NONLINEAR SYSTEM IDENTIFICATION USING TUNED GENETIC ALGORITHM FOR MODIFIED ELMAN NEURAL NETWORK

V.Kavidha
Department of Computer Science and Engineering, National Engineering College, Kovilpatti, TamilNadu,India
kavidhacse@nec.edu.in

Abstract: *Detection is the process of modeling a system based on its inputs and outputs. Detection techniques for nonlinear systems are based on linear approximations of the system and such approximations perform well for a large range of process. But complex systems need complicated identification techniques. Neural networks have been shown to outperform traditional identification techniques on complex problems. Neural networks have unique pattern recognition characteristics which enable them to identify non linear systems. Tuned Genetic algorithms (TGA) have recently been applied to the design of neural networks. Based on the principles of natural evolution, TGA leads a more directed search than a random procedure, while still exploring, the entire search space. This paper describes technique for optimizing Modified Elman Neural Networks (MENN) using TGA for the identification and control of non linear systems. Cart pole system is used as the standard for this study. MENN is optimized using TGA are applied to cart pole system. It can be safely concluded that training and optimizing MENN using TGA yield substantially robust designs. TGA Elman network will definitely outperform the one using MENN trained by Back Propagation algorithm.*

Keywords: *System Identification, Tuned Genetic Algorithm, Optimization, Elman network, Mean Square Error.*

1.Introduction

System identification is the procedure that develops models dynamic systems based on their input output signals[1]. Identification of nonlinear system can be done through a variety of methods[2]. Nonlinear models are necessary where the process to be identified cannot be approximated well by linear methods [3]. Genetic algorithms are then used tooptimize the training of the neural network [4]. The nonlinear system to be identified consists of an inverted pendulum on a cart is described [5]. Using the SIMULINK model which takes the force as the input and predicts the pendulum angle, a number of Autoregressive exogenous (ARX) models were tried [6]. Neural networks how impersonate a very simplified version of the brain in two aspects is applied where the neurons are the processing elements of the network [7]. Neural networks trained to minimize the squared error between their output and the plant input, identify an inverse transfer function [8]. A continous function can be randomly well approximated by a feed forward network with only one hidden layer [9]. The main properties that make neural networks excellent identification tools from experience [10].

Context units are used to memorize previous activations of the hidden units and act as one step time delays [11]. Number of Elman networks were setup using the newelm function in MATLAB which allows specifying the number of layers [12]. The properties of recurrent networks were described which make them more suitable system identification and to represent dynamical behavior, avoiding the need of temporal data window [13]. For dynamic system identification Elman and Jordon networks are more popular [14].Genetic Algorithm do not need auxiliary information about derivatives for optimization [15]. Genetic Algorithm search starts with a population of individual solutions represented in the form of encoded chromosomes [16].Weights and biases encoding strategy is used and real coded chromosomes are used to manage the large number of weights and biases [17]. Neural networks, which are inherently nonlinear and hence more suited to nonlinear identification are applied to the problem [18]. Common techniques involve a linear approximation of the system and have been found to give satisfactory results for a wide variety of applications [19]. Parameters of the model are adjusted until its output is similar to the output of the real system [20].

2. Non Linear Cart Pole System

The nonlinear system to be identified consists of an inverted pendulum on a cart as shown in Fig.1. This is a extremely unstable system. The system is composed of a rigid pole and a cart on which the pole is hinged. The pole is hinged to the cart through a pivot joint such that it has only one degree of freedom.

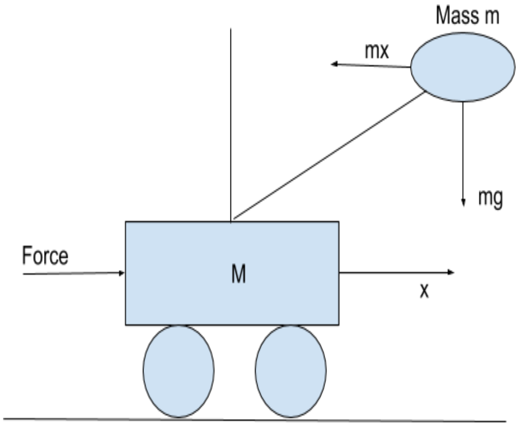

Fig. 1. Cart Pole System

For identification the output that is tracked is the pole angle θ. Output of the controller is the arranged forward or backward movement of the cart at a fixed force F. Nonlinear dynamic equations of the cart pole system are:

$$\ddot{\theta}=\frac{(M+m)g\sin\theta - \cos\theta[F+m/\theta^{2}\sin\theta]}{4/3(M+m)-m/\cos^{2}\theta} \quad (1)$$

$$\ddot{y} = \frac{F+ m/[\,\theta^{2}\sin\theta - \ddot{\theta}\cos\theta]}{(M+m)} \quad (2)$$

Parameters of the cart pole system by assuming the following values in the SIMULINK model is shown in Fig.2.

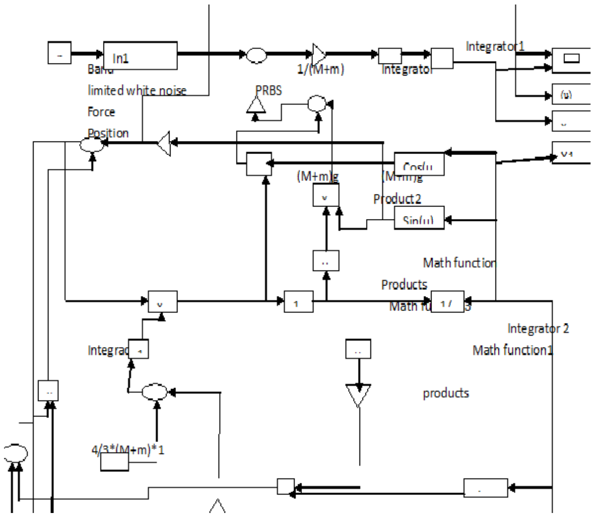

Fig.2. SIMULINK model for the nonlinear cart pole system

Mass of cart(M) = 1 kg (3)
Mass of pole(m) = 0.1kg (4)
Pole length (l) = 0.5m (5)
Inertia of the pendulum (l) = 0.006kg-m^2(6)
Gravitational acceleration (g) =9.8m/s^2 (7)

The pseudo random binary source is shown in Fig.3 used as a controlled input to the nonlinear cart pole system. The input is either +10 or -10, which provides random excitation, forward and backward movement at the cart. The data is determinedly exciting, so that the training set has to be delegate of the entire class of inputs that may excite the system. Using the SIMULINK model which takes the force as the input and predicts the pendulum angle, a number of ARX models were tried. The input output training data for predicting the linear models is as shown in Fig.4.

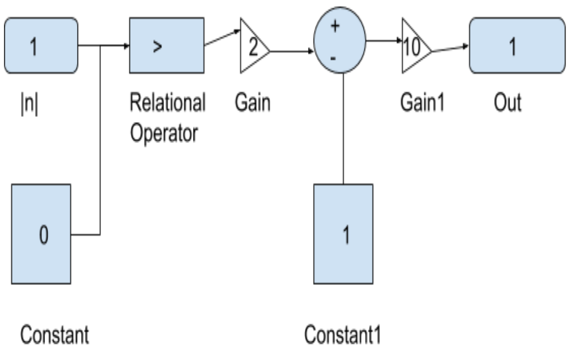

Fig.3. Pseudo random binary source created from white noise

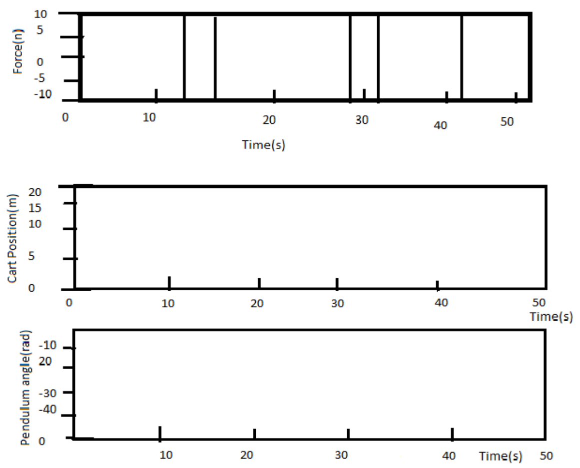

Fig.4. Input-Output training data used for nonlinear model

Using these input output patterns, ARX models of various orders were generated. The predicted output was plotted against the actual output using the training data as Validation data shown in Fig. 5, and other validation data shown in Fig. 6. Many models were generated, however it was felt that to achieve better system approximation it is necessary to move to nonlinear models. Neural networks can be used to approximate any nonlinear function.

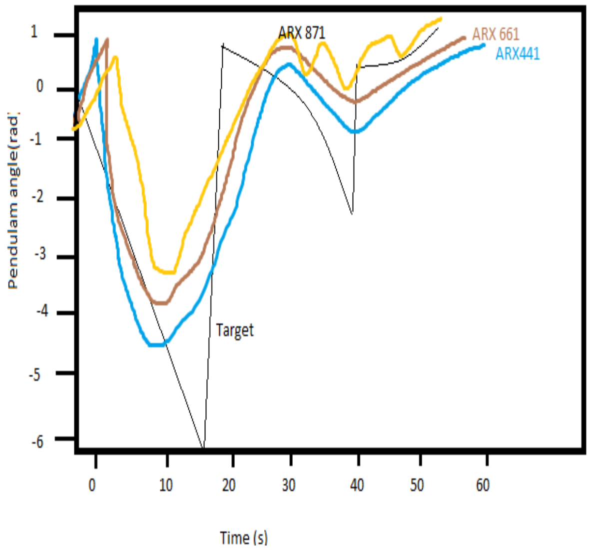

Fig.5. ARX performance using training data

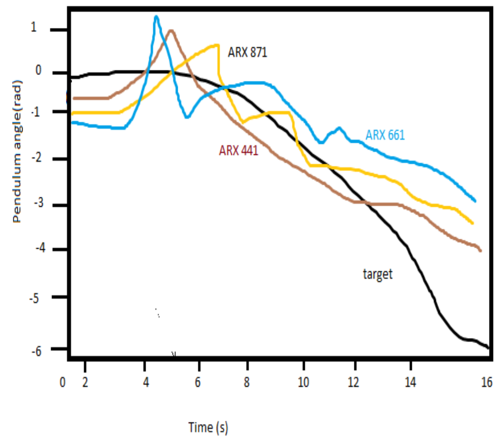

Fig.6. ARX performance using unseen validation data

3. Modified Elman Network Based Identification

A Neural network operates as a parallel distributed processor capable of extracting patterns or information's from experimental data, an applying the information so learned previously unseen problems. It outlines how neural networks imitate a very basic version of the brain in two aspects. Neurons are the processing elements of the interconnections between them determine the structure of a network. The layout of the neurons in space and the interconnections between them determine the structure of the network. Common structures can be modified to suit a particular application, or alternatively a entirely new structure can be designed. A learning algorithm decides how the weights are adjusted to achieve the desired activities of the network.

Neural networks trained to diminish the squared error between their output and the plant input, identify an inverse transfer function. The other method of classification involves designing the plant output to the network output. In this type identifier the system represents the forward transfer function of the plant. It proved that a continuous function can arbitrarily glowing approximated by a feed forward network with only one hidden layer. The main properties that make neural networks admirable identification tools include the ability to learn from experience. In addition, neural networks can simplify for untrained inputs, based on learning it will be able to forecast the output for a previously unseen input. They also have the capability to approximate to an arbitrary accuracy, given efficient number of neurons. It describes the properties of recurrent networks which make them more suitable for system identification and to represent dynamical behavior, avoiding the need of the temporal data window.

For dynamic system identification Elman Network is more popular. The Standard MENN shown in Fig.7 consists of input/output and context units. The input/output units behave similarly as in feed forward networks, the input units serve as buffers to distribute the signals without processing them, and, output units linearly sum the inputs form the preceding layer and have a linear activation function. The hidden units can have linear or nonlinear activations of the hidden units and act as one step time delays. At given time k, the inputs to the network consist of the current inputs u(k) and the previous activations of the hidden units x(k-1). These inputs are propagated forward to produce the outputs. Standard back propagation algorithm can then be used to train the network. Activations of the hidden units at time k are sent back through the recurrent links to the context units and saved there for the next training step at time (k+1). External inputs to the network are represented by U(k+1) and the network outputs by Y(k). Activations of the hidden units are represented by $X^c(k)$. MENN is able to model an nth order dynamic system if it can be trained do do so. For that 2n input units would be needed if tapped delay units is one, or n+1 if the context units are regarded as input units, hence it is significantly smaller in structure than a feed forward networks when n is large.

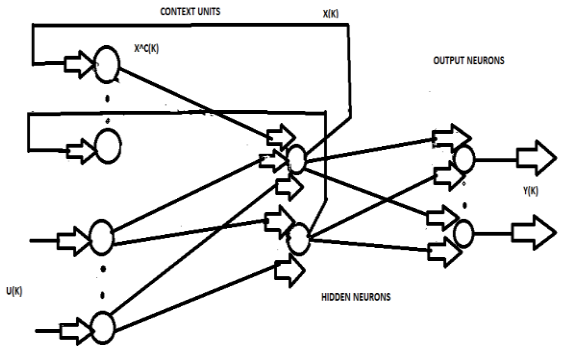

Fig. 7. Modified Elman Neural Network

4. Simulation of Modified Elman Neural Network

Using the MATLAB's neural networks toolbox a feed forward network with one layer consisting of 30 hidden neurons was set up with the command Net =newff([-1010],[30 1], {'tansig' 'purelin'},'traingdm). To increase the number of training patterns for the network a number of input output training patterns for the network a number of input output sequences were concatenated together. In addition, no upper or lower limits were placed on the pendulum angle. The training patterns were presented for 500 epochs. Fig.8 shows the predictions of the feed forward network when tested with the training data. It is observed that feed forward network completely fails to predict the actual output. Number of MENN were set up using the newwlm function in MATLAB which allows specifying the number of layers, the number of neurons in each layer and the activation function to be used as well as the learning rate, the number of epochs, and a number of other training parameters. Weight updates are done using the gradient descent back propagation rule with momentum. To incorporate the nonlinear nature of the problem in the network design neurons of hidden layer are given tan sigmoid activation functions. Neurons of the output layer are used with linear activation functions. Hidden layer consists of 10 neurons. Quality of the neural model is tested by calculating the Mean Square Error(MSE) as shown in Fig.9., which gives a good indication of its accuracy. Predictions of the networks before and after training for 500 epochs are presented in Fig.10. Even with this relatively small number of training epochs the network performance when tested using the training data is much improved over all in the previous networks. The MSE values computed for MENN by changing the number of hidden neurons and number of layers is presented in Table 1.

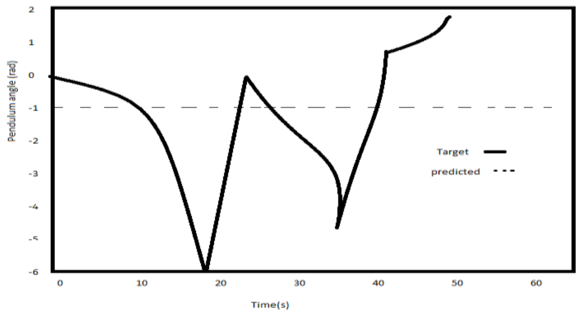

Fig.8. Feed Forward network with 30 hidden neurons

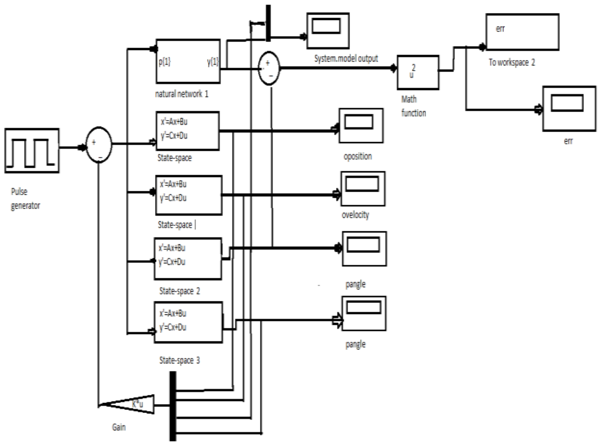

Fig. 9. SIMULINK model for quality testing

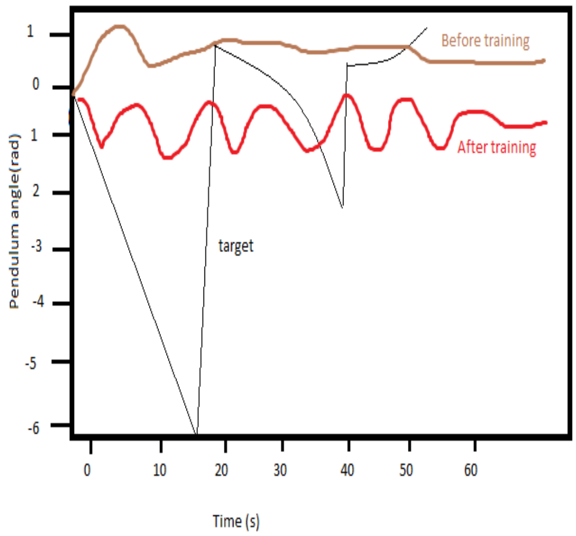

Fig.10. Modified Elman Neural Network with 1 hidden layer, 10 hidden neurons

Number of hidden neurons(HN)	Modified Elman Neural Network trained by Back Propagation algorithm	
	1 hidden layer	2 hidden layers
	MSE	MSE
15	2.9281	2.2048
20	2.7357	1.6440
30	1.9934	0.2723
40	1.2403	…….
50	0.8570	0.1508
60	0.7170	…….
70	0.1829	0.0055
80	0.1854	…….
90	0.1307	…….

Table.1. MSE values of an MENN for nonlinear cart pole system.

5.Tuned Genetic Algorithm for MENN Optimization

TGA search starts with a population of individual solutions represented in the form of encoded chromosomes. It make use a number of evolutionary operations. These include selection, reproduction, crossover and mutation and do not need auxiliary information about derivatives etc. for optimization. A technique of training an MENN using TGA is given here. Weights and biases encoding strategy is used and real coded chromosomes are used to manage the large number of weights and biases. For the MENN architecture, parameters are tuned using TGA. Table 2 shows the schematic of the real coded TGA chromosomes including the weights of input to hidden layer (HN*inputs), biases of hidden layers(HN), weights of context layer (HN*HN), weights of hidden to output layer (HN), biases of output layer = single output).

BiasH	BiasO	IW1	IW2	LE	OW
HN	1	HN	HN	HN*HN	HN*1
Biases of hidden layer	Biases of output layer	Input 1 to hidden weights	Input 2 to hidden weights	Context layer weights	Hidden to output weights

Table.2.Schematic of real coded TGA chromosomes

Algorithm for TGA for Modified Elman Neural Network

Step 1: Represent the parameter space as chromosome of fixed length. Choose the population size NIND, crossover probability XOV_RATE, mutation probability MUT_RATE, generation gap GGAP, termination criterion MAXGEN, initialize the generation counter gc=0.

Step 2: Define a fitness function to measure the performance of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction. MSE is found to be the best choice.

Step 3: Randomly generate an initial population of NIND no. of chromosomes.

Step 4: Decode each of the NIND individuals into the network represents.
Step 5: Evaluate every network obtained from Step 4 with the training data. Calculate Error=Target=predicted. Then find MSE. Objective is to minimize MSE.
Step 6: Rank the population of networks according to their fitness.
Step 7: Select a pair of chromosomes for mating the current population. Parent chromosomes are selected with a probability related to their fitness.
Step 8: Create a pair of off spring chromosomes in the new population.
Step 10: If size of new population is less than NIND, go to step 7.
Step 11: Replace the initial population with the new population. Increment generation counter gc.
Step 12: If gc is less that MAXGEN, go to Step 4.

6.Simulation of TGA for Modified Elman Neural Network

Input output pairs were generated from SIMULINK model of cart pole linear system. Objective value used to evaluate the fitness of a particular individual was the MSE. Evaluation of the objective values involved translating each individual into an MENN, and simulating that network to find the MSE between the actual and the predicted output. Because this needs to be done for each individual in the population the run time of the algorithm is long even for a small number of generations. The following TGA and MENN parameters were selected for simulation after no. of trails:
NIND = 50; HN = 15; MAXGEN =50; GGAP =0.98; XOV_RATE = 0.85; MUT_RATE = 0.01;

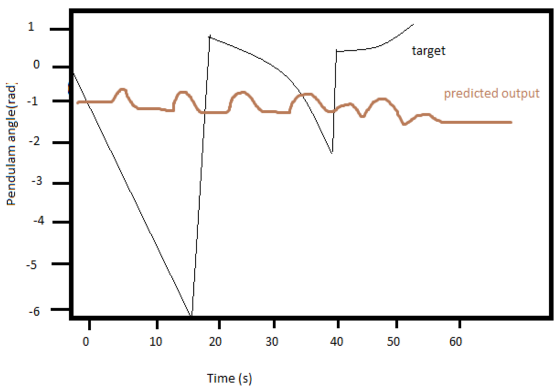

Fig.11. Modified Elman Neural Network trained by TGA for 50 generations

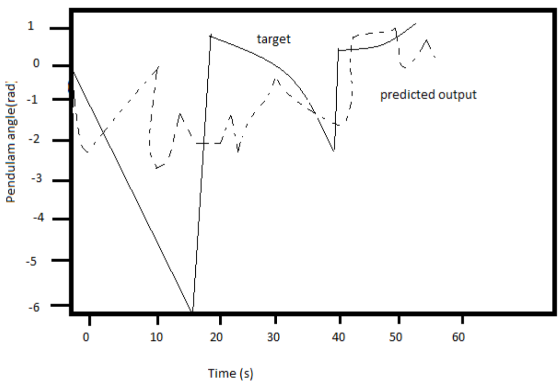

Fig.12. Modified Elman Neural Network trained by TGA for 600 generations

The responses from the system and the MENN networks trained by TGA are presented in Fig.11 and Fig.12. The MSE associated with this is significant. It is concluded that 50 generations is probably insufficient to train the network. Training was run again for 600 generations. The MSE values computed for the MENN trained by back propagation and genetic algorithm are presented in Table 3.

Number of hidden neurons ,HN	Modified Elman Network trained by back propagation algorithm		Modified Elman Network trained by genetic algorithm
	1 hidden layer , MSE	2 hidden layer, MSE	
	2.9281	2.2048	3.3316(50 gen)
			2.2789(600 gen)

Table.3.Comparison of Back propagation and TGA training

7. Conclusion

It can be deduced that use of nonlinear tan sigmoid activation functions in the hidden layers of the neural networks is particularly advantageous. MENN outperforms feed forward networks in the case of nonlinear cart pole system. This is attributed to the dynamic nature of recurrent neural networks, which make use of delay elements. It was found that neural networks could develop excellent fits to the training data, over a period of approximately 1000 epochs. In the traditional back propagation, there is chance of getting local as opposed to global mimic to make sub optimal predictions. TGA greatly reduce the probability of getting stuck in local minima because they test a population of potential solutions in parallel.

TGA algorithms useful to the selection of MENN weights lead to a enhanced selection of network weights than the back propagation rule. It is observed that the TGA for MENN give comparable sometimes better performance for this problem. Considering the fact that the TGA search is less locally concentrating and is more distributed over the investigate space, such results look very promising. Because the results are obtained for deterministic model, it can be safely concluded that for real life nonlinear systems without a deterministic model and non differentiable, irregular and noisy search spaces, the TGA for MENN will definitely outperform the one using calculus based training. The direction of research looks very promising. Presently the only significant drawback of using TGA for MENN is the very long computational time required for such training.

References

1.Luc Andro Gregnire, Handy Fortin Blanchette, Jean Belanger, Kamal Aihaddad, "*A stability and accuracy validation method for multirate digital simulation*", IEEE Transaction on Industrial Informatics, vol 13, no.12,pp. 512-519, April 2017.

2. K.J.Istrom, K.Furuta, "*Brief swinging up pendulum by energy control*", Journal Automatica, vol 6, no.2, p.287-295, Feb 2007.

3. www.mathworks.com

4. Roger Achkar,Mustafa,Elie Bassil, Rayan Fakhno, Marny Khalil, "*Voice identity finder using the back propogation algorithm of an artificial neural network"*, procedia computer science, vo1. 93, pp. 245-252,November 2016.

5. Tim Waegman,Francis wyffels, Benjamin Schrauen, "*Feedback control by online learning an inverse model"*,IEEE Transactions on Neural Networks and learning system, vol 23, no.10, pp.1637-1648,October 2012.

6. Danilo Costarelli, Renato Spigler, "*Approximation results for neural network operators activated by sigmoidal functions"*, Neural Networks Elsevier, vol 44, pp. 101-106,March 2013.

7. Dan Sheng Yu*, "Approximation by neural networks with sigmoidal functions"*, Acta Mathematica Sinica, vol 29, no.10,pp.2013-2026, October 2013.

8. V.Tomin,G.Kurbatsky, N.Denis, V.Alexy, "*Machine learning techniques for power system security assessment"*, Proceeding for International Workshop of Control of Transmission and Distribution of Smart Grids, vol 49, no.27,pp.445-459, August 2016.

9.L.C.Kiong, M.Rajeswari, "*Non linear dynamic system identification and control via constructivism inspired neural network*", Applied soft computing, vol 3, pp.191-200, November 2003.

10. O.De Jesus, J.M.Horn,M.T.Hajan, " *Analysis of recurrent network training and suggestions for improvements*", Proceedings International Conference on Neural networks, August 2002.

11. L.Ljung, " *Identification of Linear and non linear dynamics system"*, Berkley, 2005.

12. D.Schroder, C.Hintz, "*Intelligent modeling, observation and control for nonlinear system*", IEEE Transactions on Neural Networks, vol 2, no.6, pp.122-131,July 2001.

13. A.F. Atiya, A.G. Parlos, "*New Results on Recurrent Network Training: Unifying the Algorithms and Accelerating Convergence*", IEEE Transactions on Neural Networks, vol 11,no.3 pp. 697-709, May2000.

14.Qing Song, " *On the weight convergence of Elman Network*", IEEE Transactions on Neural Networks, vol 21, no.3, pp. 463-480, February 2010.

15. G. L. Plett, "*Adaptive inverse control of linear and nonlinear systems using dynamic neural networks*", IEEE Transactions on Neural Network., vol 14, no. 2, pp. 360-376, March 2003.

16. Matteo Cacciola, Giuseppe Megali, Diego Pellicano, Franscesco carto Morabito, "*Elman Network for characterizing voids in welded strips: a study"*, Journal of Network computing and Applications, vol 21, no.5, pp. 869-875, July 2012.

17. S. Hu, D. Liu, "*On the global output convergence of a class of recurrent neural networks with time-varying inputs*", Journal Elsevier Neural Networks, vol 18, pp. 171-178, November 2005.

18. B. Hu, Z. Zhao, and J. Liang, "*Multi-loop nonlinear internal model controller design under nonlinear dynamic PLS framework using ARX-neural network model*," Journal of Process Control, vol 22, no. 1, pp. 207–217, October 2012.

19. S. F. Ding, W. K. Jia, X. Z. Xu, and C. Y. Su, "*Elman neural network algorithm based on PLS*," Acta Electronica Sinica, vol 38, no. 2A, pp. 71–75, 2010.

20. Y. M. Zhang, "*The Application of Artificial Neural Network in the Forecasting of Wheat Midge*", Northwest A&F University, 2003.

Dr.V.Kavidha received her Bachelor of Engineering degree in Electrical and Electronics, Master of Engineering and PhD in Computer Science and Engineering from Manonmaniam Sundarnar University, TamilNadu, India in 1999, 2004 and 2016 respectively. He is currently working as an Associate professor in Department of Computer Science and Engineering, National Engineering College, Kovilpatti, Tamilnadu, India. She has published papers in reputed and refereed journals and Conferences. Her specific research areas include networks, application of computational intelligence techniques in control systems. She is a member in various professional bodies such as ISTE,IEI. She has got various awards by President of India for publishing best papers in journals.