

Transient Stability Assessment of a Power System Using Probabilistic Neural Network

Noor Izzri Abdul Wahab and Azah Mohamed
Department of Electrical, Electronic and System Engineering,
Faculty of Engineering, Universiti Kebangsaan Malaysia,
43600 UKM Bangi, Selangor D. E., Malaysia

Abstract: This study presents transient stability assessment of electrical power system using Probabilistic Neural Network (PNN) and principle component analysis. Transient stability of a power system is first determined based on the generator relative rotor angles obtained from time domain simulation outputs. Simulations were carried out on the IEEE 9-bus test system considering three phase faults on the system. The data collected from the time domain simulations are then used as inputs to the PNN in which PNN is used as a classifier to determine whether the power system is stable or unstable. Principle component analysis is applied to extract useful input features to the PNN so that training time of the PNN can be reduced. To verify the effectiveness of the proposed PNN method, it is compared with the multi layer perceptron neural network. Results show that the PNN gives faster and more accurate transient stability assessment compared to the multi layer perceptron neural network in terms of classification results.

Key words: Transient stability assessment, dynamic security assessment, probabilistic neural network

INTRODUCTION

Recent blackouts in the USA, some European and Asian countries have illustrated the importance and need of more frequent and thorough power system stability study. Nowadays, power systems have evolved through continuing growth in interconnection, use of new technologies and controls. Due to the increased operations which may cause power system to be in highly stressed conditions, the need for dynamic security assessment of power systems is arising. Transient Stability Assessment (TSA) is part of dynamic security assessment of power systems which involves the evaluation of the ability of a power system to remain in equilibrium under severe but credible contingencies. These evaluations aim to assess the dynamic behavior of a power system in a fast and accurate way. Methods normally employed to assess TSA are by using time domain simulation, direct and artificial intelligence methods. Time domain simulation method is implemented by solving the state space differential equations of power networks and then determines transient stability. Direct methods such as the transient energy method determine transient stability without solving differential state space equations of power systems. These two methods are

considered most accurate but are time consuming and need heavy computational effort. Presently, the use of Artificial Neural Network (ANN) in TSA has gained a lot of interest among researchers due to its ability to do parallel data processing, high accuracy and fast response.

In transient stability assessment, the Critical Clearing Time (CCT) is a very important parameter in order to maintain the stability of power systems. The CCT is the maximum time duration that a fault may occur in power systems without failure in the system so as to recover to a steady state operation. Some works have been carried out using the feed forward multilayer perceptron (MLP) with back propagation learning algorithm to determine the CCT of power systems^[1,2]. Proposed the use of radial basis function networks to estimate the CCT^[3]. Another method to assess power system transient stability using ANN is by means of classifying the system into either stable or unstable states for several contingencies applied to the system^[2,4]. ANN method based on fuzzy ARTMAP architecture is also used to analyze TSA of a power system^[5]. Ref.^[6] proposed a combined supervised and unsupervised learning for evaluating dynamic security of a power system based on the concept of stability margin. Ref.^[7] used ANN to map the operating

Corresponding Author: Noor Izzri Abdul Wahab, Department of Electrical, Electronic and System Engineering, Faculty of Engineering, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor D. E., Malaysia
Tel: +60 (3) 89464433

condition of a power system based on a transient stability index which provides a measure of stability in power systems.

In this study, a new method for transient stability assessment of power systems is proposed using Probabilistic Neural Network (PNN). The procedures of transient stability assessment using PNN are explained and the performance of the PNN is compared with the MLP so as to verify the effectiveness of the proposed method. Both the MLP and PNN networks were developed using the MATLAB Neural Network Toolbox.

MATHEMATICAL MODEL OF MULTIMACHINE POWER SYSTEM

The differential equations to be solved in power system stability analysis using the time domain simulation method are the nonlinear ordinary equations with known initial values. Using the classical model of machines, the dynamic behavior of an n-generator power system can be described by the following equations:

$$M_i \frac{d^2 \delta_i}{dt^2} = P_{mi} - P_{ei} \quad (1)$$

it is known that,

$$\frac{\delta_i}{dt} = \omega_i \quad (2)$$

By substituting (2) in (1), therefore (1) becomes

$$M_i \frac{\omega_i}{dt} = P_{mi} - P_{ei} \quad (3)$$

where:

- δ_i = Rotor angle of machine i
- ω_i = Rotor speed of machine i
- P_{mi} = Mechanical power of machine i
- P_{ei} = Electrical power of machine i
- M_i = Moment of inertia of machine i

A time domain simulation program can solve these equations through step-by-step integration by producing time response of all state variables.

PROBABILISTIC NEURAL NETWORK THEORY

PNN which is a class of Radial Basis Function (RBF) network is useful for automatic pattern recognition, nonlinear mapping and estimation of probabilities of class membership and likelihood

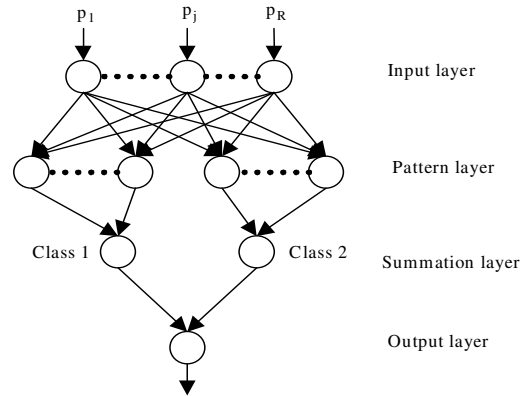


Fig. 1: PNN architecture

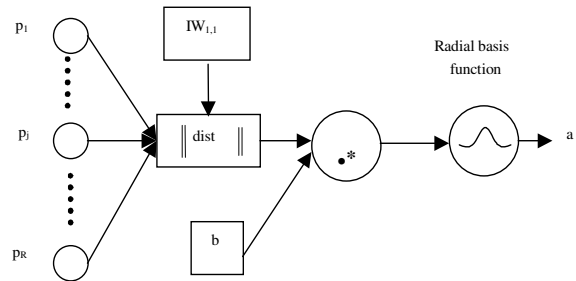


Fig. 2: PNN pattern layer

ratios^[8]. It is a direct continuation of the work on Bayes classifiers^[9] in which it is interpreted as a function that approximates the probability density of the underlying example distribution. The PNN consists of nodes with four layers namely input, pattern, summation and output layers as shown in Fig. 1. The input layer consists of merely distribution units that give similar values to the entire pattern layer.

For this work, RBF is used as the activation function in the pattern layer. Figure 2 shows the pattern layer of the PNN. The $\| \text{dist} \|$ box shown in Fig. 2 subtracts the input weights, $IW_{1,1}$, from the input vector, p and sums the squares of the differences to find the Euclidean distance. The differences indicate how close the input is to the vectors of the training set. These elements are multiplied element by element, with the bias, b , using the dot product ($.*$) function and sent to the radial basis transfer function.

The output a is given as,

$$a = \text{radbas}(\| IW_{1,1} - p \|) \quad (4)$$

where, radbas is the radial basis activation function which can be written in general form as,

$$\text{radbas}(n) = e^{-n^2} \quad (5)$$

The training algorithm used to train the RBF is the orthogonal least squares method which provides a systematic approach to the selection of RBF centers^[10].

The summation layer shown in Fig. 1 simply sums the inputs from the pattern layer which correspond to the category from which the training patterns are selected as either class 1 or class 2. Finally, the output layer of the PNN is a binary neuron that produces the classification decision. As for this work, the classification is either class 1 for stable cases or class 2 for unstable cases. Performance of the developed PNN can be gauged by calculating the error of the actual and desired test data. Firstly, error is defined as,

$$\text{Error}, E_n = |\text{Desired Output}, DO_n - \text{Actual Output}, AO_n| \quad (6)$$

where, n is the test data number. The desired output is the known output data used for testing the neural network. Meanwhile, the Actual Output (AO) is the output obtained from testing on the trained network.

From Eq. 6, the mean error can be obtained as:

$$\text{Percentage Mean Error}, (ME) = \sum_{n=1}^N \frac{E_n}{N} \times 100 \quad (7)$$

where N is the total number of test data.

The percentage classification error is given by,

$$\begin{aligned} &\text{Percentage Classification Error} \\ &= \frac{\text{no. of misclassification of test data}}{N} \times 100 \end{aligned} \quad (8)$$

MATERIALS AND METHODS

In the PNN method used for transient stability assessment, the IEEE 9-bus test system is used for verification of the method. Before the PNN implementation, time domain simulations considering several contingencies were carried out for the purpose of gathering the training data sets. Simulations were done by using the MATLAB-based PSAT software^[11]. Time domain simulation method is chosen to assess the transient stability of a power system because it is the most accurate method compared to the direct method. In PSAT, power flow is used to initialize the states variable before commencing time domain simulation. The differential equations to be solved in transient stability analysis are nonlinear ordinary equations with known initial values. To solve these equations, the

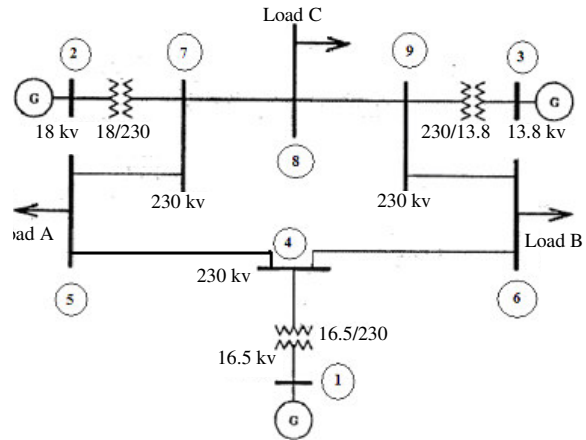


Fig. 3: IEEE 9 bus system

techniques available in PSAT are the Euler and trapezoidal rule techniques. In this work, the trapezoidal technique is used considering the fact that it is widely used for solving electro-mechanical differential algebraic equations^[12]. The type of contingency considered is the three-phase balanced faults created at various locations in the system at any one time. When a three-phase fault occur at any line in the system, a breaker will operate and the respective line will be disconnected at the Fault Clearing Time (FCT) which is set by a user. The FCT is set randomly by considering whether the system is stable or unstable after a fault is cleared. According to ref.^[13], if the relative rotor angles with respect to the slack generator remain stable after a fault is cleared, it implies that $FCT < CCT$ and the power system is said to be stable but if the relative angles go out of step after a fault is cleared, it means $FCT > CCT$ and the system is unstable.

Transient stability simulation on the test system:

Figure 3 shows the IEEE 9-bus system in which the data used for this work is obtained from ref.^[13]. The system consists of three Type-2 synchronous generators with AVR Type-1, six transmission lines, three transformers and three loads.

Figure 4 shows examples of the time domain simulation results illustrating stable and unstable cases. A three phase fault is said to occur at time $t = 1$ second at bus 7. In Fig. 4a, the FCT is set at 1.08 second while in Fig. 4b the FCT is set at 1.25 second. Figure 4a shows that the relative rotor angles of the generators oscillates and the system is said to be stable whereas Fig. 4b shows that the relative rotor angles of the generators go out of step after a fault is cleared and the system becomes unstable. It can be deduced from Fig. 4

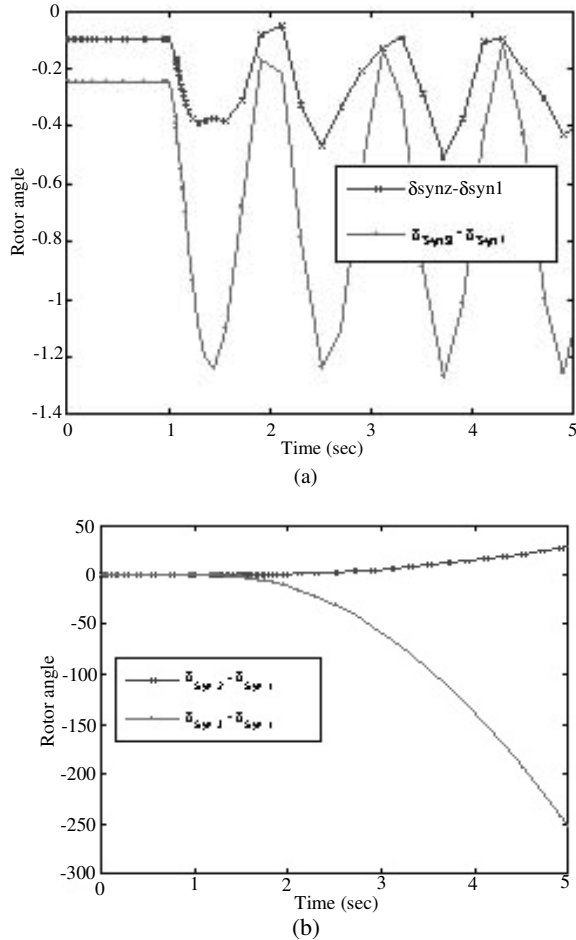


Fig. 4: Relative rotor angle curves of generators for (a): stable and (b): unstable cases

that the FCT setting is an important factor to determine the stability of power systems. If FCT is set at a shorter time than the CCT of the line, the system is stable, otherwise the system will be unstable.

Data preprocessing: The simulation on the system for a fault at each line runs for five seconds at a time step Δt , set at 0.001 sec. The fault is set to occur at one second from the beginning of the simulation. Data for each contingency is recorded in which one steady state data is taken before the fault occurs and 20 sampled data taken for one second duration after the fault occurs. There are 25 contingencies simulated on the system and this gives a size of 25×21 or 525 data collected. The collected data are further analyzed and trimmed down to 468 due to repetitions of data. The one steady state data taken before all faults occur are reduced to one only since the values will be the same

Table 1: Input Features Selected

Name of input features	No. of features
Relative rotor angles (δ_{i-1})	2
Generator speed (ω_i)	3
P_{gen} and Q_{gen}	6
P_{line} and Q_{line}	12
P_{trans} and Q_{trans}	6
Total number of features	29

for all faults. Next, the repetitions are due to the faults that occur on the same line. The FCT of the same line are set at four different times, two for stable cases and two for unstable cases. At the start of the fault, same values of data are recorded for all the four faults. A few milliseconds after the fault, the recorded data differ from each other due to different FCT settings. For the repetitions of data recorded, one data out of the four different FCT settings are kept. These data are denoted as data for stable cases. The data collected are normalized so that they have zero mean and unity variance.

Feature extraction method: The selection of input features is an important factor to be considered in the ANN implementation. The input features selected for this work are relative rotor angles (δ_{i-1}), motor speed (ω_i), generated real and reactive powers (P_{gen} , Q_{gen}), real and reactive power flows on transmission line (P_{line} , Q_{line}) and the transformer powers (P_{trans} , Q_{trans}). Overall there are 29 input features to the ANN. Table 1 shows the breakdown of the input features selected for the neural network.

Out of the 468 data collected from simulations, a quarter of the data which is 117 data are randomly selected for testing and the remaining 351 data are selected for training the neural network. Principle component analysis is used for feature extraction or dimension reduction of the input features. The analysis orthogonalizes the components of the input vectors so that they are uncorrelated with each other, orders the components so that those with largest variation come first and eliminates those that contribute less to the variation in the data set. The standard procedure when using principle components is by first normalizing the input features so that they have zero mean and unity variance. In this work, principle component analysis eliminates those principal components that contribute less than 2% to the total variation in the data set. By applying the analysis, the input features to the networks are reduced from 29 to 11 features only. These reduced features are then used for training the neural networks. For any additional features to be considered, similar PCA procedure needs to be carried out.

TEST RESULTS

In this section, the results obtained from the PNN for transient stability assessment are presented. Initially, the PNN results using 29 and 11 input features are given and discussed. For the purpose of evaluating the effectiveness of the PNN, the results of the multi layer perceptron neural network (MLPNN) are also presented. Finally, a comparison is made between PNN and MLPNN results for transient stability assessment.

PNN results for transient stability assessment: The PNN developed in this work is used for classifying power system transient stability states in which the PNN classifies 1 for stable cases and 2 for unstable cases. The architecture of the PNN is such that it has 29 input neurons, the hidden layer neurons equal the number of input features which is 29 and with a single

output neuron. Table 2 shows the PNN testing results using the 29 (PNN-29) and 11(PNN-11) input features. The shaded cells in the Table denote the misclassification of test data. For the PNN-29, looking into Table 2, it can be deduced that the false alarm rate is 0.86% and the false dismissal rate is 0.86%. False dismissal rate is the rate of unstable cases assigned to the stable cases and the false alarm rate is the rate of stable cases assigned to the unstable cases. Thus, the total error of misclassification and the mean error are both 1.71%.

For the PNN-11 results, the shaded cells in the table denote the misclassification of test data. From the table, it is noted that the false alarm rate is 0.86% and the false dismissal rate is 0.86%. Hence, the total error of misclassification and the mean error are both 1.71%. It can be deduced that, with or without PCA, the performance of the PNN remains the same. As shown

Table 2: PNN Testing Results Using 29 and 11 Input Features

Test data	Desired output	PNN-29	PNN-11	Test data	Desired output	PNN-29	PNN-11	Test data	Desired output	PNN-29	PNN-11
1	1	1	1	40	1	1	1	79	1	1	1
2	1	1	1	41	1	1	1	80	2	1	1
3	1	1	1	42	2	2	2	81	2	2	2
4	1	1	1	43	2	2	2	82	2	2	2
5	1	1	1	44	2	2	2	83	2	2	2
6	2	2	2	45	2	2	2	84	1	1	1
7	2	2	2	46	1	1	1	85	1	1	1
8	2	2	2	47	1	1	1	86	1	1	1
9	2	2	2	48	1	1	1	87	1	1	1
10	1	1	1	49	1	1	1	88	1	1	1
11	1	1	1	50	1	1	1	89	2	2	2
12	1	1	1	51	2	2	2	90	2	2	2
13	1	1	1	52	2	2	2	91	2	2	2
14	1	1	1	53	2	2	2	92	2	2	2
15	2	2	2	54	2	2	2	93	2	2	2
16	2	2	2	55	1	1	1	94	1	1	1
17	2	2	2	56	1	1	1	95	1	1	1
18	2	2	2	57	1	1	1	96	1	1	1
19	1	1	1	58	1	1	1	97	1	1	1
20	1	1	1	59	1	1	1	98	1	1	1
21	1	1	1	60	2	2	2	99	1	1	1
22	1	1	1	61	2	2	2	100	1	1	1
23	1	1	1	62	2	2	2	101	1	1	1
24	2	2	2	63	2	2	2	102	1	1	1
25	2	2	2	64	2	2	2	103	1	1	1
26	2	2	2	65	1	1	1	104	2	2	2
27	2	2	2	66	1	1	1	105	2	2	2
28	1	2	2	67	1	1	1	106	2	2	2
29	1	1	1	68	1	1	1	107	2	2	2
30	1	1	1	69	1	1	1	108	1	1	1
31	1	1	1	70	2	2	2	109	1	1	1
32	1	1	1	71	2	2	2	110	1	1	1
33	2	2	2	72	2	2	2	111	1	1	1
34	2	2	2	73	2	2	2	112	1	1	1
35	2	2	2	74	2	2	2	113	2	2	2
36	2	2	2	75	1	1	1	114	2	2	2
37	1	1	1	76	1	1	1	115	2	2	2
38	1	1	1	77	1	1	1	116	2	2	2
39	1	1	1	78	1	1	1	117	2	2	2

Table 3: MLPNN Results Using 29 and 11 Input Features

Test data	Desired output	MLPNN-29	MLPNN-11	Test data	Desired output	MLPNN-29	MLPNN-11	Test data	Desired output	MLPNN-29	MLPNN-11
1	1	1	1	40	1	1	1	79	1	1	1
2	1	1	1	41	1	1	1	80	2	2	X
3	1	1	1	42	2	2	2	81	2	2	2
4	1	1	1	43	2	2	2	82	2	2	2
5	1	1	1	44	2	2	2	83	2	2	2
6	2	2	2	45	2	2	2	84	1	1	X
7	2	2	2	46	1	1	1	85	1	1	1
8	2	2	2	47	1	1	1	86	1	1	1
9	2	2	2	48	1	X	1	87	1	1	1
10	1	1	1	49	1	1	2	88	1	1	1
11	1	1	1	50	1	X	1	89	2	2	2
12	1	1	1	51	2	2	2	90	2	2	2
13	1	1	1	52	2	2	2	91	2	2	2
14	1	1	1	53	2	2	2	92	2	2	2
15	2	2	2	54	2	2	2	93	2	2	2
16	2	2	2	55	1	1	1	94	1	X	X
17	2	2	2	56	1	1	1	95	1	X	X
18	2	2	2	57	1	1	1	96	1	X	1
19	1	1	1	58	1	1	1	97	1	X	1
20	1	1	1	59	1	1	1	98	1	1	1
21	1	1	1	60	2	X	X	99	1	1	1
22	1	1	1	61	2	X	X	100	1	1	1
23	1	1	1	62	2	X	1	101	1	1	1
24	2	2	2	63	2	2	2	102	1	1	1
25	2	2	2	64	2	2	2	103	1	1	1
26	2	2	2	65	1	1	1	104	2	2	2
27	2	2	2	66	1	1	1	105	2	2	2
28	1	X	X	67	1	1	1	106	2	2	2
29	1	1	X	68	1	1	1	107	2	2	2
30	1	1	1	69	1	1	1	108	1	X	X
31	1	X	X	70	2	X	X	109	1	1	1
32	1	1	1	71	2	2	2	110	1	1	1
33	2	2	2	72	2	2	2	111	1	1	1
34	2	2	2	73	2	2	2	112	1	1	X
35	2	2	2	74	2	2	2	113	2	2	2
36	2	2	2	75	1	1	1	114	2	2	2
37	1	1	1	76	1	1	1	115	2	2	2
38	1	1	1	77	1	1	1	116	2	2	2
39	1	1	1	78	1	1	1	117	2	2	2

in Table 2, the misclassifications of test data occur at the same data set. Thus, it concluded that no information is lost when using PCA to extract the input features to the PNN. The time taken to train the PNN-29 and PNN-11 are 1.32 seconds and 0.76 seconds, respectively. The time taken to train the PNN-11 is less compared to using the complete input features, but however the difference in the training times is for a small test system such as the 9-bus system.

MLPNN results for transient stability assessment:

The testing results of the MLPNN using the complete 29 input features (MLPNN-29) and reduced 11 input features (MLPNN-11) are given in Table 3. The architecture of the MLPNN-29 is such that it has 29 input neurons representing the 29 input features, one hidden layer with 13 neurons of hyperbolic tangent transfer function and a single output neuron. The mean

squared error is used as a goal for training the neural network which is set at 0.03. As for the architecture of the MLPNN-11 is such that it has 11 input features, one hidden layer with 5 neurons of hyperbolic tangent transfer function and a single output neuron. Its error goal is set at 0.04. The training algorithm used for both network is the resilient back propagation algorithm^[14].

The performance goal was met at 41,050 epochs after a training time of 25min 32sec for MLPNN-29. As for the MLPNN-11, the performance goal was met at 11,150 epochs after 3 min 36.2 sec of training time. The calculated mean error is 6% for MLPNN-29 and 6.9% for MLPNN-11. The MLPNN outputs are not crisp 0 or 1 but in the range 0 to 1, where '0' indicates the system is stable and 1 when the system is unstable. So for classification purpose, a decision rule is used such that if the MLPNN output is in the range of 0.9 to 1.1 (±10%), it will indicate that the system is stable

Table 4. Summary of PNN and MLPNN Results

Network	PNN-29	PNN-11	MLPNN-29	MLPNN-11
Input features	29	11	29	11
Mean error	0.0171	0.0171	0.06	0.069
Misclassification	2 (1.71%)	2 (1.71%)	13 (11.1%)	14 (11.9%)

(class 1 or 1) whereas if the MLPNN output is in the range of -0.1 to 0.1 ($\pm 10\%$), it means that the system is unstable (class 2 or 2). For MLPNN output outside this range of values, it is considered as misclassified and denoted as X in the table. Classes 1 and 2 are used in column MLPNN in Table 3 instead of 1 and 0 for stable and unstable classification so that the results conformed to the results obtained from PNN. By using this decision rule the number of misclassified data for MLPNN-29 is 13 out of 117 test data, which is 11%. The number of misclassified data for MLPNN-11 in this case is 14, thus giving a percentage classification error of 11.9%. Comparing these results with the MLPNN-29, the MLPNN-11 is less accurate and this may be due to some information lost when the PCA is applied.

Comparison of neural network results in transient stability assessment: Table 4 summarizes the neural network results in which the columns indicated by MLPNN-11 and PNN-11 are the networks that use the PCA for feature extraction.

From the table, comparing the results of PNN-29 and PNN-11, it is shown that both neural networks give similar performance. The PCA applied on PNN does not affect the performance of the PNN-11 in terms of its mean error and percentage of classification error. In terms of training time, the time for training PNN-11 is shorter compared to training PNN although it is not significant.

Comparing the results of MLPNN-29 and MLPNN-11, it is noted that the mean error and percentage classification of the MLPNN-11 are slightly greater than that of MLPNN-29. This is due to the fact that some information is lost when PCA is applied to the MLPNN network. In terms of training time, the MLPNN-29 takes much longer time (25 min 32 sec) compared to the MLPNN-11 (3 min 36.2 sec).

As for results of PNN and MLPNN, it can be concluded that the performance of PNN is better compared with MLPNN for both with and without the application of PCA. The mean error for both PNN-29 and PNN-11 are 0.017 compared to 0.06 and 0.069 for MLPNN-29 and MLPNN-11, respectively. The percentage classification errors are also less for both PNN-29 and PNN-11 (1.71%) compared to 11.1% and 11.9% for MLPNN-29 and MLPNN-11, respectively. In terms of training time, both PNN-29 and PNN-11 are

significantly shorter than the time taken to train MLPNN-29 and MLPNN-11.

CONCLUSION

The use of PNN has been proposed for transient stability assessment of electrical power system by means of classifying the system into either stable or unstable states for several three phase faults applied to the system. Time domain simulations were first carried out to generate training data for the PNN and to determine transient stability state of a power system by visualizing the generator relative rotor angles. Principle component analysis is also applied to extract the useful input features to the PNN. The PNN is then compared with the MLPNN so as to evaluate its effectiveness in transient stability assessment. Results show that the PNN gives better performance than the MLPNN in terms of transient stability classification. Another advantage of PNN compared to MLPNN is that the training time is significantly faster. Thus, the PNN is a promising neural network technique for the transient stability assessment of power systems.

REFERENCES

1. Pothisarn, C. and S. Jiriwibhakorn, 2003, Critical clearing time determination of EGAT system using artificial neural networks. IEEE Power Engineering Society General Meeting, 2: 735-738.
2. Sanyal, K. K., 2004, Transient stability assessment using neural network. IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, 2: 633-637.
3. Bettiol, A.L., A. Souza, J.L. Todesco and J.R. Tesch Jr., 2003. Estimation of critical clearing times using neural networks. IEEE Bologna Power Tech Conference, 3: 6.
4. Krishna, S. and K.R. Padiyar, 2000. Transient stability assessment using artificial neural networks. IEEE International Conference on Industrial Technology, 1: 627-632.
5. Silveira, M.C.G., A.D.P. Lotufo and C.R. Minussi, 2003. Transient stability analysis of electrical power systems using a neural network based on fuzzy ARTMAP. IEEE Bologna PowerTech Conference, 3: 7.
6. Boudour, M. and A. Hellal, 2005. Combined use of supervised and unsupervised learning for power system dynamic security mapping. Eng. Appl. Artificial Intel., 18 (6): 673-683.

7. Sawhney, H. and B. Jeyasurya, 2004. On-line transient stability assessment using artificial neural network. Large Engineering Systems Conference on Power Engineering, 76-80.
8. Specht, D.F, 1992. Enhancements to probabilistic neural networks. International Joint Conference on Neural Networks, 1: 525-532.
9. Burrascano, P., 1991. Learning vector quantization for the probabilistic neural network. IEEE Trans. Neural Networks, 2 (4): 458 - 461.
10. Chen, S., C.F.N. Cowan and P.M. Grant, 1991. Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Networks, 2: 302-309.
11. Milano, F., 2005. An open source power system analysis toolbox. IEEE Trans. Power Syst., 20: 1199-1206.
12. Milano, F., 2007. Documentation for power system analysis toolbox.<<http://www.power.uwaterloo.ca/~fmilano/downloads.htm>>
13. Anderson, P.M. and A.A. Fouad, 2003. Power System Control and Stability. IEEE Press USA, pp: 37-48.
14. Riedmiller, M. and H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP Algorithm. IEEE International Conference on Neural Networks, 1: 586-591.