Chapter 6
Online Dynamic Security Assessment

Jorge L. Jardim

Abstract This chapter describes an implementation of an online dynamic security assessment system based on time-domain simulation, in which dynamic models are the same as those used offline for planning studies with all necessary details. It contains a review of the adopted methods and algorithms (power flow, continuation power flow, time-domain simulation, energy functions, single machine equivalent, and Prony spectral decomposition) focusing on the main issues related to their numerical and computational performance. The utilization of these methods to execute complex security tasks such as dynamic contingency analysis and security region computations is described. Aspects of high-performance computation (fine- and coarse-grain parallelization) are discussed. Some practical results obtained online and comparisons of online with offline planning cases are shown.

6.1 Introduction

Power system security limits are typically computed offline and stored as nomograms and tables to be monitored by system operators in the real-time environment. Several uncertainties exit during such computations and consequently reasonable security margins must be taken into account in the final limits. Despite this, unplanned outages may cause operational conditions not considered at planning stages and consequently system operators are left with no proper security information under such circumstances. Online security assessment has been proposed as an additional line of defense in which security limits can be computed based on the actual power system state, which eliminates most of the uncertainties, thus providing more accurate limits (Debs and Benson 1975; Dy Liacco 1968; Hayashi 1969; Limmer 1966). Of course, this approach assumes that reasonably accurate online data are available.

Power system security may, and typically does, require evaluation of several different aspects such as thermal limits, steady-state and transient voltage levels, transient stability, etc. The steady-state aspects are generally evaluated through

J. L. Jardim (✉)
High Performance Power System Applications, Rio de Janeiro, Brazil
e-mail: jljardim@live.com

S. C. Savulescu (ed.), *Real-Time Stability in Power Systems,*
Power Electronics and Power Systems, DOI 10.1007/978-3-319-06680-6_6,

power flow contingency analysis, which, for the currently available computational resources, can be easily done online, even for very large networks. The transient aspects are traditionally evaluated by visual inspection of time-domain simulation trajectories. In this case, two major difficulties exist in doing it online. One is that the computational costs of these simulations are normally several orders higher than those of steady-state analysis, i.e., the problem presents a much higher computational complexity, in particular for large systems. Therefore, efficient simulation methods and parallel processing are required. The other is that the evaluation cannot depend on visual inspection as typically done in offline studies. However, this difficulty can be circumvented with post-processing algorithms to derive the essential information from monitored time-simulation trajectories.

Basically, two main lines of research have been proposed in the literature and applied in the industry to deal with the computational complexity. One is to simplify the problem by using reduced network and/or dynamic models and computing stability indices based on faster calculations. In some cases, heuristics are used to further simplify the problem. Such indices are not expected to be quite accurate, but just provide a degree of proximity to transient instability. Several different approaches have been developed in this line, such as Transient Energy Functions (TEF; Pai 1989), single machine equivalent (Pavella et al. 2000), and steady-state stability indices (Molina et al. 2009).

The other line of research is to use full time-domain simulations with detailed models, efficient simulation algorithms, and parallel processing (Jardim 2000; Jardim et al. 2004; Jardim et al. 2006; Jardim 2009). Detailed models are used at least for the main area of interest, but when necessary external network and dynamic equivalents can be used. Some advantages of this kind of approach are the following:

- Close compatibility with limits computed offline
- Quite accurate assessment
- Not only transient stability but also all other dynamic security aspects can be evaluated; and
- Easier validation of online assessments

The online security assessment can be done for the (real-time) operating point only or additionally for a region around this point. An operating point is said to be secure if no predefined contingency causes violation of the security criteria. Time-domain simulations can tell whether the system is transiently stable or not, but do not provide a quantifiable degree of stability/instability. For example, if the system is stable, how close is it to be unstable? Additional methods, such as energy function methods, can be embedded in the time simulation to estimate stability margins and answer these types of questions. These estimates cannot be very accurate due to the nonlinear nature of models and phenomena involved. On the other hand, by successively stressing the system in a particular fashion and reprocessing contingencies, it is possible to obtain more accurate limits for such conditions, but at the expense of much more computation. In practice, this is the approach used offline for computing security limits (nomograms).

The main disadvantage of time-domain simulations, in particular when detailed models are used, is the inherently high computational cost. Therefore, the adoption of quite efficient numerical integration methods can make a huge difference in performance for this type of application.

This chapter presents the main aspects of an online Static And Dynamic Security Assessment System (SDSA) that is based on time-domain simulation with detailed models and is able to calculate security regions (nomograms). Essentially, it is an automation of offline procedures, and therefore its relevant characteristics are in the details of the adopted algorithms, the automation process, and the methods used to verify all security aspects. SDSA has been online for several years in the control centers of the Brazilian National Operator (ONS).

6.2 Simulation Methods

Whenever there is a numerical failure in offline simulations, the analyst can normally circumvent the problem by examining its results, changing parameters, and trying it again. For example, if a power flow does not converge, it is possible to change the starting procedure, change or block specific controls, etc. In the case of time-domain simulation with fixed time step, reducing the step or changing the load characteristics during a fast transient may overcome a numerical instability.

As online processes are not supervised by an analyst, they demand careful choice of the numerical methods and of procedures for circumventing failures. In this section, the traditional simulation methods are revisited with focus on aspects to improve their performance and to avoid numerical problems. Nevertheless, it is always important to remember that despite the importance of the methods and their implementation, any online power system application based on detail modeling depends fundamentally on good data and models, not only for the accuracy of the results but also for the software performance.

6.2.1 Power Flow

Power flow calculation is the most basic method used in a security assessment system. For online SDSA, the power flow is used for computing the initial system condition with reasonable accuracy and steady-state contingency analysis. Theoretically speaking, the solved power flow case retrieved from the Energy Management System (EMS) should be fully converged and ready to be used by the security assessment; in practice, though, this may not be true due to differences in tolerance and/or inaccuracies that might have been introduced by data truncation, depending on the data exchange format. Also, and more important, the power flow calculation is the engine for steady-state contingency analysis.

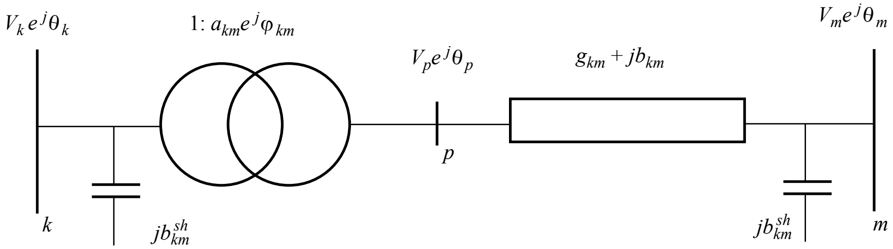


Fig. 6.1 Generic branch model

The power flow formulation is well known (Stott 1974), largely described in textbooks, e. g., Monticelli (1999) and is summarized in the following.

A network branch can be modeled as in Fig. 6.1, where for a transmission line $a_{km} = 1$ and $\varphi_{km} = 0$, for an in-phase transformer $\varphi_{km} = 0$, and $b_{km}^{sh} = 0$ for transformers.

The general power flow equations for this generic branch are given by

$$P_{km} = a_{km}^2 V_k^2 g_{km} - a_{km} V_k V_m (g_{km} \cos(\theta_{km} - \varphi_{km}) + b_{km} \sin(\theta_{km} + \varphi_{km})) \tag{6.1}$$

$$Q_{km} = -a_{km}^2 V_k^2 \left(b_{km} + b_{km}^{sh}\right) + a_{km} V_k V_m (b_{km} \cos(\theta_{km} + \varphi_{km}) - g_{km} \sin(\theta_{km} + \varphi_{km})) \tag{6.2}$$

and

$$P_{mk} = V_m^2 g_{km} - a_{km} V_k V_m (g_{km} \cos(\theta_{km} + \varphi_{km}) - b_{km} \sin(\theta_{km} + \varphi_{km})) \tag{6.3}$$

$$Q_{mk} = -V_k^2 \left(b_{km} + b_{km}^{sh}\right) + a_{km} V_k V_m (b_{km} \cos(\theta_{km} + \varphi_{km}) + g_{km} \sin(\theta_{km} + \varphi_{km})). \tag{6.4}$$

The power flow formulation is obtained by applying Kirchhoff's nodal law to all nodes of the network, which results in the following reciprocal power equations:

$$0 = -Pg_k + Pl_k + \sum_{m\ \Omega_k} P_{km} \tag{6.5}$$

$$0 = -Qg_k + Ql_k + \sum_{m\ \Omega_k} Q_{km} \tag{6.6}$$

for $k = 1, \ldots, n$, where n is the number of buses in the system

Pg_k and Qg_k are respectively the active and reactive generation at bus k, and Pl_k and Ql_k are the active and reactive load at bus k, respectively. Loads can be constant power or voltage dependent.

The solution of the set of equations (6.5 and 6.6) requires at least one reference voltage and one reference angle, which are normally set at a specific bus, called swing or slack bus. Considering, for example, that the injected power for all remaining buses is known, the problem consists of solving a set of $2n-2$ nonlinear equations with $2n-2$ variables (V_i,θ_i), typically by the Newton–Raphson method. By the nature of the problem, the load imbalance (generation—load—losses) is allocated to the slack bus. Therefore, for some uses of power flow calculation, such as in contingency analysis, it may be desirable to distribute the imbalance among some or all generators in the system.

Generators, switchable shunts, FACTS, and Tap Changing Under Load (TCUL) transformer taps are set to control terminal or remote buses, and phase shifters and FACTS can be set to control power flow. Possible formulations for each control are the suppression of one variable, making it constant, and one equation. Alternatively, an equation specifying the control logic and the respective control variable can be added. Also, additional equations must be included when multiple devices (e. g., generators) control the same variable (e. g., bus voltage) to define the participation of each control and avoid multiple solutions. For example, if generators at buses i and j control the same bus voltage, an additional equation establishing how they will participate in the control can be defined as

$$K_i Q_i - K_j Q_j = 0,$$

where Q and K are the generator MVAr output and participation factor.

Of course, control limits must be enforced and it is imperative that they are correctly specified. Controls must be able to back-off limits whenever possible.

DC Links are represented by additional power injections in (6.5, 6.6; Smed et al. 1991).

The set of equations (6.5, 6.6) and additional control equations can be represented in a simplified form as:

$$0 = f(x), \tag{6.7}$$

where $f(x)$ is a vector function of independent variables x (typically, V, θ, a, ϕ).

The Newton method solves iteratively (6.7) approximated by a first-order truncated Taylor series, i.e.,

$$f(x^{\nu} + \Delta x^{\nu}) \cong f(x^{\nu}) + f'(x^{\nu})\Delta x^{\nu}, \tag{6.8}$$

where $f'(x)$ is the Jacobian matrix and v is the iteration number.

At a solution point $\Delta x = 0$ and $f(x) = 0$ which leads to the following iterative process:

$$\Delta x^{\nu} = -\left[J(x^{\nu})\right]^{-1} f(x^{\nu}) \tag{6.9}$$

$$x^{\nu+1} = x^{\nu} + \alpha\Delta x^{\nu} \tag{6.10}$$

until $\Delta x^{\nu} < \varepsilon$ or $f(x) < \varepsilon$, where ε is a small tolerance.

Although the power flow problem is relatively simple, some practical considerations must be observed to minimize the possibility of failure, as follows:

- The full Newton–Raphson method, in which all variables are solved simultaneously, is the preferred solution algorithm because of its better convergence properties. The alternate approach where some variables (e.g., taps and phase shifts) are solved between iterations of the Newton method generates interface errors that can be amplified or sustained, thus preventing convergence. The fast decoupled algorithms were very attractive with respect to computational speed when computer power was quite limited, but, due to its inferior convergence properties when compared with the full Newton, unless for specific applications, there is no reason to adopt it today. Again, reliable algorithms should be the primary concern in the implementation of online applications
- It is critical to scale the update by a factor α to improve convergence, as shown in (6.10). For ill-conditioned cases, some of the elements in Δx can be very high, thus violating the assumption in (6.8), i.e., that Δx is small. Of course, for well-behaved situations α can be set to 1 without causing any problem. Several methods have been proposed to find a suitable α, including one-dimension minimization methods (Braz et al. 2000)
- It is well known that the Newton–Raphson method has a very good local convergence, but may fail for initial conditions far from the solution. This should not be critical for online base cases as long as the state estimator is able to deliver a converged power flow. For offline applications, a cold start procedure, such as DC power flow combined or not with one iteration of the fast decoupled method can be quite useful
- It is important to remember that a solution to the power flow problem may not be, and normally is not, unique. To make a long discussion around this subject short and assuming the problem can be solved, if only continuous controls and constant power loads are represented, there should be one useful solution and several nonrealistic ones, but if dead-band controls are represented there may be several useful solutions. The possibility of finding nonrealistic solutions also emphasizes the need of starting close to the solution. Of course, the concern is related to unrealistic solutions that may yield false security violations. For online applications, starting too far away from the solution can be a problem in contingency analysis and when re-dispatching generation to stress the operating condition. Thus, it is useful to implement methods to move smoothly between two different operating points. The continuation power flow method is quite effective in dealing with this problem and should be the preferred choice whenever possible
- A problem frequently observed in solved power flow cases consists of conflicting or wrongly defined controls. There are several situations in which this may happen. A few examples are the following: (a) two parallel tap-controlled transformers controlling voltages at opposite sides; (b) a voltage control device controlling the voltage of a very remote bus or a bus in a different electrical island; (c) two voltage control devices controlling the same bus, but at different voltage levels; (d) a tap controlled transformer with no voltage source device on the low voltage side trying to control the voltage of a high voltage side bus; etc. Such situations may occur because of data errors or due to a topological network

change caused by a contingency. Then, it is important to implement routines to verify and correct or disable controls, if necessary, at the data input level and during the iterative process.

6.3 Continuation Power Flow

For the reasons explained in the previous section, moving as smooth as possible between different power flow solutions helps power flow convergence and avoids convergence to unrealistic solutions. The Continuation Power Flow (CPF) method is very effective and efficient for this, and it is used in SDSA for the process of searching security boundaries, i.e., stressing the system pre-contingency operating condition in a given direction. Also, although it is not the rule, there may be situations in which the security boundary is not affected by the specified contingencies, but instead it is found on the pre-contingency case. For these cases, for example, the CPF can provide the maximum loadability point with good accuracy whereas the regular power flow is likely to fail at lower stress/load level.

The efficiency of the CPF method comes from the fact that it allows larger steps when moving the operating point in a particular direction. Also, the prediction phase, with the cost of one power flow iteration, saves more than one iteration in the corrector phase.

The tangent vector method approach (Ajjarapu and Christy 1991; Seydel 1994) is adopted in SDSA. It consists of two phases, linked through a continuation parameter. In the first phase, called *predictor,* the power flow equations are parameterized and sensitivities of the power flow variables with respect to the parameters are computed. These sensitivities are used to estimate a new operating point, given a uniform change in the parameters. Then, in the second phase, called *corrector,* the Newton–Raphson method is used to find the solution with a good accuracy, considering one of the variables, called *continuation parameter,* fixed. The predictor–corrector cycle is repeated until the desired solution (target operating point or maximum loadability) is obtained.

6.3.1 Tangent Vector Method

The following description of the method assumes that generations and loads will be re-dispatched according to a given pattern (direction) defined by changing factors. The parameterization is set as

$$\begin{aligned} Pg &= Pg_0 + \lambda\, Kp_g \\ Pl &= Pl_0 + \lambda\, Kp_l \\ Qg &= Qg_0 + \lambda\, Kq_g \\ Ql &= Ql_0 + \lambda\, Kq_l \end{aligned},$$

where λ is the generation/load increment variable applied to all buses of the system, Pg_0, Pl_0, Qg_0, *and* Ql_0 are the generation and load values at the initial operating point, and Kp_g, Kp_l, Kq_g, and Kq_l are the generation/load change factor or parameters, defined for each bus in the system.

Considering $\lambda = 0$, the set of nonlinear equations system defined in (6.7) becomes:

$$f(x, \lambda) = 0. \tag{6.11}$$

Linearizing (6.11) at a solution, one gets

$$\begin{aligned} f(x+\Delta x, \lambda + \Delta\,\lambda) &\cong f(x,\lambda) + f'(x,\lambda)(\Delta x, \Delta\lambda)^T \\ &= f'(x,\lambda)(\Delta x, \Delta\lambda)^T = 0 \end{aligned} \quad . \tag{6.12}$$

From (6.12), the sensitivity of the state variables with respect to λ can be computed and then prediction of these variables for a step increase/decrease of λ can be calculated.

The predicted values, supposedly close to the final solution are used as initial condition for a slightly modified power flow calculation. In this process, one of the variables (called continuation parameter) in (6.11) is kept constant. The predictor–corrector cycle is repeated until the solution (target point or maximum loadability) is obtained.

The continuation parameter may be the increment variable, lambda, or the voltage at one bus. The decision is based on which one has the highest sensitivity, which is computed in predictor phase. Figure 6.2 illustrates the process. In practice, voltage is selected only near maximum loadability points (nose tip).

The first task in the prediction process is to calculate the tangent vector. This tangent calculation is derived by the augmented Jacobian matrix, which has one extra column, associated with the additional unknown variable lambda (λ). One additional equation must be added to match the number of variables. This can be done by considering the sensitivity of the variables to a step change (increase or decrease) in the increment variable, i.e.

$$\partial\lambda = \pm 1.$$

The set of equations then becomes

$$\begin{pmatrix} f'(x,\lambda) \\ e_\lambda \end{pmatrix} \begin{pmatrix} \partial x \\ \partial\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \pm 1 \end{pmatrix}, \tag{6.13}$$

Where e_λ is a row vector with all elements equal to zero except in λ position, which is one.

Once the (sensitivity) tangent vector $t = (\partial x, \partial\lambda)$ has been found, the step size should be chosen so that the predicted solution is within the radius of convergence of the corrector. A possible choice is the inverse of the norm of the tangent vector, as follows:

$$\begin{pmatrix} \Delta x \\ \Delta\lambda \end{pmatrix} = \alpha \left\| t \right\|^{-1} \begin{pmatrix} \partial x \\ \partial\lambda \end{pmatrix}, \tag{6.14}$$

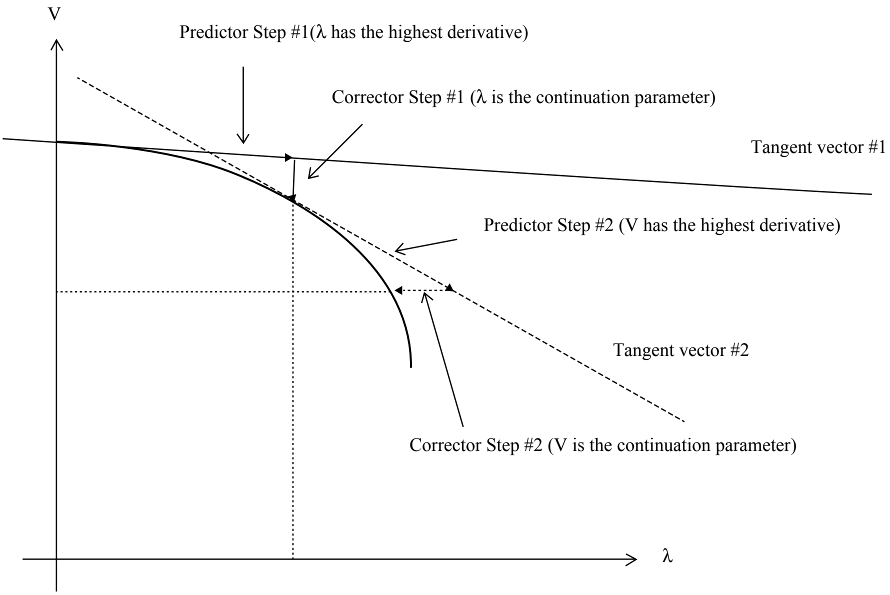

V
Predictor Step #1(λ has the highest derivative)
Corrector Step #1 (λ is the continuation parameter)
Tangent vector #1
Predictor Step #2 (V has the highest derivative)
Tangent vector #2
Corrector Step #2 (V is the continuation parameter)
λ

Fig. 6.2 Example of two predictor–corrector cycles with different continuation parameters

where α can be adjusted to reach specific points, or guarantee that the initial condition will be sufficiently close to the solution.

The corrector step solves (6.11) using the Newton power flow method, but forcing the continuation parameter at the specified value. The continuation parameter can be either voltage at a specific bus or lambda. Considering that η is the specified value for the continuation parameter (V_k or λ) the new system to be solved can be expressed as:

$$\begin{bmatrix} f(x,\lambda) \\ V_k - \eta \end{bmatrix} = [0]$$

or

$$\begin{bmatrix} f(x,\lambda) \\ \lambda - \eta \end{bmatrix} = [0].$$

6.4 Time-Domain Simulation

The most computationally expensive task in a detailed modeling approach for dynamic security assessment is that performed by the time-domain simulation engine. Consequently, overall performance is very much affected by the numerical integration methods adopted.

Most of the existing time-domain simulation programs use numerical integration based on fixed time step and alternate solution of control and network equations (Arrillaga et al. 1983). This kind of technology was acceptable, and possibly a good choice, for computers with low memory, nonstiff system of equations (only synchronous machines with relatively large time constants), and offline studies performed much ahead of real time. For those cases, computational speed was not a critical factor.

However, adapting these programs for online applications is not a good choice now because computer memory is not a limitation anymore, power electronic devices are increasingly being represented in dynamic studies, which stiffen equations, security assessment requires huge amount of fast computation, and there are much better and well-recognized methods for numerical integration.

Despite the fact that more advanced methods have been recommended for quite some time (Stott 1979), the majority of the commercial software still have not adopted them.

The time-domain simulation in SDSA is based on the Adams–Bashforth–Moulton and Backward Differentiation Formulas (ABM–BDF) numerical integration method (Astic et al. 1994, Brenan et al. 1989, Lambert 1991) associated with the Variable-Step-Variable-Order (VSVO) approach and the simultaneous solution of the algebraic and differential equations.

These techniques yield high numerical stability and improved performance (several times faster) compared with more traditional nonsimultaneous fixed-time-step approaches. The shortcomings of the latter are explained as follows. To avoid numerical instability (Lambert 1991), the fixed step alternated solution methods have to use very small integration time steps. Roughly speaking, their step size should not be greater than the smallest time constant in the dynamic models. But under stiff numerical conditions, even smaller time steps may be required to avoid numerical instability. Obviously, the impact on the performance is severe, particularly when fast acting control devices such as static VAr compensators or DC links need to be represented.

By contrast, in a simultaneous solution approach, the size of the time step is bounded by the accuracy of the simulation rather than its numerical stability. In practical terms, the desired accuracy can be met with small-time steps during fast transients and larger time steps on smoother trajectories. The time step size is optimized by a dynamic adjustment mechanism as described below.

The differential and algebraic equations describing a power system model are represented by the following equations:

$$\dot{y} = f(y, x, t) \tag{6.15}$$

$$0 = g(y, x, t), \tag{6.16}$$

where:
$y \in R^n$ is the vector of the state variables (or phase variables) that represent the dynamic models of control components such as synchronous machines, voltage regulators, DC links, etc.

$x \in R^m$ is the vector of algebraic variables, which are basically network voltages, current injections, and some control variables.
The ABM and BDF can be represented as

$$\sum_{i=0}^{j} \alpha_i y_{n+i} = h \sum_{i=0}^{j} \beta_i f_{n+i}, \tag{6.17}$$

where:
α_i and β_i are parameters dependent on the specific integration method, j is the number of steps of the method, and h is the time step.

Using the appropriate parameters, the first-order ABM and BDF methods correspond to the Euler method:

$$y_{n+1} = y_n + hf_n \qquad \text{(predictor)} \tag{6.18}$$

$$y_{n+1} = y_n + hf_{n+1} \qquad \text{(corrector)} \tag{6.19}$$

The second order ABM is usually known as the Trapezoidal method:

$$y_{n+1} = y_n + 0.5h(3f_n - f_{n-1}) \qquad \text{(predictor)} \tag{6.20}$$

$$y_{n+1} = y_n + 0.5h(f_{n+1} + f_n) \qquad \text{(corrector)} \tag{6.21}$$

The second order BDF is given by

$$y_{n+1} = 2y_n - 3y_{n-1} + y_{n-2} \qquad \text{(predictor)} \tag{6.22}$$

$$y_{n+1} = \frac{4}{3} y_n - \frac{1}{3} y_{n-1} + \frac{2}{3} f_{n+1} \qquad \text{(corrector)} \tag{6.23}$$

The ABM method is used for most of the differential equations, whereas the BDF method is used for algebraic equations and differential equations with very small time constants (< 10 ms). For improved efficiency, the current and past information are stored in Nordsieck (Lambert 1991) vector form.

By applying the numerical integration formulae (6.18–6.23) to the model equations (6.15–6.16), the following set of algebraic equations is obtained:

$$0 = y_{n+1} - \beta_{n+1} hf(y_{n+1}, x_{n+1}) - C, \tag{6.24}$$

$$0 = g(y_{n+1}, x_{n+1}) \tag{6.25}$$

where:

C is the weight sum of y_n and $\dot{y}_n$ terms

β_{n+1} is the constant that multiplies f_{n+1} in the integration formulae (6.18–6.23).

The solution of this set of equations is obtained by a "dishonest Newton method," in which the Jacobian matrix is updated only if: there is a time step change, or the algorithm does not converge, or if a large hard discontinuity occurs. Typically, two to three iterations are needed to converge at a time step.

The mechanism to change the time step is based on the estimation of the Local Truncation Error (LTE) at the end of each time step. If the LTE is smaller than the required tolerance, the current step is accepted and the possibility of increasing its size is evaluated. If the LTE is above the tolerance, the current step is rejected and the step size is sufficiently reduced to bring the error to half of the tolerance.

The LTE estimation is based on the first neglected term of the Taylor series:

$$E_k = \frac{h^{k+1} y^{k+1}}{k+1},$$

where k is the current integration order (1 or 2). The maximum time step $\bar{h}$ is calculated by considering the truncation error equal to the tolerance

$$\bar{h} \approx h \left[\frac{\tau}{E_k} \right]^{\frac{1}{k+1}},$$

where τ is the tolerance. Considering that there is no error margin in this estimation, a conservative approach is adopted for the next step, say half of the estimated value.

Whenever the time step is changed, the best order is also evaluated. The criterion to choose the best order is the decreasing pattern of the truncated Taylor terms. Thus, the Taylor series expansion behaves as expected for the second order if the magnitude of the third-order terms form a decreasing sequence. Otherwise, first-order integration is used (Brenan et al. 1989).

A difficult problem in time-domain simulation is the treatment of discontinuities. The one-step (self-starting) methods can handle this better, but are generally less efficient as they require smaller time steps when compared to higher order methods. The multistep methods need step size and order changes to deal effectively with such situations. Re-initialization with first-order integration is one of the possible approaches.

This implementation deals with discontinuities in different ways depending on their types. The main sources of discontinuities are network and control switching and state variable nonlinearity. Switching operations can be specified by the users (e.g., in the contingency definition) or are automatically activated by controls such as excitation limiters, or protection systems such as line tripping. Depending on the severity of the switching operation, the program re-initializes the numerical integration process by zeroing the time step. The severity is measured by the norm of the first-order derivative of the state variables. The re-initialization of the integration process nullifies the past (previous steps) information and the integration order is set to one. If necessary, the time step is reduced so that a switching operation occurs at the specified time.

The severity of the effect of nonlinearity on state variables, such integrator saturation, is taken into account at the end of the step. If the LTE is greater than the tolerance, the step is rejected and decreased.

6.5 Diagnostic Methods

Assessing security through time-domain simulation without visual inspection of trajectories requires specific functions for monitoring the security criteria. Some of the most frequently adopted criteria include transient stability (or stability margin), electromechanical oscillation damping, transient voltage behavior, and frequency limits. Computing stability margins is not trivial and typically requires energy function methods. Computing oscillation damping also requires special algorithms (e.g., Prony analysis), but it is a simpler problem. Practically, all the other criteria can be assessed by trivial procedures.

The classical TEF methods are based on simplified dynamic models (e.g., classical synchronous generator models) with the purpose of estimating stability margins. For SDSA, which is based on detailed modeling approach, a much more suitable technique, single machine equivalent (SIME; Pavella et al. 2000) is adopted, as it does not impose restriction on the models, can be easily embedded in time-domain simulation programs, and has a negligible computational cost. But some key concepts of the TEF methodology (Pai 1989) are adopted in SDSA to detect instability and early terminate simulations, support some of the SIME features, and select generators for modal spectral analysis.

6.5.1 Energy Functions

SDSA uses numerical energy functions and a modified version of the SIME method for energy/power margin computation, instability detection, and identification of oscillatory machines.

System stability can be detected via the following dot product (Pai 1989):

$$f_{ip} = P_{ac}^{T}\Delta\theta, \tag{6.26}$$

where:

$$P_{ac}^{T} = (pac_1, pac_2, \ldots, pac_{ng})$$
$$\Delta\theta = (\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_{ng})$$
$$\theta_i = \delta_i - \theta_{coi}$$
$$\theta_{coi} = \frac{1}{M_t}\sum_{i=1}^{ng} M_i \delta_i$$
$$pac_i - pm_i - pe_i - \frac{M_i}{M_t} pcoi_i \;\; i = 1,2,\ldots,ng ,$$
$$pcoi = \sum_{i=1}^{ng} (pm_i - pe_i)$$
$$M_i = \sum_{i=1}^{ng} M_i$$

where M_i, δ_i, pe_i, pm_i, and pac_i are, respectively, the inertia constant, rotor angle, electrical output, mechanical power, and accelerating power of machine *i*; *ng* is the number of synchronous generators; θi is rotor angle of machine *i* referred to the center of inertia θ_{coi}; and P_{ac} and $\Delta\theta$ are the vectors of generator accelerating power and angle deviation respectively. Both quantities are referred to the center of inertia.

For classical synchronous machine models, system instability is detected when $f_{ip} < 0$. For higher-order synchronous machine models, a lower level is used, i.e., $f_{ip} < \tau$, $\tau < 0$.

Individual energy functions are also computed to determine machines with low damping. The potential energy function is given by

$$Vpe_i = \int pac_i d\,\theta,$$

the kinetic energy function by

$$Vke_i = \frac{1}{2} M_i \omega_i^2 ,$$

and the total energy is

$$Vt_i = Vpe_i + Vke_i .$$

The rate of decay of Vt_i indicates those machines with lower damping. These are selected for Prony decomposition analysis.

6.5.2 Modified SIME Method

The SIME method can be used to estimate security margins at a given operating point. In the following implementation, it can also be used for contingency filtering. A brief review of the SIME method (Pavella et al. 2000) in both the original and the modified versions is presented as follows.

6.5.3 Original SIME Method

It is well known that a power system transient instability is caused initially by the separation of only two generation areas. Certainly, cascading effects may lead to further separations, but the interest is obviously to avoid the initial separation, which is caused by a power imbalance in which generators of one area accelerate (or decelerate) with respect to the others. This leads to the concept of two coherent groups of generators, denominated critical and noncritical clusters. By definition here, the critical cluster is the one with smaller inertia. If the critical cluster accelerates with respect to the noncritical cluster, it is said that it swings *forward,* if it decelerates then it swings *backward.*

The critical cluster is composed of the generators with angle increasing (decreasing) with respect of the Center Of Angle (COA) of an electrical island, if it swings forward (backward). It is important to know whether the critical cluster mode is forward or backward because the corrective measure will be to reduce or to increase respectively its generation.

The determination of the critical cluster can be done by selecting several candidate sets and testing for the one with lowest margin. The original SIME method proposes the use of the most advanced angles (large angle excursions) to classify machines in candidate sets. Rotor speeds are also a good measure for this classification. In SDSA, and for stable cases, the identification of critical clusters occurs only when the total kinetic energy reaches a minimum, which correspond to a point of maximum separation between critical and noncritical clusters.

Given the critical and noncritical groups, the respective machines are aggregated into their respective COA and then the COAs are replaced by a One-Machine Infinite Bus (OMIB) equivalent, as follows.

Compute the quantities of the aggregated groups

$$\delta_c(t) = \frac{1}{M_c} \sum_{k \in C} M_k \, \delta_k(t) \tag{6.27}$$

$$\delta_N(t) = \frac{1}{M_N} \sum_{j \in N} M_j \, \delta_j(t) \tag{6.28}$$

$$M_C = \sum_{k \in C} M_k \tag{6.29}$$

$$M_N = \sum_{k \in N} M_k \tag{6.30}$$

$$\omega_C(t) = \frac{1}{M_C} \sum_{k \in C} M_k \, \omega_k(t) \tag{6.31}$$

$$\omega_N(t) = \frac{1}{M_N} \sum_{j \in N} M_j \, \omega_j(t) \tag{6.32}$$

$$Pe_C(t) = \frac{1}{M_C} \sum_{k \in C} Pe_k(t) \tag{6.33}$$

$$Pm_C(t) = \frac{1}{M_C} \sum_{k \in C} Pm_k(t) \tag{6.34}$$

$$Pe_N(t) = \frac{1}{M_N} \sum_{j \in N} Pe_j(t) \tag{6.35}$$

$$Pm_N(t) = \frac{1}{M_N} \sum_{j \in N} Pm_j(t), \tag{6.36}$$

where the subscript *C* denotes the group of critical machines and *N* the noncritical machines.

Compute the equivalent OMIB

$$\delta(t) = \delta_C(t) - \delta_N(t) \tag{6.37}$$

$$\omega(t) = \omega_C(t) - \omega_N(t) \tag{6.38}$$

$$Pm(t) = M(Pm_C(t) - Pm_N(t)) \tag{6.39}$$

$$Pe(t) = M(Pe_C(t) - Pe_N(t)) \tag{6.40}$$

$$Pa(t) = Pm(t) - Pe(t) \tag{6.41}$$

$$M = \frac{M_C \times M_N}{M_C + M_N} \tag{6.42}$$

Equations (6.37–6.42) represent the mapping of a multi-machine system into an OMIB system, which allow us to apply the principle of the Equal Area Criterion (EAC).

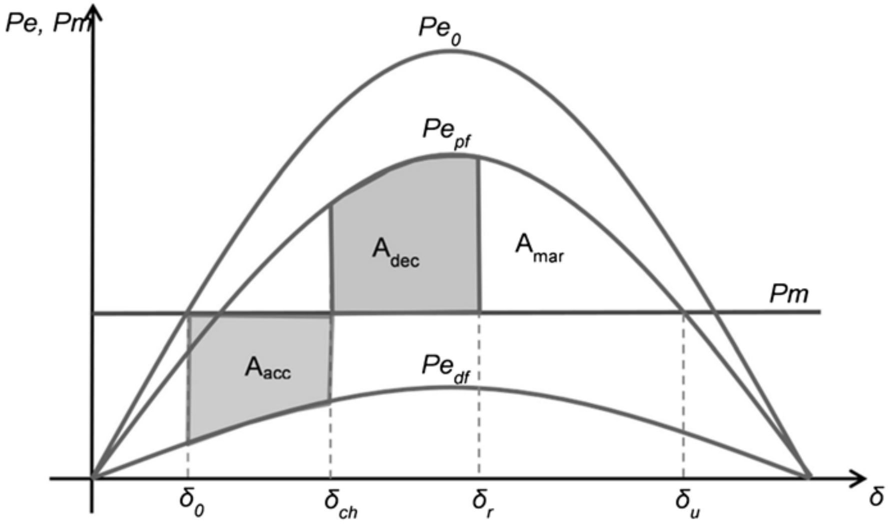


Fig. 6.3 Equal area criterion

Figure 6.3 illustrates the EAC concept, where Pe_0, Pe_{df}, and Pe_{pf} are the OMIB power transfer characteristics for pre-fault, during the fault, and post-fault conditions, respectively, δ_0 is the pre-fault rotor angle, δ_{ch} is the post-fault rotor angle in which the accelerating power changes from positive to negative, δ_r is the angle of return, i.e., the maximum angular excursion for a stable scenario, δ_u is the unstable equilibrium point, A_{acc} is the acceleration area ($Pm > Pe$), A_{dec} is de deceleration area ($Pe > Pm$), and A_{mar} is the margin area, i.e., the *energy margin* of the system for the particular fault. To simplify the analysis, *Pm* is considered constant. The sum ($A_{dec} + A_{mar}$) is the total potential energy available to absorb the kinetic energy introduced into the system by the fault. Computation of A_{mar} requires that the function of Pe_{pf} versus angle is known (or estimated with good accuracy).

During the fault, the machine accelerates because the accelerating power ($Pac = Pm - Pe$) is positive. At the point in which the accelerating power becomes negative (most of the time it is the fault clearing time), the machine speed is maximum and it starts decelerating.

The total energy gained by the system can be determined as the kinetic energy at this point, as follows:

$$A_{acc} = \frac{1}{2} M \, \omega_{ch}^2 \tag{6.43}$$

The condition for the system to be transiently stable is that the decelerating area A_{dec} must be greater than A_{acc}. In other words, the angle of return δ_r must be smaller than the unstable angle δ_u.

Instability is detected by the crossing of the unstable equilibrium point, which is characterized by the accelerating power changing from negative to positive and by the derivative of the angle being positive. At this point, the remaining energy in the system, not dissipated in the decelerating area, is the negative energy margin and can be accurately computed by

$$\eta_u = \frac{1}{2} M \omega_u^2, \tag{6.44}$$

where ω_u is the equivalent machine speed at the crossing point.

A key aspect on the accuracy of the SIME method is the computation of the positive margin A_{mar} for stable cases. In Pavella et al. (2000), two methods are proposed to estimate this margin. The first approach consists of the triangle approximation, formulated as:

$$A_{mar} = \frac{1}{2}(Pe_{pf}(\delta_r) - Pm)(\delta_u - \delta_r). \tag{6.45}$$

Obviously, this requires the knowledge of the unstable angle δ_u, but this angle is not known if the system is stable. In practice, repeated simulations with increasing stress are necessary to find the unstable angle. This makes the triangle approximation inefficient and of little interest.

The other suggested approach is to approximate the $Pe_{pf}(\delta)$ as follows:

$$Pe(\delta) = A\delta^2 + B\delta + C, \tag{6.46}$$

where a, b, and c are computed through weighted least-square approximation using three or more successive time steps. In practice, $Pe_{pf}(\delta)$ is not as well behaved as in Fig. 6.1 and, for stable scenarios with large margins, the points may not be representative of the real characteristic. Additionally, if the case is quite stable the angle excursion is small and the available points for the curve fitting may be insufficient. Consequently, large errors can occur. Again, to use this method effectively it would be necessary to run successive simulations with increasing stress level to find a good approximation, which is again quite inefficient.

6.5.4 Modified SIME Method

An improved, meaning faster and more accurate, method for computing positive margins is crucial to the SIME approach in order to obtain a reliable and quick assessment of the system stability. The objective is to be able to estimate the energy margin (positive or negative) a few milliseconds, for example, 200 ms, after the fault is cleared.

The central idea is to approximate the $Pe_{pf}(\delta)$ characteristic by the power transfer function of the OMIB system:

$$Pe(\delta) = \frac{E_m(\delta)E_\infty}{X_e}\sin\delta + P_0, \tag{6.47}$$

where E_m is the equivalent machine voltage behind its transient reactance, which is modeled as a function of the rotor angle, E_∞ is the infinite bus voltage, which is assumed to be constant, δ is the equivalent machine rotor angle, and P_0 represents a local power referred to the OMIB equivalent. To use this approximation, E_m, E_∞, and X_e need to be estimated. If this is possible, P_0 can be calculated to fit the equation at a particular point.

Remark The approximation $Pe(\delta) = P_{12}\sin\delta + P_0$ was also tried, where P_{12} is a constant, and P_{12} and P_0 are computed using values at successive time steps. But this leads to the same problems of the weight least-square approximation of (6.46). Also considering P_{12} constant is a source of error as generator excitation can change significantly from nonstressed to stressed scenarios.

$E_m(\delta)$ is estimated in the adopted approach as the average of the critical cluster voltages behind the transient reactance:

$$E_m(\delta) = \frac{1}{nc}\sum_{k\in C} E_k(\delta). \tag{6.48}$$

where nc is the number of generators in the critical cluster. E_∞ is estimated in the same fashion for the noncritical cluster, but in the tests performed so far, it has been estimated at the returning angle and left constant, i.e., it is not considered as a function of the angle displacement.

Xe is estimated in the proposed approach as the weighted average of the external impedance seen by each generator plus its own transient reactance.

$$Xe = \frac{1}{M_C}\sum_{k\in C} M_k(xe_k + xd_k'), \tag{6.49}$$

where xd_k' and xe_k are the transient reactance and the external impedance seen by generator k, respectively. Thus, the missing piece of information to complete the proposed approximation is the external impedance seen by each generator.

One way of finding this information is by the explicit computation of the Thévenin impedance seen by each generator in the cluster, considering the other generators of the cluster as open circuit, but depending on the number of generators in the critical cluster, this computation can be quite expensive. Then the following alternative approach was used.

Assuming that the interconnection between the critical and noncritical clusters is through a reactance and that the machines in the cluster oscillate coherently, one can write

$$\bar{V}_{mk}(t) = j(xe_k)\bar{I}_k(t) + \bar{E}_\infty \tag{6.50}$$

where $\bar{V}_{mk}(t)$ and xe_k are the terminal voltage of and external impedance seen by machine *k,* respectively. Considering also that $\bar{E}_\infty$ is constant (infinite bus), one can write

$$|xe_k| = \left| \frac{\frac{d\bar{V}_{mk}(t)}{dt}}{\frac{d\bar{I}_k(t)}{dt}} \right| \tag{6.51}$$

Remark The approximations in this model are quite reasonable compared with the overall approximation of the SIME model. Generally, there is no infinite bus, but in a multi-machine system for a single machine the rest of the system typically behaves as an infinite bus. Also, the interconnection between critical and noncritical clusters is not purely reactive but typically the resistive component is relatively small.

The external impedance seen from each individual machine xe_k can be computed at any post-fault time step, but in the current implementation it is being computed as an average over a time range.

Note that the external impedance can be theoretically estimated immediately after the fault is cleared. Consequently, the entire decelerating area can be readily estimated. The maximum kinetic energy (accelerating area) is known, as mentioned above, as soon as the accelerating power changes from positive to negative.

For not very stressed conditions, this occurs at fault clearance; for stressed conditions, it can take a few milliseconds after fault clearance, and for very stressed conditions it may not happen at all. But this last case can be easily flagged as a severe condition without requiring too long time simulation. Then for the other two conditions, the energy margin (total decelerating area minus the accelerating area) can be estimated at most a few milliseconds after fault clearance, which results in a very fast approach for contingency ranking and simulation early termination.

The form of implementation of the SIME algorithm depends on the purpose of its use. For example, for contingency screening it is desirable to estimate the stability margin just few milliseconds after fault clearance, but for the diagnostic of a full time-domain simulation the urgency is not needed and the estimation can be more conveniently performed at returning angles or instability detection.

The following implementation strategy was used for diagnosis of time-domain simulation:

1. Start the time-domain simulation.
2. For the post-fault system condition, check for instability or angle of return (minimum kinetic energy) at each time step.
3. If instability is detected, determine the critical cluster and the negative energy margin and stop the simulation.
4. If a point of return is found, determine the critical cluster and compute the positive energy margin.

For contingency ranking, the strategy used is the following:

1. Start the time-domain simulation.
2. If the system is in post-fault condition and instability is detected, compute the negative energy margin and stop the simulation.
3. If the system has been simulated for a minimum time interval (few milliseconds) in post-fault condition and the kinetic energy has reached a maximum, compute the positive (or negative) energy margin and stop the simulation.

6.5.5 Prony Analysis

The Prony method (Castanié 2011; Hauer 1991) is used for spectral analysis (damping assessment) of synchronous machine angle trajectories. The objective is to compute the following spectral decomposition for a given signal, say rotor angle, $\hat{y}(t) = \hat{\delta}(t)$:

$$y(t) = \sum_{i=1}^{n} R_i e^{\lambda_i t} \tag{6.52}$$

or in the discrete time

$$\begin{aligned} \hat{y}_k &= \sum_{i=1}^{n} R_i z_i^k \\ z_i &= e^{\alpha_i \tau} \quad , \\ t &= k\tau \end{aligned} \tag{6.53}$$

where $y(t)$ approximates $\hat{y}(t)$, $R_i \in C$ is the residue for pole $\lambda_i \in C$. The objective is to identify residues, poles, and the order n of the model to minimize the least square of $y(t)$.

Assuming that $y(k)$ can be described by a combination of n past values

$$y(k) = a_1 y(k-1) + a_2(k-2) + \ldots + a_n y(k-n) \tag{6.54}$$

Let the set of sample vectors $\{\overline{y}_i\}, i = 1, \ldots, n+1$, where

$$\overline{y}_i = [y(i), y(i+1), \ldots, y(i+N-n-1)]^T . \tag{6.55}$$

By repeatedly applying (6.28), the following system of linear equations is formed:

$$\begin{aligned} Y\overline{a} &= [\overline{y}_n, \overline{y}_{n-1} \cdots \overline{y}_1]\overline{a} = \overline{y}_{n+1} \\ \overline{a} &= [a_1, a_2, \ldots, a_n]^T \end{aligned} \tag{6.56}$$

Y is a Toeplitz matrix that can be solved with 2*n* data samples. From (6.53) and (6.54), and considering *k=n*, $\{z_i\}$ represent the roots of the characteristic polynomial

$$C(z) = z^n - a_1 z^{n-1} \ldots - a_n = 0 and \lambda_i = \ln(z_i) / \tau. \tag{6.57}$$

Then the residues $\{R_i\}$ can be found by

$$Z\bar{R} = \bar{y}_i, \tag{6.58}$$

where

$$Z = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \vdots & \vdots & \vdots & \vdots \\ z_1^{N-1} & z_2^{N-1} & \cdots & z_n^{N-1} \end{bmatrix} \text{ and } R = \left[R_1, R_2, \ldots, R_n\right]^T.$$

The algorithm can be summarized as follows:

1. Given a sampled signal y_i, $i=0, 1, \ldots, m$, determine the coefficients a_i of the characteristic polynomial by fitting a linear prediction model.
2. Using the a_i's, compute the discrete-time eigenvalues by solving for the roots of the characteristic polynomial z_i's.
3. Compute the R_i's by solving (6.58).
4. Calculate the continuous time eigenvalues λ_i's using $z_i = e^{\lambda_i T}$.

The cost of computing this spectral decomposition to all generator rotor angles is very high and must be avoided. In SDSA, just the least damped rotor angles are selected based on the rate of decay of the machine total energy.

6.6 Security Functions

The automation of any security assessment process is dependent on the operation planning practices of the relevant utility or system operator entity. In the present SDSA case, three basic assessment functions apply:

- Contingency analysis at the operating point (is the operating point secure?)
- Maximum transfer (interchange) between two subsystems
- Maximum security region or transfer nomogram for three subsystems

In addition, functions are provided for recommending preventive actions on the basis of such assessments. Preventive action is needed when one or more contingencies will bring the system state to a condition in which at least one security criterion is violated. This action, such as generation re-dispatch, moves the system state to a new secure operating point.

6.6.1 Contingency Analysis at an Operating Point

This is the most basic component of the SDSA system. Contingency analysis is normally performed for both the steady and dynamic states.

In steady-state contingency analysis, the critical aspect is the robustness of the power flow engine. For example, if a solvable contingency fails to converge, the interpretation might be that the operating point is beyond the maximum loadability limit. This can wrongly trigger an uneconomical preventive re-dispatch. Therefore, the care with respect to the implementation of power flow solutions, as pointed out in Sect. 6.2 must be observed.

In dynamic contingency analysis, a stability diagnosis may not be produced at all if the numerical integration algorithm fails to converge. Again, it is critical to avoid numerical problems, in particular when fast controls are represented in the model. The simultaneous solution of differential and algebraic equations associated with variable time step is key to avoid this problem, as discussed in Sect. 6.4.

In addition, the ability of changing the time step has two fundamental advantages. One of course is to increase it, whenever possible, to speed up the simulation. Thus the computation, depending on the stiffness of the equations, can be one or two orders faster. The other advantage is to reduce the time step below regular values whenever a difficult numerical condition occurs. This can significantly slow down the simulation, but prevents the worst scenario, i.e., the computation failure.

If the number of contingencies is large, screening methods can be applied to improve performance. For steady-state analysis, this is usually not necessary for today's computer power. For dynamic analysis, the key technique is to terminate as early as possible those simulations that are estimated to be "quite" stable. As presented in Sect. 6.5, SDSA adopts early termination for unstable cases and provides stability indices based on the modified SIME method that allows filtering contingencies on the fly.

However, it is important to remember that most, if not all, of the screening methods proposed so far deal only with the transient stability aspect. It is then assumed that if the system is "quite" stable, it should comply with all other criteria. This is hard to prove, of course, and care should be taken. In practice, power system analysts know the set of contingencies that can cause any harm to the system. Today's operational planning is based on such knowledge. This may not be possible in the steady-state analysis where contingencies all over the network can cause some security violation, such as thermal or voltage limit violation. On the other hand, the dynamic problems are invariably associated with transmission bottlenecks (relatively weak interconnections), with location well known by planners. These locations can change over time as system topology changes, but it is still possible to predefine effective contingency sets. Thus, the "blind" approach, typically advocated and used for steady-state contingency screening, should be avoided or not used at all in online dynamic security assessments because it is an unnecessary waste of computer resources and the proposed techniques do not cover all security criteria.

6.6.2 Import–Export Transfer Capacity

This SDSA function is useful for assessing the maximum secure transfer between two interconnected areas. It basically consists of performing contingency analysis at successively increasing/decreasing power transfer levels between two areas. The pre-contingency operating point at which a security criterion becomes violated defines the maximum transfer capacity for that specific criterion. To identify this transfer level with relatively good accuracy, a binary search is used.

To speed up this search, specific quantities (voltages, flows, etc.) and indices (MW stability margin, damping, etc.) are stored along the one-dimensional search and are used to estimate the violation point by interpolation and extrapolation. The search stops when two consecutive estimated points are sufficiently close to each other.

The changes in transfer level are effected by re-dispatching generation in the exporting and importing areas. All other generation and loads in the system remain constant. Then, neglecting changes in losses, the security region per criterion is defined by sets of points (line segments) belonging to the line $Pa+Pb=K$, where Pa and Pb are the respective generations in each area, and K is constant. The direction of search from the operating point is given by $\Delta Pa+\Delta Pb=0$.

This function demands much more computation than the single contingency analysis at the operating point as the contingencies will be simulated at various operating points. On the other hand, the number of contingencies simulated for this kind of function is typically small, since only those that affect the transfer between the two areas are of interest. The function is useful for monitoring critical transmission corridors and it automatically provides the security margins for the current operating point, which is very desirable information at real time.

The operating point change in the direction of search is implemented with the CPF, Sect. 6.3. For steady-state transfer capacity assessment, the contingencies are computed by the Newton power flow with care to restart and approach the solution "slowly" if the first direct attempt fails. For dynamic transfer capacity assessment, contingencies are simulated with the methods presented in Sect. 6.4.

6.6.3 Security Regions

Situations arise where transfer limits are highly dependent on the generation patterns in three areas. Therefore, re-dispatching generation in only two of them may provide inaccurate transfer limit estimates. At this point, the objective is to find secure regions in the two-dimensional surface defined by $Pa+Pb+Pc=K$, where Pa, Pb, and Pc are the respective generations in the three areas and K is constant, if changes in losses are neglected. This surface is embedded in three-dimensional space and it is bounded by the generation limits in each area. Alternatively, it is also possible to replace one of the generations (*Pa*, *Pb*, or *Pc*) by a load set. In this case, the security region is embedded in the surface defined by $Pa+Pb-L=K,$ where L is the total load of the set.

The display of the security surface in a three-dimensional space is possible, but it has been found that it is better visualized by its projections on a two-generation subspace. For example, Fig. 6.1 shows the projections relative to three generation groups named Paranaiba (*Pa*), Grande (*Pb*), and Parana (*Pc*).

This figure shows three superimposed regions. The green region is secure. The yellow region represents post-contingency thermal limit violation and the red region is unstable or inaccessible due to generation limits. The operating point is a gray dot inside the green region. Instability means that if the system is operating at any point in the red region, at least one of the evaluated contingencies will cause system transient stability.

The violation of any other monitored criterion (oscillation damping, transient overvoltage, frequency limit, etc.) can similarly be depicted by a contour. For example, the blue contour in Fig. 6.4 shows violation of after fault voltage drop. Whenever the operating point approaches a contour, some preventive actions may be need. For example, these actions could entail generation re-dispatch to keep a security margin or energizing a shunt device to avoid undesired post-contingency voltages.

The process of finding the boundary in a particular direction is similar to that used for the import–export transfer capacity, but now the directions of search are given by $\Delta Pa + \Delta Pb + \Delta Pc = 0$. These are radial directions from the operating point.

Finding the boundary as in the import–export method, i.e., by binary search, is a brute-force approach. Based on a huge number of simulated cases so far, it has been observed that the contours are convex. Irregularities on the border may occur only as a result of a numerical problem, typically power flow data errors. This convexity characteristic can be exploited to significantly reduce the number of contingencies in the search of contour boundaries. For example, pre-contingency operating points may be targeted near the expected boundary, avoiding several unnecessary simulations, once two adjacent directions have been completed.

Also, and very useful, is to filter contingencies in a particular direction if two relatively close directions have been completed and found that these contingencies do not present threat to the system, i.e., their stability margins are high all over the adjacent directions. The performance gain with this and other heuristic approaches vary from case to case, but can be huge.

To achieve a good plotting contour, it is important to maintain a straight-line direction of search, keeping the ratios $\Delta Pa/\Delta Pb$ and $\Delta Pb/\Delta Pc$ constant. As the operating point moves along this direction, contingencies are processed and evaluated. The final contours represent the intersection of all violations resulting from all contingencies.

The number of directions used for each contour plot determines the precision of the contour. But as this number increases, more computation is required and, and given limited computational resources, consequently performance decreases. Thus, a good compromise is required for online applications.

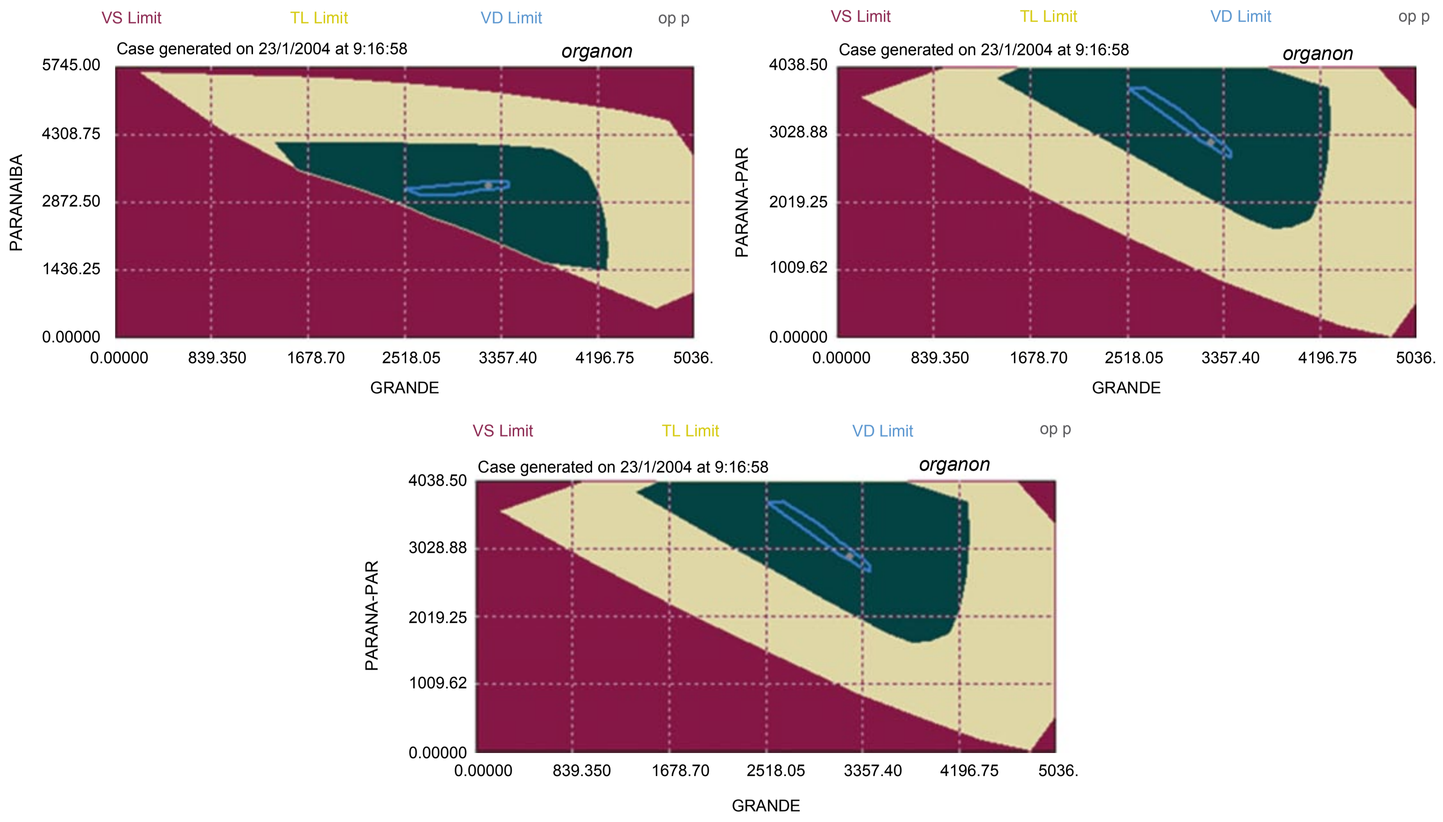


Fig. 6.4 Security region projected on two-generation subspaces

If better accuracy in defining the security border is required, the generation units may be allowed to switch on and off along the re-dispatch direction to keep the spinning reserve within a realistic range (excessive spinning reserve would lead to optimistic security margins, since the total inertia and MVAr margin would be bigger than expected in practice).

Before re-dispatching the generation, the spinning reserve at the new operating point is checked against the range and adjusted if possible. This feature has been used more frequently for operational planning. For online security assessment, this is not used because the focus is on what are the security boundaries for the current state and energized devices. In this case, committing generation units can artificially extend the boundaries.

The complexity of the security region calculation is obviously greater than that for import–export transfer capacity, but only a small set of contingencies is typically required. The security region computation for a transmission corridor typically requires only a few dozen contingencies. Thus, for example, for a case with 10 contingencies, 20 radial directions, and an average of 7 contingencies checks on each direction, approximately 1400 contingencies are computed.

One of the main benefits of computing and displaying a security region is the powerful and immediate insight that it provides to system operators. For example, if the operating point lies outside in the insecure (red) region, the required generation re-dispatch (where and how much) to correct the problem can directly be retrieved from the region display. Each line segment of a contour is associated with the contingency that caused the violation. This information is available in tables, with full report for each line search, and on the region via mouse hovering on contour boundaries. It is also possible to retrieve the pre-contingency operating point at any point in the security region to be analyzed in study mode or to be displayed on the operator's one-line diagrams. The process is simple and requires just a few mouse clicks.

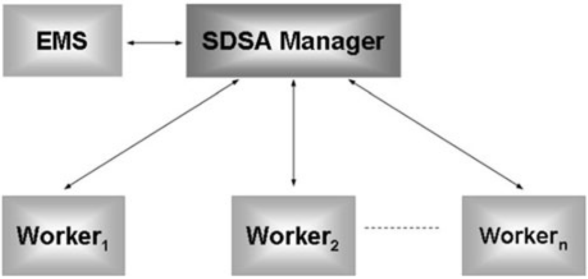
A security region is focused in a specific transmission bottleneck. For a large system, it may be necessary to monitor several bottlenecks. This is done by multiple instances implementation of SDSA. The instances run concurrently, using the same real-time information, but with a different set of contingencies and different generation areas. For the Brazilian system implementation, each of its four control centers currently monitors four different transmission bottlenecks, totaling 16 different security regions.

6.7 Solution Architecture

6.7.1 Parallelization

In order to meet stringent performance requirements, the SDSA adopts parallel processing techniques in a manager/worker (master/slave) configuration, as shown in Fig. 6.5.

Fig. 6.5 Manager–worker distributed processing environment



The manager process contains the high-level instructions to perform the security assessment functions. Most of the calculations (power flow, time-domain simulation, etc.) are done in the worker processes. The manager is responsible for generating base cases, distributing tasks among servers, collecting the respective reports, communicating with the supervisory control and data acquisition (SCADA)/EMS, managing distributed resources, and storing/displaying results and plots. Workers receive tasks from the manager on a first-to-ask-first-to-get basis, process them using the task-specified power system simulation tool, send the respective results back to the manager, and ask for another task. The workload per processor tends to be well balanced as the number of tasks increase. This is certainly the case for security region calculation and contingency analysis with a large number of contingencies. The idle time per processor is relatively small and occurs only at the end of an assessment cycle when some have finished their tasks and there is no additional work to be done. Depending on the number of processes (workers), initial data (operating point base case) broadcasting from manager to workers may have a nonnegligible effect on performance.

This effect can be minimized by broadcasting only data changes from the previous assessment. For example, dynamic models do not change and can remain in memory over subsequent assessments. It has been observed that in up to 48 processes, the data exchange (task assignment and report) between workers and the manager during the calculations does not cause significant overhead. If more processes are necessary, the parallelization strategy can be modified to decentralize the manager role and balance the workload. Worker processes can run in silent mode or be attached to a console. The manager can run in silent mode, attached to a console or the graphical user interface (GUI).

This level of parallelization is implemented via the Message Passing Interface (MPI) mechanism (Gropp et al. 2000). The initial data are broadcasted to workers. Task assignments and diagnosis reports for security regions are implemented via send/receive communication with the tasks assigned in a first-to-ask-first-to-get order. For contingency analysis, tasks are assigned in chunks and results are gathered at the end of the assessment. The send/receive strategy adopted for security regions was found to be more convenient because the manager process can take decisions on the fly to terminate processes based on results achieved by other processes.

In addition to MPI parallelization, the software can also be compiled with loop-level and working share (Chandra 2001) directives for lower granularity

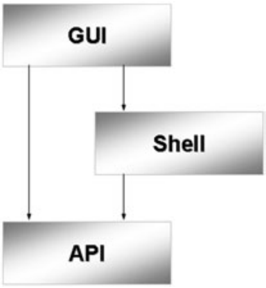


Fig. 6.6 Software layers

parallelization, allowing improved performance in multiple core architectures. This type of parallelization can provide significant speed up of a time-domain simulation because the most costly tasks in this type of simulation are the computation of the vector functions (6.15 and 6.16), which can be shared among processes/threads.

However, for computing a large number of contingencies with limited computational resources the best overall performance is obtained with large granularity parallelization only. This observation can be explained by the fact that there is no data interdependence for coarse granularity parallelization (contingency simulations), whereas significant interdependence in fine-grain parallelization (loop parallelization within a contingency simulation), which adds a significant overhead to the total computation, and more importantly, for fine-grain parallelization there are significant parts of the codes that have to be executed sequentially.

Moreover, the need for improving the performance of a time-domain simulation is not so critical when using variable time step, as this can be several orders faster than fixed-time-step approaches. Therefore, if a limited number or cores are available, the loop-level parallelization would be only recommended for very large networks and few contingencies to be simulated.

6.7.2 Software Layers

Internally, the software is organized in layers as shown in Fig. 6.6, where the arrows indicate the data dependence relationship. The (GUI) can be removed if not desired. The shell consists of input/output interfaces.

The API communicates with the security assessment functions as well as allowing direct access to their analytical engines, such as power flow, contingency analysis, etc. This design facilitates different levels of both SCADA/EMS integration and the offline use of the software.

6.7.3 Integration with SCADA/EMS

The general design for SDSA hardware integration with the SCADA/EMS systems is shown in Fig. 6.7.

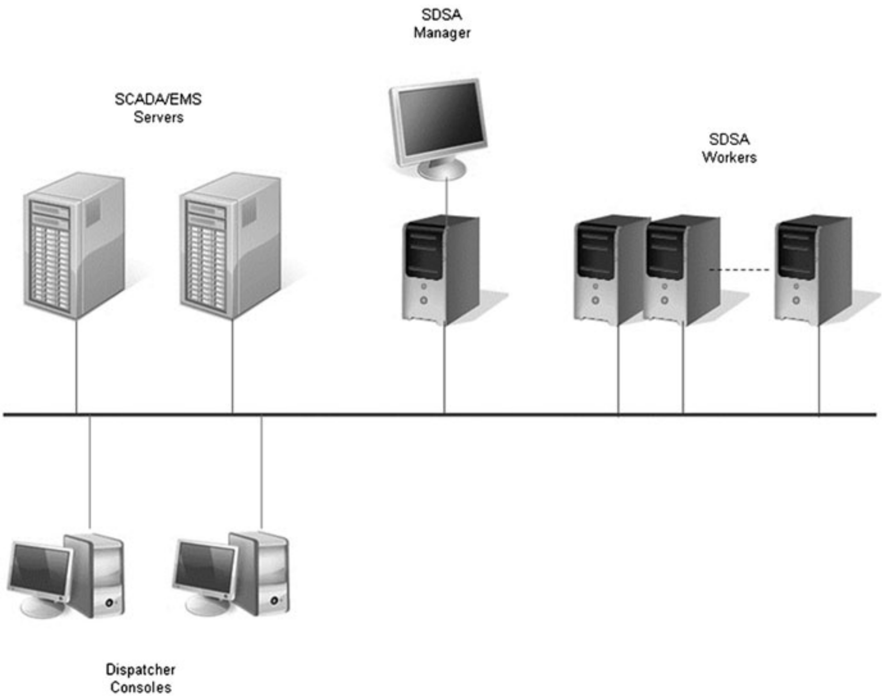


Fig. 6.7 SCADA/EMS network for SDSA

In a loose integration approach, the base cases generated by the SCADA/EMS real-time network analysis subsystem are saved in the form of flat files and are retrieved by the SDSA periodically but asynchronously. The security assessment results are displayed in the SDSA GUI. They can also be either viewed on dispatcher consoles or projected on the control room displays. This kind of integration is relatively simple to implement. The SDSA can also periodically retrieve base cases and be used in study mode.

In tight software integration mode, the SCADA/EMS servers and SDSA servers are in the same local network, as shown in Fig. 6.7. Any of the nodes dedicated to SDSA can be the manager process. There is no communication between the SCADA/EMS and SDSA workers.

A failover procedure can be implemented. The general idea is the following. Failure of a worker process can be detected by the manager process, which reassigns the task to another process. A monitor process in the SCADA/EMS detects failures of the manager process and restarts the SDSA, reallocating the manager process to another node if necessary.

The sequence of events for security assessment is as follows:

- The SCADA/EMS generates a bus-branch model based on the state estimation output.
- The SCADA/EMS sends a message to the SDSA to start a new assessment cycle.

- The SCADA/EMS sends the data and calculation parameters to the SDSA.
- The SDSA performs the assessment and sends the results back to the SCADA/EMS to be stored in its database.

Results are displayed on the dispatcher's consoles and control room projection board. As most of the time the system operates with sufficient security margin, it may not be necessary to display nomograms and other results permanently. One way of circumventing this dilemma is to generate alarms whenever the system operating condition approaches one of the security boundaries.

SDSA can be implemented in multiple instances, i.e., when several different assessments run concurrently, e. g., two instances with online nomograms for different transmission corridors and one instance running system-wide contingencies.

SDSA can also be used to monitor near real-time operating conditions. For this, it is important to be able to generate base cases that represent the near future operating condition with reasonable accuracy. It is necessary to access databases with information regarding generation scheduling, load forecast, and outage schedule. A combination of power flow and CPF methods can be used to solve the problem or, preferably, an optimal power flow method (Granville 1994).

6.7.4 Performance

It is difficult to establish accurate performance figures given that it depends on several factors such as:

a. Network size and number of dynamic models represented
b. Complexity of the model (e. g., representation of several remedial action or special protection schemes) that can extend transient periods
c. Stiffness of equations (e. g., representation of DC links, series controlled compensators, static VAr compensators, etc.
d. Number of available CPUs
e. Number of contingencies
f. Type of assessment (operating point or security region)

It is desired that the response time for an online security assessment system should not be >5 min and preferably around 2 min. This can be achieved by properly dimensioning the computer resources.

An SDSA performance example, using the same number of contingencies (brute force) and different number of cores is shown in Table 6.1.

The simulation conditions for this example are as follows:

- *Model characteristics*: It consists of the computation of a security region for the 500/765 kV south–southeastern corridor of the Brazilian Grid. The network comprises 5306 buses, 7604 branches, and 1173 generators in service. As most of the power plants are hydro with several similar units operating in parallel, the respective generator models are trivially aggregated resulting in 867

Table 6.1 SDSA performance for different number of CPUs

Number of instances	Number of CPUs	Total time (s)	Effective time per contingency (s)
2	32	189	0.166
3	48	141	0.124
4	60	126	0.111

generation buses. The case represents a heavy load condition with approximately 78,000 MW. The dynamic model includes two LCC HVDC (six monopoles) links, four TCSC, several SVCs, out-of-step protection systems, and a highly complex generation shedding protection scheme

- *Contingencies and stiffness*: The five most severe N − 1 contingencies in this corridor are considered. The simulation time for each contingency is 20 s. Several of these contingencies result in commutation failure and/or generation shedding. Consequently, the respective simulations are computationally demanding because of several hard discontinuities and fast transients to which the system is subjected. Whenever there is a discontinuity, the time step reduces significantly and remains low for the duration of the fast transient. Thus, more steps per simulation are necessary, which decrease the performance. (Note: The effect of stiffness on simulations based on fixed time step is even worse, as it may force the time step to be quite small for the total duration of the simulation to avoid numerical instability.)
- *Computational effort*: The security region was computed with 20 boundary search direction and required a total of 1137 time-domain simulations, which correspond to an average of 56.85 simulations per searching direction, or 11.37 simulations per searching direction per contingency. Only brute-force approach, i.e., binary search method, was used in this process to obtain a worst-case scenario (as explained in Sect. 6.6, more intelligent heuristic methods can be used to significantly speed up the process). Loop-level parallelization was not used.
- *Security Criteria*: In addition to the traditional criteria (transient stability margin, minimum acceptable damping, transient voltage levels and duration, thermal limits, and frequency levels), transient impedance trajectories seen by out-of-step relays are also monitored. If a trajectory approaches the relay's tripping threshold by a given margin, the criterion is violated.
- *Platform*: Instances of Amazon cloud (virtual) cluster cc2.8xlarge with two processors ES-2670 (physical 8 cores each), which means a total of 16 physical cores per instance.

It was assigned one SDSA process per CPU. Even using a brute-force approach it is possible, in this case, to meet the requirements with only two instances. If optional heuristic methods are used, some of the simulations are avoided, speeding up the assessment. However, the performance gain can change from case to case and would be better quantified by statistical methods.

6.8 Practical Implementation Aspects

6.8.1 Bus Numbering

The regular bus-branch network model used in planning studies needs to be adapted for online dynamic security assessment. The locations of the dynamic models are usually specified by bus numbers, but these numbers are not fixed in real-time models. Therefore, mapping is necessary. To help with this, a bus identification scheme was implemented. It consists of extending the bus number by adding the section number. For example, bus section 1 of bus 100 can be represented by 100.1. If there is only one section in this bus, it is represented by 100 as usual without section information. This also helps to model bus split/merge events in contingencies without losing the original bus number identity.

6.8.2 Network Size and Observability

For small- to mid-sized network models, say < 10,000 buses, network reduction can help improve performance, but it is not critical for today's computational capacity. However, for very large networks, model reduction can be necessary to achieve online response requirements, in particular for dynamic security assessment. Also, for situations in which there is no real-time observability of external networks, the use of external equivalents is necessary. Several effective methods to reduce networks and derive external equivalents have already been proposed (Monticelli et al. 1979; Savulescu 1981). SDSA currently adopts the Ward method (Monticelli et al. 1979) with flexibility to allow some of its variants. For steady-state contingency analysis, it can yield very accurate results with neglectful computational cost. The issue is how to couple the external equivalent with the network model that has been observed online. If the EMS allows, this can be done during the state estimation process using pseudo-measurements. Otherwise, it can be done externally via a similar process or, as in SDSA, via nonlinear programming (Granville 1994).

The network reduction must take into account important aspects for dynamic security analyses. Depending on the network size, a simple and effective practice is to retain in the network equivalent all or most of the generation buses. For very large systems, quite remote generation buses can be aggregated and simplified dynamic models adopted for the aggregated generator. For the Brazilian network, for example, the real-time network is smaller than the used for planned studies because some portions of the low voltage grid (under 138 kV) are not represented, but dynamic models are the same in both cases.

6.8.3 Contingency Set

For transfer capacity and security region studies, the set of contingencies to be simulated is relatively small, since the contingencies are restricted to the transmission path of interest. The contingencies are predefined by planning engineers based on their experience and can be enabled/disabled by operators on line.

Security assessment by contingency analysis for the entire network demands a relatively large set of contingencies. In this case, a screening process can significantly improve the performance. Considering the quite efficient time-domain simulation, as described in Sect. 6.4, there may be no need for screening or the screening can be performed on the fly. The modified SIME method, as described earlier in this chapter, is quite suitable for this. However, it is important to remember as noted Sect. 6.6 that:

- The proposed screening methods generally estimate only energy margins, consequently do not assess other security criteria.
- These methods are devised for simple simulations where the fault is removed after a few milliseconds along some equipment. For more complex simulation patterns and models with automatic action of protection relays or schemes, subsequent events can be triggered invalidating the screening.

6.8.4 Real-Time Data

Real-time data with reasonable quality must be available for online security assessment, of course. Among the well-known prerequisites are sufficient observability of the internal system, robust state estimation, and a carefully maintained database. When used to initialize security assessment, a state estimator's requirements are much higher than merely providing correct values for voltages and flows. The estimator needs to produce a solved model that is valid for power flow solutions including all controls. For example, it is not uncommon to see real-time cases with the following problems:

- Unrealistic MVAr injections in neighboring buses, sometimes canceling each other
- Generators with MW outputs significantly above their specified maximum capacities
- Generators with unbounded (very high) capacity
- Generators, shunts or on-load tap changing transformers controlling extremely remote buses, etc.

Some of these problems originate from the state estimation process and some are related to errors in the database. At the cost of some extra computation, various conflicting control problems can be intercepted and resolved by extensive input data-checking functions of the security assessment software itself. This is preferably

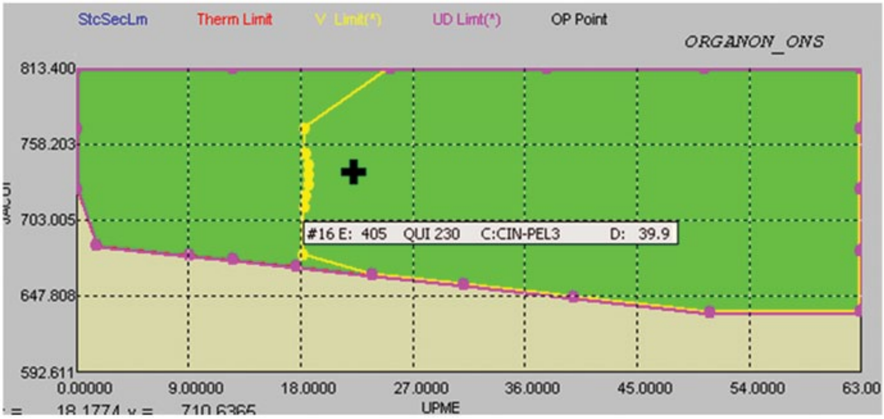


Fig. 6.8 Operating point near voltage limit contour

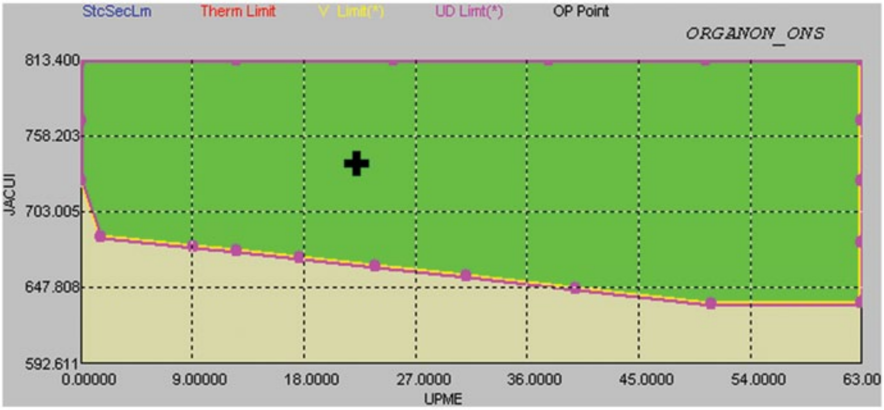


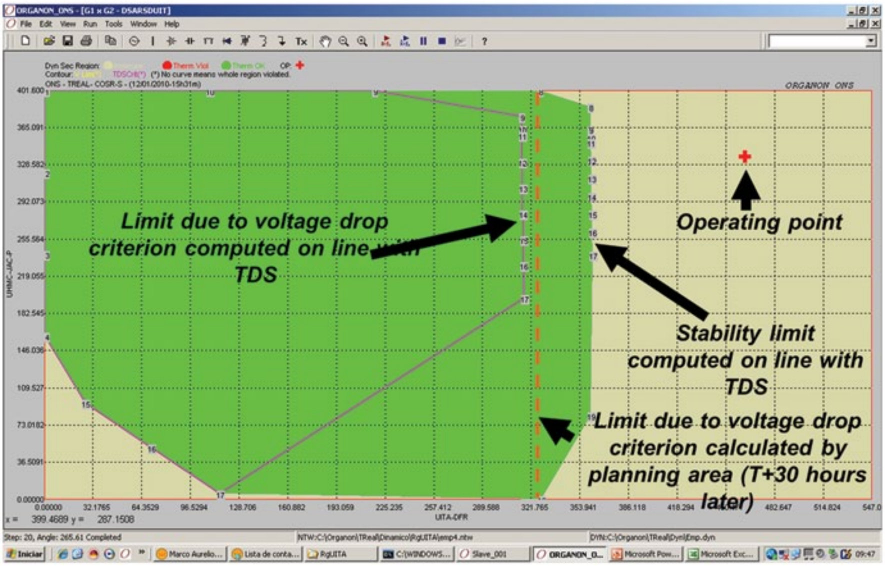
Fig. 6.9 Assessment after preventive action (capacitor switching)

done offline, of course. Problems of poor observability (generally, an inadequate measurement set) are more difficult to solve and, depending on the required accuracy, may be a major barrier for the online implementation.

6.9 Examples

Figure 6.8 shows an example where the real-time operating point (black cross), monitored by one of ONS control centers, was approaching a (yellow) contour indicating violation of voltage limit.

In Fig. 6.9, the same operating condition of Fig. 6.8 is assessed, but after switching on a capacitor bank. If contingencies had only been computed at the operating point, the proximity to violation would have passed unnoticed.



Limit due to voltage drop criterion computed on line with TDS

Operating point

Stability limit computed on line with TDS

Limit due to voltage drop criterion calculated by planning area (T+30 hours later)

Step: 20, Angle: 265.61 Completed

NTW:C:\Organon\TReal\Dinamico\RgUITA\emp4.ntw

DYN:C:\Organon\TReal\Dyn\Emp.dyn

Fig. 6.10 Unsecure operating point for not planned condition

Figure 6.10 shows the real-time operating point outside the security region due to an N−2 condition for which no security limits had been computed offline. System operators were able to correct the problem based on online nomogram computation, i.e., the generation was re-dispatched to correct the violation. The online case was saved and after the fact (T+30 h) studies confirmed the online calculations.

Figure 6.11 shows a security region validation (online vs offline) for one of the most important transmission corridors in the south–southeastern part of Brazil.

Figure 6.11a presents the security region computed online with no generation commitment. Figure 6.11b shows results computed offline for the same transmission bottleneck with the base case data adjusted for conditions quite close to the online case and allowing generation commitment to keep reasonable spinning reserve.

Figure 6.11c shows the online results superposing the offline results. The light green contour represents voltage drop limitation and the red contour represents the operation of out-of-step relays. In Fig. 6.11c, it is noticeable that the stability regions (nonred regions) in the upper right corner are very close in both cases.

The limiting factors are the proximity of out-of-step triggering and voltage drop (A and B in Fig. 6.11c), which are practically the same in both cases. The extended lower boundary in the offline security region, depicted by C in Fig. 6.11c, is because unit commitment is enable in this case.

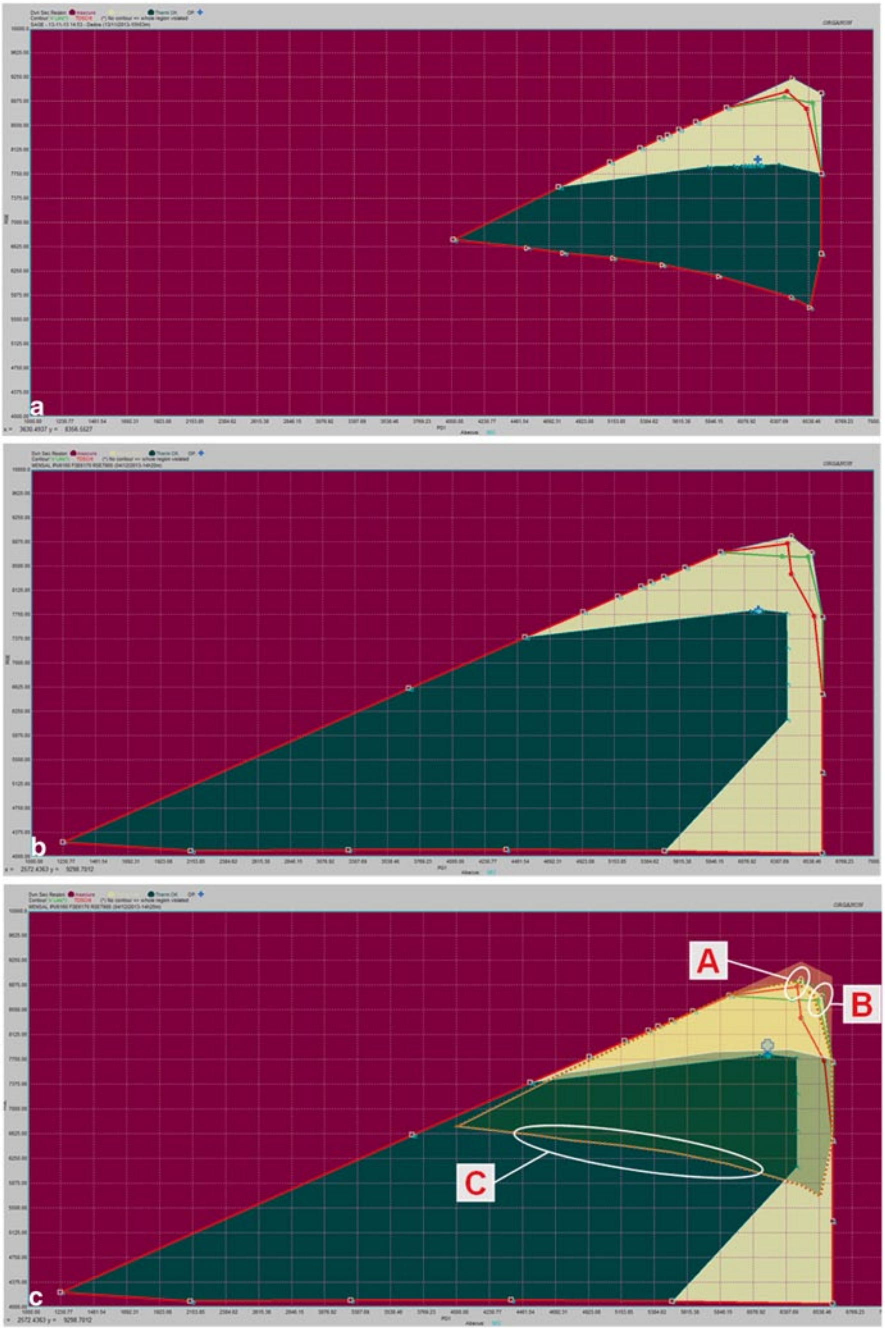


Fig. 6.11 Security regions: **a** using real-time data, **b** using offline planning data, **c** real-time results superimposed on planning results

6.10 Conclusions

This chapter has described an online security assessment system whose development was dominated, particularly on the dynamics side, by the need for consistent and solid analytical methods, process automation, and high-performance computing. The system is able to perform detail and sufficiently fast security assessments. The full detail time-domain simulation approach keeps close compatibility with offline studies, provides accurate assessments, and can be easily validated. It can also be combined with faster simplified methods to reduce the computational burden and improve response time under fast changing operating conditions.

In summary, until recently it was very difficult, if not impossible, to implement online dynamic security assessment for medium-sized to large power systems because the analytical methods for security margin estimation were not mature enough and low-cost computing power was not available, either. These barriers no longer exist. Today, the state of the technology allows us to perform huge numbers of detailed simulations in a few minutes. This is of use not only for online security assessment, but also for power system planning. It is a change in paradigm, which brings benefits to all areas of power system analysis.

6.11 Acknowledgments

The author is grateful to ONS, in particular to Dr. Luiz Claudio de A. Ferreira and Dr. Carlos Alberto da S. Neto for providing examples of SDSA application and performance figures. Portions of this chapter have been reprinted, with permission, from Jardim, J, 2009, Online Security Assessment for the Brazilian System—A Detailed Modeling Approach, in *Real-Time Stability Assessment in Modern Power System Control Centers*, IEEE Press and John Wiley & Sons, Inc., Hoboken, NJ, 2009, ©2009 IEEE.

References

Ajjarapu V, Christy C (1991) The continuation power flow: a tool for steady state voltage stability analysis, IEEE PICA conference proceedings, May 1991, pp 304–311

Arrillaga J, Arnold CP, Harker BJ (1983) Computer modelling of electrical power systems. Wiley, London

Astic JY, Bihain A, Jerosolimski M (1994) The mixed Adams—BDF variable step size algorithm to simulate transient and long term phenomena in power systems. IEEE Trans Power Syst 9(2):929–935

Braz LMC, Castro CA, Murati CAF (2000) A critical evaluation of step size optimization based load flow methods. IEEE Trans Power Syst 15(1):202–207

Brenan KE, Campbell SL, Petzold LR (1989) Numerical solution of initial-value problems in differential-algebraic equations. SIAM Classics in Applied Mathematics, North-Holland, New York

Castanié F (2011) Digital spectral analysis: parametric, non-parametric and advanced methods. Wiley, Hoboken

Chandra R (2001) Parallel programming in OpenMP. Academic, San Diego

Debs AS, Benson AR (1975) Security Assessment of Power Systems. In: Proceedings of system engineering for power: status and prospects, Henniker, NH

Dy Liacco TE (1968) Control of power systems via the multi-level concept, Case Western Reserve University System Research Center, Report no SRC-68-19, June 1968

Granville S (1994) Optimal reactive dispatch through interior point methods. IEEE Trans Power Syst 9(1):136–146

Gropp W et al (2000) MPI: the complete reference. MIT Press, Cambridge

Hauer JF (1991) Application of Prony analysis to the determination of modal content and equivalent models for measured power system response. Proceedings of the IEEE winter meeting, 215-4 PWRS

Hayashi S (1969) Power system security assessing by digital computer simulation—basis control. In: Proceedings of PICA conference, Denver, Colorado, 18–21 May 1969

Jardim JL (2000) Online dynamic security assessment: implementation problems and potential use of artificial intelligence. Proceedings of IEEE power engineering society summer meeting, vol 1, 16–20 July 2000

Jardim J (2009) Online security assessment for the Brazilian system—a detailed modeling approach. In: Savulescu SC (ed) Real-time stability assessment in modern power system control centers. IEEE Press and Wiley, Hoboken

Jardim JL, Neto CS, Kwasnicki WT (2004) Design features of a dynamic security assessment system. Proceedings of IEEE power system conference and exhibition, New York, 13–16 Oct 2004

Jardim JL, Neto C, Santos MG (2006) Brazilian system operator online security assessment system. Proceedings of IEEE power system conference and exhibition, Atlanta, GA, 30 Oct–02 Nov 2006

Lambert JD (1991) Numerical methods for ordinary differential systems: the initial value problem. Wiley, New York

Limmer HD (1966) Security applications of on-line digital computers. Second power systems computation conference, Stockholm, 27 June 1966

Molina RDM, Cassano M, Savulescu SC (2009) Dimo's approach to steady-state stability assessment: methodology overview, numerical example, and algorithm validation. In: Savulescu SC (ed) Real-time stability assessment in modern power system control centers. IEEE Press and Wiley, Hoboken

Monticelli A (1999) State estimation in electric power systems: a generalized approach. Kluwer, Norwell

Monticelli A, Deckmann S, Garcia A, Stott B (1979) Real-time external equivalents for static security analysis. IEEE Trans Power Syst 98(2):498–508 (PAS)

Pai MA (1989) Energy function analysis for power system stability. Kluwer, Norwell

Pavella M, Ernst D, Ruiz-Vega D (2000) Transient stability of power systems: a unified approach to assessment and control. Kluwer, Norwell

Savulescu SC (1981) Equivalents for security analysis of power systems. IEEE Trans Power Syst 100(5):2672–2682 (PAS)

Seydel R (1994) Practical bifurcation and stability analysis: from equilibrium to chaos. Springer-Verlag, New York

Smed T, Andersson G, Sheblé GB, Grigsby LL (1991) A new approach to AC/DC power flow. IEEE Trans Power Syst 6(3):1238–1244

Stott B (1979) Power system dynamic response calculations. Proc IEEE 67(2):219–241

Stott B (1974) Review of load flow calculation methods. Proc IEEE 62:916–929