Composite Security Boundary Visualization

Guozhong Zhou  James D. McCalley
gzhou@iastate.edu  jdm@iastate.edu
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011

Abstract—Based on a recently developed security assessment methodology using automatic training data generation software and neural networks, a composite security boundary visualization approach is proposed for power system operations. This approach can draw a single continuous boundary given multiple individual security problems. It can be conveniently accessed on-line and can be easily integrated into the energy management system. Examples are given for demonstration of the effectiveness of the proposed approach.

Keywords—Boundary visualization, security assessment, operations planning, power systems.

I. INTRODUCTION

In the open access environment, utilities face a number of challenges to be more competitive. One of these challenges is that typical operating conditions tend to be much closer to security boundaries. This is because transmission use is increasing in sudden and unpredictable directions, and competition together with other regulatory requirements make new transmission facility construction more difficult. Consequently, security assessment for the transmission network must be accurate and easily accessed on-line by system operators. A security boundary delineates acceptable and unacceptable postcontingency performance under a given contingency for the most restrictive credible contingency in terms of precontingency operating parameters. Obtaining clear, compact visual security boundaries is very useful for operators to quickly ascertain operating point proximity to insecurity.

Neural networks have been found to be an effective technique for on-line security assessment. A number of efforts have gone toward classification of the current operating point as secure or insecure under a specific contingency [1], [2], [3], [4]. Another approach, which we take, is to use neural networks in a function approximation of the security boundary. This approach is advantageous because it directly provides classification capability as well as boundary proximity information.

This paper considers how to visualize the composite boundary given multiple boundary characterizations, each of which corresponds to the constraint imposed by a different security problem. It includes the following sections.

Section II provides an overview of the traditional boundary characterization approach and the recently developed improved approach using automatic security assessment software and neural networks. Section III formulates the boundary visualization problem. Section IV describes a general automatic boundary visualization algorithm for an individual boundary. Section V discusses update functions used for boundary visualization. Section VI gives conceptual formulation and describes the developed algorithm for composite boundary visualization. Section VII shows examples, and Section VIII draws conclusions.

II. OVERVIEW OF TRADITIONAL AND AUTOMATIC BOUNDARY CHARACTERIZATION APPROACHES

A. Traditional Boundary Characterization Approach

The traditional boundary characterization approach results in a two-dimensional graph called a nomogram which is used on-line by operators [7], [8], [9]. The nomogram is represented graphically using two coordinate axes where each axis corresponds to a critical parameter. A critical parameter must be a precontingency operating parameter that can be monitored by the operator on-line such as voltage magnitude, real power flow, generation level or load level. A precontingency operating parameter is chosen as a critical parameter such that it is a good indicator of the postcontingency system performance for the specific security problem. From a power flow perspective, the critical parameters that are input variables to the power flow program such as generation or load levels are called independent critical parameters; those from the power flow solutions such as flows or voltage magnitudes at PQ buses are called dependent critical parameters. From a nomogram perspective, the critical parameters to be represented by the coordinate axes are called presented critical parameters; those not represented by the coordinate axes are called covered critical parameters. A precontingency operating parameter that is not a critical parameter and therefore not a good indicator of the postcontingency system performance for the specific security problem, is called a noncritical parameter.

A performance measure for the specific security problem and contingency is used to quantify the postcontingency performance level. Selection of the performance measure depends on the type of security problem under study. For example, current flow on the overloaded circuit is often used as the performance measure for a thermal overload...
problem, reactive power margin can be used for a voltage instability problem and damping ratio for an oscillatory problem. A performance measure for transient stability is not easily defined, but possibilities include energy margin [10] and others [11].

To develop a nomogram, two presented critical parameters are required. All of the covered critical parameters are set to selected values within a typical operating range. The noncritical parameters are set to constant values. Points on the nomogram curve are determined by repeating computer simulations, varying one presented critical parameter while keeping the other constant.

B. Automatic Boundary Characterisation Approach

We denote the critical parameters by vector \( \mathbf{x} \), the performance measure by \( R \), and the threshold value by \( R_0 \) that delineates between acceptable and unacceptable performance measures. For a given contingency and network configuration, the value of \( R \) is uniquely determined by the precontingency operating condition, i.e., all of the critical and noncritical parameters \( \Omega \). It is assumed, however, that \( R \) may be computed more compactly and with a good level of accuracy using only a small subset of \( \mathbf{x} \) taken from \( \Omega \), resulting in \( R = f(\mathbf{x}) \). Given that function \( f \) can be obtained and a threshold performance measure \( R_0 \) is specified beyond which system performance is unacceptable, the boundary associated with the contingency can be expressed as the set of feasible operating conditions which satisfies \( f(\mathbf{x}) - R_0 = 0 \). The function is shifted and normalized to force \( R_0 = 0 \) so that \( R < 0 \) denotes acceptable performance and \( R > 0 \) denotes unacceptable performance. The functional relationship between \( \mathbf{x} \) and \( R \) can be obtained by neural network training.

Compared with the traditional nomogram approach, the automatic boundary characterization approach provides more accurate boundary representation, reduces the labor burden, and can be more easily integrated into the energy management system (EMS) [6].

Fig. 1 shows the complete procedure to develop the automatic security boundary characterization. Steps 1 to 5 are performed off-line, but the final product in Step 6 is used on-line. Step 1 is no different from what has been done in the past for manually generated nomograms. It is the responsibility of the engineer to have a good understanding of the security constraints in order to choose candidate critical parameters that most influence the problem at hand. Baseline construction, Step 2, is performed in much the same way as the traditional approach. However, the need for a conservatively contracted baseline is unnecessary. Traditionally, labor requirements limit the engineer to selecting no more than about five parameters as the critical parameters. Other operating parameters of significant influence to the security problem were set to credible extreme values. With our procedure, however, the automatic procedure eliminates the guesswork involved in choosing these extremes and instead uses the entire range of operations. The security assessment simulations are performed in Step 3, data generation, using an automatic procedure. This procedure generates a large database, with each entry corresponding to a set of precontingency operating conditions and their respective performance measures. Step 4 uses the database generated in Step 3 and selects a subset of candidate critical parameters that best describe the postcontingency performance measure. Step 5 then uses this best critical parameter subset to train a neural network, which can in turn take a set of precontingency operating conditions and give the respective performance measure. Step 6 incorporates the trained neural network into the visualization software. This software is then used on-line to create a current system nomogram so that the operator in the control room can assess the operating conditions as secure or insecure and take the appropriate actions.

Since Step 6 is performed on-line, the boundary is continuously updated as operating conditions change. Because the neural network provides boundary characterization in terms of a relatively large number of critical parameters, the visualization software used in Step 6 is robust. It requires revision, and consequently repetition of Steps 1 to 5, only if unanticipated operating conditions or network configurations occur that deviate from the range of the training data set. Depending on the nature of the deviation, it may be possible to account for it very quickly by augmenting the original training data set and simply repeating Step 5, a procedure that would require only a few hours or less. For more substantive deviations, we would recreate an entirely new data set and neural network, repeating Steps 1 to 5, a procedure that can be done in one day or less, depending on how distributed are the computing resources as well as the nature and number of the security problems.

This paper deals with the last step, visualization.

Fig. 1. Steps for automatic boundary characterization.

III. PROBLEM FORMULATION FOR BOUNDARY VISUALIZATION

After the neural network is trained, the mapping between the critical parameters and the postcontingency performance measure is formed. This mapping function is used to draw a security boundary, i.e., all points on the boundary must satisfy this function. In addition, all points on the boundary must satisfy the power flow equations. Therefore, in drawing a security boundary, the following equa-
tions must be satisfied simultaneously

\[ f(x) - R_0 = 0 \]  
\[ h(u) = 0 \]

where (1) represents the mapping function, (2) is the power flow equations, \( x \) is the critical parameter vector, and \( u \) is the input parameter vector to the power flow program. The vector \( x \) may include both independent critical parameters and dependent critical parameters. A mapping exists between \( x \) and \( u \). If \( x \) does not include any dependent parameters, \( x \) is simply a subset of \( u \), i.e., \( x \subset u \). Otherwise, more generally, \( x \) is a function of \( u \).

In this paper, we use only independent critical parameters as presented critical parameters. In drawing a boundary in the space of the two presented critical parameters, we will search the values of one presented critical parameter that satisfy equation (1) for given values of the other presented critical parameter. However, we must ensure that the critical parameter vector \( x \) provided to the mapping function satisfies the power flow equations whenever we solve (1). There are two different approaches to solving this problem, depending on whether \( x \) contains any dependent critical parameters.

If \( x \) contains no dependent critical parameters, i.e., all of the critical parameters are independent critical parameters that would be part of the input parameters to the power flow program, then the boundary is drawn in the space of the presented critical parameters with all of the covered critical parameters fixed. In this case, the power flow equations are satisfied and do not need to be explicitly enforced because variations of the two presented critical parameters do not influence the covered critical parameters. If a covered critical parameter value changes, the boundary needs to be redrawn to account for this change.

If one or more critical parameters are dependent, we cannot fix their values in drawing the boundary. This is because their values are dependent on the independent critical parameters. Variations of the two presented critical parameters in drawing the boundary will influence the dependent critical parameters subject to the power flow equation constraints. We could account for this influence by introducing a full power flow solution into the visualization software; however, the increase in the computational cost would outweigh the improvement in accuracy. One simple way to model this dependency is to use appropriate update functions from linearization of power flow equations.

IV. INDIVIDUAL BOUNDARY VISUALIZATION

After the neural network is trained to compute \( R \) as a function of \( x \), the obtained mapping function equation (1) is used to visualize the boundary. Suppose that there are \( n \) independent critical parameters and \( l \) dependent critical parameters. Let the critical parameter vector be \( x = [x^T, y^T]^T \), \( z = [z_1, \ldots, z_n]^T \), \( y = [y_1, \ldots, y_l]^T \) where \( z \) is the independent critical parameter vector and \( y \) is the dependent critical parameter vector. Then the mapping function can be used to identify a two-dimensional boundary with axes of \( z_1 \) and \( z_2 \) where \( z_{1,\text{min}} \leq z_1 \leq z_{1,\text{max}}, \ z_{2,\text{min}} \leq z_2 \leq z_{2,\text{max}} \) and \( z_l (l = 3, \ldots, n) \) is constant. We assume in this algorithm that the boundary is monotonic with respect to \( z_2 \). This implies that given \( f(z_2) - R_0 = 0 \) may be solved uniquely for \( z_2 \). The algorithm is initialized from the current operating point \( x^{(0)} = (z^{(0)}, y^{(0)}) \). When drawing the boundary, values of \( z_1 \) and \( z_2 \) are varied. But variations of \( z_1 \) and \( z_2 \) must be used to update \( y \), which also represents neural network inputs. We assume that simple, approximate expressions may be developed to perform this update (see Section V). We denote these expressions as \( g_y(y^{(0)}, \Delta z_1, \Delta z_2) \), where \( \Delta z_1 \) is the change in \( z_1 \) from the value \( z_1^{(0)} \) from which the algorithm was initialized, i.e.,

\[ \Delta z_1 = z_1^{(k)} - z_1^{(0)} \]

so that

\[ y^{(k)} = g_y(y^{(0)}, \Delta z_1, \Delta z_2) \]  

Substitution of equation (3) into equation (1) allows the neural network mapping function to be expressed as a function of \( z \) only, i.e.,

\[ f(z^{(k)}, g_y(y^{(0)}, \Delta z_1, \Delta z_2)) - R_0 = 0 \]

Based on the analysis above, the basic procedure of the algorithm can be described. The algorithm starts from the minimum value of \( z_1 \), solves equation (4) for \( z_2 \), then increases \( z_1 \) by a step and updates \( y \) accordingly. This process is repeated until the maximum value of \( z_1 \) is reached. For more detailed description of the algorithm, see reference [6].

Use of dependent critical parameters depends on whether they can more compactly supply information compared with use of independent critical parameters alone and on whether simple update expression \( g_y \) can be developed. Approximations in update expressions \( g_y(y^{(0)}, \Delta z_1, \Delta z_2) \) may influence the boundary accuracy for portions of the boundary far away from the current operating point, since in this case, \( \Delta z_1 \) and \( \Delta z_2 \) are large. However, the accuracy of this portion of the boundary is of less importance unless the system is moving toward it, a situation that can be handled by frequent reinitialization.

V. OBTAINING UPDATE FUNCTIONS

As mentioned earlier, we model the influence of variations of the two presented critical parameters on dependent critical parameters by linearization of power flow equations. When the presented critical parameters are real power injections, we must define an allocation rule regarding how to compensate for changes in real power generation or demand. Suppose we have \( M_g \) load following generators. We define the allocation coefficients \( A_n \) as the percentage of the power imbalance caused by \( \Delta z_1 \) or \( \Delta z_2 \) that is redispached to load following generator \( m \). These factors may be obtained from economic dispatch base points and participation factors [12, pp. 44-46].
The update function is
\[
y^{(k)}_i = y^{(0)}_i + \sum_{m=1}^{M} [A_{m} K_{i,m1} (\Delta z_{1} + A_{m} K_{i,m2} (\Delta z_{2})]
\]
where the distribution coefficients \( K_{i,mj} \) may be computed using DC load flow approximations. They can also be obtained via sensitivity analysis using an AC power flow program where, for example, \( z_{1} \) is increased by \( \Delta z_{1} \), and this increase is compensated by load following generator \( m \) according to the allocation coefficients, and the change in \( \Delta y_{i} \) of flow on circuit \( i \) or voltage magnitude at PQ bus \( i \) is obtained (an increase is positive and a decrease is negative), so that \( K_{i,m1} = \Delta y_{i} / \Delta z_{1} \). \( b_{i,mj} = A_{m} K_{i,mj} (j = 1, 2) \) indicates the change in circuit flow or PQ bus voltage magnitude \( y_{i} \) due to the change in \( z_{j} \), which is compensated by load following generator \( m \).

VI. COMPOSITE BOUNDARY VISUALIZATION

A. Conceptual Formulation

Our goal is to draw a single continuous boundary, i.e., find a closed secure region, given several individual security boundary characterizations so that each portion of the composite boundary corresponds to the most restrictive security problem under the given contingency. Suppose we have \( M \) individual boundary characterizations that are represented as
\[
f_{i}(x_{i}) = R_{i}, \quad i = 1, \ldots, M
\]
A neural network is extracted for each of these problems and C-code is extracted that characterizes the functional relationship \( f_{i}(x_{i}) = R_{i} \) between critical parameter \( x_{i} \) and the performance measure \( R_{i} \) specific to problem \( i \). In general, \( x_{i} \neq x_{j} \) if \( i \neq j \). However, they have at least one common critical parameter, otherwise there is no reason to use them together to draw a composite boundary. For the purpose of illustration of the developed automatic security boundary characterization approach, we assume that there are two common critical parameters denoted by \( z_{1} \) and \( z_{2} \).

In order to obtain a continuous composite boundary (or equivalently, a closed secure region), we add following four boundaries:
\[
\begin{align*}
z_{1} - z_{1,min} &= R_{M+1} \\
z_{1} - z_{1,max} &= R_{M+2} \\
z_{2} - z_{2,min} &= R_{M+3} \\
z_{2} - z_{2,max} &= R_{M+4}
\end{align*}
\]
These boundaries form a rectangle enclosing the region of interest in the study and correspond to the bounds on the presented parameters. Fig. 2 shows a composite boundary formed by various individual boundaries. Clearly, the problem is to identify and draw only the portions of the boundaries enclosing the shaded region. Suppose \( R_{i} < 0 \) corresponds to the secure operating conditions and \( R_{i} > 0 \) to insecure conditions. The problem is then equivalent to finding the region \( S \) in which for every point \( s(z_{1}, z_{2}) \in S \)
\[
f_{i}(x_{i}) < 0, \quad \forall \ i = 1, \ldots, M
\]

B. Algorithm Description

As shown in Fig. 2, some points on some individual boundaries do not contribute to forming the composite boundary. To reduce computational cost, we desire to avoid drawing these portions of the boundaries. Therefore, starting from the minimum value of \( z_{1} \), for each interval, as in Fig. 3, we first identify the two or more individual boundary functions that are binding. To do this, we rank the functions in descending order of \( z_{2} \), which is obtained by solving
\[
f_{j}(z_{1}, z_{2}, \ldots) = 0, \quad j = 1, \ldots, M
\]
using a bisection search technique. Neighboring boundary functions are paired according to which one follows the other in this ranking. For each pair of neighboring functions, we perform an interior check for security, i.e., we check an arbitrarily selected point (marked with crosses) between two neighboring boundaries to see if it satisfies (7). If so, this point is inside the secure region, and the corresponding neighboring individual boundary functions must be the binding functions for the composite boundary for this interval. The composite boundary is therefore identified as the pair of individual boundaries, and all other individual boundaries are ignored in this interval.

If, in the next interval, there are no other individual boundary functions between the two binding functions identified in the previous interval, then these functions are also binding for the new interval. In this case, it is not necessary to perform the interior check for this interval. This is why there are no crosses for some intervals in Fig. 3. Once it is no longer possible to find any secure point, i.e., there are no neighboring individual boundary functions for which (7) is satisfied, then \( z_{1} \) has exceeded its feasible or secure range, and the algorithm stops. At this point, a secure region is formed and a continuous boundary is obtained, as in Fig. 3. This algorithm is applicable to both
linear and nonlinear individual boundaries as long as they form one single continuous region. Cases where individual boundaries form several disjoint regions are not considered because they are rare in practice.

![Fig. 3. Illustration of the algorithm to determine the composite boundary.](image)

### VII. Examples

A sample system is shown in Fig. 4. This is a simplified illustration of the model used in our work, a 2500-bus model of Northern California. The load in the subarea, during high loading conditions, is greater than the generation capacity in the subarea. Therefore, a significant amount of power must be imported into the subarea to meet the demand. There are several ties between the subarea and the remaining part of the system. Based on experience and knowledge of the system, we know that operation of generators A, B, and C under high subarea loadings is constrained by several different security problems.

A **Line thermal overload** occurs on tie line 5 when tie line 3 is outaged under certain conditions. The postcontingency performance measure for this problem is the flow in amperes on tie line 5. A **transformer overload** also may occur in the subarea of this system, and the performance measure is the flow of the transformer. In addition, the outage of tie line 3 can cause a **voltage instability** problem in the subarea. One load bus in the subarea has been identified as the most reactive deficient bus. The performance measure is MVAr margin at this load bus. A secure operating point on the boundary is defined as one for which this bus has at least 200 MVAr of reactive margin following loss of tie line 3. The performance measure is MVAr margin at this load bus.

The critical parameters for each of the three problems are chosen by feature selection software [13]. They are indicated in Tables I and II together with their values corresponding to the initial operating point. They are load, generations, line flows, bus voltage magnitudes, and available reactive power supplies. There are three available reactive power supplies corresponding to three different groups of generators, \( Q_{max1} \), \( Q_{max2} \), and \( Q_{max3} \), each of which is the sum of the reactive capacities for the committed generators in that group. Therefore, these values change with the unit commitment status in each group. We can also see that there are four common critical parameters for the three problems. Fig. 5 - 7 show the boundaries for the

![Fig. 4. A sample system](image)
instability problems, respectively. The corresponding composite boundary for all of the three problems is shown in Fig. 8. The area enclosed by the composite boundary indicates the secure operating region. Note that only a part of each individual boundary is used to obtain the composite boundary. As boundaries are different for different initial operating conditions, the initial operating conditions for these examples are shown in Tables I and II. The accuracy for the line thermal overload problem is between 2.5% - 5.5% of the threshold level of current flow [6]. The boundaries tested for the transformer overload and the voltage instability problems have similar accuracy levels.

VIII. Conclusions

We have developed an automatic boundary characterization procedure that presents accurate boundaries to operators. It is more efficient and more flexible compared with the manual approach. In our approach, we draw a continuous composite boundary based on the automatic individual boundary characterizations. The approach can be used on-line by operators for system operations. Since it can be applied to any type of security problem, it provides a general way to integrate boundaries corresponding to problems of different types.

ACKNOWLEDGEMENTS

This work was supported by Pacific Gas and Electric Company.

REFERENCES


Biographies

Guozhong Zhou received the B.S. and M.S. degrees in electrical engineering from Tianjin University, Tianjin, P. R. China, in 1985 and 1988, respectively. He worked as a researcher at Tianjin University from 1988 to 1992, as an engineer at Northeast China Electric Power Company from 1992 to 1994, and as a visiting researcher at the University of Porto in Portugal from 1994 to 1995. He is currently working toward his Ph.D. degree at Iowa State University.

James D. McCalley is Assistant Professor in Electrical and Computer Engineering at Iowa State University, where he has been employed since 1992. He worked for Pacific Gas and Electric Company from 1986 to 1990. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Georgia Institute of Technology, in 1982, 1986, and 1992, respectively. Dr. McCalley is a registered professional engineer in California and a member of the IEEE.