

FAST ASSESSMENT OF TRANSIENT STABILITY MARGINS BY A NEURAL NETWORK APPROACH

J. N. Fidalgo, J. A. Peças Lopes, Vladimiro Miranda
DEEC - Fac. de Engenharia da Universidade do Porto
INESC, Largo Mompilher 22 - 4000 Porto, PORTUGAL

L. B. Almeida
DEEC - Instituto Superior Técnico
INESC - Lisboa - PORTUGAL

Keywords - Dynamic Security Assessment; Neural Network Applications.

Abstract: This paper reports the use of Artificial Neural Networks (ANN) for fast and accurate evaluation of the transient stability degree for each contingency in a multimachine power system, using only real time monitorable system values. The output of an ANN provides an emulation of the transient stability energy margin. Preventive control measures can be suggested through generation load redispatch, by getting the sensitivities of the energy margin directly from a trained ANN. The approach was tested with several contingencies in the CIGRE test system, giving better results than other methods so far reported in the literature.

1. INTRODUCTION

Transient stability evaluation is a problem of major concern in the on-line security assessment of multimachine power systems. This stability evaluation deals with the system behavior after the occurrence of a given disturbance and its main concern is to predict whether the system will collapse or not.

For each disturbance to be considered the traditional and most accurate analysis involves the numerical solution of a system of non-linear differential equations, which is a very demanding and time consuming computational task and is therefore unsuitable for on-line purposes. The need for the development of on-line assessment methods has therefore been considered by the utilities and research institutions as a priority problem to be tackled. On the other hand the transient stability assessment by itself is an incomplete task, because when an unstable state is detected fast control actions or preventive control actions must be suggested to the system operators in order to prevent the system to collapse.

During the last thirty years these issues have been a matter of intensive research using Lyapunov, transient energy function methods and other related energetic approaches [1,2], Pattern Recognition (PR) approaches [3,4,5] and Artificial Intelligence techniques (like decision trees [6]).

In spite of the maturity already attained by direct methods, they present however some drawbacks derived mainly from the great computing effort needed to estimate the stability boundary and the use, as input, of a large set of state variables that cannot be directly monitored in real time applications. This has strongly motivated the effort on the research for the use of other methodologies.

This paper describes successful exploratory efforts in transient stability analysis for real time applications under two main concerns:

1. Adopting a very fast and reliable tool to emulate the stability behavior of power systems: an artificial neural network (ANN) approach;
2. Building models based mainly on monitorable system variables.

The feasibility of the application of an ANN approach to be used as a real time tool is demonstrated below, with several contingencies simulated in the CIGRE test system, for:

1. Classifying the stability/unstability status of a given operating point for each specified contingency;
2. Presenting a numerical indication of the transient stability degree of each operating point and for each selected contingency, through the emulation of the transient energy function margin ΔV ;
3. Providing fast and reliable suggestions for preventive control strategies when unstability is detected.

2. ANN PREVIOUS ESSAYS

2.1. Abridged overview

In the last few years the application of Artificial Neural Network (ANN) approaches has been tried in order to deal with particular problems, because they provide a tool with the characteristics needed for real time response and have the ability to store very complex relationships.

Sobajic and Pao have presented a first paper about this subject [7] where critical clearing times (CCT) for a

given system disturbance were evaluated when using a multilayer perceptron with six neurons in a single hidden layer. Also in that paper the authors have shown that with the same ANN it was possible to correctly estimate CCT even when different topological configurations were considered, which is quite difficult to obtain when using pattern recognition methods where every new topology requires a new classifier.

Later Pao and Sobajic [8] have proposed a combined use of an unsupervised and supervised learning procedure together with the use of a functional link net architecture, in order to obtain an ANN for CCT evaluation.

Recently, Djukanovic et al. [9] presented an application of an ANN to be used in support of the decision making process towards the fast stabilization of multi-machine systems through the determination of generator shedding requirements when instability was detected.

A good discussion of the problems to be solved when implementing the ANN methodology for the dynamic security assessment in real dimension systems is presented in Kumar et al. [10].

2.2. Neural network basics

An Artificial Neural Network is a massively parallel architecture, with a high degree of connectivity between its elementary processors.

A most interesting characteristic of this kind of system is its ability to learn from training data rather than from a set of rules. The ANN's structure is dictated, among other factors, by the complexity of the "transfer function" inherent to the training set. An introduction to ANNs is given in the references [11,12,13].

A typical ANN structure can present one input layer, one or more hidden layers and one output layer. The input layer receives the input vector that stimulates the network. The contribution of the element i of this vector in the unit j of the next layer is given by the product $x_i w_{ij}$, where w_{ij} is the weight of the connection between the units i and j . The net input of unit j is then:

$$\text{net}_j = \sum_i (w_{ji} x_i) \quad (1)$$

The value net_j is then passed through a non-linear function (activation function) that establishes the output state of unit j . The function selected in this study was the hyperbolic tangent tgh , and we will have:

$$O_j = f(\text{net}_j) = \text{tgh}(\text{net}_j + q_j) \quad (2)$$

where q_j is the bias value of unit j , whose role is to shift the transition region of function f .

The outputs of all hidden units are added in the output layer units to decide their output states. The ANN is trained by iteratively being shown the training data.

In complicated classification problems the hidden units allow the generation of complex decision regions able to interpolate among the training data. On the other hand if the classification boundaries are simple, the ANN won't need many hidden units.

2.2.1. Training the ANN with the Backpropagation method

In ANNs, the interconnect weights are initially unknown parameters. The learning rule used to obtain the ANN in this paper is a variant of the steepest decedent method, which is known as "backpropagation" (BP).

The parameter usually chosen for representing the network performance is:

$$E = \frac{1}{2} \sum_{k=1}^{N_p} (T_k - O_k)^2 \quad (3)$$

where T_k is the desired output. O_k represents the actual output, calculated with the current values of weights and input vector and N_p is the number of training patterns. One desires that E will be reduced to a minimum, by learning adequately.

In the BP algorithm, the weights are updated iteratively according to a gradient optimization procedure using the entire data set:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \quad (4)$$

where η is the learning rate parameter. A way to increase the learning speed, without causing oscillations, is to modify expression (4) to include a momentum term:

$$\Delta w_{kj}(n+1) = -\eta \frac{\partial E}{\partial w_{kj}} + \alpha \Delta w_{kj}(n) \quad (5)$$

A more detailed description of the training algorithm can be found in [11,12,13].

2.2.2. Making Backpropagation faster

The ANN's most "unpleasant" characteristics are related to the large amount of data and training time required. Another negative point related to the method, is the possibility of reaching a local minimum, where the error criterion is not satisfied. However, in the great majority of cases, a point can be reached which leads to

good global performance results.

To increase the efficiency of the training process we used the Adaptive Backpropagation (ABP) algorithm [15]. This method is based on individual adaptation of the learning rate parameter of each synapse. If in a learning process, the sign of a given component of the gradient remains equal for several iterations, that is because the surface error in this axis has a monotonous path at least in the neighborhood of these points. Then this particular learning rate should be increased. On the other hand, if the sign of some component changes consecutively, then that learning rate should be decreased to avoid oscillation.

To implement this idea we substitute eq. (5) by

$$\Delta w_{kj}(n) = -\eta_{kj}(n) \cdot v_{kj}(n) \quad (6)$$

where

$$v_{kj}(n) = \frac{\partial E}{\partial w_{kj}}(n) + \alpha v_{kj}(n-1) \quad (7)$$

and $\eta_{kj}(n)$ is the learning rate parameter of the synapse between units k and j at epoch n .

The learning rate adaptation process must be made at each epoch according to:

$$\eta_{kj}(n) = \begin{cases} u \eta_{kj}(n-1) & \text{if } \frac{\partial E}{\partial w_{kj}}(n) \cdot \frac{\partial E}{\partial w_{kj}}(n-1) > 0 \\ d \eta_{kj}(n-1) & \text{if } \frac{\partial E}{\partial w_{kj}}(n) \cdot \frac{\partial E}{\partial w_{kj}}(n-1) < 0 \end{cases}$$

where u and d are positive constants with values slightly above and below unity, respectively. For a full description of the algorithm, see [15].

The ABP algorithm provides a much faster learning process. In some complex mapping cases we experienced only 1/1000 (or even less) of the training epoches required by the traditional BP algorithm, with the same initial conditions.

It is known that if the patterns presented to the ANN have the statistical properties of mean zero and unity variance, the learning process is faster. So, this pre-processing was implemented in the training data sets used in this work.

There are several possible weight initialization techniques. The one adopted in the work reported below was based on getting a few samples of the training set (say 5 to 10%), and starting the learning phase with only this training subset. With only 5% of the training patterns, the learning is incomplete but, on the other hand, each iteration is 20 times faster than if we had the whole set.

This method provides, together with the ABP, a fast

weight initialization. After a few epoches (presentations to the ANN of the training set), the amount of data shown to the ANN was enlarged by another share (10 to 20%) of the training set. This procedure was followed until the whole training set was presented to the ANN.

3. ANNs FOR TRANSIENT STABILITY EVALUATION

The success of the application of an ANN approach to the Transient Stability assessment problem depends on the verification that there are some pre-fault system characteristics that give rise to a stable or unstable post-fault system.

Like in the traditional Pattern Recognition approaches the use of ANNs requires the following phases: an initial feature selection stage; a training set generation; a learning procedure; and performance evaluation tests.

3.1. Feature Selection

Each operating point is characterized by a set of system measures (ANN inputs). Input data must appropriately characterize the system state, but one must try to keep the input set as small as possible while having still enough discriminating ability.

In the work reported here, the following set of system parameters and measures were used:

- generated active power outputs (P_i);
- emfs (E_i);
- machine inertia constants (M_i);

where $i=1, \dots, NG$ with NG - number of generation buses.

These variables share the unique characteristic that they consist of values available in an EMS data base, either because they are known preset values or monitorable values, injected by a data acquisition process. Note that when using traditional PR approaches, and in order to obtain reduced classification errors, the features that were used are usually variables directly related with the transient stability evaluation like kinetic and potential energies at the fault elimination moment [3].

In spite of the apparent simplicity of the chosen input variables, they implicitly contain the information needed to derive the values of other variables with greater discriminating power, like the accelerating power or even other kinetic or potential related variables. The use of an ANN approach enables us to exploit the capability of reproducing internally the relationships between the selected variables and other more complex ones that are not explicitly used in this work.

3.2. Training and Test Set Generation

This stage consists in generating a great number of feasible operating situations and analyzing them for each considered contingency. For that purpose we have used an algorithm described in [4], where the classical reduced dynamic model for transient stability studies was used [1], adopting in this model the center of angle reference frame.

For each selected contingency several load levels were considered and for each load level different unit-commitment alternatives were examined. Each scenario was submitted to an optimum dispatch exercise. Furthermore, new non optimally dispatched scenarios were built from the previous optimal ones, by changing the injected power at each busbar by a certain amount and distributing this difference among the other buses.

In order to completely characterize each operating point a load flow was then performed for each scenario. A selected contingency was then simulated and system stability analyzed by the Transient Energy Function method [1,2], which allows the assignment to each operating point of a transient energy margin value $\Delta V = V_{cr} - V_{tel}$ (where V_{cr} is the transient energy function value at the stability boundary and V_{tel} is the energy function value at the fault elimination moment).

When using this direct method, the stability region evaluation, needed to determine V_{cr} , was computed through the potential energy boundary surface method [1,2]. In the transient energy function that was used the conductances were included and the path dependent term was numerically evaluated using a trapezoidal rule as proposed by Athay et al. [2].

The ΔV margin was used for labeling each operating point for the contingency under analysis:

$$\Delta V < 0: \text{unstable}; \quad \Delta V > 0: \text{stable}.$$

3.3. ANN Learning for Power System Transient Stability Classification and Transient Security Index Derivation

It is well known that a simple classification procedure leads to incomplete information and therefore obtaining a stability index is of greatest importance. The CCT (Critical Clearing Time) determination is the traditional way to quantify the system stability. However, the derivation of control actions stemming directly from this index seems to be a very difficult and heavy task.

The transient energy function margin ΔV , besides enabling system stability classification also produces a measure of the system transient stability for a given contingency. Therefore the idea was to build an ANN to reproduce the ΔV margin value for each operating point and contingency, thus providing a ΔV emulation, ϕ ,

able to give a real time stability indication.

In the training set generation phase, the ΔV value for each operating point included in the training set is therefore kept, to be used as the desired target output value when training the ANN.

In order to obtain the best ANN structure, different ANN topologies (number of layers and number of neurons in each hidden layer) have been tested. This report will refer to the one which displayed the best results.

For stability classification, the performance evaluation index for each structure was the total classification error in a test set. For ΔV emulation, the root mean squared error (Erms) and the average error (Ema), obtained in a test set for each contingency, were used instead.

4. DERIVATION OF PREVENTIVE CONTROL STRATEGIES

4.1. The problem

When an unstable situation is detected, ($\phi < 0$), a change in the control variables, in order to increase the security factor ϕ , is required. For this preventive control strategy only changes in P_i values were allowed, corresponding to a generation redispatch. The general problem can be formalized as a typical non-linear constrained optimization problem:

$$\min f(\mathbf{P}) \quad (8)$$

$$\text{subj. } \mathbf{P} \in \mathbf{D} \quad (9)$$

$$\phi \geq \phi_{\text{sec}} \quad (10)$$

where \mathbf{D} is the domain of the feasible power flow solutions, $f(\mathbf{P})$ is the generation cost function and ϕ_{sec} is the security factor to be reached, corresponding to a desired transient stability margin with $\Delta V > 0$. An initial solution may be found by any optimizing dispatch technique, neglecting the last constraint. In order to find, in a next step, a feasible solution to the global problem, whenever the optimal dispatch solution violates $\phi \geq \phi_{\text{sec}}$, a technique similar to the reduced gradient method was adopted. This procedure requires the knowledge of the partial derivatives of ϕ relative to each P_i .

4.2. Estimating derivatives with ANNs

Neural networks provide a simple and efficient way to evaluate the derivatives of the output relative to the input variables. In this case, this is equivalent to the calculation of the sensitivities of the transient energy margin with respect to the chosen control variables.

Based on eqs. (1) and (2), it is possible to calculate the derivative of the output activation in order to obtain the activation of the units of the previous layer:

$$\frac{\partial O_j}{\partial x_k} = \frac{\partial O_j}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial x_k} = f' w_{kj} \quad (11)$$

where k refers to any of the units that connects directly to the output j .

Now, a chain rule can be applied, calculating iteratively the successive derivatives from the output layer back to the input. Note that if a unit feeds two or more units in the next layer, those influences must be added in the calculation. So, for a unit in a layer behind the penultimate layer we have:

$$\frac{\partial O_j}{\partial x_i} = \sum_k f'_k w_{ki} \frac{\partial O_j}{\partial x_k} \quad (12)$$

where k now refers to all the units in the next layer that are directly fed by unit i .

This process is very similar to the one used in the BP algorithm, where the derivative of the output error relative to each weight is calculated.

4.3. Algorithm

For a given operating point and for a selected contingency for which a trained ANN is available, if the transient energy function emulation ϕ is less than ϕ_{sec} , then the preventive control algorithm is applied (see figure 1).

Each algorithm step is based on the redistribution of the powers to be generated in each busbar. Following a gradient technique, for each generating bus, a correction ΔP_i is given by

$$\Delta P_i = H \beta_i, \quad i=1, 2, \dots, NG-1 \quad (13)$$

where

$$\beta_i = \frac{\partial \phi}{\partial P_i} - \frac{\partial \phi}{\partial P_{NG}} \quad (14)$$

NG is chosen as a reference bus. ΔP_{NG} is calculated using the constraining equation derived from an incremental DC approach to the power flow problem:

$$\sum_i \Delta P_i = 0$$

The gradient vector $\mathbf{\beta}$ is normalized, as usual in these algorithmic approaches. H is a step size control parameter and is changed according to the evolution of the algorithm [if $\phi(n+1) > \phi(n)$ for several consecutive iterations, then H is increased; if P_i exceeds one of the production limits or if $\phi = \phi_{\text{sec}}$ then H is decreased]. In general, small starting values for H are adopted, so that small relative changes are generated in each step to avoid instability in the iterative process.

The constraints concerning the limits of power to be generated in each bus (P_i^{\min} and P_i^{\max}) are checked and

dealt with during the iterative procedure, so that the dispatch corrections always lead to feasible load flow solutions.

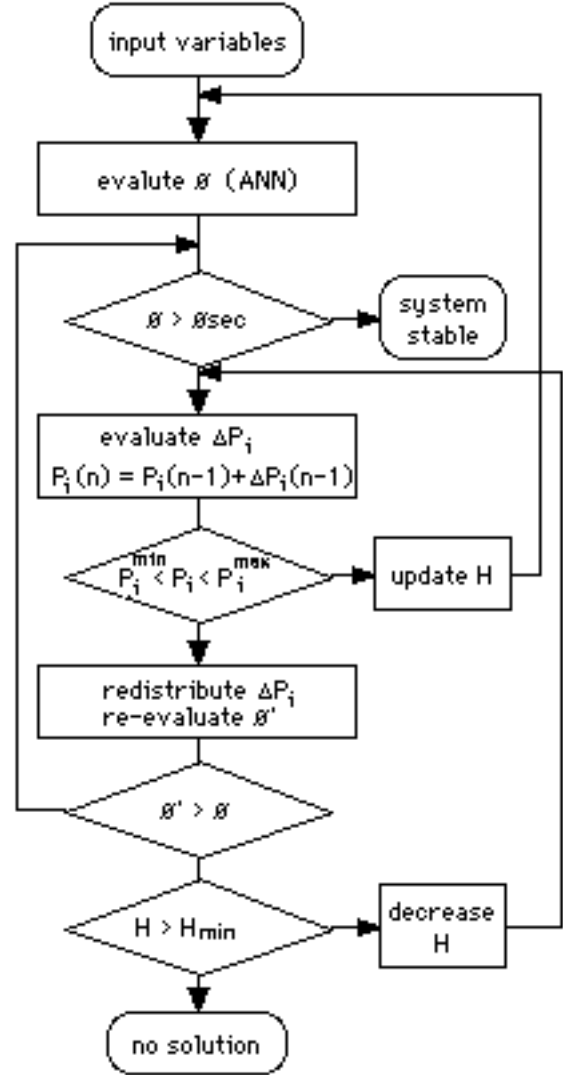


Fig. 1. - Algorithm of the preventive control procedure

In situations where the system is very unstable it might be necessary to largely reduce the generated power from the most critical machines; however it might happen that the other machines are unable to generate that power value: this indicates that the situation may not be controllable and therefore no solution is obtained through the described method.

The algorithm described above must be seen only as the first step of a more general non linear programming procedure, that would seek the optimal solution of problem (8 - 10). For instance, in a hemstitching strategy, one would in alternate steps use the gradient of the objective function and of the violated $\phi \geq \phi_{\text{sec}}$ constraint until a convergence criterion would be met.

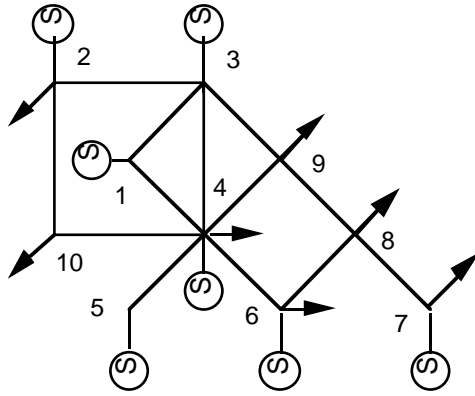


Fig. 2. - Single diagram of CIGRE test system

5. NUMERICAL RESULTS

5.1 Data

Numerical results obtained with four contingencies simulated in the CIGRE test system are described and discussed next. Figure 2 shows the single line diagram of the system; numerical data can be found in Peças Lopes et al. [4]. The simulated contingencies are described in Table 1.

Table 1 - Simulated contingencies

contingency nr.	3phase fault on bus	elimin time (s)	line switched
1	1	0.3	1-3
2	3	0.37	3-2
3	5	0.31	no
4	6	0.43	6-8

In order to obtain the corresponding training sets several load levels were considered, with the total system load varying linearly between 1190 MW and 1850 MW. Each operating point was thus initially characterized by a set of 21 variables: P_i ; E_i ; M_i ($i=1, \dots, 7$).

5.2. Stability classification

The data set was then divided in two sub-sets: the training set and the test set. The number of patterns for each contingency and subset is mentioned in Table 2, as well as the test error (number of misclassifications divided by the total number of test patterns) obtained with the most successful ANN topology (an ANN with a structure of 21 inputs, 4 neurons in the first hidden layer and 2 neurons in the second hidden layer is mentioned as having a topology 21-4-2-1).

Other variables, like...

- accelerating powers at fault inception moment (P_{aci});

Table 2 - Classification results

conting nr	nr. train patterns	nr. test patterns	ANN topology	test error (%)
1	1393	348	21-4-2-1	0
2	1161	290	21-4-2-1	1.0
3	1108	185	21-4-2-1	0.5
4	1681	406	21-4-2-1	0.7

- accelerating power derivatives relative to the machine rotor angle for all generators ($\partial P_{aci} / \partial \theta_i$);
- total load level consumption (PL);
- total inertia constant of the system ($M_t = \sum M_i$);
- center of angle accelerating power immediately after fault occurrence (P_{coa} / M_t);

...were also tried in order to characterize each operating point in the training and test sets. In spite of their better dynamic characterization, compared to the first mentioned reduced set of variables, their use as input led to similar performances of the ANNs. The noticed advantage in using these variables usually associated with the stability phenomenon was that the number of epoches needed to train the ANN was found to be smaller than in the case where the reduced set of controllable variables was used.

The quality of the results obtained must be strongly stressed in two main aspects:

- the ANN approach is most adequate for real time applications, providing a faster response than any other method available, for the same accuracy level;
- furthermore, the discriminating ability obtained in all the tried examples is largely superior to previously reported results, namely those based on pattern recognition techniques, for which the classification errors are greater (order of magnitude 1-5%), even when using variables with, in principle, higher discriminating capabilities [3,4].

5.3. Energy function emulation

When training the ANN in order to obtain the emulation of the transient energy function values for the specified contingencies, the following results were obtained:

Table 3 - Performance of ΔV emulation

contig. nr.	train. pat.	test pat.	ANN topol.	Erms	Ema
1	1161	290	21-5-3-1	.013	.089
1	1161	290	21-4-2-1	.006	.004
2	1161	290	21-4-2-1	.024	.015
3	1034	326	21-4-3-1	.019	.034
4	1681	410	21-4-2-1	.031	.023

Table 4 presents some examples of the transient energy margins calculated directly and through the trained ANN for contingency number 1.

Table 4 - Transient energy margin values obtained with the ANN and with the transient energy function

Energy margin:		ϕ -ANN value;		ΔV - analytic value	
ϕ	ΔV	ϕ	ΔV	ϕ	ΔV
6.198	6.191	-0.019	-0.016	0.097	0.100
5.117	5.122	-0.116	-0.127	-6.248	-6.253
4.632	4.631	-0.897	-0.890	0.967	0.977
3.740	3.744	-1.125	-1.134	-5.387	-5.381
2.301	2.291	-2.032	-2.046	1.807	1.811
1.901	1.900	-3.918	-3.925	-4.839	-4.835

5.4. Preventive control actions

Several system unstable states for different contingencies were tested with the preventive control algorithm. An illustrative example is described next.

An initial state system was characterized by the measures indicated in table 5, namely the "Old P" values. If contingency 1 would occur, the system would become unstable, with $\phi = -2.087$. The values "New P" in this table show the corrected powers to be generated as a result of the preventive control algorithm, giving a new feasible stable solution. The new value of the transient energy function is now $\phi = 0.084$, assuring therefore the stability of the system. This result was confirmed by numerical step by step integration of the machine differential equations.

Again, with high quality results, one has a tool adequate for real time applications, due to the unique characteristics of ANNs.

Table 5 - NN inputs and results from preventive control

bus bar	1	2	3	4	5	6	7
E (pu)	1.16	1.20	1.10	1.08	1.12	1.12	1.07
M	.060	.021	.076	.121	.064	.068	.095
Old P (pu)	2.71	2.05	3.03	3.80	2.72	2.16	2.32
New P(pu)	2.53	2.10	3.08	3.85	2.78	2.20	2.33

6. CONCLUSIONS

Neural networks provide a simple, fast and reliable tool for power systems dynamic security classification, emulation and control. The results show that is possible to reach a classification performance very near 100% in many cases. In the emulation of the security function, ANN based methods can reach a very good level of approach to the referred function values.

Simple but effective ANN methods can also be implemented to develop preventive control measures. The algorithm presented here, for this purpose, shows a way of reaching system stable states from unstable ones. The results show that, in many cases, a very small change in the optimal economic dispatch is sufficient to provide a new stable state.

The main contributions of this work can be summarized

as follows:

- The ANN approach developed here is able to produce real time accurate measures of the degree of the system stability when using, as ANN inputs, controllable system variables available directly from the SCADA systems;
- Effective preventive control measures can be suggested on-line through generation load redispatch, when exploiting the possibility of getting, directly from the trained ANN, the sensitivities of the stability index relative to the input variables.

REFERENCES

- M. A. Pai, "Energy function analysis for power system stability", Kluwer Academic Publishers, 1989
- T. Athay, V. R. Sherkat, R. Podmore, S. Virmani, and C. Puech, "Transient stability analysis", Conf. on 'Systems Engineering for Power: Emergency Operating State Control - Section IV', Davos, Switzerland, 1980
- H. Hakimmashadi, "Fast Transient Security Assessment", Ph. D. Thesis, Purdue University, 1982
- J. A. Peças Lopes, F. Maciel Barbosa, and J. Marques de Sá, "Fast transient security assessment by pattern recognition techniques", Proc 9th PSCC, Lisboa, 1987
- M. A. Matos, J. A. Peças Lopes and F. Maciel Barbosa, "Fuzzy classification for on-line transient stability assessment of electric power systems", Proc 10th PSCC, Graz, 1990
- L. Wehenkel, T. Van. Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for on-line transient stability assessment of power systems", IEEE Trans. on PWRS, Vol. 4, nr. 2, May 1989
- D. J. Sobajic and Y. H. Pao, "Artificial neural-net based security assessment for electric power systems", IEEE Trans. on PWRS, Vol. 4, nr. 1, February 1989
- Y. H. Pao and D. J. Sobajic, "Combined use of unsupervised and supervised learning for dynamic security assessment", PICA' 91, Baltimore 1991
- M. Djukanovic, D. J. Sobajic and Y. H. Pao, "Neural net based determination of generator shedding requirements in electric power systems", IEE Proc. C - Vol 139, nr. 5, September 1992
- A. B. R. Kumar, A. Ipakchi, V. Brandwajn, M. El-Sharkawi and G. Cauley, "Neural network for dynamic security assessment of large-scale power systems: requirements overview", Proc. 1st International Forum on Applications of Neural Networks to Power Systems, Seattle, July 1991
- Rumelheart and MacClelland, "Parallel Distributed Processing", University of California, MIT Press, Cambridge, London, England, 1988
- Lipman, "An Introduction to computing with neural networks", IEEE ASSSP Magazine, April 1987
- R. Hecht Nielsen, "Neurocomputing", Addison Werley, 1990
- Mitsuo Kawato, "Feedback error-learning neural network for supervised motor learning", Neural Network for Sensory and Motor Systems, Mar. 1990, Amsterdam
- F. M. Silva, L. B. Almeida, "Acceleration Techniques for the Backpropagation Algorithm", in "Neural Networks", L. B. Almeida and C. J. Wellekens (eds.), Springer-Verlag, 1990.