 July 2009; Accepted 16 October 2009

Recommended by Zhigang Zeng

A focused time lagged recurrent neural network (FTLR NN) with gamma memory filter is designed to learn the subtle complex dynamics of a typical CSTR process. Continuous stirred tank reactor exhibits complex nonlinear operations where reaction is exothermic. It is noticed from literature review that process control of CSTR using neuro-fuzzy systems was attempted by many, but optimal neural network model for identification of CSTR process is not yet available. As CSTR process includes temporal relationship in the input-output mappings, time lagged recurrent neural network is particularly used for identification purpose. The standard back propagation algorithm with momentum term has been proposed in this model. The various parameters like number of processing elements, number of hidden layers, training and testing percentage, learning rule and transfer function in hidden and output layer are investigated on the basis of performance measures like MSE, NMSE, and correlation coefficient on testing data set. Finally effects of different norms are tested along with variation in gamma memory filter. It is demonstrated that dynamic NN model has a remarkable system identification capability for the problems considered in this paper. Thus FTLR NN with gamma memory filter can be used to learn underlying highly nonlinear dynamics of the system, which is a major contribution of this paper.

Copyright © 2009 S. N. Naikwad and S. V. Dudul. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

In any manufacturing process, where there is a chemical change taking place, a chemical reactor is at the heart of the plant. In size and appearance it may often seem to be one of the least impressive items of equipment, but its demand and performance are usually the most important factors in the design of whole plant. Depending on mode of operation, reactors are classified as batchwise or continuous. In batchwise mode, reactors are charged at the beginning of the reaction and products are removed at the end of the reaction. In continuous stirred tank reactor (CSTR), an agitator is deliberately introduced to disperse the reactants thoroughly into the reaction mixture immediately after they enter the tank. The stirred tank reactors are by their nature well suited to liquid-phase

reactions. Stirred tank reactors, by virtue of their large volume, provide a long residence time. This, combined with isothermal nature of reactor, permits operation at the optimum temperature for a long reaction time. Thus, stirred tank reactors have been employed on a commercial scale mainly for liquid-phase reaction systems at low or medium pressures. Stirring tank reactor exhibits nonlinear operations where reaction is exothermic. Thus, performance prediction becomes difficult due to high degree of nonlinearity hence exact mathematical modeling is not possible. However, due to development of neural networks, it is possible to develop learning machine based on neural network model that can learn from available experimental data. Thus a system model can be constructed by estimating unknown plant parameters using neural networks (as discussed elsewhere [1–8]).

Inspired from the structure of the human brain and the way it is supposed to operate, neural networks are parallel computational systems capable of solving a number of complex problems in such a diverse areas of as pattern recognition, computer vision, robotics, control and medical diagnosis, to name just few (as discussed by Haykin [9]). Neural networks are an effective tool to perform any nonlinear input output mappings. It was the Cybenko (as discussed by Cybenko [10]), who first proved that, under appropriate conditions, they are able to uniformly approximate any continuous nonlinear function to any desired degree of accuracy. It is these fundamental results that allow us to employ neural network for system identification purpose. One of the primary reasons for employing neural network was to create a machine that was able to learn from experience. They have the capability to learn the complex nonlinear mappings from a set of observations and predict the next outcome (“as discussed by Dudul [11]”).

The present paper carries out neural network based identification and modeling of a typical continuous stirred tank reactor using famous neural network like focused time lag recurrent neural network (FTLR NN) with gamma memory filter. The optimal model is estimated on the basis of performance measures like MSE (Mean Square Error), NMSE (Normalized Mean Square Error), r (Correlation coefficient), and visual inspection of regression characteristics on the testing data sets. Finally, it is shown that dynamic NN model has a remarkable system identification capability for the CSTR processes.

2. Estimation of NN Model

A CSTR in Reaction Engineering laboratory of College of Engineering & Technology, Akola (Maharashtra, India) is used for experimentation.

It consists of a universal motor of 4000 rpm, 0.6 A, 220–230 V manufactured by REMI Motors, Bombay. The input/output experimental data has been obtained through rigorous experimentation carried out on CSTR in a laboratory by varying flow rate of input reactant from zero to maximum, and corresponding output (concentration of liquid) is tested for each instance. The simulation data constitutes 383 samples. In fact the process is multi-input single output, where the output variable is concentration of liquid and input variables are stirring speed, temperature, and flow rate. In this experiment, as the stirring speed and temperature are held constant to their normal values the system could be created as single input-single output (SISO). Another benchmark data for CSTR process is obtained from Internet which is contributed by Jairo ESPINOSA ESAT-SISTA KULEUVEN, Kardinaal Mercierlaan 94, B-3001, Heverlee, Belgium. Here the process is a Continuous Stirring Tank Reactor where reaction is exothermic and the concentration is controlled by regulating coolant flow. It consists of 7500 samples. (“as discussed elsewhere [12, 13]”).

As process exhibits time relationship in the input-output mappings, versatile FTLR NN model is particularly used to describe the system behaviour. The weights are adjustable

parameters of the system and they are determined from a set of examples through a process called training. The exemplars, or the training data as they are usually called, are the sets of inputs and corresponding desired outputs. When NN has been trained, the next step is to evaluate it. This is done by standard method in statistics called Independent validation. This method divides the available data into a training set and a test set. The entire data is usually randomized first. The training data is next split into two partition; the first partitions is used to update the weights in the network, and the second partition is used to assess (or cross validate) the training performance. The test data is then used to assess how well the network has generalized. The learning and generalization ability of the estimated NN based model is assessed on the basis of certain performance measures such as NMSE, correlation coefficient, and the regression ability of the NN by visual inspection of the regression characteristics for different output of the system under study (“as discussed by Narendra and Parthasarathy [14]”). Neurosolutions (version 5.0) is specifically used for obtaining results.

2.1. Performance Measures

2.1.1. *MSE (Mean Square Error)*. The formula for the mean squared error is

$$\text{MSE} = \frac{\sum_{j=0}^P \sum_{i=0}^N (d_{ij} - y_{ij})^2}{NP}, \quad (1)$$

where P = number of output PEs (processing elements), N = number of exemplars in the data set, y_{ij} = network output for exemplar i at PE j , and d_{ij} = desired output for exemplar i at PE j .

2.1.2. *NMSE (Normalized Mean Square Error)*. The normalized mean squared error is defined by the following formula:

$$\text{NMSE} = \frac{PN \text{MSE}}{\sum_{j=0}^P \left(\left(N \sum_{i=0}^N d_{ij}^2 - \left(\sum_{i=0}^N d_{ij} \right)^2 \right) / N \right)}, \quad (2)$$

where P = number of output PEs, N = number of exemplars in the data set, MSE = mean square error, and d_{ij} = desired output for exemplar i at PE j .

2.1.3. *r (Correlation Coefficient)*. The size of the mean square error (MSE) can be used to determine how well the network output fits the desired output, but it does not necessarily reflect whether the two sets of data move in the same direction. For instance, by simply scaling the network output, we can change the MSE without changing the directionality of the data. The correlation coefficient (r) solves this problem. By definition, the correlation coefficient

TABLE 1: Parameters of FTLR NN (Lab. data). Number of Exemplars: Training = 268, Testing = 115, Max epoch = 1000, Focused, Gamma, Norm \rightarrow L-infinity.

Sr. no.	Parameters	Hidden layer # 1	Output layer
1	Processing Elements	42	1
2	Transfer function	tan h	tan h
3	Learning rule	Momentum	Momentum
4	Step size	0.1	0.1
5	Momentum	0.7	0.7

between a network output x and a desired output d is

$$r = \frac{\sum_i (x_i - \bar{x})(d_i - \bar{d})/N}{\sqrt{\sum_i (d_i - \bar{d})^2/N} \sqrt{\sum_i (x_i - \bar{x})^2/N}}, \quad (3)$$

where $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$, $\bar{d} = \frac{1}{N} \sum_{i=1}^N d_i$.

The correlation coefficient is confined to the range $[-1, 1]$. When $r = 1$, there is a perfect positive linear correlation between x and d , that is, they covary, which means that they vary by the same amount. When $r = -1$, there is a perfectly linear negative correlation between x and d , that is, they vary in opposite ways (when x increases, d decreases by the same amount). When $r = 0$, there is no correlation between x and d , that is, the variables are called uncorrelated. Intermediate values describe partial correlations. For example a correlation coefficient of 0.88 means that the fit of the model to the data is reasonably good.

2.2. Modeling of CSTR Using FTLR NN Model. As there is a time structure underlying the data collected after rigorous experimentation, dynamic modeling will certainly help to improve the performance. Dynamic NNs are topologies designed to explicitly include time relationships in the input-output mappings. Time constitutes an indispensable component of the learning process. It is through the inclusion of time into operation of NN that it is enabled to follow statistical variations in nonstationary processes. Time lagged recurrent networks (TLRNs) are MLPs extended with short-term memory structures. Here, a ‘‘Static’’ NN (e.g., MLP) is endowed with dynamic properties (as discussed by Dudul [15]). This, in turn, makes the network reactive to the temporal structure of information bearing signals. For an NN to be dynamic, it must be given memory. This memory may be classified into ‘‘short-term’’ and ‘‘long-term’’ memory. Long-term memory is built into an NN through supervised learning, whereby the information content of training data set is stored (in part or in full) in the synaptic weights of the network (as discussed by Principe et al. [16]). However, if the task at hand has a temporal dimension, some form of ‘‘short-term’’ memory is needed to make the network dynamic.

The input processing elements of an MLP are replaced with a tap delay line, which is followed by an MLP NN. This topology is called focused time-delay NN (TDNN).

TABLE 2: Parameters of Fully Recurrent NN (Lab. data). Number of Exemplars: Training = 306, Testing = 77, Max epoch = 1000, memory-axon, Norm \rightarrow L1.

Sr. no.	Parameters	Hidden layer # 1	Output layer
1	Processing Elements	18	1
2	Transfer function	tan h	tan h
3	Learning rule	Momentum	Momentum
4	Step size	0.1	0.1
5	Momentum	0.7	0.7

TABLE 3: Parameters of FTLR NN (Benchmark data). Number of Exemplars: Training = 3000, Testing = 4500, Max epoch = 1000, Partially recurrent, Gamma, Norm \rightarrow L-infinity.

Sr. no.	Parameters	Hidden layer # 1	Output layer
1	Processing Elements	39	1
2	Transfer function	tan h	sigmoid
3	Learning rule	dbd	dbd

TABLE 4: Parameters of Fully Recurrent NN (Benchmark data). Number of Exemplars: Training = 3000, Testing = 4500, Max epoch = 1000, Partially recurrent, Gamma, Norm \rightarrow L-infinity.

Sr. no.	Parameters	Hidden layer 1	Hidden layer 2	Output layer
1	Neurons	16	27	1
2	Transfer function	tan h	tan h	Linear tan h
3	Learning rule	dbd	dbd	dbd

The focused topology only includes the memory kernels connected to the input layer. This way, only past of the input is remembered. The delay line of the focused TDNN stores the past sample of the input. The combination of tap delay line and the weights that connect the tap to the PEs of the first hidden layer is simply linear combiners followed by a static nonlinearity.

Typically, a gamma short-term memory mechanism is combined with nonlinear PEs in restricted topologies called focused. Basically, the first layer of the focused TDNN is a filtering layer, with as many adaptive filters as PEs in the first hidden layer. The outputs of the linear combiners are passed through a nonlinearity (of the hidden-layer PE) and are then further processed by the subsequent layers of the MLP for system identification, where the goal is to find the weights that produce a network output that best matches the present output of the system by combining the information of the present and a predefined number of past samples (given by the size of the tap delay line).

Size of the memory layer depends on the number of past samples that are needed to describe the input characteristics in time. This number depends on the characteristics of the input and the task. This focused TDNN can still be trained with static backpropagation, provided that a desired signal is available at each time step. This is because the tap delay line at the input layer does not have any free parameters, so the only adaptive parameters are in the static feedforward path.

TABLE 5: Comparison of NN models.

NN Models	Performance Measures on Test Data Set					
	CSTR lab data			Benchmark CSTR data		
	MSE	NMSE	r	MSE	NMSE	r
FTLR NN with Gamma memory	1.13E-05	0.470	0.9132	1.40E-06	0.00777	0.9961
Fully Recurrent NN	8.01E-06	0.754	0.702	4.15E-06	0.0230	0.988

The memory PE receives in general many inputs $x_i(n)$, and produces multiple outputs $\mathbf{y} = [y_0(n), \dots, y_D(n)]^T$, which are delayed versions of $y_0(n)$, the combined input,

$$y_k(n) = g(y_{k-1}(n)), \quad y_0(n) = \sum_{j=1}^p x_j(n), \quad (4)$$

where $g(\cdot)$ is a delay function.

These short-term memory structures can be studied by linear adaptive filter theory if $g(\cdot)$ is a linear operator. It is important to emphasize that the memory PE is a short-term memory mechanism, to make clear the distinction from the network weights, which represent the long-term memory of the network.

There are basically two types of memory mechanisms: memory by delay and memory by feedback. We seek to find the most general linear delay operator (special case of the Auto Regressive Moving Average model) where the memory traces $y_k(n)$ would be recursively computed from the previous memory trace $y_{k-1}(n)$. This memory PE is the generalized feedforward memory PE. It can be shown that the defining relationship for the generalized feedforward memory PE is (as discussed by Principe et al. [16])

$$g_k(n) = g(n) * g_{k-1}(n), \quad k \geq 1, \quad (5)$$

where $*$ is the convolution operation, $g(n)$ is a causal time function, and k is the tap index. Since this is a recursive equation, $g_0(n)$ should be assigned a value independently. This relationship means that the next memory trace is constructed from the previous memory trace by convolution with the same function $g(n)$, the memory kernel yet unspecified. Different choices of $g(n)$ will provide different choices for the projection space axes. When we apply the input $x(n)$ to the generalized feedforward memory PE, the tap signals $y_k(n)$ become

$$y_k(n) = g(n) * y_{k-1}(n), \quad k \geq 1, \quad (6)$$

the convolution of $y_{k-1}(n)$ with the memory kernel. For $k = 0$, we have

$$y_0(n) = g_0(n) * x(n), \quad (7)$$

where $g_0(n)$ may be specified separately. The projection $\hat{x}(n)$ of the input signal is obtained by linearly weighting the tap signals according to

$$\hat{x}(n) = \sum_{k=0}^D w_k y_k(n). \quad (8)$$

The most obvious choice for the basis is to use the past samples of the input signal $x(n)$ directly, that is, the k th tap signal becomes $y_k(n) = x(n - k)$. This choice corresponds to

$$g(n) = \delta(n - 1). \quad (9)$$

In this case, $g_0(n)$ is also a delta function $\delta(n)$ (delta function operator used in the tap delay line). The memory depth is strictly controlled by D , that is, the memory traces store the past D samples of the input. The time delay NN uses exactly this choice of basis.

The gamma memory PE attenuates the signals at each tap because it is a cascade of leaky integrators with the same time constant. The gamma memory PE is a special case of the generalized feedforward memory PE where

$$g(n) = \mu(1 - \mu)^n, \quad n \geq 1, \quad (10)$$

and $g_0(n) = \delta(n)$. The gamma memory is basically a cascade of lowpass filters with the same time constant $1 - \mu$. The overall impulse response of the gamma memory is

$$g_p(n) = \binom{n-1}{p-1} \mu^p (1 - \mu)^{n-p}, \quad n \geq p, \quad (11)$$

where $\binom{n}{p}$ is a binomial coefficient defined by $\binom{n}{p} = n(n-1) \cdots (n-p+1)/p!$ for integer values of n and p , and the overall impulse response $g_p(n)$ for varying p represents a discrete version of the integrand of the gamma function (as discussed by De Vries and Principe [17]), hence the name of the memory.

The gamma memory PE has a multiple pole that can be adaptively moved along the real Z-domain axis, that is, the gamma memory can implement only lowpass ($0 < \mu < 1$) or highpass ($1 < \mu < 2$) transfer functions. The highpass transfer function creates an extra ability to model fast-moving signals by alternating the signs of the samples in the gamma PE (the impulse response for $1 < \mu < 2$ has alternating signs). The Depth in Samples parameter (D) is used to compute the number of taps (T) contained within the memory structure(s) of the network.

2.3. Modeling of CSTR Using Recurrent NN Model. Fully recurrent networks feedback the hidden layer to itself. Partially recurrent networks start with a fully recurrent net and add a feedforward connection that bypasses the recurrency, effectively treating the recurrent part as a state memory. These recurrent networks can have an infinite memory depth and thus find relationships through time as well as through the instantaneous input space. Most



FIGURE 1: Continuous stirred tank reactor.

real-world data contains information in its time structure. Recurrent networks are the state of the art in nonlinear time series prediction, system identification, and temporal pattern classification.

There are four input layer structures to choose from. If your data is multidimensional, you should try a simple Axon first. If your data is a one-dimensional time series, then use one of the memory axons. There are two recurrent structures to choose from. The fully recurrent structure connects the first hidden layer to itself through a recurrent synapse connection. The partially recurrent structure adds a feedforward connection, through a synapse, from the input axon to the layer after the 1st hidden layer. In this case, the recurrent structure acts as a state for the feedforward structure.

3. Simulation of Recurrent NN Models

An exhaustive and careful experimental study has been carried out to determine optimal configuration of the different NN models. All possible variations are tried to decide number of hidden layers and number of neurons in each hidden layer on the basis of performance measures. Training and testing percentages of exemplar are then varied to get optimum training-testing exemplars for each NN model. Different supervised learning rules, different transfer functions, and different transfer functions in output layer are investigated in simulation. Finally effects of different norms are tested on the model to decide optimal neural network. After meticulous examination of performance measures like MSE, NMSE, correlation coefficient, and the regression ability of the NN models on test data set, the optimal parameters are decided for the model as listed in Tables 1, 2, 3, and 4.

A rigorous experimental study has been undertaken in order to determine the optimal value of the gamma parameter. Again, for every variation, the network is run three times with different random weight-initialization. In computer simulation, a gamma parameter is gradually varied from 0.0 to 1.8 in the interval of 0.1 while maintaining all other parameters of the FTLRNN at their nominal default values. The results of variation of gamma parameter are

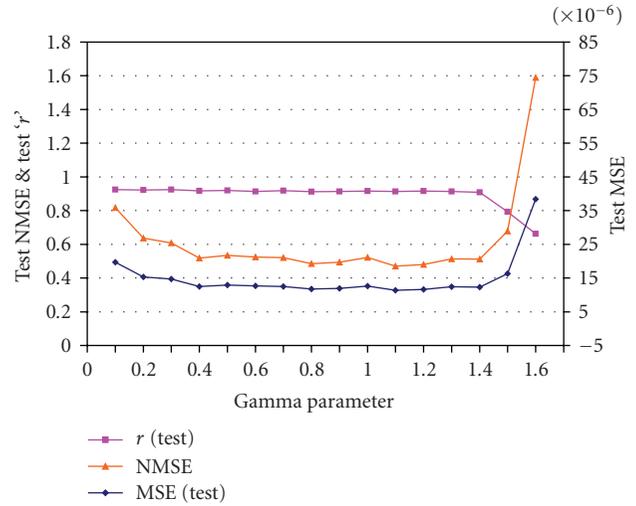


FIGURE 2: Choice of Gamma (CSTR lab. data).

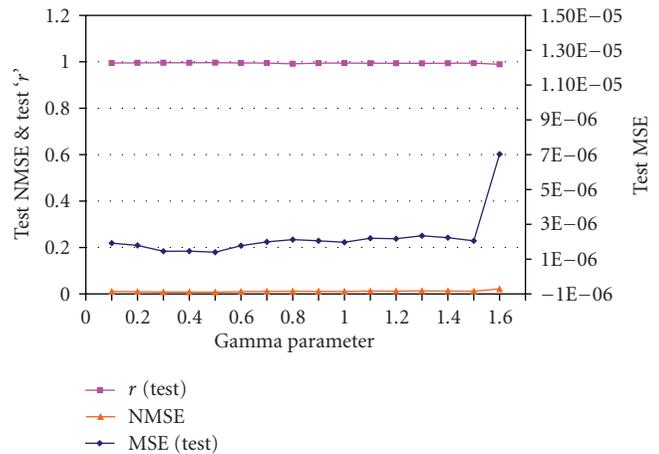


FIGURE 3: Choice of Gamma (CSTR Benchmark Data).

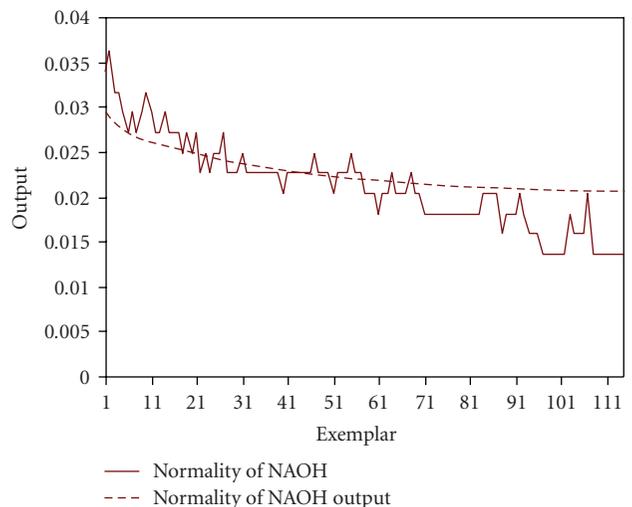


FIGURE 4: Desired output and Network output for testing data set (FTLR NN model-Lab. data).

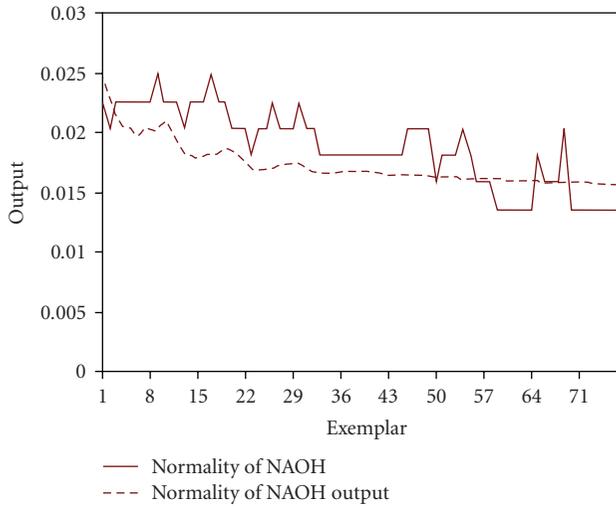


FIGURE 5: Desired output and Network output for testing data set (Fully recurrent NN model-Lab. data).

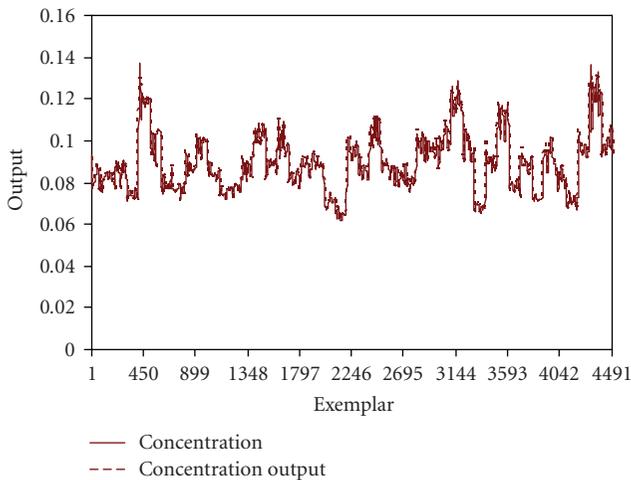


FIGURE 6: Desired output and Network output for testing data set (FTLR NN model-Benchmark data).

graphed in Figures 2 and 3. Their careful inspection reveals that with the initial increase in the gamma, MSE and NMSE on the test data set start decreasing. At the same time, the correlation coefficient shows little effect. This enthusiastic trend continues until a threshold value of gamma is reached, beyond which MSE and NMSE begin increasing along with little decrease in the correlation coefficient. From close observation of Figures 2 and 3, the threshold is noticed as 1.1 and 0.5, respectively, for the best identification performance of the FTLR NN models with gamma memory.

Figures 4 and 6 demonstrate the regression ability of the FTLR NN models on testing data set with gamma memory at 1.1 and 0.5, respectively. Here desired output is compared with actual output produced by neural network models with gamma memory and from close visual inspection, it is noticed that this model (with gamma memory) elegantly

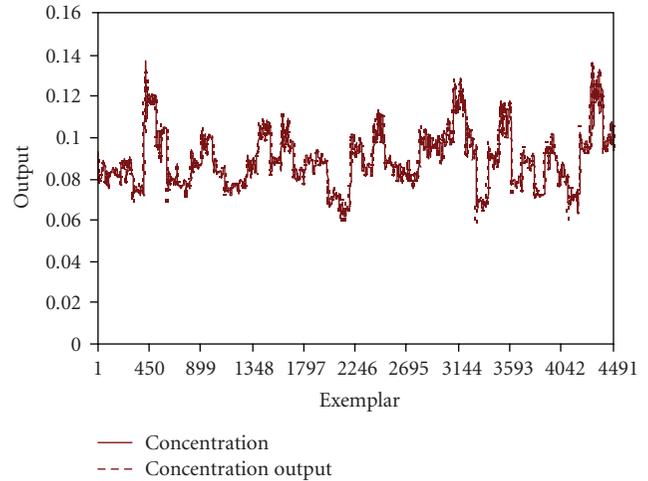


FIGURE 7: Desired output and Network output for testing data set (Fully recurrent NN model-Benchmark data).

learns the rich nonlinear dynamics of the system. Comparison of Figures 4 and 5 (CSTR lab. data) and Figures 6 and 7 (CSTR benchmark data) reveals that regression ability of FTLR NN models is far better than Fully recurrent NN models. This fact is also confirmed from comparison of NN models with respect to their performance measures as given in Table 5.

4. Conclusion

It is seen that FTLR NN model is capable of learning nonlinear dynamics of two CSTR processes. In this paper, it is demonstrated that FTLR NN models with gamma memory filter very closely follow desired output of CSTR processes for the testing instances. From the results presented, it is seen that FTLR NN models with gamma memory filter at 1.1 for laboratory data and at 0.5 for net data have an edge over fully recurrent NN models when performance measures and visual inspection of regression characteristics are taken into consideration. It is thus concluded that for identification of CSTR process using neural networks, FTLR NN model with gamma memory filter can be used to learn underlying highly nonlinear dynamics of the system which is the major contribution of this paper.

References

- [1] S. S. Ge, J. Zhang, and T. H. Lee, "Direct MNN control of continuous stirred tank reactor based on input output model," in *Proceedings of the 41st SICE Annual Conference (SICE '02)*, vol. 5, pp. 2770–2775, 2002.
- [2] O. C. Daniel and M. H. Elder, "A chemical reactor benchmark for parallel adaptive control using feed forward neural networks," in *Proceedings of the 6th Brazilia Symposium on Neural Network*, Riode Janeiro, Brazil, 2000.
- [3] S.-R. Lin and J.-S. Yu, "Robust control based on neuro-fuzzy systems for a continuous stirred tank reactor," in *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1483–1488, 2002.

- [4] Y. Tian, J. Zhang, and A. J. Morris, "Neural network based optimal control of fed batch reactor," in *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT '98)*, vol. 1, pp. 308–312, 1998.
- [5] S. N. Jyothi and M. Chidambaram, "Identification of Hammerstein model for reactors with input multiplicity," *Indian Chemical Engineer*, vol. 45, p. 33, 2003.
- [6] I. S. Baruch, P. Georgieva, L. A. P. Hernandez, and B. Nenkova, "An adaptive integral plus states neural control of aerobic continuous stirred tank reactor," in *Proceedings of the IEEE International Conference on Intelligent Systems*, vol. 1, pp. 337–343, 2004.
- [7] F. Liu and J. Chen, "A design of constrained fuzzy controller for CSTR's," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 2249–2253, 2004.
- [8] M. Khodabandeh and H. Bolandi, "Model predictive control with state estimation and adaptation mechanism for a continuous stirred tank reactor," in *Proceedings of the International Conference on Control, Automation and Systems*, pp. 1466–1471, 2007.
- [9] S. Haykin, *Neural Network : A Comprehensive Foundation*, Pearson Education, Delhi, India, 2nd edition, 2003.
- [10] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [11] S. V. Dudul, "Prediction of a Lorenz chaotic attractor using two-layer perceptron neural network," *Applied Soft Computing Journal*, vol. 5, no. 4, pp. 333–355, 2005.
- [12] J. D. Morningred, B. E. Paden, D. E. Seborg, and D. A. Mellichamp, "An adaptive nonlinear predictive controller," in *Proceedings of the American Control Conference (ACC '90)*, vol. 2, pp. 1614–1619, 1990.
- [13] G. Lightbody and G. W. Irwin, "Nonlinear control structures based on embedded neural system models," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 553–567, 1997.
- [14] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic system using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1992.
- [15] S. V. Dudul, "Identification of a liquid saturated steam heat exchanger using focused time lagged recurrent neural network model," *IETE Journal of Research*, vol. 53, no. 1, pp. 69–82, 2007.
- [16] J. C. Principe, N. R. Euliano, and C. Lefebvre, *Neural and Adaptive Systems-Fundamental through Simulation*, John Wiley & Sons, New York, NY, USA, 1st edition, 2000.
- [17] B. De Vries and J. C. Principe, "The gamma model—a neural model for temporal processing," *Neural Networks*, vol. 5, pp. 565–576, 1992.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

