Indian Journal of Science and Technology Vol. 3 No. 4 (Apr. 2010) ISSN: 0974- 6846

Novel FTLR NN model with gamma memory filter for identification of a typical magnetic stirrer

S.N. Naikwad and S.V. Dudul[1]

Dept. of Electrical Engineering, College of Engineering & Technology, Babhulgaon, Akola-444 104, India
[1]Dept. of Applied Electronics, Faculty of Engg. & Technol., Sant Gadgebaba Amaravati Univ., Amravati-444 602, India
snnaikwad@rediffmail.com; dudulsv@rediffmail.com

Abstract

In this paper, a novel focused time lagged recurrent neural network (FTLR NN) with gamma memory filter is designed to learn the subtle complex dynamics of a typical magnetic stirrer process. Magnetic stirrer exhibits complex nonlinear operations where reaction is exothermic. It appears to us that identification of such a highly nonlinear system is not yet reported by other researchers using neural networks. As magnetic stirrer process includes time 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, effect of different norms are tested along with variation in gamma memory filter. It is shown that dynamic NN model has a remarkable system identification capability for the problem 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 major contribution of this paper.

Keywords: Magnetic stirrer; focused time lag recurrent neural network; gamma memory filter.

Introduction

In any manufacturing process a chemical reactor is at the heart of the plant. Magnetic stirrer is one of the most widely used utility device for pH measurement, reaction rates, distillation and agitation purposes. Magnetic stirrer is a chemical reactor which in size and appearance seems to be one of the least impressive item in manufacturing process but its demand and performance are usually the most important factors in the design of a whole plant. In designing a reactor the aim is to produce a specified product at a given rate from known reactants. As magnetic stirrer exhibit nonlinear operations, performance prediction becomes difficult due to high degree of non-linearity 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.

Inspired from the structure of the human brain and the way it is supposed to be operate, neural networks are parallel computational systems capable of solving number of complex problems in such a diverse areas of as pattern recognition, computer vision, robotics, control and medical diagnosis, to name just few (Haykin, 2003). Neural networks are an effective tool to perform any nonlinear input output mappings. It was the Cybenko (1989) who first proved that, under appropriate conditions, they are able to uniformly approximate any continuous 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 future (Dudul, 2007).

The present paper carries out neural network based modeling of a typical magnetic stirrer 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 magnetic stirrer process.

Estimation of NN model

A typical magnetic stirrer in Reaction Engineering laboratory of College of Engineering and Technology, Akola is used for experimentation. Magnetic stirrer having speed and temperature control facility is manufactured by REMI Equipments Pvt. Ltd., Bombay. The input/output experimental data has been obtained through vigorous experimentation carried out on magnetic stirrer in a laboratory by exciting the system parameters to its complete range. The simulation data constitute 151 samples in each set. In fact, the process is multi-input-single output, where the output variable is concentration

Indian Journal of Science and Technology Vol. 3 No. 4 (Apr. 2010) ISSN: 0974- 6846

of liquid and input variables are stirring speed, temperature and time for mixing. 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). As process exhibits time relationship in the input-output mappings, versatile FTLR NN models is 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 training set and a test set. The entire data is usually randomized first. The training data are 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 are 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. Neurosolutions (version 5.0) is specifically used for obtaining results.

Performance measures

MSE (Mean Square Error): The formula for the mean squared error is:

$$MSE = \frac{\sum_{j=0}^{P}\sum_{i=0}^{N}\left(d_{ij}-y_{ij}\right)^2}{NP} \qquad (1)$$

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, d_{ij} = desired output for exemplar i at PE j.

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

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

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

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 doesn't 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 between a network output x and a desired output d is:

$$r = \frac{\frac{\sum_i\left(x_i-\bar{x}\right)\left(d_i-\bar{d}\right)}{N}}{\sqrt{\frac{\sum_i\left(d_i-\bar{d}\right)^2}{N}}\sqrt{\frac{\sum_i\left(x_i-\bar{x}\right)^2}{N}}}$$

where, $\bar{x} = \frac{1}{N}\sum_{i=1}^{N}x_i$ and $\bar{d} = \frac{1}{N}\sum_{i=1}^{N}d_i$ (3)

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, i.e. the variables are called uncorrelated. Intermediate values describe partial correlations. For an example, a correlation coefficient of 0.88 means that the fit of the model to the data is reasonably good.

Modeling of a magnetic stirrer 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 non-stationary 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 (Dudul, 2007). This, in turn, makes the network reactive to the temporal structure of information bearing signals. For a 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 a 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 (Principe *et al.*, 2000). 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). The focussed 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 are simply linear combiners followed by a static non-linearity.

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 back propagation, provided that a desired signal is available at each time step. This is because the tap delay line at the input layer doesn't have any free parameters, so the only adaptive parameters are in the static feed forward path.

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

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

Where, g (.) is a delay function.

These short-term memory structures can be studied by linear adaptive filter theory if $g(.)$ 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 feed forward memory PE. It can be shown that the defining relationship for the generalized feed forward memory PE is (Principe *et al.*, 2000).

$$g_k(n) = g(n) * g_{k-1}(n) \qquad k \ge 1 \qquad (5)$$

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 feed forward memory PE, the tap signals $y_k(n)$ become

$$y_k(n) = g(n) * y_{k-1}(n) \qquad k \ge 1 \qquad (6)$$

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

$$y_0(n) = g_0(n) * x(n) \qquad (7)$$

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

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

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

$$g(n) = \delta(n-1) \qquad (9)$$

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 feed forward memory PE where

$$g(n) = \mu(1-\mu)^n \qquad n \ge 1 \qquad (10)$$

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

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

where (:) is a binomial coefficient defined by $\binom{n}{p} = \frac{n(n-1)...(n-p+1)}{p!}$ for integer values of *n* and *p*. the overall impulse response $g_p(n)$ for varying *p* represents a discrete version of the integrand of the gamma function (Principe *et al.*, 1992), 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 low pass $(0 < \mu < 1)$ or high pass $(1 < \mu < 2)$ transfer functions. The high pass transfer function creates an extra ability to model fast-moving signals by alternating the signs of the

Indian Journal of Science and Technology Vol. 3 No. 4 (Apr. 2010) ISSN: 0974- 6846

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.

Table 1. Parameters of FTLR NN

Parameters	Hidden layer # 1	Output layer
Neurons	33	1
Transfer function	tan h	tan h
Learning rule	Momentum	Momentum
Step size	0.1	0.1
Momentum	0.7	0.7

No. of Exemplars: Training = 60; Testing = 90; Max epoh =1000, Focused, Gamma axon, Norm L1

Simulation of FTLR NN model

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 layer and number of neurons in each hidden layer on the basis of performance measures. Training and testing percentage 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, effect 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 Table 1. Fig.1 demonstrate the regression ability of the FTLR NN model on testing data set whose performance measures are MSE = 0.000127, NMSE = 0.0452 & r = 0.992. Here from the close visual inspection of the figure it is clear that FTLR NN model very well learn nonlinearity in the system and desired output closely follows actual output of the system. Also,

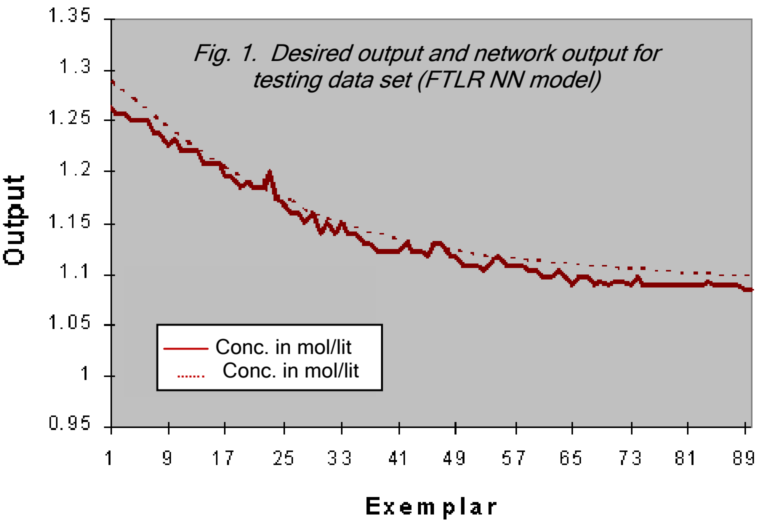

Fig. 1. Desired output and network output for testing data set (FTLR NN model)

Table 2. Training report (optimal FTLR NN model)

Best network	*Training*
Run #	3
Epoch #	179
Minimum MSE	0.093637995
Final MSE	0.131667001

performance measures clearly indicates capability of FTLR NN model for system identification of magnetic stirrer process.

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 graphed in Fig. 2. Its 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 an encouraging increase. This enthusiastic trend continues until a threshold value of gamma is reached, beyond which MSE and NMSE begin increasing along with an

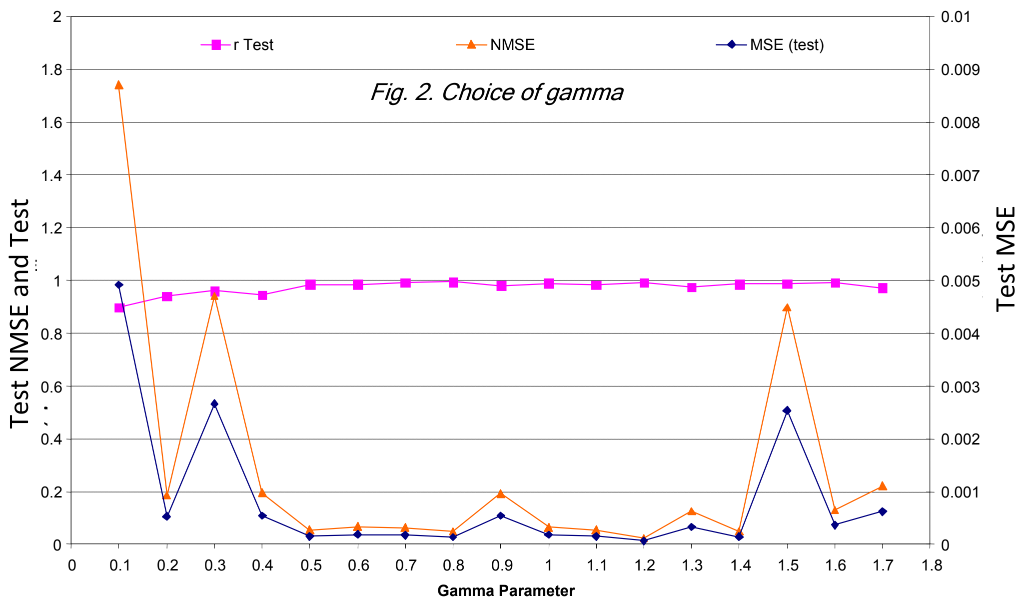

Fig. 2. Choice of gamma

little decrease in the correlation coefficient. From close observation of Fig. 2, the threshold is noticed as 1.2 for the best identification performance of the dynamic NN model. Table 2 displays the training report of the best chosen focused time lagged recurrent neural network model with gamma memory filter.

Fig.3 demonstrate the regression ability of the FTLR NN model with gamma memory set at 1.2 on testing data set whose performance measures are MSE = 6.3279E-05, NMSE= 0.02239 & r = 0.9933. Here desired output is compared with actual output produced by neural network model with gamma memory and from close visual inspection, it is noticed that this model (with gamma memory) elegantly learns the rich nonlinear dynamics of the system and performance measures also shows significant improvement.

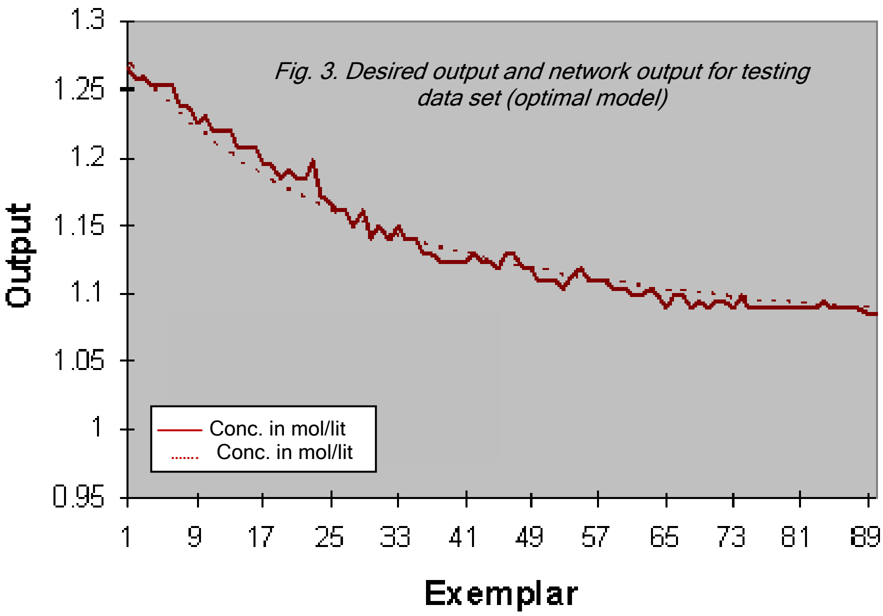

Fig. 3. Desired output and network output for testing data set (optimal model)

Conclusion

Previously it is seen that FTLR NN model is capable of learning nonlinear dynamics of magnetic stirrer process. In this paper, it is demonstrated that FTLR NN model with gamma memory filter very closely follows desired output of magnetic stirrer process for the testing instances. From the results presented, it is seen that FTLR NN model with gamma memory filter at 1.2 has an edge over former model when performance measures and visual inspection of regression characteristics are taken into consideration. It is thus concluded that for identification of magnetic stirrer process using neural networks, FTLR NN model with gamma memory filter can be used to learn underlying highly nonlinear dynamics of the system is the major contribution of this paper.

References

1. Cybenko G (1989) Approximations by superposition of a sigmoidal functions. *J. Math. Control Signals Syst*. 2, 303-314.
2. Dudul SV (2007) Identification of a liquid saturated steam heat exchanger using focused time lagged recurrent neural network model. *IETE J.* 53, 69-82.
3. Haykin S (2003) Neural Network: A Comprehensive Foundation. 2nd Ed., Pearson Education, India, ISBN: 81-7808-300-0, p443.
4. Principe JC, Euliano NR and Lefebvre C (2000) Neural and Adaptive Systems-Fundamental Through Simulation. 1st Edn., John Wiley & Sons, NY.