Application of a Novel Fuzzy Neural Network to Real-Time Transient Stability Swings Prediction Based on Synchronized Phasor Measurements

Chih-Wen Liu* Mu-Chun Su** Shuenn-Shing Tsay* Yi-Jen Wang*

*Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

**Department of Electrical Engineering.
Tamkang University
Taipei, Taiwan

Abstract-The ability to rapidly acquire synchronized phasor measurements from around the system opens up new possibilies for power system protection and control. In this paper we develop a novel class of fuzzy hyperrectangular composite neural networks which utilize synchronized phasor measurements to provide fast transient stability swings prediction for use with high-speed control. Primary features of the method include constructing a fuzzy neural network for all fault locations, using a short window of realistic-precision post-fault phasor measurements for the prediction, and testing robustness to variations in the operating point. From simulation tests on a sample power system, it reveals that the proposed tool can yield a highly successful prediction rate in real-time.

I. INTRODUCTION

With the advent of phasor measurement units (PMUs)[9,10] capable of tracking the dynamics of an electric power system, and with modern telecommunication abilities, utilities are becoming able to response intelligently to an event in progress. By synchronizing sampling of microprocessor-based systems, phasor calculations can be placed on a common reference [11]. The magnitudes and angles of these phasors comprise the state of the power system and are used in state estimation and transient stability analysis. By communicating time-tagged phasor measurements to a central location the dynamic state of the system can be tracked in real time. An emerging application of this technology is to track the state of the system immediately following a transient event to select an appropriate remedial control action. One such real-time control strategy is already being implemented at the Florida-Georgia interface[3], and others are currently under developments[5]. More specifically, use can be made of these measurements to predict a developing swing and instantaneously initiate important relay operations such as out-of-step blocking and tripping[15], or other control actions such as fast-valve control of turbines, dynamic braking, superconducting magnetic energy storage system[2], system switching, or modulation of High Voltage Direct Current (HVDC) link power flow[9].

PE-500-PWRS-0-05-1998 A paper recommended and approved by the IEEE Power System Dynamic Performance Committee of the IEEE Power Engineering Society for publication in the IEEE Transactions on Power Systems. Manuscript submitted August 12, 1997; made available for printing May 18, 1998.

There are obvious differences between the real-time stability prediction problem and on-line dynamic security assessments. In conventional dynamic security assessments [6,8,13], in which the critical clearing time (CCT) is to be found, there are three stages that a power system goes through: prefault, fault-on and postfault stages. In the prediction problem, the CCT is not of interest. Instead, one can monitor the progress of the transient in real time, thanks to the technique of synchronized phasor measurements. Moreover, we assume in a real-time problem that protection systems for faulted transmission lines are extremely fast and the fault is removed immediately at the fault inception. By ignoring the short fault-on stage (in which the transient phasor measurements are discarded) in real time, the prediction problem only involves prefault and postfault stages.

On the other hand, many existing transient stability assessment techniques, while simple in on-line application, are too complicated for real-time use. What is required is a computationally efficient way of processing the real-time measurements to determine whether an evolving swing will ultimately be stable or unstable. In [7,12,17] the possibility of using short-term prediction algorithm for this purpose was explored. In this paper a novel two-layer fuzzy hyperrectangular composite neural network (FHRCNN) is presented for real-time transient stability prediction using synchronized phasor measurements. This neuro-fuzzy approach can learn in off-line from training set and are used in on-line to predict future behavior of new data much faster than would be possible by solving the model analytically. We present a novel class of FHRCNN for prediction use in section II. Finally, simulation results are shown to demonstrate the effectiveness of the proposed method.

II. THE PROPOSED FUZZY NEURAL NETWORK

In a real-time environment it is desirable to predict the stability of transient swings before asynchronism occurs then appropriate control actions such as tripping of generators or tie-lines can be initiated to protect system equipments or an identification of recoverable swings in order to avoid unnecessary tripping to maintain the integrity of the system. That would require that computations be carried out and decision be made in very short periods of time. This is beyond the computational capability of current transient stability analysis methods.

Therefore a novel two-layer fuzzy hyperrectangular composite neural network (FHRCNN) is proposed for real-time stability prediction. The FHRCNN is a type of classifier that can be constructed off-line from a training set of examples. The FHRCNN is used to classify a transient swing as either stable or unstable on the basis of synchronized phasor measurements. Rather than attempting to solve the power system model in real-time, extensive simulations are performed off-line in order to capture the essential features of system behavior. The network building process statistically analyzes this data and constructs a FHRCNN designed to correctly classify new, unseen examples. The resulting FHRCNN classifier is compact and extremely fast, thus well suited for on-line use.

A description of basic elements of two-layer FHRCNN (shown in Fig.1) follows.

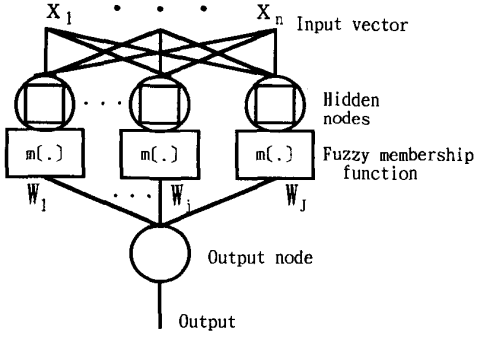

Figure 1 A two-layer FHRCNN

Input Vector(The Predictor)

$\hat{x}(=[x_1, x_2, \cdots, x_n])$ is an input vector for FHRCNN and is also the predictor for real-time transient stability prediction task. In this paper the predictor $\hat{x}$ is made by the following process.

Stability prediction is based on an eight cycle window of phasor measurements which begins at fault clearing time, T_c. Three consecutive measurements, four cycles apart, are taken from each of the total generator angles: The first measurement at T_c, another at T_c+ 4/60, and the last at T_c+ 8/60 (show in Fig.2). The generator angles, measured in radians and in center of angle coordinates, are first written to a data file in a format with three degits after the decimal.

Two velocities and one acceleration are computed from generator angles, for a total of six predictors per generator. Denoting the three angle measurements from the i'th generator $\delta_i(0)$, $\delta_i(1)$, $\delta_i(2)$, we compute

$$V_i(0) = 10*[\delta_i(1) - \delta_i(0)] \quad (1)$$

$$V_i(1) = 10*[\delta_i(2) - \delta_i(1)] \quad (2)$$

$$a_i(0) = 20*[\delta_i(2) - 2*\delta_i(1) + \delta_i(0)] \quad (3)$$

Thus the predictor, $\hat{x}$, is of the following form :

$$\hat{x} = [\delta_1(0), \delta_1(1), \delta_1(2), V_1(0), V_1(1), a_1(0), \cdots\cdots, \delta_n(0), \delta_n(1), \delta_n(2), V_n(0), V_n(1), a_n(0)] \quad (4)$$

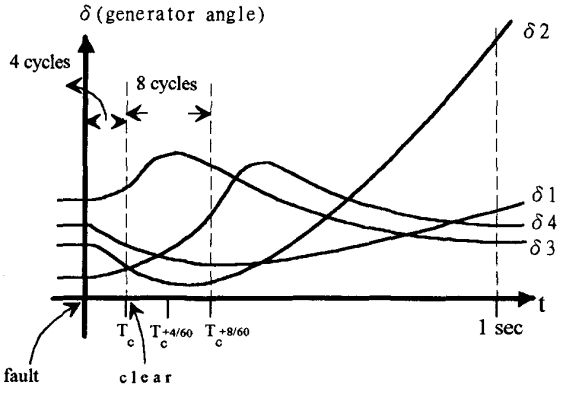

Figure 2 Reading of generator angles

Remark 1 : We have used the phrase "generator rotor angles" interchangably with "generator voltage angles" and "phasor measurements". What we are intending is that when you have a phasor measurement unit (PMU) located on the EHV or HV bus of a generating plant, that you will have a suitably detailed model of the generator for simulation purposes, but also have a classical generator model consisting of a voltage source behind a transient reactance. The internal voltage in the classical model is easily computed from the terminal voltage and current phasor measurements:

$$E_{\text{internal}} = E_{\text{terminal}} - I_{\text{terminal}} * X'_d, \quad (5)$$

where X'_d is the transient reactance. In the more general situation where phasor measurements are not taken directly from generator buses, the reduced Y_{bus} matrix relating the measured voltages V_m and the generator (internal) voltages and currents V_g , I_g can be used solve for the generator (internal) voltages g. From

$$\begin{bmatrix} I_g \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_g \\ V_m \end{bmatrix} \quad (6)$$

the second set of equations

$$Y_{21}V_g + Y_{22}V_m = 0 \quad (7)$$

can then be solved for the generator voltages by least squares. For a pattern recognition approach, a classifier could likely be trained to associate generator terminal phasor measurements directly with the prediction of future behavior. It would need to be tested whether a classifier would perform better using generator terminal phasor measurements as classifier inputs, or else using the computed internal voltage angles of the associated classical models. The proposed FHRCNN do have an advantage of being able to select the profitable variables from an overinclusive set of potential input variables. It is likely that including other variables such as line power flows would bring an improvement in the proposed FHRCNN performance without an overly large increase in the amount of computation required for training.

Remark 2 : Depending on the swings to be predicted, local devices compute bus voltage angles or generator angles from terminal bus angles, then transmit these angles to central center where transient stability predictions are conducted. Although synchronized phasor measurements are performed constantly at a high sampling rate, say, 12 times per cycle[10], we should consider the time for the transmission of angle data. It is reasonable to assume that we can obtain all synchronized data of interest at a rate of once every four cycles.

Output Variable

The magnitude of output variable is employed to label stability status of transient swing. We use 2 integers to label the status like the following :

1 → unstable swing
0 → stable swing

I-O mapping

The relationships between input vector and output variable of FHRCNN can be described by the following equation.

$$output = \sum_{j=1}^{J} w_j m_j(\hat{x}) + \theta \quad (8)$$

Where

$$m_j(\hat{x}) = \frac{vol_j}{vol_j + s_j^2(vol_j(\hat{x}) - vol_j)}$$

= fuzzy membership function ,

$$vol_j = \prod_{i=1}^{n}(M_{ji} - l_{ji})$$

= volume of the j-th hyperrectangular hidden node ,

$$vol_j(\hat{x}) = \prod_{i=1}^{n}\max(M_{ji} - l_{ji}, x_i - l_{ji}, M_{ji} - x_i)$$

And, w_j , θ , s_j , M_{ji} , l_{ji} : scalar parameters adjusted by hybrid training algorithm.

Hybrid Training Algorithm

The supervised decision-directed learning (SDDL) algorithm and back-propagation (BP) algorithm are combined to train FHRCNN. The SDDL is based on an approach that divides the input vector space into proper subsets (i.e. hyperrectangles). Each hyperrectangle, corresponding to a hidden node, is n-dimensional and defined by $[l_{j1}, M_{j1}] \times \cdots \times [l_{jn}, M_{jn}]$. Once the number of hyperrectangles and initial parameters, $l_{j1}, M_{j1}, \cdots, l_{jn}, M_{jn}$, are determined by SDDL. The BP algorithm is used to adjust parameters, w_j , θ , s_j , M_{ji} , l_{ji} , such that the following error function is minimized.

$$E = \sum_p E_p = \sum_p \frac{1}{2}(Out_p - t_p)^2 \quad (9)$$

where

t_p : desired output value of the p-th input vector,

Out_p : output value from FHRCNN.

A detailed description of the hybrid training algorithm is provided as follows:.

The SDDL Algorithm

The supervised decision-directed learning (SDDL) algorithm generates a two-layer feedforward Hyper-rectangular Composite Neural Network (HRCNN) in a sequential manner by adding hidden nodes as needed. As long as there are no identical data over different classes, we can obtain 100% recognition rate for training data. First of all, the training patterns are split into two sets: (1) a *positive* class from which we want to extract the **concept** and (2) a *negative* class which provides the counterexamples with respect to the concept. A *seed* pattern is used as the base of the initial concept (the seed pattern can be arbitrarily chosen from the positive class). When a pattern is select as the seed pattern, a kind of generalization occurs automatically, since the seed pattern naturally partitions the space into the two regions: (1) *a relevant region* surrounding the seed pattern contains all points closer to that seed pattern than from (2) a *non-relevant region*. Fig.3 illustrates this kind of generalization represented in terms of a rectangle for the two-dimensional case.

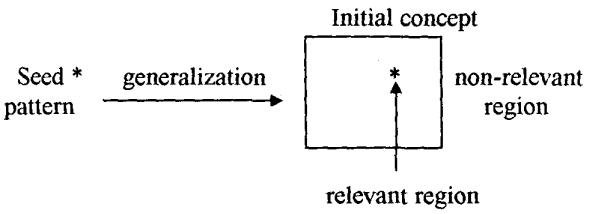

Figure 3 An initial rectangular concept for a two-dimensional case.

After a seed pattern $\hat{x}$ is chosen the weights are initialized in the following way:

$$l_i(0) = x_i - \varepsilon \quad for \quad 1 \le i \le n \tag{10}$$

$$M_i(0) = x_i + \varepsilon \quad for \quad 1 \le i \le n \tag{11}$$

where ε , a small positive real number, is the control parameter which decides the degree or generalization contributed by a positive pattern. After that, we use all counterexamples from the negative class to prevent overgeneralization (the induced concept should not be so general as to include any counterexample) formed by the initialization of weights. Fig. 4 illustrates the process of preventing the overgeneralization formed by an initial concept.

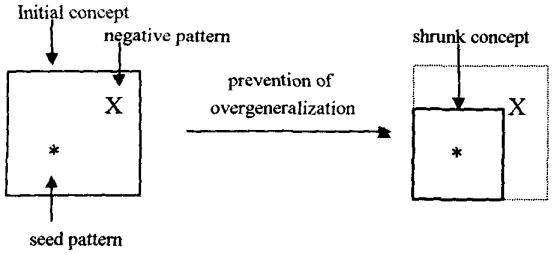

Figure 4 The process of preventing overgeneralization formed by an initial rectangular concept.

The following step is to fetch the second pattern from the positive class and to generalize the initial concept to include the new positive pattern. This process involves expanding the original hyperrectangle to make it larger to include the new positive pattern. Fig.5 illustrates the process of generalization. Here we assume every positive pattern contribute some degree of generalization to the concept so that the new pattern lies inside the expanded hyperrectangle. After the process of generalization, we again use patterns from the negative class to prevent overgeneralization.

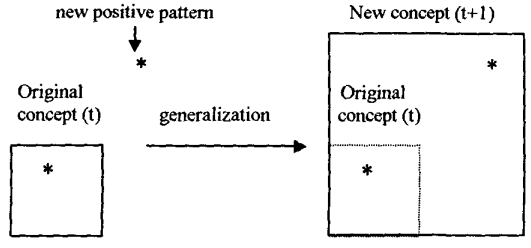

Figure 5 The process of generalization for a rectangular concept

Fig.6 shows the method of preventing overgeneralization; this process is repeated for all the remaining patterns in the positive class.

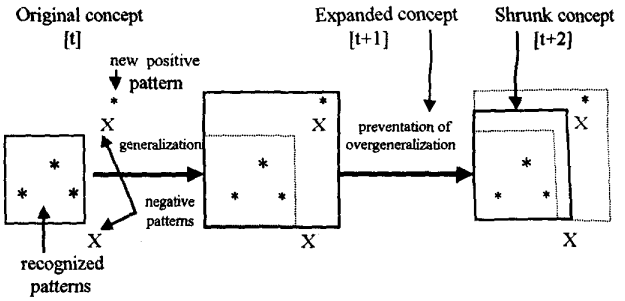

Figure 6 The process of preventing overgeneralization for an expanded rectangular concept.

Here it should be pointed out that during the process of preventing overgeneralization, the most important thing is that whenever we shrink the expanded hyperrectangle (i.e. at time t+1) in order to exclude any pattern from the negative class, the new shrunk hyperrectangle (i.e. at time t+2) should include the original hyperrectangle (i.e. at time t). This criterion guarantees that at least one positive pattern (seed pattern) be recognized after the training procedure. In other words, it prevents the new learning from washing away the memories of prior learning. Thus, after one cycle of presenting positive patterns, if there is any unrecognized positive pattern, another hidden node is self-generated and the process of learning is repeated again and again until all positive patterns are recognized. In the worst case, the number of hidden nodes is equal to the number of positive patterns. In fact the number of the extracted intermediate rules is decided by the inherent properties of the patterns. If we hope that hyperrectangles do not overlap each other too much, we can make the recognized positive patterns negative and then continue the process of the learning process. The flowchart of the algorithm is shown in Fig.7. Here pseudo-code descriptions of two important procedures are given:

- Procedure of generalization

begin ($\hat{x}$ is a positive pattern)
  for i from 1 to dimensions-of-input
    begin
      if $x_i \ge M_{ji}(t)$
      then $M_{ji}(t+1) = x_i + \varepsilon$
      else if $x_i \le l_{ji}(t)$
      then $l_{ji}(t+1) = x_i - \varepsilon$
    end;
  end;
end.

- Procedure of prevention-of-overgeneralization

begin ($\hat{x}$ is a counterexample)
  for i from 1 to dimensions-of-input

begin
if $x_i > M_{ji}(t)$
then $M_{ji}(t+2) = x_i - \delta$ (δ should be chosen to ensure $x_i - \delta \geq M_{ji}(t)$);
else if $x_i < l_{ji}(t)$
then $l_{ji}(t+2) = x_i + \delta$ (δ should be chosen to ensure $x_i + \delta \leq l_{ji}(t)$);
end;
end;
end.

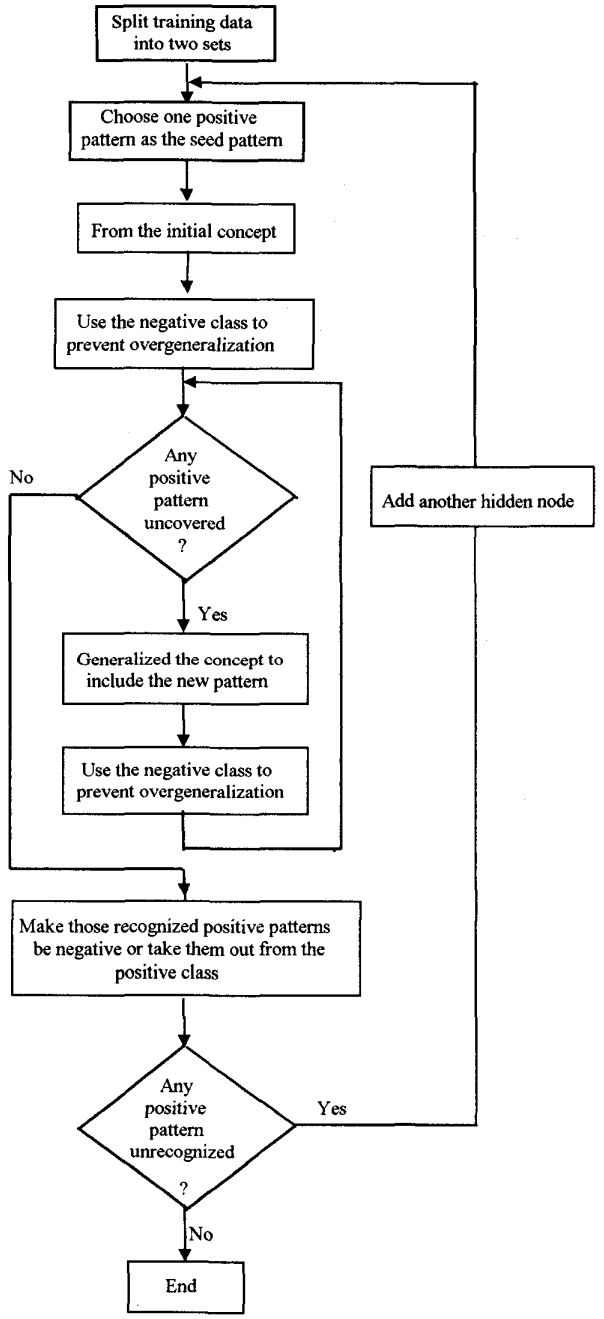

Figure 7 The flowchart of the SDDL algorithm.

The value of the ε can be equal to or greater than zero. As for the specification of the value of the parameter δ, one simplest method is to make δ be equal to $\frac{1}{2}(x_i - M_{ji}(t))$ if $x_i > M_{ji}(t)$ or $\frac{1}{2}(l_{ji}(t) - x_i)$ if $x_i < l_{ji}(t)$.

Fuzzification

A crisp classification attempts to determine the exact boundary inside which patterns of the same class lie. However much of the information we have to deal with in engineering applications or in real life is in the form of imprecise, disturbed, or uncertain patterns. Obviously, it is not appropriate to use crisp classification rules to manipulate such kinds of data because vagueness may exit in the decision space, therefore, fuzzy logic into classifications tool to cope with the kinds of data. The incorporation of fuzzy logic into classifications provides flexibility and robustness. Fuzzy classification rules regard the decision surface between two different classes as a grey region. A pattern falling in the grey region has partial membership value representing the degree to which class it belongs. As discussed in the preceding section, the values of the synaptic weights of a trained HRCNN can be utilized to extract crisp classification rules of which antecedents are represented in the form of a set of hyperrectangles. In order to fuzzify these crisp rules, a membership function is required to measure the degree to which an input pattern is close to a hyperrectangle. As the membership function, $m_j(\hat{x})$, approaches 1, and input pattern $\hat{x}$ is more close to the j^{th} hyperrectangle defined by $[l_{j1}, M_{j1}] \times \ldots \times [l_{jn}, M_{jn}]$ with the value representing complete hyperrectangle containment. Basically, any function producing a value between 0 and 1 can be a membership function. Membership functions can have different shapes (trapozoidal, triangular, or bell-shaped) depending on the designer's preference or experience. According to our simulations on many pattern recognition problems such as speech recognition, character recognition, etc., we found the following function is a good choice in terms of computation burden and recognition rates.

$$m_j(\hat{x}) = \frac{Vol_j}{Vol_j + S_j^2 (Vol_j(\hat{x}) - Vol_j)} \qquad (12)$$

After a fuzzification, HRCNN evolves to FHRCNN.

The BP algorithm

The BP algorithm is derived based on the concept of the gradient descent search to minimize the error function, E, through the adjustment of w_j , θ , S_j , M_{ji} , and l_{ji}. Individual w_j adjustments are computed by Eq.(13-14).

$$\frac{\partial E}{\partial w_j} = \sum_p \frac{\partial E_p}{\partial Out_p}\frac{\partial Out_p}{\partial w_j} = \sum_p (Out_p - t_p) m_j(\hat{x}) \tag{13}$$

$$w_j(t+1) = w_j(t) + \eta \frac{\partial E}{\partial w_j} \tag{14}$$

θ adjustment is computed by Eq.(15-16).

$$\frac{\partial E}{\partial \theta} = \sum_p \frac{\partial E_p}{\partial Out_p}\frac{\partial Out_p}{\partial \theta} = \sum_p (Out_p - t_p) \tag{15}$$

$$\theta(t+1) = \theta(t) + \eta \frac{\partial E}{\partial \theta} \tag{16}$$

Individual S_j adjustments are computed by Eq.(17-18).

$$\frac{\partial E}{\partial s_j} = \sum_p \frac{\partial E_p}{\partial Out_p}\frac{\partial Out_p}{\partial m_j(\hat{x})}\frac{\partial m_j(\hat{x})}{\partial s_j} = \sum_p (Out_p - t_p) w_j \frac{2}{s_j} m_j(\hat{x})(m_j(\hat{x}) - 1) \tag{17}$$

$$s_j(t+1) = s_j(t) + \eta \frac{\partial E}{\partial s_j} \tag{18}$$

Individual M_{ji} adjustments are computed by Eq.(19-20).

$$\frac{\partial E}{\partial M_{ji}} = \sum_p \frac{\partial E_p}{\partial Out_p}\frac{\partial Out_p}{\partial m_j(\hat{x})}\frac{\partial m_j(\hat{x})}{\partial M_{ji}} = \sum_p (Out_p - t_p) w_j \begin{cases} \dfrac{m_j^2(\hat{x}) Vol(\hat{x}) s_j^2}{Vol_j(x_i - M_{ji} + l_{ji}^2)} & \text{if} \quad x_i > M_{ji} \\ 0 & \text{if} \quad M_{ji} - l_{ji}^2 \le x_i \le M_{ji} \\ -\dfrac{m_j^2(\hat{x}) s_j^2 Vol(\hat{x})}{Vol_j(x_i - M_{ji})} & \text{if} \quad x_i < M_{ji} - l_{ji}^2 \end{cases} \tag{19}$$

$$M_{ji}(t+1) = M_{ji}(t) + \eta \frac{\partial E}{\partial M_{ji}} \tag{20}$$

Individual l_{ji} adjustments are computed by Eq.(21-22).

$$\frac{\partial E}{\partial l_{ji}} = \sum_p \frac{\partial E_p}{\partial Out_p}\frac{\partial Out_p}{\partial m_j(\hat{x})}\frac{\partial m_j(\hat{x})}{\partial l_{ji}} = \sum_p (Out_p - t_p) w_j \begin{cases} 2m_j^2(\hat{x}) Vol(\hat{x}) s_j^2 \left(\dfrac{1}{l_{ji}} - \dfrac{l_{ji}}{x_i - M_{ji} + l_{ji}^2}\right)\dfrac{1}{Vol_j} & \text{if} \quad x_i > M_{ji} \\ 0 & \text{if} \quad M_{ji} - l_{ji}^2 \le x_i \le M_{ji} \\ 2m_j^2(\hat{x}) Vol(\hat{x}) s_j^2 \Big/ (l_{ji} Vol_j) & \text{if} \quad x_i < M_{ji} - l_{ji}^2 \end{cases} \tag{21}$$

$$l_{ji}(t+1) = l_{ji}(t) + \eta \frac{\partial E}{\partial l_{ji}} \tag{22}$$

where η learning rate

III. SIMULATION RESULTS

The FHRCNN was investigated using the IEEE 39-bus test system as shown in Fig.8. In our simulation model, the generator is modeled by 7th-order differential equations and the loads are modeled as constant impedance. A detailed description of the above model can be referred to [1]. Three-phase short-circuit-to-ground faults with four-cycle clearing time are simulated to occur on various transmission lines. The postfault system configuration is the same as the prefault system except that the faulted line is removed. Each examples of a fault contains the simulated post-fault phasor measurements along with whether the particular fault results in instability. Large numbers of examples are aggregated together into training and test sets, from which FHRCNN are constructed and tested.

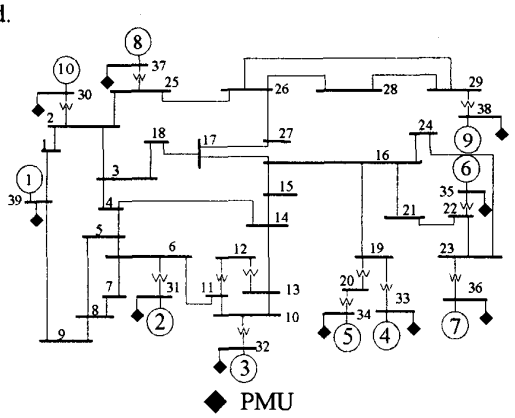

Figure 8 One-line diagram of IEEE 39-bus system equipped with 10 PMUs.

Instability Criterion

The criterion for instability is whether the difference between any two generator angles exceeds π radian in the 1 second after clearing time. Otherwise the fault is declared as stable. We would like to emphasize that the instability criterion depends heavily on the characteristics of specific power systems. Here, the chosen π radian is just for illustration of our scheme.

Training Sets and Test Sets

For a given fault location, duration and clearing action, the fault-on and postfault swings are obtained from PSS/E computer package developed by Power Technologies, Inc.(PTI). Several operating points were generated in order to test the FHRCNN method on a stressed system, and in order to study the method's robustness to variations in the operating point. Our base case was obtained by increasing the real powers of the individual load by 35%. The extra power generation was spread uniformly among the generators. We choose an increase of 35% because it lowered critical clearing times, while maintaining an

acceptable load-flow solution. Fifty operating points are generated from base case by considering random changes of key parameters like load, shunt compensation, active and reactive generation scheduling, and topology. The distribution of the random numbers was uniform rather than Gaussian, and a different string of random numbers was used for each operating point. Faults of varying location, duration and clearing action were simulated for each operating point. A bus fault refers to a fault on the end of a transmission line which is cleared by removing the line. A mid-line fault refers to a fault in the middle of a transmission line. For the training set, we simulate 650 bus faults and 350 line faults per operating point, in which 330 faults are unstable and 670 faults are stable. For the test set, we simulate 350 bus faults and 150 line faults, in which 165 faults are unstable and 335 faults are stable, separate from training set per operating point.

Results

The simulation program were developed on a SUN SPARC II in C++. For a comparison, we conducted simulation results on traditional feedword neural network (FNN) with the same neurons, training set, and test set. The proposed FHRCNN and FNN were trained on the training set using 61 neurous. It takes about 2 hours to train FHRCNN and about 3 hours to train FNN. Both networks are tested on the test set. It takes just a few hundredths of 1 second to predict the swings in the 1 second after clearing time for both networks. The classification rates of both networks are given in Table 1. The numbers indicate the percentage of predictors correctly classified.

From simulation results, one has the following observations :

(1)The FHRCNN has a pretty high classification rate over 99% for training set and over 94% for test set.

(2)The FHRCNN has a better performance than traditional FNN.

(3)The FHRCNN has the potential to be an on-line tool for real-time transient stability swings prediction in power systems.

Table 1 Classification rates of FHRCNN and FNN for stability swings prediction

Network Type	Training Set		Test Set	
	Stab	Unstab	Stab	Unstab
FHRCNN	99.8	99.5	95.1	94.2
FNN	73.5	73.3	72.7	72.4

IV.CONCLUSION

We have demonstrated the success of properly trained FHRCNN in predicting transient stability swings from a short window of post-fault phasor measurements. Extensive testing was performed on the IEEE 39-bus system under heavy loading conditions. The computational burden proved to be quite reasonable, and larger systems could be handled. Since individual faults are generated independently, parallel implementation is trivial.

Current financial and environmental trends portend a power system forced to operate under more stressed condition than in the past. This leave little room for traditional avoidance principles and will no doubt lead to the use of more control. The transient stability swings prediction method discussed in this paper is designed to provide timely information for power system control in a real-time. We suggest that a neuro-fuzzy approach can automate the process of transforming off-line simulation studies into on-line decision rules.

V.ACKNOWLEDGEMENT

This work was supported in part by National Science Council under the grant NSC 85-2213-E-002-071.

VI.REFERENCES

[1]P.M. Anderson, and A.A. Fouad, Power System Control and Stability, Iowa State University Press, Ames, 1977.

[2]H.J. Boenig and J.F. Hauer, "Commissioning Tests of the Bonueville Power Administration 30 MJ SMES unit", *IEEE Trans. on Power Appratus and Systems*, vol. PAS-104, No.2, pp.301-312, Feb. 1985.

[3]V. Centeno et al., "Adaptive Out-of-Step Relaying Using Phasor Measurement Techniques", *IEEE Computer Applications in Power*, vol.6, No.4, pp.12-17,1993.

[4]Chung-Liang Chang, System Planning Department, *Taiwan Power Company*, Personal communication, Aug.1995.

[5]Ph. Denys et al., "Measurement of Voltage Phase for the French Future Defence Plan Against Losses of Synchronism", *IEEE Trans. on PWRD, PWRD-7*, No.1, pp.62-69,1992.

[6]A.A. Fouad et. al., "Dynamic Security Assessment Practices in North America", *IEEE Trans. on Power Systems, PWRS-3*, No.3, pp.1310-1321, 1988.

[7]C.-W.Liu and J.S.Thorp, "Application of Synchronized Phasor Measurements to Real-Time Transient Stability Prediction", *IEE Proc.-Gener. Transm. Distrib.*, vol.142, No.4, pp.355-360, 1995.

[8]M.A. Pai, Energy Function Analysis for Power System Stability, Kluwer, Boston, 1989.

[9]A.G. Phadke and J.S. Thorp, Computer Relaying for Power Systems, John Wiley and Sons Inc., New York, 1988.
[10]A.G. Phadke, "Synchronized Phasor Measurements in Power Systems", *IEEE Computer Applications in Power*, vol.6, No.2, pp.10-15, 1993.
[11]A.G. Phadke et al., "Synchronized Sampling and Phasor Measurements for Relaying and Control", *IEEE PES Winter Meeting, Columbus, Ohio*, Feb.1993(93 WM 039-8-PWRD).
[12] S. Rovnyak, C.-W. Liu, J.Lu, W. Ma and J. Thorp, "Predicting future Behavior of Transient Events Rapidly Enough to Evaluate Remedial Control Options in Real-Time", *IEEE Trans, on Power Systems*, vol.10, No.3, pp.1195-1203, 1995.
[13]D.J. Sobajic, and Y.H. Pao, "Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems, PWRS-4, No.1, pp.220-228, 1989.
[14]M.C. Su, A Neural Network Approach to Knowledge Acquisition, Ph.D. Thesis, University of Maryland, Aug.1993.
[15]C.W. Taylor, J.M. Haner, L.A.Hill, W.A. Mittelstadt, and R.L. Cresap, "A New Out-of-Step Relay with Rate of Change of Apparent Resistance Augmentation", *IEEE Trans. on Power Apparatus and Systems*, vol.PAS-102, No.3, pp.631-639, March 1983.
[16] J.S. Thorp, A.G. Phadke, S.H. Horowitz and M.M. Begovic, "Some Applications of Phasor Measurements to Adaptive Protection", *IEEE Tran. on Power Systems*, vol.3, No.2 pp.791-798, May 1988.
[17] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision Trees for Real-Time Transient Stability Prediction", *IEEE Trans. on Power Systems*,vol.9,No.3,pp. 1417-1426, Aug. 1994.

Chih-Wen Liu was born in Taiwan in 1964. He received the B.S. degree in electrical engineering from National Taiwan University in 1987, Ph.D. degree in electrical engineering from Cornell University in 1994. Since 1994, he has been with National Taiwan University, where he is associate professor of electrical engineering. His research interests lie in power system computer application.

Shuenn-Shing Tsay was born in Pingtung, Taiwan in 1965. He received his B.S. degree in Industrial Education from National Taiwan Normal University, Taipei, Taiwan in 1990, M.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 1996. He is a teacher in Taiwan Provincial Pingtung Industrial Vocational High School, Pingtung, Taiwan. His research interests include application of synchronized phasor measurements to power system transient stability and Artificial Intelligence in power systems.

Yi-Jen Wang was born in Taiwan in 1961. He received his B.S. degree in electrical engineering from National Taiwan Institute of Technology in 1986, M.S. degree in electrical engineering from National Taiwan University in 1990. He is currently working toward his Ph.D. degree at National Taiwan University. His research interests include power system analysis and optimal theory.

Mu-Chun Su received the B.S. degree in electronics engineering from National Chiao Tung University, Taiwan, R.O.C. in 1986, and the M.S. and Ph.D. degrees in electrical engineering from University of Maryland, College Park, MD, in 1990 and 1993, respectively. He was Franklin V. Taylor Award recipient for the most outstanding paper presented to 1991 IEEE Systems, Man, and Cybernetics Conference. He is an Associate Professor of Electrical Engineering at Tamkang University, Taiwan, R.O.C. His current research interests include neural networks, fuzzy systems, medical informatics, pattern recognition and speech recognition.