

NEURAL NETWORK BASED IDENTIFICATION OF MULTIMACHINE POWER SYSTEM

Sheela Tiwari¹, Ram Naresh², Rameshwar Jha³

¹ Department of Instrumentation and Control Engineering, Dr. B R Ambedkar
National Institute of Technology, Jalandhar, Punjab, India

² Department of Electrical Engineering, National Institute of Technology,
Hamirpur, Himachal Pradesh, India

³ IET Bhaddal Technical Campus, Ropar, Punjab, India

ABSTRACT

This paper demonstrates an effective application of artificial neural networks for online identification of a multimachine power system. The paper presents a recurrent neural network as the identifier of the benchmark two area, four machine system. This neural identifier is trained using the static Backpropagation algorithm. The trained neural identifier is then tested using datasets generated by simulating the system under consideration at different operating points and a different loading condition. The test results clearly establish a satisfactory performance of the trained neural identifier in identification of the power system considered.

KEYWORDS

System Identification, Neural Networks, Backpropagation (BP)

1. INTRODUCTION

With electric power consumption set to increase many fold worldwide, there is an imminent need of augmenting the power carrying capacity of the existing power grid. Continuous monitoring and intelligent control of the grid activities is becoming important to improve the reliability, security and efficiency of the electric power system. However, with increasing complexities in the modern power grid, it is becoming difficult and time consuming to generate acceptable approximate mathematical models of the system and the uncertainties involved therein, required by the conventional controllers. Artificial neural networks (ANNs) have been known to have the capability to learn the complex approximate relationships between the inputs and the outputs of the system and are not restricted by the size and complexity of the system [1]. Since these approximate relationships are learnt on the basis actual inputs and outputs, they are generally more accurate as compared to the relationships based on assumptions. This makes ANNs a promising and attractive tool for modern power system applications. ANNs have been proposed for detection of power system harmonics [2-4], fault section estimation [5-6], fault diagnostics of power plant [7] and in protection strategies [8-9]. Applications of ANNs in reactive power transfer allocation [10] and ATC estimation [11] have been reported. Stability issues like damping of oscillations [12-14], prediction of loadability margins [15] and voltage contingency screening [16] have been successfully addressed by ANN based solutions. ANNs have the potential of application for real-time control as reported [17]. In today's time, accurate system identification is of great impor-

tance in power system operation and control. The capability of the ANNs to approximate non linear functions sufficiently accurately makes them a suitable choice for use in identification of non linear systems. Multilayer feedforward artificial neural networks using Backpropagation algorithm for training have been proposed for successful online model identification of synchronous generator [18] and a UPFC equipped single machine infinite bus system [19]. A neural network based estimation unit has been proposed to estimate in real time, the parameters for an interfacing scheme for grid-connected inverters and simultaneously estimating the grid voltage [20]. The authors have employed neural network for system identification for predictive control of a multimachine power system operating under widely varying operating conditions and subjected to transient conditions [21].

The work undertaken proposes to use a recurrent neural network for online identification of a multimachine power system. This work aims to investigate the performance of such a neural network in online identification of a multimachine power system. The testing performances of the trained neural identifier are also investigated to establish the accuracy of the trained identifier.

2. SYSTEM DESCRIPTION

The system under consideration for the undertaken work is a two area system with active power flowing from Area 1 to Area 2 [22]. In spite of the small size of the system, its behavior mimics the behavior of a large power system in actual operation. Each area comprises of two 900 MVA machines and the two areas are connected by a 220 kV double circuit line of length 220 km. The load voltage profile is improved by installing additional 187 MVar capacitors in each area. The system under study is equipped with PSS and has a UPFC installed between bus 11 and bus 12 with bus 11 common to the shunt and series converters and the other side of the series converter connected to bus 12 as shown in Figure 1.

The effective utilization of the UPFC in the system requires implementation of various control schemes, many of which require the system to be identified. In this work, the active power at bus 12, P_{12} corresponding to a specific value of the quadrature component, V_q of the series voltage injected by the UPFC is to be predicted under various operating conditions. This next step value of P_{12} is predicted using the values of V_q and P_{12} at some preceding instants.

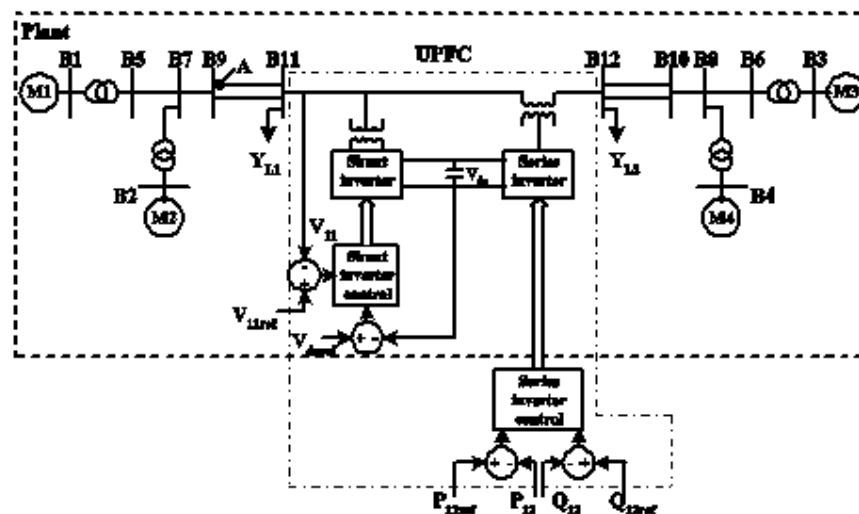


Fig.1. 2-Area system equipped with UPFC

3. DESIGN OF THE NEURAL IDENTIFIER

Recurrent neural networks have the capability to predict the future values based on the values at the preceding instants. The nonlinear autoregressive network with exogenous (independent) inputs i.e. NARX, is a recurrent dynamic network defined by

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)) \quad (1)$$

Where $y(k)$ and $u(k)$ are the outputs and inputs at the k th instant and n_y, n_u are the number of time steps for which the current output is regressed on the output and input respectively. A diagram showing the implementation of the NARX model using a feedforward neural network to approximate the function f in (1) is given in Figure 2.

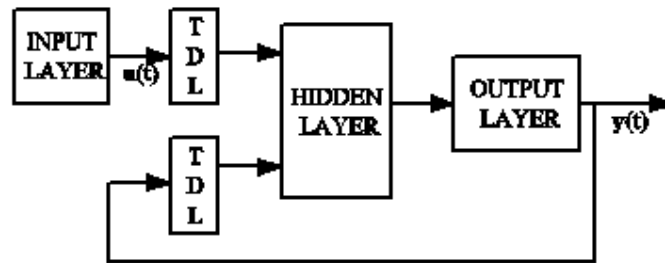


Fig.2. Implementation of NARX model

A neural identifier that predicts the next step value of P_{12} on the basis of the values of V_q and P_{12} at four preceding instants is proposed. As the objective clearly requires that the output of the neural network should depend on the current input as well as on current and/or previous inputs and outputs, the NARX model is used. A two layer neural network with sigmoidal hidden layer neurons and linear output layer neurons can identify any system with any degree of accuracy, subject to the availability of sufficient number of hidden neurons [23]. Therefore, the NARX model is implemented using the two layer feedforward neural network. Since the true value of the output P_{12} for the preceding instants are available, the series parallel architecture is used. As the values of V_q and P_{12} at four preceding instants are to be used, the total number of inputs to the neural network is eight as shown in Figure 3. The number of sigmoidal neurons in the hidden layer has been fixed at thirteen using trial and error approach and the output layer has one linear neuron.

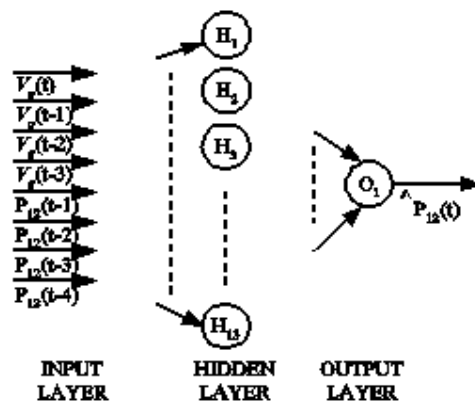


Fig.3. Architecture of the proposed neural identifier

The system under consideration is modeled and simulated using MATLAB / SIMULINK to generate data for training and testing the proposed neural identifier. The operation of the system is simulated by applying V_q restricted within the range +0.1 pu and -0.1 pu (restricting the quadrature component of the series injected voltage to 10% of the nominal line-to-ground voltage) and sampling the input, V_q and output, P_{12} at the rate of 32 samples per second. The neural network is presented with the following inputs

$$\mathbf{V}_q = [V_q(t) V_q(t-1) V_q(t-2) V_q(t-3)]^T \quad (2)$$

And

$$\mathbf{P}_{12} = [P_{12}(t-1) P_{12}(t-2) P_{12}(t-3) P_{12}(t-4)]^T \quad (3)$$

for predicting $P_{12}(t)$. For linear input neurons, the output of the input neurons is same as the input given by

$$\mathbf{a}_{8 \times 1}^0 = [\mathbf{V}_q \ \mathbf{P}_{12}]^T \quad (4)$$

The output of the hidden layer (layer 1), consisting of 13 sigmoidal neurons is given by

$$\mathbf{a}_{13 \times 1}^1 = \tan \operatorname{sig}(\mathbf{W}_{13 \times 8}^1 \mathbf{a}_{8 \times 1}^0 + \mathbf{b}_{13 \times 1}^1) \quad (5)$$

where, $\mathbf{W}_{13 \times 8}^1$ is the weight matrix between input and hidden layers and $\mathbf{b}_{13 \times 1}^1$ is the bias to the hidden layer neurons. Similarly, the output of the output layer (layer 2), comprising of one linear neuron is given by

$$\hat{P}_{12} = \operatorname{purelin}(\mathbf{W}_{1 \times 13}^2 \mathbf{a}_{13 \times 1}^1 + b^2) \quad (6)$$

where, $\mathbf{W}_{1 \times 13}^2$ is the weight matrix between hidden and output and layers and b^2 is the bias to the output layer neurons.

4. TRAINING OF THE NEURAL IDENTIFIER

Identification requires setting up a suitably parameterized identification model and adjustment of these parameters of the model to optimize a performance function based on the error between the outputs from the plant and the identification model [24]. It is assumed that the weight matrices of the neural network proposed as the identifier exists, for which, both plant and the identifier have the same output for any specified inputs, for the same initial conditions [24].

The system under consideration is simulated for different operating conditions ranging over a wide range of steady state active power flow level between the two areas to generate data for training. The training data set consists of 1288 data points spread over a wide range of operation. This training dataset is employed in training the proposed neural identifier offline through simulation to make it learn the forward dynamics of the plant. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The performance function used for the neural identifier under consideration is the mean square error, mse, given by

$$V = \frac{1}{N} \sum_{q=1}^N (P_{12_q} - \hat{P}_{12_q})^2 = \frac{1}{N} \sum_{q=1}^N (e_q)^2 \quad (7)$$

where, N is the size of training dataset, P_{12} and \hat{P}_{12} are the target and predicted value of the output of the neural network when the q^{th} input is presented and e_q is the error (difference between the target and predicted value) for the q^{th} input. The performance index V in (7) is a function of weights and biases, $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ and can be given by

$$V(x) = \frac{1}{N} \sum_{q=1}^N e_q^2(x) \quad (8)$$

The performance of the neural network can be improved by modifying \mathbf{x} till the desired level of the performance index, $V(\mathbf{x})$ is achieved. This is achieved by minimizing $V(\mathbf{x})$ with respect to \mathbf{x} and the gradient required for this is given by

$$\nabla V(x) = \mathbf{J}^T(\mathbf{x}) \mathbf{e}(\mathbf{x}) \quad (9)$$

where, $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix given by

$$J(x) = \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \dots & \frac{\partial e_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_N(x)}{\partial x_1} & \dots & \frac{\partial e_N(x)}{\partial x_n} \end{bmatrix} \quad (10)$$

and $\mathbf{e}(\mathbf{x})$ is the error for all the inputs.

The gradient in (9) is determined using backpropagation, which involves performing computations backward through the network. This gradient is then used by different algorithms to update the weights of the network. These algorithms differ in the way they use the gradient to update the weights of the network and are known as the variants of the Backpropagation algorithm. This work compares the performance of the basic implementation of the Backpropagation algorithm i.e. Gradient descent algorithm with the Levenberg-Marquardt algorithm. A brief overview of the different algorithms considered in this work is given under:

4.1 Gradient Descent Algorithm

The network weights and biases, \mathbf{x} is modified in a direction that reduces the performance function in (8) most rapidly i.e. the negative of the gradient of the performance function [25]. The updated weights and biases in this algorithm are given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla V_k \quad (11)$$

Where, \mathbf{x}_k is the vector of the current weights and biases, ∇V_k is the current gradient of the performance function and α_k is the learning rate.

4.2 Levenberg-Marquardt Algorithm

Since the performance index in (8) is sum of squares of non linear function, the numerical optimization techniques for non linear least squares can be used to minimize this cost function. The Levenberg-Marquardt algorithm (LM), which is an approximation to the Newton's method is said to be more efficient in comparison to other methods for convergence of the Backpropagation algorithm for training a moderate-sized feedforward neural network [26]. As the cost function is a

sum of squares of non linear function, the Hessian matrix required for updating the weights and biases need not be calculated and can be approximated as

$$\mathbf{H} = \mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) \quad (12)$$

The updated weights and biases are given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + \mu\mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x})\mathbf{e}(\mathbf{x}) \quad (13)$$

where, μ is a scalar and \mathbf{I} is the identity matrix.

This work compares the performance of the basic implementation of the Backpropagation algorithm i.e. Gradient descent algorithm with the LM algorithm.

5. SIMULATION RESULTS AND DISCUSSIONS

5.1 Training

The neural network proposed in section 3 was trained using the training set and the training algorithms described in section 4. A Pentium (R) Dual-Core CPU T4400 @2.20 GHz was used to train the proposed neural identifier. The proposed neural identifier was trained 50 times each with the two training algorithms, with random initial weights taken for each trial to rule out the weight sensitivity of the performance of the two training algorithms. The network was trained in each case till the value of the performance index in (8) was 0.0001 or less. The training trials established the inability of the Gradient Descent algorithm to converge for the required value of the performance index but the LM algorithm converges successfully during each of the 50 trials. The average time (from the 50 trials) required for training the network using the Levenberg-Marquardt algorithm using the entire training set consisting of 1288 data points is 5.8675 seconds. The minimum and maximum time required for training is 2.9393 and 9.4187 seconds respectively.

5.2 Testing

The trained neural network was tested on the same CPU. The test datasets consisted of data points not included in the training set. The system under consideration was simulated at two such operating points for which no data point was included in the training set. The operation of the multi-machine power system under consideration at these two operating points was simulated using MATLAB/SIMULINK for a period of 13 seconds each. This period also included a 3-phase short circuit fault at point A at $t=10$ s for a duration of 200 ms with the circuit breakers auto reclosing after 12 cycles. The load on the system was then increased and the operation of the system under the new load condition was then simulated for a period of 45 seconds with the same fault at $t=30$ s with auto reclosing circuit breakers. The data during these three simulations was sampled at the rate of 32 samples per second to form three test sets: Test Set I and Test Set II, corresponding to the two operating points at the initial loading condition and Test Set III, at the higher load operating point. As the Gradient Descent algorithm failed to converge for the desired value of the performance index, the neural network that was trained using the LM algorithm was tested using these three test sets.

Test Set I. Test Set I consists of 417 data points. The first four data points are used to predict the output at the next instant. Therefore, the number of predicted outputs for this test set is 413. A

three-phase short circuit fault is simulated at $t=10$ s which corresponds to the sample point number 321 in the test set. The actual output for the system after autoreclosure of the circuit breaker is available in the sample point number 329 of the test set. The actual values of the output power and the values predicted for the same using the neural identifier during a part of the steady state and transient period are shown in Figure 4 and 5 respectively. The effect of the 3-phase short circuit fault on the system is captured in the sample point 322 of the actual output power as shown in Figure 5. However, the circuit breakers operate and an improved system performance is reflected in subsequent samples. As the neural network has been trained to use the information at four preceding instants to predict the next step output, the effect of decrease in the actual output in sample 322 is reflected immediately in the values predicted by the neural identifier in sample 323. Figure 5 clearly shows that the values predicted by the neural identifier follow the actual values closely in the transient period. The effect of autoreclosure of the circuit breakers on the power level is visible in sample number 329 of the actual output power. The increased value of the actual output power in sample 329 due to the autoreclosure of the circuit breakers is reflected in the value predicted by the neural identifier in the subsequent instant. The average absolute error in the predicted values is determined as a measure to establish the predictive quality of the neural identifier. The value of the average absolute error over the entire set is 0.0103.

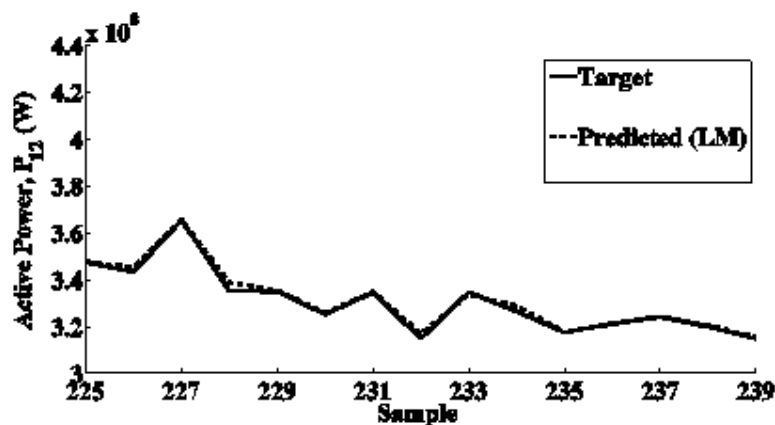


Fig.4. Actual and predicted values of output power during steady state for test set I

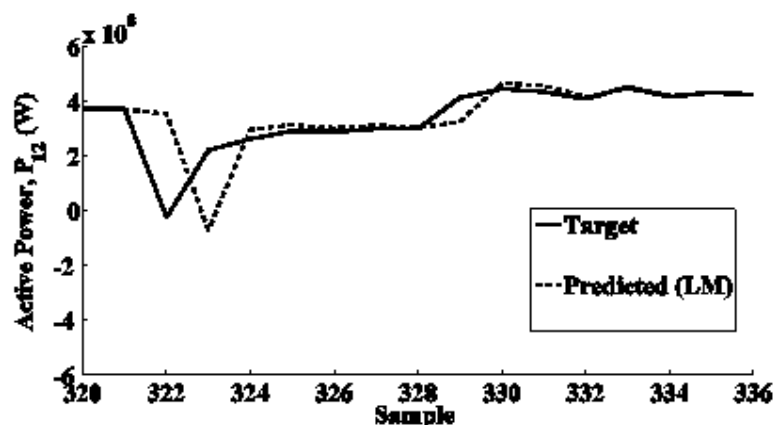


Fig.5. Actual and predicted values of output power during transient period for test set I

Test Set II. Test Set II also consists of 417 data points and 413 predicted outputs. This test set is generated by simulating the system at an operating point corresponding to a different tie line power level and subjected to the same fault as in Test Set I. Figures 6 and 7 show the actual and

predicted output values for a section of steady state and transient period of the system respectively, at the operating point under consideration. It is clear from these figures that the actual power output values and the values predicted using the neural identifier in steady state and transient period are in close proximity even at this operating point. The average absolute error over this test set is 0.0078.

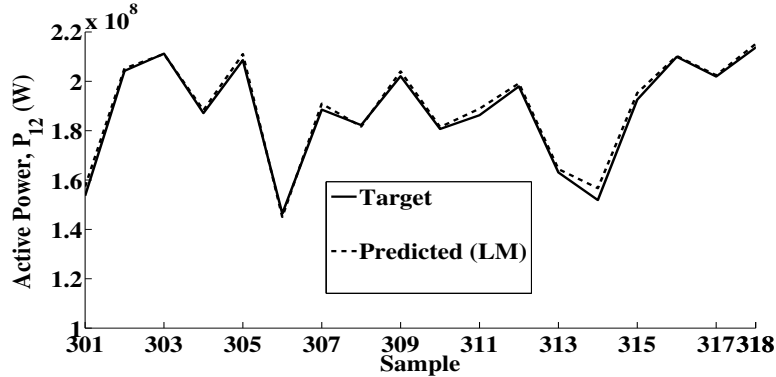


Fig.6. Actual and predicted values of output power during steady state for test set II

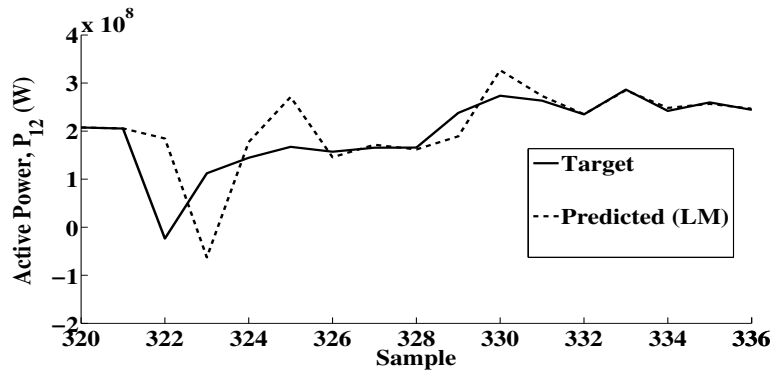


Fig.7. Actual and predicted values of output power during transient period for test set II

Test Set III. Test Set III consists of 1441 data points. The actual and predicted output values for a section of steady state and transient period of the system are shown in figures 8 and 9 respectively. These figures demonstrate the same trend as in Test Set I and II. The value of average absolute error for this test set is 0.0367.

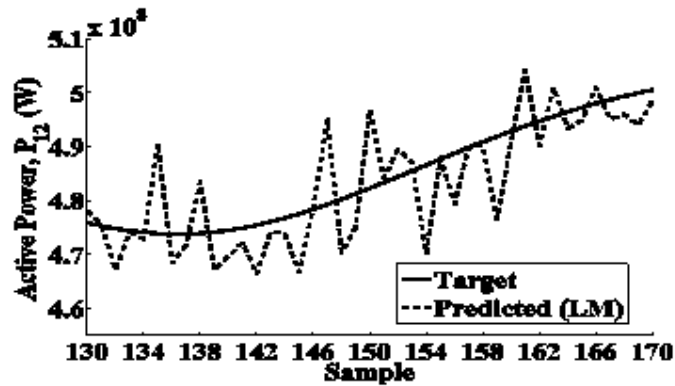


Fig.8. Actual and predicted values of output power during steady state for test set III

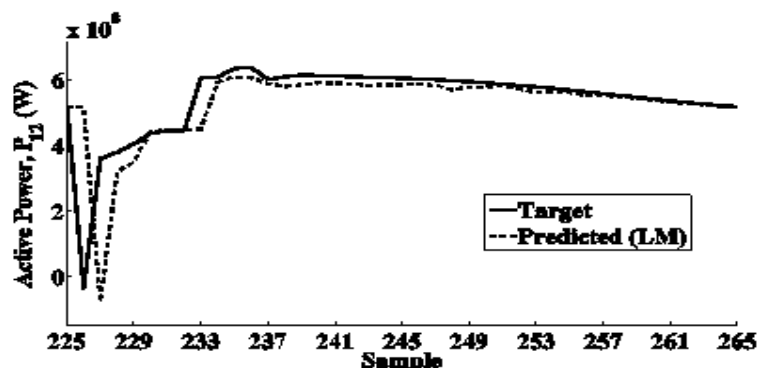


Fig.9. Actual and predicted values of output power during transient period for test set III

6. CONCLUSION

A neural network has been proposed to predict the next step value of the output power on the basis of the values of the control input and output power at preceding time instants. The proposed neural network is trained using the Levenberg-Marquardt algorithm. The trained neural identifier is tested over a range of operating conditions and the test results establish a satisfactory performance of the trained neural identifier over the entire range of testing conditions. The availability of fast computing machines in current times and the accurate predictions reported in this work clearly establish the scope for online application of neural networks for identification of multimachine power systems.

REFERENCES

- [1] Halpin, S.M., Burch, R.F.: Applicability of neural networks to industrial and commercial power systems: a tutorial overview. *IEEE Trans. Industry Applications*. 33(5), 1355-1361 (1997)
- [2] Hsiung, C.L.: Intelligent Neural Network-Based Fast Power System Harmonic Detection. *IEEE Trans. Industrial Electronics*. 54(1), 43-52 (2007)
- [3] Mazumdar, J., Harley, R.G., Lambert, F.C., Venayagamoorthy, G.K.: 'Neural Network Based Method for Predicting Nonlinear Load Harmonics. *IEEE Trans. Power Electronics*. 22(3), 1036-1045 (2007)
- [4] Mazumdar, J., Harley, R.G., Lambert, F.C., Venayagamoorthy, G.K., Page, M.L.: Intelligent Tool for Determining the True Harmonic Current Contribution of a Customer in a Power Distribution Network. *IEEE Trans. Industry Applications*. 44(5), 1477-1485 (2008)
- [5] Cardoso, G., Rolim, J.G., Zurn, H.H.: Application of neural-network modules to electric power system fault section estimation. *IEEE Trans. Power Delivery*. 19(3), 1034- 1041(2004)
- [6] Cardoso, G., Rolim, J.G., Zurn, H.H.: Identifying the Primary Fault Section After Contingencies in Bulk Power Systems. *IEEE Trans. Power Delivery* 23(3), 1335-1342 (2008)
- [7] Liangyu Ma, Yongguang Ma, Lee, K.Y.: An Intelligent Power Plant Fault Diagnostics for Varying Degree of Severity and Loading Conditions. *IEEE Trans. Energy Conversion* 25(2), 546-554 (2010)
- [8] Negnevitsky, M., Pavlovsky, V.: Neural networks approach to online identification of multiple failures of protection systems. *IEEE Trans. Power Delivery* 20(2), 588- 594 (2005)
- [9] Thalassinakis, E.J., Dialynas, E.N., Agoris, D.: Method Combining ANNs and Monte Carlo Simulation for the Selection of the Load Shedding Protection Strategies in Autonomous Power Systems. *IEEE Trans. Power Systems* 21(4), 1574-1582 (2006)
- [10] Mustafa, M.W., Khalid, S.N., Shareef, H., Khairuddin, A.: Reactive power transfer allocation method with the application of artificial neural network. *IET Proc.- Genr. Transm. Distrib.* 2(3), 402-413 (2008)
- [11] Pandey, S.N., Pandey, N.K., Tapaswi, S., Srivastava, L.: Neural Network-Based Approach for ATC Estimation Using Distributed Computing. *IEEE Trans. Power Systems* 25(3), 1291-1300 (2010)

- [12] Changaroon, B., Srivastava, S.C., Thukaram, D.: A neural network based power system stabilizer suitable for on-line training-a practical case study for EGAT system. *IEEE Trans. Energy Conversion* 15(1), 103-109 (2000)
- [13] Chao Lu, Si, J., Xiaorong Xie.: Direct Heuristic Dynamic Programming for Damping Oscillations in a Large Power System. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics* 38(4), 1008-1013 (2008)
- [14] Nguyen, T.T., Gianto, R.: Neural networks for adaptive control coordination of PSSs and FACTS devices in multimachine power system. *IET Proc.- Genr. Transm. Distrib.* 2(3), 355-372 (2008)
- [15] Gu, X., Canizares, C.A.: Fast prediction of loadability margins using neural networks to approximate security boundaries of power systems. *IET Proc.- Genr. Transm. Distrib.* 1(3), 466-475 (2007)
- [16] Jain, T., Srivastava, L., Singh, S.N.: Fast voltage contingency screening using radial basis function neural network. *IEEE Trans. Power Systems* 18(4), 1359- 1366 (2003)
- [17] Ray, S., Venayagamoorthy, G.K.: Real-time implementation of a measurement-based adaptive wide-area control system considering communication delays. *IET Proc.- Genr. Transm. Distrib.* 2(1), 62-70 (2008)
- [18] Jung-Wook Park, Venayagamoorthy, G.K., Harley, R.G.: MLP/RBF neural-networks-based online global model identification of synchronous generator. *IEEE Trans. Industrial Electronics* 52(6), 1685-1695 (2005)
- [19] Venayagamoorthy, G.K., Kalyani, R.P.: Two separate continually online-trained neurocontrollers for a unified power flow controller. *IEEE Trans. Industry Applications* 41(4) 906- 916 (2005)
- [20] Mohamed, Y.A.-R., El-Saadany, E.F.: Adaptive Discrete-Time Grid-Voltage Sensorless Interfacing Scheme for Grid-Connected DG-Inverters Based on Neural-Network Identification and Deadbeat Current Regulation. *IEEE Trans. Power Electronics* 23(1), 308-321 (2008)
- [21] Tiwari, S., Naresh, R., Jha, R.: Neural network predictive control of UPFC for improving transient stability performance of power system. *Appl Soft Comput* 11, 4581-4590 (2011)
- [22] Kundur, P.: *Power System Stability and Control*. Tata McGraw-Hill, New Delhi (1994).
- [23] Hagan, M.T., Demuth, H.B.: Neural networks for control. In: *American Control Conference, 1999*
- [24] Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks* 1(1), 4-27 (1990)
- [25] Hagan, M.T., Demuth, H.B., Beale, M.H.: *Neural Network Design*. MA: PWS Publishing, Boston (1996)
- [26] Hagan, M.T., Menhaj, M.: Training feed-forward networks with the Marquardt algorithm. *IEEE Trans. Neural Networks* 5(6), 989-993 (1994)