Craig A. Jensen, Russell D. Reed, Mohamed A. El-Sharkawi, Robert J. Marks II,
"Location of Operating Points on the Dynamic Security Border Using Constrained Neural Network Inversion",
Proceedings of the International Conference on Intelligent Systems Applications to Power Systems (ISAP), pp.209-217, Seoul, Korea, July 6-10, 1997.

Location of Operating Points on the Dynamic Security Border Using Constrained Neural Network Inversion

Craig A. Jensen **Russell D. Reed** **Mohamed A. El-Sharkawi** **Robert J. Marks II**

Department of Electrical Engineering
University of Washington
Seattle, WA 98195

Abstract

A method is proposed in which a neural network trained to predict the dynamic security of a power system is inverted to provide information regarding nearby operating regions. Given an initial operating state, the nearest operating state of a given security status can be located. This information is useful in the everyday operation of a power system in that it outlines operating regions that should be avoided and provides information regarding operating regions that enhance system security. The proposed system is tested on the IEEE 17 generator transient stability test system.

Keywords: *Power system security assessment, constrained neural network inversion, query learning, dynamic security border.*

Introduction

The goal of dynamic security assessment (DSA) is to determine if a given power system operating configuration will be able to survive the transient period immediately following a contingency. Contingencies usually result from the unexpected loss of equipment such as transmission lines or generators. An insecure system can result in equipment damage, excessive voltage, current or frequency violations. In severe cases, a blackout or brownout may occur.

Presently, several methods are commonly used to predict the dynamic security of power systems including:

1) Time domain simulations
2) Energy function methods
3) Eigenvalue methods
4) Intelligent systems methods

Time domain simulations require the simultaneous solution of a set of algebraic and differential equations that describe the power system [1-2]. This method is accurate but its computational complexity and inability to determine a relative stability ranking of a given case limit its applicability.

In the energy function methods, the power system is described by its transient energy at fault clearing [3-4]. The magnitude of the transient energy is compared to a critical value and used to determine the system energy margin. The energy margin is a measure of the amount of excess energy that if stored in the system will cause instability. This method is less computationally demanding than the time domain methods, but less accurate.

Eigenvalue methods require the computation of eigenvalues of the linearized system equations [5-6]. These methods are accurate only for small disturbances and can be computationally demanding for large systems.

Intelligent systems methods include training neural networks to predict the stability of a power system. These systems require extensive off-line simulations to acquire a set of training data. After the system is trained, neural networks can provide extremely fast solutions. Properly trained neural networks have proven their ability to generalize from the training data and accurately classify the security of new power system operating conditions.

Presently, many of the above mentioned DSA methods are in use by several utilities in the day-to-day operation of their systems. These DSA systems can classify operating states based on their relative stability, but fail to offer any information regarding nearby operating regions. The system proposed in this paper not only has the ability to predict the dynamic stability of a power system operating state, but also provides insight into nearby operating regions. Specifically, this new system has the ability to determine the nearest operating state where the system could be unstable, thus informing operators of regions to avoid.

The proposed system starts with a standard multi-layer perceptron (MLP) neural network (NN) trained to predict the dynamic security characteristics of a given power system. Next, a border tracking algorithm is introduced that, given the current operating point, can locate the nearest point that satisfies a given stability criteria. An example might be to find the nearest unstable operating point to the current operating point. The border tracking algorithm is essentially a neural network inversion routine which conducts a constrained search of the input space of a trained neural network.

Neural networks for Security Assessment

Multi-layer perceptron neural networks have been successfully applied to the area of dynamic security assessment of power systems. In Pao, *et al.*, [7], a technique was proposed where a MLP was trained to predict the critical clearing time (CCT) for a fault based on the pre-fault system attributes, such as the acceleration powers and relative load angles of individual generators. The training patterns were generated for different load levels and base topologies via dynamic simulation of the test power system. It was shown that a neural network can generalize its knowledge and predict the critical clearing time of previously unseen topologies and load levels with reasonable accuracy. El-Sharkawi, *et al.*, [8], used a similar approach where the inputs to the neural network were the various real and reactive generator bus injections, and the output was the corresponding security status. The trained neural network was then used to determine various 2-dimensional security contours with respect to selected security attributes. British Columbia Hydro, used a similar approach but instead used their 'Second Kick' method to determine the stability ranking of each case [14]. El-Sharkawi and Huang [9], presented a query based method in which a partially trained neural network was inverted to produce additional training patterns that lie on or near the security border, thus improving the networks ability to classify a given operating condition as secure or insecure.

One of the greatest challenges in applying neural networks to dynamic security assessment is the selection of good training features. Several methods have been proposed to select good training features. Class-Mean-Separation [13] uses the notion of interclass distance to remove redundant features from a training set. Another proposed method is Karhunen-Loe′ve Expansion [10] which selects the most prominent set of features based on an eigenvalue approach.

Concept of Security Border/Contours

Neural network training data for DSA consists of a set of power system features and corresponding security rankings. The neural network is trained via the standard back-propagation algorithm to predict the security ranking. A rank of 0.0 is assigned to the least stable cases and 1.0 is assigned to the most stable cases. Marginally stable cases are assigned a rank of 0.5.

As a result of training, the network learns a mapping from the input features to the stability ranking. Contours of the map are operating states of equal stability ranking. The 0.5 contour is of interest because it corresponds to marginally stable operating states. This is defined here as the stability border. All contours with values above 0.5 are stable operating states, while contours with values less than 0.5 are unstable cases. Figure 1 shows a possible feature map of one generator in a power system. The input features in this case are the generator real power output and reactive power output. Several possible contours are shown.

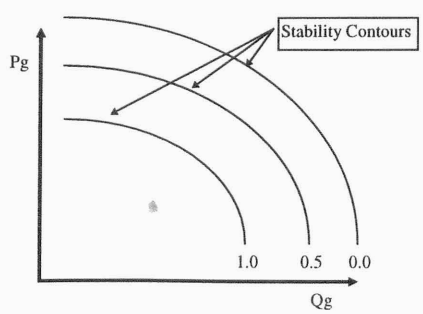

Figure 1 Feature Map of a Single Generator

The network mapping contains the information relating the input space to the output space. The idea is to query this map to extract useful information for the everyday operation of the power system. Ideally, the system would be able to answer questions such as:

- What is the nearest unstable operating point?
- What is the most stable operating configuration?
- Given recent trajectory, when is instability likely to occur?

Border Tracking Algorithm (BTA)

The neural network input-output mapping is a function $f(x)$. The stability border is the set of points x such that $f(x) = c$; in this case, $c=0.5$. Given an operating point x_o, the task is to find the nearest point on stability border.

Since $f(x)$ is differentiable, one possible approach is to use gradient information to reach the border from an initial search point at x_o. Gradient descent on a function such as $(f(x) - c)^2$ starting from x_o would give the same results. Gradient information is calculated by inverting the neural network [19] about the security border, i.e. $f^{1}(c)$. Some potential problems include; (1) If there are local minima, the point may never reach the border. (2) Since $f(x)$ is nonlinear, the gradient at x_o might not point in even the general direction of the nearest border point. (3) The resulting trajectory is likely to be a curved path so endpoint is unlikely to be the point on the border nearest to x_o.

The first two problems are addressed by doing repeated searches from different starting points around x_o. Initial points are obtained by adding noise to x_o with the variance chosen a significant fraction of the points fall on either side of the border. This ensures that the initial points sample a `region of interest' that contains both x_o and some part of the boundary. If $f(x)$ is not too badly behaved, most of the searches will reach some site on the border. The third problem is then addressed by doing constrained gradient descent on $| x_o - x |$ to move x towards x_o while staying on the surface.

Briefly, the algorithm is as follows. Generate random points x around x_o using a variance large enough so that a significant fraction lie on either side of the border. For each point, iterate:

1. If x is on the same side of the surface as x_o, follow the gradient to the surface.
2. If x is on the opposite side of the surface, use interval halving or a similar scheme to locate the surface.
3. Once x_i is on the surface, do constrained gradient descent on $| x_o - x |$ to move x towards x_o while staying on the surface.

Search for Operating Points on the Security Border

The border tracking algorithm conducts a constrained search of the input space to locate the operating state on the stability border that lies closest to the current operating point. To assure the feasibility of the new point, additional constraints must be imposed. These constraints insure that the new operating state does not result in any limit violations for the power system.

Additional constraints can be imposed via an iterative process whereby the border tracking algorithm is coupled with a standard power flow. The power flow is used to check for limit violations. If limit violations are discovered, the search algorithm will adjust the variables with violations and continue on the new path toward the border. The process repeats until an operating point is found that satisfies both the border tracking and feasibility search algorithms.

A graphical example of the iterations is shown in Figure 2. The initial search point corresponds to the current operating configuration. First, border tracking algorithm locates the nearest point on the security border. This point is not

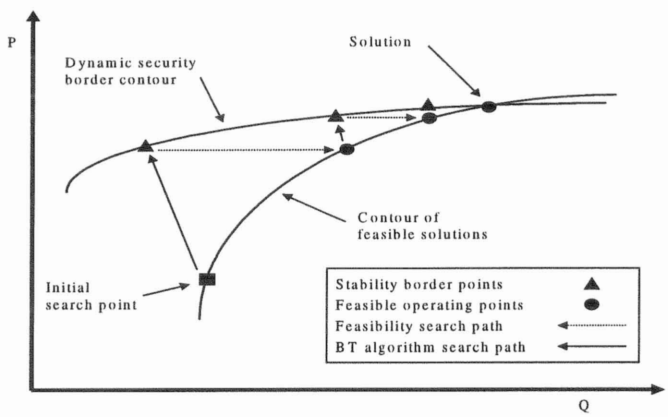

Figure 2 Graphical Representation of Solution

guaranteed to be feasible in the sense that it may contain limit violations or may not satisfy the power flow algorithm. Next, the feasibility search algorithm removes all limit violations and the process is repeated until the final point is both feasible and on the security border.

Query Learning

The overall accuracy of the proposed technique depends entirely on how well the neural network resembles the actual power system. If the neural network cannot accurately classify a given operating condition, the new points found by the inversion process will be meaningless. Query learning is used to enhance the accuracy of the NN inversion and hence the entire algorithm. Query-based learning [16] is the process of asking a partially trained neural network to respond to questions. The response of the NN is then compared with the answers computed by an oracle. The oracle has the ability to respond with the correct answer to a given question. Any questions the NN fails to answer correctly are then added to the training set and the network is retrained.

One of the key issues affecting the performance of the final NN is the proper selection of the training data set. A NN can only accurately respond to patterns that are similar to patterns in the training set. For this reason the training data must adequately cover all feasible regions of the input space. For applications where the input dimension is high, such as for power systems, it is not possible to collect enough training data to cover the entire input space. In such cases, a relatively small subset of the entire input space must be used to initially train the NN. A query learning procedure can then be used to reveal areas that are poorly learned [9]. For each poorly learned area, examples are added to the training set and the network is retrained.

A flowchart of the entire query learning process is shown in Figure 3. The query learning process begins with a partially trained neural network. A set of testing data are then generated and the border tracking algorithm is used to determine the nearest operating point that lies on the stability border for a given operating point. The actual stability ranking of the new operating state is then verified via dynamic simulation of the power system. Any points that have large errors represent areas in the input space that have not been properly learned by the NN. These data points are then added to the training data, the network is retrained, and the process repeats until the system produces acceptable results.

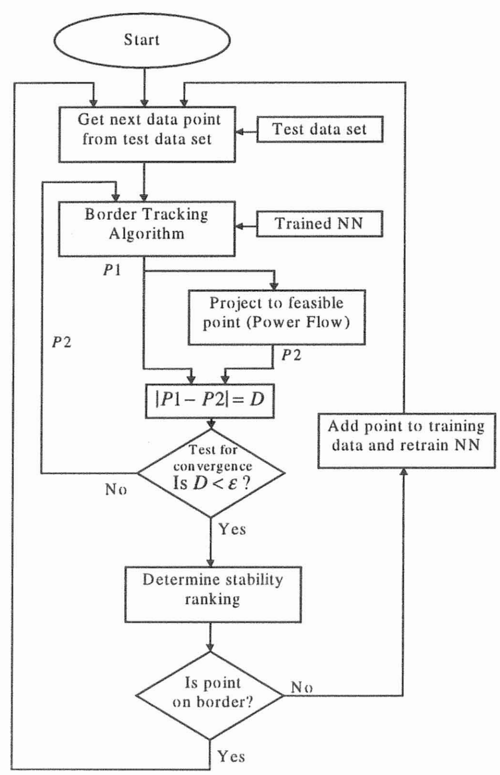

Figure 3 Flowchart of Query Learning

Case Study

System Description

The system studied is based on the IEEE 17-generator transient stability test system [12]. It consists of 17 generators and 162 buses. Classical machine data are available for each of the 17 generators. The data consists of the rotor inertia time constant and transient reactance. The first swing stability criterion is used to classify the system stability. The criterion monitors the rotor angle of each generator during the first oscillation. If any generator angle loses synchronism with the rest of the system, the system is said to be unstable.

Feature Selection

The trained neural network is queried to determine new power system operating points. Since the new points are used to govern the future operation of the system, the chosen features must satisfy the following requirements; 1) the features must define the power system, and 2) the features must correspond to power system variables that can be controlled by system operators. The features must fully describe the following; 1) system topology, 2) bus power injections, and 3) system control parameters such as relay settings, load shedding, etc. The use of dynamic features and non-controllable static features are thus rejected. BC Hydro reports that removing the dynamic features from the training data set has a very minor effect on the accuracy of the neural network [15].

The training features chosen are the real and reactive power outputs of each of the 17 generators and the total system load. These quantities fully define the system state and thus satisfy the above requirements. The fault critical clearing time (CCT) is used as the stability index. The critical clearing time is defined as the maximum time the fault can exist on the system before instability occurs. To determine the CCT, each system configuration was subjected to a series of faults of increasing duration until instability occurred. The faults ranged in duration from 0.25 to 0.425 seconds, resulting in 9 quantized stability ranges from 0.0 to 1.0 in 0.125 increments.

System Simulations

The Extended Transient/Mid-Term Stability Program (ETMSP) written by the Electric Power Research Institute (EPRI) is used to conduct the simulations [17]. This software can accurately simulate the dynamic response of very large scale power systems to any number of contingencies.

Table 1 Distribution of Training Patterns

CCT (Seconds)	Stability Classification	Number of Patterns
< 0.250	0.000	49
0.250 - 0.275	0.125	41
0.275 - 0.300	0.250	43
0.300 - 0.325	0.375	59
0.325 - 0.350	0.500	54
0.350 - 0.375	0.625	43
0.375 - 0.400	0.750	38
0.400 - 0.425	0.875	31
> 0.425	1.000	142

The test system was studied in response to a single contingency with a fixed system topology. A three-phase fault was placed at bus #75 and cleared by removing the line between bus 75 and bus 9 [12]. An initial data set of 500 patterns was created by varying the system loading and generation levels between 60% and 120% of their nominal values. The stability of each case was determined based on the first swing stability criterion. The data was divided into 400 training patterns and 100 test patterns and normalized before training the neural network. Table 1 shows the distribution of patterns in the training data set.

Neural Network Synthesis

A standard feed-forward neural network was trained with varying numbers of hidden neurons. It was determined through experimentation that a single hidden layer with 8 neurons gave the best results. It was noticed that the standard back-

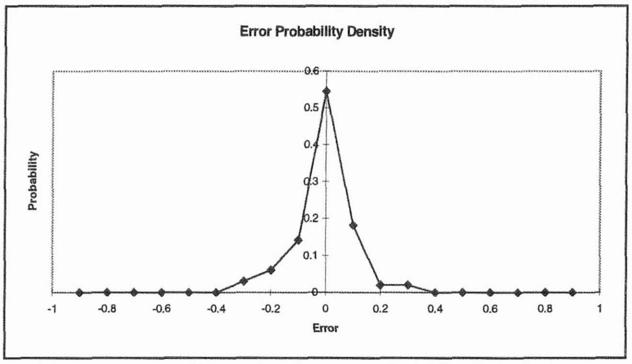

Figure 4 NN Testing Error Probability Density

propagation algorithm resulted in large maximum training errors (~50% error). To reduce these large errors, a new training algorithm was used that stresses learning on patterns with large errors. The new algorithm varies the learning rate, (*eta*), based on the magnitude of the error on a pattern by pattern basis. Patterns with large errors are assigned higher learning rates than patterns with small errors. A similar technique was used in [18]. The final neural network resulted in a maximum testing error of 0.296250 and RMS error of 0.102251. Figure 4 shows a plot of the error probability density function of the testing data file. The testing data contains patterns that have not been used in NN training.

Boundary Tracking

The border tracking algorithm was then tested with the 100 test patterns. It was found that the iterative process converged in 94/100 cases, the 6 failures were traced to power flow divergence problems. The 94 new points were then validated via ETMSP simulations. A 50% threshold was allowed for correctly classified patterns. Any patterns with an error greater than 50% were considered misclassified.

The validation of the initial neural network revealed an accuracy of 74.4% correctly classified and 25.6% misclassified. The incorrectly classified patterns were then added to the training data set and the neural network was re-trained. A new testing data set (100 patterns) was then generated and the query learning process was repeated. After 4 iterations the query learning process converged to an accuracy of approximately 98.7% correct classification and 1.3% misclassification. Table 2 shows the classification results for each iteration of query learning.

Table 2 Border Tracking Accuracy

Iteration	% Correctly Projected	% Incorrectly Projected
0	74.4	25.6
1	85.2	14.8
2	87.0	13.0
3	87.7	12.3
4	90.1	9.9
5	98.7	1.3

Conclusions

A technique is proposed to locate nearby operating states of a large scale power system. The technique starts with a NN trained to predict the dynamic security of a power system. Operational constraints are enforced to assure the feasibility of the final operating state. A query based learning procedure is used to overcome limitations due to insufficient data and enhance the overall accuracy of the system. Results are given for a test power system.

References

[1] G. W. Stagg, A. H. El-Abiad, *Computer Methods in Power System Analysis*, McGraw-Hill, New York, 1968.

[2] H. W. Dommel and N. Sato, "Fast Transient Stability Solutions," IEEE Transactions on Power Apparatus and Systems, vol. 91, July/August 1972, pp. 1643-1650.

[3] M. Ribbens-Pavella and F. J. Evans, "Direct Methods for Studying Dynamics of Large Scale Electric Power Systems," Automatica, vol. 21, no. 1, 1985, pp. 1-21.

[4] H. F. Williams, S. A. Louieand, G. W. Bills, "Feasibility of Liapunov Functions for the Stability Analysis of Electric Power Systems having up to 60 Generators," IEEE Transactions on Power Apparatus and Systems, vol. 91, May/April 1972, pp. 1145-1157.

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

[6] J. M. Undrill, "Dynamic Stability Calculations for an Arbitrary Number of Interconnected Synchronous Machines," IEEE Transactions on Power Apparatus and Systems, vol. 87, March 1968, pp. 835-844.

[7] D.J. Sobajic and Y. H. Pao, "Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems", IEEE Transactions on Power Systems, vol. 4, no. 1, February 1989, pp. 220-228.

[8] M. A. El-Sharkawi, R. J. Marks II, M. E. Aggoune, D. C. Park, M. J. Damborg and L. E. Atlas, "Dynamic Security Assessment of Power Systems using Back Error Propagation Artificial Neural Networks," Second Symposium on Expert System Application to Power Systems, Seattle, WA, 1989.

[9] M. A. El-Sharkawi and S. S. Huang, "Query-Based Learning Neural Network Approach to Power System Dynamic Security Assessment," International Symposium on Nonlinear Theory and Its Applications, Waikiki, Hawaii, December 5-10, 1993.

[10] S. Weerasooriya and M. A. El-Sharkawi, "Use of Karhunen-Loe've Expansion in Training Neural Networks for Static Security Assessment," First Symposium on Applications of Neural Networks to Power Systems, Seattle, WA, July 1991, pp. 59-71.

[11] R. D. Reed and R. J. Marks II, "An Evolutionary Algorithm for Function Inversion and Boundary Marking," IEEE International Conference on Evolutionary Computation, Perth, West Australia, December 1995.

[12] IEEE Committee Report, "Transient Stability Test Systems for Direct Stability Methods," IEEE Transactions on Power Systems, vol. 7, no. 1, February 1992, pp. 37-44.

[13] C. K. Pang, F. S. Prabhakara, A. H. El-Abiad and A. J. Koivo, "Security Evaluation in Power Systems Using Pattern Recognition," IEEE Transactions on PAS, Vol. 93, pp. 969-976, May/June, 1974.

[14] A. Vaahedi, et. al., "Enhanced "Second Kick" Methods for On-line Dynamic Security Assessment", IEEE Winter Meeting, Baltimore, MD, 1996.

[15] Yakout Mansour, Ebrahim Vaahedi, A.Y. Chang, B.R. Corns, Jeyant Tamby and M. A. El-Sharkawi, "Large Scale Dynamic Security Screening and Ranking Using Neural Networks," IEEE Transaction on Power Systems, in print

[16] J.N. Hwang, J.J. Choi, S. Oh and R. J. Marks II, "Query Based Learning Applied to Partially Trained Multilayer Perceptrons", IEEE Transactions on Neural Networks, Vol. 2, 1991, pp.131-136.

[17] EPRI, "Extended Transient Midterm Stability Program: Version 3.0", Vol. 1-6, EPRI TR-102004, Palo Alto, California, 1993.

[18] J. P. Cater, "Successfully Using Peak Learning Rates of 10 (and greater) in Back-Propagation Networks with Heuristic Learning Algorithm", Proceedings of the IEEE First International Conference on Neural Networks (San Diego), Vol. II, pp. 645-651, 1987.

[19] J. N. Hwang, J. J. Choi, S. Oh, and R. J. Marks II, "Classification Boundaries and Gradients of Trained Multilayer Perceptrons", IEEE International Symposium on Circuits and Systems, 1990, pp. 3256-3259.