ORGANON - Volume III
Introduction
Methodology

Version 1.1

11/4/2005

Index

1 Introduction

Organon is a power system analysis and security assessment software. It can be used as standalone mode and/or integrated to an Energy Management Systems – EMS, which makes it suitable for planning and real-time environments.

SECURITY ASSESSMENT

Security assessment is a complex problem, as it demands a very high computational effort. Traditionally, security assessment is performed by planning engineers using conventional power system analysis tools. The results are saved in operating orders in the form of tables, rules and graphics, which guide dispatchers in real time power system operation. The main disadvantage of that approach is that it is impossible to assess security for all possible real time operating conditions because of combinatorial explosion of possible cases to be computed. Moreover, even the processing of a large number of operational conditions is impractical because it is a very slow process. The time required by an engineer to preparing data, processing simulations, analyzing results and writing a report can take many hours or even days. Despite of that, the technology available until a few years ago would not allow the implementation of a different process. As a consequence the security assessment is done for a relatively small set of operating conditions. Usually worst scenario cases are evaluated, which tends to impose conservative constraints on security margins. These in turn may have undesirable economical effects, either because of the need for more investments (mainly in transmission lines) or the loss of opportunity to sell more power.

The limitation in the number of cases evaluated may also have serious consequences on system reliability because of the high probability of missing critical operating conditions and non-predicted emergency situations. To prevent such risk, it is necessary to adopt very conservative reliability criteria. However, current environmental, economical and financial constraints do not allow much flexibility for that. The consequence may be risky operation of stressed networks.

In other words, the operation can be unnecessarily conservative for predictable situations, but unsafe for unpredictable ones.

For the last three decades, there have been many research efforts to develop methods for real time on line security assessment. The advantage of such facility would be that the complexity of the problem reduces significantly, as there is no uncertainty regarding the operating condition, which is known with good accuracy through SCADA system. Then the security assessment perfectly matches the current operating condition. The major difficulty to implementing an online security assessment was the insufficient computational power available for the complexity of the problem. Then the focus of the research concentrated in two main lines. One was the use of approximated algorithms and methods for power system simulation to speed up the computation. The other was the adoption of simplified power system models to reduce the size of the problem. Some very fast methods have been devised, but they have failed to reach production grade and being implemented commercially because they either fail to produce accurate results or fail to complete the calculations due to numerical problems.

This scenario has changed dramatically as low cost High Performance Computation – HPC became available in the last few years. For example, a power system simulation can be processed more effectively today in a personal computer than it used to be in a workstation some years ago.

ORGANON DESIGN CRITERIA

Organon was designed to take advantage of this technological evolution for improving power system planning and operation processes. It combines HPC, robustness and ability of detail modeling representation to provide reliable, accurate and fast security assessment. The system design is nurtured first by simulation fidelity, followed by robustness and then computational speed. This priority order distinguishes Organon from most of the similar tools.

Simulation fidelity implies the use of detail model representation, which is a characteristic of tools used in planning environment. Most of the tools developed for real time environment aim at fast computation in detriment of simulation accuracy. This is a major drawback as results are not accurate, i.e., it is hard to trust the assessment. Detail modeling presents no technical difficult, as it is a standard procedure in general purpose off-the-shelve power system analysis tools. Organon just recognizes this basic requirement for effective security assessment.

A security assessment tool is useless if it is not reliable. For example, if a load flow calculation fails to converge it is not possible to conclude the assessment. In this case dispatchers would be in the dark. So robustness is paramount for effective power system computation. The choice of numerical methods and algorithms has been guided by robustness criterion.

HPC is achieved in Organon with a scalable distributed processing architecture. Tasks in a security assessment process are concurrent. Thus they don't necessarily need to be processed sequentially. Contingency analysis, for example, is a typical functionality suitable for distributed processing. The various contingencies to be evaluated can be distributed among many processors, which process them in parallel. The results from all simulations are then combined to produce a global assessment.

ORGANON METHODOLOGY

Organon methodology is based on the automation of traditional planning procedures. It contains some built in functionalities, but in some cases customization may be necessary.

Automation usually arises suspicions about the ability of the system to deal with non-ordinary situations. In other words, the sometimes lack of flexibility or adaptability of an automated process is generally seen with skepticism. However, these feelings are generally unsubstantiated, as the industrialization history has shown. In the particular case of power system security assessment process the benefits of automation are overwhelming. An automated process can be 200 times faster than a manual one using a single processor. With 20 processors it can be 3000-4000 times faster. Such productivity gain cannot be disregarded in the planning environment and is sine qua non condition for real time security assessment. Moreover, the rules and methods can evolve as the system conditions change and/or more is leaned about the system.

An automated process is also auditable and not ambiguous. It is well know in planning environment that given a calculation procedure, two analysts can reach different conclusions because of individual interpretations. In an automated process there is no such risk.

An automated process can help increase expertise and the knowledge base about the system, as much more system condition can be evaluated faster. Experts and specialists, usually spend most of their time with routine work (preparing data, inspecting results, etc). Usually a small percentage of their time is allocated to real analysis. Automation improves their working conditions and allows them to contribute more effectively and proactively with their expertise.

The essential components of Organon automated security assessment are the follow:

- Tools to perform simple tasks such as run a power flow, change the operating point to a new desired condition and run a time domain simulation.
- Procedures to evaluate the critical aspects of the system. These procedures are based on the practices adopted in planning environment.
- Reliability criteria, which are different for each system or coordination council.
- Methods of extracting diagnosis from simulations without human interpretation or visual inspection.

FUNCTIONALITIES

The main functionalities are the following.

- Contingency analysis.
- Generation and load shedding calculations.
- Security margins in MW (e.g., distance to collapse).
- Security regions (nomograms).
- Long-term dynamic simulation.
- Alarms.

- Case-by-case processing or study mode.

OUTPUT AND USER INTERFACE

Various forms of output results are available depending on the implemented functionalities. Organon generates reports containing standard and user defined output channels, tables and plotting.

The use of the system has to be easy and intuitive particularly for real time environment where direct information about security state has to be readily available. Organon has its own graphical user interface, which is the standard for a standalone application, but it can also run in background mode at EMS if required.

TECHNOLOGY

Programming Languages – The system is written basically in Fortran-90/95 and C++ for Windows© platforms. Windows API is used for graphical user interface. Message Passing Interface – MPI is used for distributed processing.

Power Flow – The full Newton-Raphson method is used because of its better convergence properties. All the controls (tap, remote generation, shunts, HVDC, etc.) are solved simultaneously by Newton method. Newton step control is implemented to improve convergence of difficult cases. Initialization procedures and other non-published methods to improve converge are also employed. Built in sensitivity analysis.

Continuation Power Flow – The tangent vector methods is used. Facilities to fully control trajectories in parametric spaces are also used.

Time Domain simulation – ABM predictor corrector integration method is used for differential equations and BDF is used for algebraic equations. The solution of all equations is simultaneous and variable-step-variable-order approach is adopted, which provides both the highest computational efficiency and robustness. Simulations are automatically terminated either by detected instability or convergence to an equilibrium point.

Signal Processing – Prony analysis and energy function algorithms are used for automatic diagnosis of time domain simulations.

Hardware – The system can run either on a single personal computer under Windows NT, 2000 or XP© or on a network of personal computers (multi-processing). For multi-processing it is possible to use an existing network (e.g., standard offices computers), which is suitable for ‘overnight supercomputer’ configuration, or a dedicated cluster of computer for heavy duty full time processing system.

MANUALS

Volume I shows how to install and use the program. Volume II shows the time domain simulation models and respective input parameters. This volume (III) presents the concepts and algorithms adopted in the program.

2 Conventional Power Flow

The load flow formulation [1,2,3] is based on the Kirchhoff current law for every node (bus) in the system. In terms of power equations this means that the bus generation minus the bus load has to be equal to the power flowing through the branches connected to the bus, This is expressed by:

$$0 = Pg_k - Pl_k - \sum_{m \in \Omega_k} P_{km} \ , \tag{2.1}$$

$$0 = Qg_k - Ql_k + Qcap_k - Qrea_k - \sum_{m \in \Omega_k} Q_{km} \ , \tag{2.2}$$

for $k= 1, \ldots, N$,

where k is a generic node index, Ω_k is the set of nodes adjacent to k, and N is the number of nodes in the network.

These equations are non-linear functions of the following variables: voltage magnitude(V_k) and angle (θ_k) for every bus in the system, transformer's taps (a_{km}) and phase shift angles (ϕ_{km}), bus generation (Pg_k, Qg_k) and bus load (Pl_k, Ql_k).

The solution method adopted is the full Newton-Raphson with the following enhancements:

- All the controls (HVDC [7], OLTC [4], remote voltage control, etc…) are solved within the Newton iteration. They are embedded in the Jacobian matrix;
- Jacobian factorisation is based on both fill-in and numerical stability criteria (off-diagonal pivoting);
- Optimisation of Newton step using quadratic interpolation [5].

2.1 Newton-Raphson Method

The set of equations (2.1, 2.2) and variables can be formulated as a general problem of finding the solution of a set of non linear equations.

$$0 = f(x) . \tag{2.3}$$

where $f(x)$ is a multidimensional function array of the dependent variables $x = (V, \ \theta, \ a, \ \varphi)$.

The Newton-Raphson method has been recognised as the most effective for the solution of (2.3). Its success is mainly due to two main reasons: good convergence properties and simple implementation. It is an iterative process based on the truncated Taylor series:

$$f(x+\Delta x) \cong f(x) + f'(x)\Delta x \tag{2.4}$$

where $f'(x)$ is the Jacobian matrix.

Considering that at the solution $f(x+\Delta x) = 0$,

$$\Delta x \; = -\left(\frac{\partial f(x)}{\partial x}\right)^{-1} f(x) = -J^{-1} f(x) \tag{2.5}$$

The method consists in solving the following equations, in one or more iterations:

$$\Delta x^i = -\left[J(x^i)\right]^{-1} f(x^i) \tag{2.6}$$

$$x^{i+1} = x^i + \Delta x^i \tag{2.7}$$

until $f(x^i) < \varepsilon$, where *i* is the iteration counter and ε is a small tolerance.

To improve convergence [5], Newton step may be corrected as follows:

$$x^{i+1} = x^i + \alpha^i \Delta x^i \tag{2.8}$$

For well-behaved cases, this factor alpha is approximately one. However, for the ill-conditioned cases, it is usually less than one. Alfa is computed through a one-dimensional minimisation process, where the norm of the mismatch array $|f(x)|$ is minimised in the direction of Δx.

Alpha calculation requires a greater computational cost per iteration. As the system is usually well conditioned and the optimum alpha is close to one, the alpha calculation is only performed if the convergence rate is less than a pre-specified threshold.

3 Continuation Power Flow

The continuation power flow solution is implemented based on the tangent vector method [11]. This method allows for a smooth transition between two different operating conditions (conventional power flow solutions). It is useful to apply a load-level to the system and/or change the generation pattern. The last one means reducing generation in one area, while increasing in a different one. It can also be used to find the maximum loadability of the system (saddle node point), if this solution exists. This is possible due to the reformulation of the power flow equations that prevents the ill conditioning of the Jacobian matrix.

The only problem with the continuation power flow implementation is that it assumes that the system is continuous. This is not true with power flow equations because of MW/Mvar limitations, discrete tap control and discrete shunt control. The limits are not a major problem as the equations can be considered piece wise continuous, although this process slows down the computational speed. The discrete taps and shunts are considered continuous during the continuation process. If the continuation is used for load level and generation pattern change purposes, the shunts and taps are fixed at the end. If the continuation is used for maximum loadability purpose, the final condition is obtained with continuous taps and shunts. Therefore it is only an approximation of the maximum loadability. For most practical purposes the accuracy is very good despite the approximation.

The tangent vector method consists of two major steps, linked through a continuation parameter. The first step is called Predictor, while the second is called Corrector. In the predictor step, variables such as voltage modules and angles and generation and load are changed, to define an approximation of the new operation point. The corrector step is, basically, the process of running a conventional power flow, starting from this approximation, and keeping one variable constant. This variable is called Continuation Parameter. The Predictor-Corrector cycle is repeated until the solution (load-level, maximum loadability, etc…) is obtained.

3.1 Tangent Vector Method

This method results from the application of a continuation algorithm to the power flow problem. The first procedure is to reformulate the power flow equations in order to add an increment factor for loads and generations resulting in:

$$Pg = Pg_0 + \lambda KPg \tag{3.1}$$

$$Pl = Pl_0 + \lambda Kpl \tag{3.2}$$

$$Qg = Qg_0 + \lambda KQg \tag{3.3}$$

$$Ql = Ql_0 + \lambda KQl \tag{3.4}$$

where lambda is the generation/load increment factor applied to all buses of the system, Pg_0, Pl_0, Qg_0 *and* Ql_0 are the values of the initial operating point and *KPg, KPl, KQg, KQl* are the generation/load change factors, defined for each bus in the system.

The linear equations system defined in (2.3) becomes:

$$f(x, \lambda) = 0 \tag{3.5}$$

To solve the problem, the continuation algorithm starts from a known solution and uses a predictor-corrector scheme, to find subsequent solutions at different load levels and/or generation patterns.

Taking an appropriately sized step in a direction tangent to the solution path makes the prediction of the next solution.

$$f(x + \Delta x, \lambda + \Delta\lambda) \cong f(x, \lambda) + f'(x, \lambda)(\Delta x, \Delta\lambda)^T = 0 \rightarrow f'(x, \lambda)(\Delta x, \Delta\lambda)^T = 0 \tag{3.6}$$

In the Corrector Step a conventional power flow is run, starting from the approximation given in Predictor Step, and keeping one variable constant. This variable is called continuation parameter. The Predictor-Corrector cycle is repeated until the solution (load-level, maximum loadability, etc…) is obtained.

The continuation parameter may be either the increment factor lambda or the voltage at one bus. The decision is based on which one has the highest derivative, computed in Predictor Step. Figure 3.11 shows how it works:

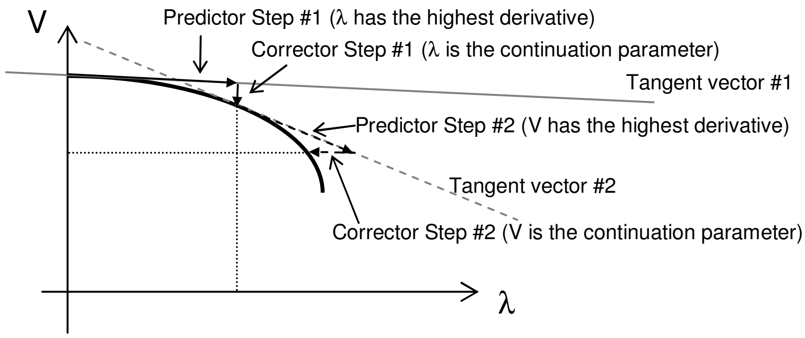

Fig. 3.1 – Example of 2 Predictor-Corrector Cycles with different continuation parameters

3.1.1 Predictor Step

The first task in the predictor process is to calculate the tangent vector. This tangent calculation is derived by the augmented Jacobian, which has one extra column, associated with the additional unknown variable lambda. To balance the number of variables, one extra equation must be added to the problem. This can be done by choosing a non-zero magnitude (say one) for one of the components of tangent vector. In other words, if t is used to denote the tangent vector:

$$t = \begin{pmatrix} \partial x \\ \partial \lambda \end{pmatrix} \rightarrow t_k = \pm 1$$

This results in:

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

where e_k is an appropriately dimensioned row vector with all elements equal to zero except the k^{th}, which equals one. If the index k is chosen correctly, letting $t_k = \pm 1$ imposes a non-zero norm on the tangent vector and guarantees that the augmented Jacobian will be non-singular at the critical point. Whether +1 or –1 is used depends on how the state variable is changing. Once the tangent vector has been found, the step size should be chosen so that the predicted solution is within the radius of convergence of the corrector. A good estimate is to use the inverse of the norm of the tangent vector. Thus:

$$\begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = \|t\|^{-1} \begin{pmatrix} \partial x \\ \partial \lambda \end{pmatrix}$$

3.1.2 Corrector Step

The process of correcting the approximate solution is based on a local parameterisation, where the original set of equations is augmented by one equation, which specifies the value of one of the state variables. This variable is called the continuation parameter and can be either the voltage at one bus or the increment factor lambda. Considering that η is the value of the continuation parameter (V_k or λ) the new system to be solved can be expressed as follows:

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

or

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

The Newton-Raphson method is used to solve this problem.

4 Synthetic Dynamic Power Flow - SDPF

The SDPF is a very robust power flow solution method. It solves a dynamic problem, which steady-state solution is the desired power flow solution [33].

4.1 SDPF Method

A. General Expression of the Power Flow Problem

The problem can be expressed as the set of equalities:

$$g\,(x, y) = 0 \tag{4.1}$$

and the inequalities:

$$m_a\,(y) \leq 0 \tag{4.2a}$$

$$m_b\,(x, y) \leq 0 \tag{4.2b}$$

Set (4.1) contains the system's continuous power flow and control equations in terms of decision variables y and state variables x. Set (4.2a) comprises simple limits on decision variables such as controlled transformer taps and angles, and shunt admittances. Set (4.2b) comprises functional inequalities such as generator VAR limits. The enforcement or release (limit back-off) of any of the inequalities can swap control equations in and out of set (4.1).

B. Synthetic Dynamic Approach

Let us synthesize a dynamic equation for each decision variable:

$$y' = z\,(x, y) \tag{4.3}$$

such that the quiescent state is the power flow solution.

Let us then perform a time domain solution of the resulting dynamic system using a stable, closed (implicit) step-by-step integration method. Thus, at integration step n, the generic form of the solution of (4.3) is:

$$y_n = \beta .\, h .\, z\,(x_n, y_n) + C \tag{4.4}$$

where β is a constant coefficient, and C is a term that usually includes previous values of y or y′.

In this time domain solution, we can rewrite (4.1) and (4.3) respectively as:

$$f\,(x_n, y_n) = y_n - \beta .\, h .\, z\,(x_n, y_n) - C = 0 \tag{4.5}$$

and

$$g\,(x_n, y_n) = 0 \tag{4.6}$$

Equations (4.5) and (4.6) together comprise a nonlinear simultaneous set in the variables (x_n, y_n) whose solution is the system state at the end of integration step n. A predictive formula is used to provide good starting values for x_n and y_n.

In this application, the concept of time t is notional. The solution starts from t = 0 and a set of initial variable values. It proceeds in steps of length h in the usual dynamic stability solution manner [6].

C. Solution of Equations

At each time step, the set of nonlinear simultaneous equations (4.5) with (4.6) is solved iteratively by Newton's method to provide the current state of the system. The form of the Jacobian equation at each iteration is:

$$\begin{bmatrix} \partial f / \partial y & \partial f / \partial x \\ \partial g / \partial y & \partial g / \partial x \end{bmatrix} \bullet \begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix} = - \begin{bmatrix} f \\ g \end{bmatrix}$$

Fig. 4.1: The Jacobian Equation

The solution of the Jacobian equation by sparse triangular factorization produces the corrections Δx and Δy to the respective variables.

Until the quiescent condition is reached, there is no need for high accuracy convergence. We are not interested in the trajectory of the dynamic response, but only the eventual equilibrium state. The use of a very high stability integration method with step control (see section V.D) limits the error propagation.

4.2 SDPF Details

A. Dynamic Modeling

This section describes the synthetic dynamic modeling approach for the most common power flow components. As explained above, each controlled power system device is represented by a simple integrator model, involving a feedback controller acting to maintain a quantity such as MWs or voltage at its specified value.

1. Generators

Here, there are two main choices for the synthetic dynamics:

Model 1: The terminal voltage components θ and V become the decision variables.

Model 2: The generator is represented by a voltage behind an arbitrary reactance (which can be very small). The internal voltage components δ and E become the decision variables. Because the present implementation is an adaptation of an existing dynamic stability program, it was more convenient to use this model.

The active power dynamics for a generator with MW output P(x,y) are then represented by the differential equation:

$$\theta' = K_p (P_{sp} - P(x,y)) \tag{4.7a}$$

or

$$\delta' = K_p (P_{sp} - P(x,y)) \tag{4.7b}$$

This ensures that, when the dynamic system has reached equilibrium, the generator has the specified MW output.

Similarly, the reactive power dynamics are represented by:

$$V' = K_v (V_{sp} – V_c) \tag{4.8a}$$

or

$$E' = K_v (V_{sp} – V_c) \tag{4.8b}$$

where V_c is the controlled voltage, whether at the generator terminal or at a remote bus.

If however the generator is operating at a fixed VAR output Q_{sp} (e.g. it is on a VAR limit), eq.(4.8) is replaced by:

$$V' = K_q (Q_{sp} – Q(x,y)) \tag{4.9a}$$

or

$$E' = K_q (Q_{sp} – Q(x,y)) \tag{4.9b}$$

2. LTC Transformers

An LTC transformer with tap T controlling a terminal or remote bus voltage V has the differential equation:

$$T' = K_t (V_{sp} – V) \tag{4.10}$$

3. Controlled Phase Shifters

A phase shifter whose angle ϕ regulates the MW flow P through itself becomes:

$$\phi' = K_\phi (P_{sp} - P(x,y)) \tag{4.11}$$

4. Controlled Shunt Reactances

A variable shunt admittance B controlling a bus voltage V becomes:

$$B' = K_b (V_{sp} - V) \tag{4.12}$$

5. Other Controls

The above approach can be applied to any other controlled devices, such as SVCs, controlled series reactors, FACTS, and HVDC links, as well as to system-wide controls such as area interchange. Special modeling features can readily be incorporated. For example, the generator VAR limits as above can be continuous or piecewise, and functions of voltage. Control targets, dead bands, etc. can be interpreted and incorporated as desired.

6. Modeling Options

The approach offers considerable flexibility for variants, since in principle a given control can be modeled either dynamically or it can be included in set (4.6) in the usual power flow manner (i.e. incorporated into the Jacobian equations, or adjusted in between integration steps). Generally, dynamic modeling seems to be the most effective.

The same applies even to non-controls. Consider for example the modeling of loads. We could represent their voltage components as decision variables in the manner of (4.7) and (4.9). At present, however, we have chosen to incorporate loads in the normal Newton manner in the power flow equations (4.6). Another example of a non-control that could be modeled dynamically is distributed slack. The relative merits of the modeling approaches need to be studied.

B. Nonlinear Equation Solutions

At each integration step, equations (4.5) and (4.6) are solved simultaneously, using the Jacobian matrix equation of Fig. 4.1. Vectors x and g in Fig 4.1 represent the voltages and mismatches respectively of the network buses. Therefore $\partial g/\partial x$ is the familiar Newton power flow Jacobian matrix.

The following refers to our implementation using generator model 2. The situation with model 1 is somewhat different, since the generator mismatch equations in (4.6) will be replaced by their counterparts from (4.5).

Vector y contains all the decision variables δ and E, and T, ϕ, B, etc. Each element of vector f corresponds to (4.5) with the right-hand side of an equation such as (4.6) - (4.12) substituted for function z.

The matrix rows and columns of Fig. 4.1 that pre-border the Newton power flow Jacobian matrix $\partial g/\partial x$ are extremely sparse and their diagonal elements are never zero. Therefore the factorization of each such row and column is trivial, and except for certain remote controls no matrix fill-in occurs. An option would be to construct only the standard power flow Jacobian equation, and embed into it the factor terms from the pre-bordered rows and columns. As desired, incidence symmetry can be preserved by adding dummy zeros.

Since the step-by-step time domain "simulation" proceeds in relatively small steps, each solution based on Fig. 4.1 converges very quickly. Frequently only one iteration is needed. In fact, the Jacobian matrix factors are often reused (dishonest Newton method). However, if convergence is slow or not obtained, the matrix is reformed and refactorized, and/or the step length is reduced.

C. Inequality Constraints

The inequalities (4.2) are tested at the end of each integration step. When a differential variable subject to (4.2a), such as in (4.10) - (4.12), violates its limit it is simply pegged at the limit. When a functional inequality (4.2b) is violated, an equation swap takes place. For example, when a PV generator VAR limit is violated, Eq.(4.9) is substituted for Eq.(4.8). The normal power flow rules for limit back off apply.

D. The Integration Method

The present implementation uses predictor-corrector Backward Difference Formulas [22] with automatic step-length control. As indicated in Section III.B, the correctors are implicit. They are stiffly stable, basically meaning that they can handle high-frequency oscillations without resorting to relatively very small time steps. The step-length control is particularly important because (a) it carefully avoids numerical instability, and (b) it can expand the step length h when the dynamic response smoothes out, to reduce the overall computation.

5 Sensitivity Analysis

The sensitivity analysis solution is implemented based on the computation of sensitivity indexes using the Jacobian matrix [1].

The conventional power flow problem, that is to balance active and reactive power at all nodes:

- $$\begin{cases} 0 = P_g - P_l - P_e \\ 0 = Q_g + Q_{cap} - Q_{rea} - Q_l - Q_e \end{cases}$$

Can be generalized as, $f(x,u)=0$, where *x* is the set of dependent variables (Ex. Voltage modules and angles), while *u* is the set of control variables (Ex. MW generation).

The relation between these two sets can be defined as:

- $$0 = \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial u}\Delta u \rightarrow \Delta x = -\left(\frac{\partial f}{\partial x}\right)^{-1}\frac{\partial f}{\partial u}\Delta u$$

Thus, to define the sensitivity indexes of one entity (Ex. MW flow from bus *k* to bus *m:* P_{km}) with respect to the set of control variables *u*, we may write:

- $$P_{km} = g(x) \rightarrow \Delta P_{km} = \left(\frac{\partial g}{\partial x}\right)^T \Delta x \rightarrow \Delta P_{km} = -\left(\frac{\partial g}{\partial x}\right)^T \left(\frac{\partial f}{\partial x}\right)^{-1}\frac{\partial f}{\partial u}\Delta u$$

Having this, it is possible to say that the sensitivity index array *s* is defined as:

- $$\Delta P_{km} = [s_1 \quad s_2 \quad \ldots \quad s_n]\begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ : \\ \Delta u_n \end{bmatrix},$$ where: $$s = \left(\frac{\partial g}{\partial x}\right)^T \left(\frac{\partial f}{\partial x}\right)^{-1}\frac{\partial f}{\partial u}$$

It is important to note that for any injection ΔP in the system there is a corresponding change -ΔP in slack bus. Thus, the sensitivity analysis results are dependent on the slack bus location.

6 Time Domain Simulation

Organon uses a variable-step-variable-order integration algorithm and the simultaneous solution of algebraic and differential equations. These algorithms are suitable for efficient simulation of short and long-term dynamics.

The time step is controlled by the accuracy of the simulation. The 'local truncation error' at every step is used to measure the accuracy. For example, if high frequency oscillations are present in the simulation trajectory, the time step will be forced to a small value. On the other hand, if the trajectories become smother the time step is allowed to increase. Also, in case of system instability the step will be forced to very small values. Typically the time step varies from 0.001s to 40s.

The main advantages of a variable time step are:

- More accurate simulation.
- More robustness.
- Simulations are usually faster.
- The same models can be used for fast and slow dynamic phenomena.

Organon also has Energy Function algorithms embedded. This allows for early termination facilities and calculation of energy margins.

Power system model is represented by a set of differential and algebraic equations as follows

$$\dot{y} = f(y, x, t) \tag{6.1}$$

$$0 = g(y, x, t)\,. \tag{6.2}$$

The predictor-corrector Linear Multistep – LM approach is used for numerical solution of (6.1-6.2). The LM integration formulae is given by

$$\sum_{i=0}^{j} \alpha_i y_{n+i} = h \sum_{i=0}^{j} \beta_i f_{n+i} \tag{6.3}$$

where α_i e 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.

Organon uses a combination of LM integration methods of Adams-Bashforth-Moulton - ABM - e BDF families [8-11]. The methods are implemented with variable step and variable order (1 or 2). Typically the time step varies from 0.001 to 40s. The parameters for the integration methods, Eq. (6.3), are given in Table I and II.

Table I- Predictor

Parameter	ABM		BDF	
	1^a ord	2^a ord	1^a ord	2^a ord
α_0	-1	-1	-1	-1
α_1	1	1	1	3
α_2	-	-	-	-3
α_3	-	-	-	1
β_0	1	-0.5	1	-
β_1	-	1.5	-	-

Table II - Corrector

Parâmetros	ABM		BDF	
	1^a ord	2^a ord	1^a ord	2^a ord
α_0	-1	-1	-1	1/3
α_1	1	1	1	-4/3
α_2	-	-	-	1
β_0	-	0.5	-	-
β_1	1	0.5	1	-
β_2	-		-	2/3

The ABM method is used for differential equations. The BDF method is used for algebraic equations and first order differential equations with time constant below a specified threshold.

As can be observed the first order ABM and BDF methods are the same and well known as Backward Euler method. The second order ABM is known as Trapezoidal method.

6.1 Solution Approach

Applying the integration method to (6.1,6.2), the following set of algebraic equations result.

$$0 = y_n - \beta_n h f(y_n, x_n) - C \tag{6.4}$$

$$0 = g(y_n, x_n)\,, \tag{6.5}$$

where *C* is the weight sum of y and $\dot{y}$ at previous steps ($\leq t_{n-1}$).

Newton method is used to solve (6.4,6.5), as it provides superior performance and is fundamental for variable time-step implementation.

6.2 Variable Order Variable Step

The right choice of integration method and algebraic equation approach allow accurate simulation and optimization of time-step.

The algorithm must be able to correctly decide about the possibility of increasing the step or the need of decreasing it. If the step is larger than allowed there is the simulation error increases and may end up in numerical instability. If the time-step is smaller than necessary the simulation is inefficient. The time step control is based on keeping local truncation error below a given tolerance. O controle do passo de integração se baseia na estimação do erro de truncamento local [11] a cada passo.

The mechanism for time step change is as follows. At the end of every step, it is verified if the estimated local truncation error is below the tolerance. If it is, the step is accepted and it is checked if there is room for increasing it (very low error). If it is not, the step is rejected and a smaller step is calculated.

Every time the step is going to change or there is a detection of numerical instability, the integration order is also evaluated. The order that results in smaller error is selected.

The local truncation error is estimated based on [22]

$$E_k = h^{k+1} y^{(k+1)} / (k+1)! \tag{6.6}$$

where *k* is the current order.

The largest time step $\bar{h}$ to keep the error within tolerance is calculated as

$$\bar{h} \approx h\left[\tau / E_k\right]^{1/(k+1)} \tag{6.7}$$

where τ is the tolerance.

7 Energy Function

Organon uses energy function and related techniques for computing energy margins, system stability and preventive actions.

System stability is determined by the use of the following Dot Product

$$f = Pac^T \Delta\theta \tag{7.1}$$

where P_{ac} and $\Delta\theta$ are the vector of generators accelerating power and angle deviation respectively. Both quantities are referred to the Center of Inertia – COI.

Organon also uses the individual energy function concept where each generator is mapped into a one-machine-infinite-bus (OMIB) equivalent. This approach allows for individual machine energy margin computation.

Organon usa funções de energia para computação de margens de energia, estabilidade do sistema e ações preventivas.

Individual and cluster stability margins are computed by means of individual and cluster numerical energy functions. The potential energy is given by

$$Vpe_i = \int Pac_i d\theta \tag{7.2}$$

and the kinetic is given by

$$Vke_i = \frac{1}{2} M\omega_i^2 \tag{7.3}$$

The energy is computed along simulation. If one or more synchronous machines loose synchronism, the not dissipated energy is computed (negative energy margin). If the system is stable, the energy distance to instability (positive margin) is computed. Those values are converted to MW to generate individual or cluster generation limits. Computation of negative margins is much simpler than positive margins, which is in fact estimated. The generation limits are obtained through the compensation scheme.

These techniques allow knowing the critical machines and respective margins without visual inspection of angle trajectories (plottings).

8 Prony Analysis

Prony analysis is used to decompose the angle trajectories in their spectral representation, allowing for modal analysis of system disturbances. In Organon this can be done to any selected trajectory, but for automated processes the angle trajectories are used. Prony Analysis is given by.

$$\hat{\theta}(t) = \sum_{i=1}^{n} B_i e^{\lambda_i t} \tag{8.1}$$

Where $B_i \in C$ 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 $\theta(t)$ least square.

The first step consists of representing the angle trajectories by means of a linear prediction model. Then this model is mapped onto the signal spectral representation. Given the sequence

$$(\theta_i), \quad i = 0,1,....,n \tag{8.2}$$

where θ_i is the machine angle at instant *i*, a linear prediction model (17) is fitted to it.

$$\theta_{k+p} = a_1\theta_{k+p-1} + \cdots + a_p\theta_k \tag{8.3}$$

The complex poles λ_i are the roots of polynomial (8.3). Replacing $\theta(t)$ by $\hat{\theta}(t)$ in (8.1), it can be solved for B_i.

These residues can be used to identify the dominant poles. The oscilation models are given by λ_i. It can be also seen that (8.1) can give the mode shape for the modes.

9 Security Assessment

This Section provides a basic characteristic of the Security Assessment (SA) activity and describes its operation and use. The SA activity has been developed for both the on-line and off-line applications. In this document, the focus is on the off-line use but some features essential for the real-time on-line operation are also outlined.

Security assessment is a process that takes part of a daily power system operation and planning. The need for security assessment becomes more critical because power systems become more stressed and deregulation brings uncertainties to the operation environment. The main benefits of security assessment are improved reliability, better use of system resources, and maintenance schedule flexibility.

For any practical method of solving security assessment problem, computational speed and accuracy must be cautiously compromised. The computational speed is a very important requirement because of the complexity of the problem, which may require solving hundreds of cases for a particular scenario in a reasonably short period of time. Accuracy is also important for security and economic reasons because it can improve the typically conservative assessments imposed as a precaution for uncertainty of the simulation results.

The SA activity is a comprehensive system of analytical tools that is capable of handling static and dynamic security assessment problems. The approaches adopted to cover all aspects of those problems within one program are the conventional power flow (PF) and continuation power flow (CPF) methods for the static assessment and the time-domain simulation (TDS), that encompasses medium-term effects, for the dynamic assessment.

The process of generating voltage stability and thermal limits is fully automated within the current Version. The transient stability limits can be calculated at present only manually. The automation of the TDS process as well as integration of Optimum Power Flow (OPF) and Sensitivity Analysis tools are planned for the next development and will be included in following versions of the SA program. Because SA has been developed as a parallel-processing program, it can be executed either sequentially on a single processor workstation or concurrently on a multiprocessing hardware such as distributed processing networks. The latest feature is particularly important for online application to meet the real-time or faster-than-real-time requirements under energy management system (EMS). A special interface program, Static Translator, has also been developed to capture the post State Estimator system data from EMS and to convert it to a standard PSS/E format that can be used as an input to SA system.

In the following sections SA characteristics and instructions on how to use the program are provided in details.

9.1 Methodology

The SA activity is designated for execution on a cluster of dedicated computers. A group of processors forms a Distributed Processing System (DPS), operating in a Master-Slave configuration as shown below:

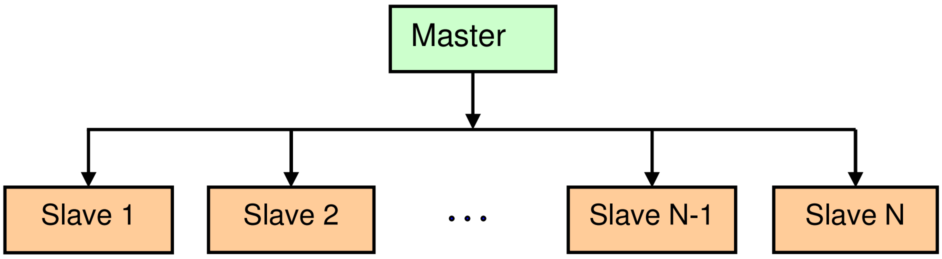

Fig. 9.1 - DSA Processing Network Configuration

Inter-process communication within SA processing network is based on the Message Passing Interface (MPI), a standardized and portable message passing system.

This general configuration, in one extreme, can be reduced to two processes running on a single processor machine such as Windows NT computer. In this case, Master process assigns tasks in a sequential manner to the Slave process, which performs each task and reports back to Master. Based on the collected results, Master generates diagnostics regarding the system security assessment.

In the other extreme case, the configuration may consist of many processors operating in parallel, which can perform all tasks simultaneously and report to Master as soon as they are finished. The overall time of collecting results is short and thus Master can come out with diagnostics much faster.

For a specific problem such as Contingency Analysis, Master forms a Task Assignment list, which are then assigned to Slaves for execution. The Task Assignment specifies what task (Contingency) must be solved and what tool need to be used for the solution. The available tools within DSA systems consist of conventional power flow (PF) and continuation power flow (CPF) methods for the static security assessment and the time-domain simulation (TDS) for the dynamic security assessment.

Two layers can be distinguished in the DSA system architecture:

A. Simulation Shall – Master controlling the whole concurent execution

B. Applications – Slaves using tools to solve assigned problems

Figure below shows a general overview of the DSA system.

The bottom layer is formed by general-purpose applications, which function is to process a single task at a time and report back the results to the upper layer (simulation shell). The results includes the identified problem, its location and respective margin. The shell layer assigns the tasks to the application layer, as defined by the methods, collects the results, generates diagnostics and decides what to do next according to the criteria.

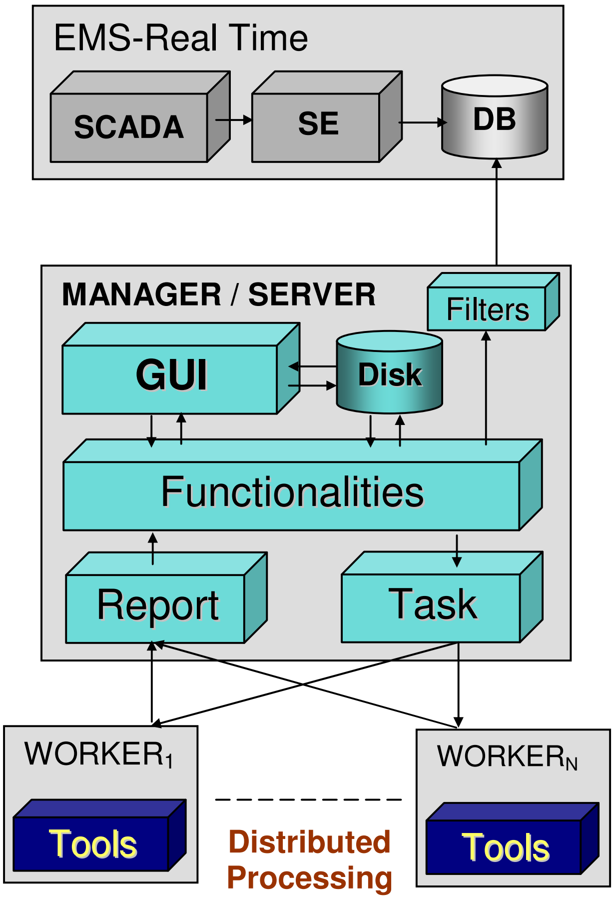

Fig. 9.2 - DSA System for Real Time

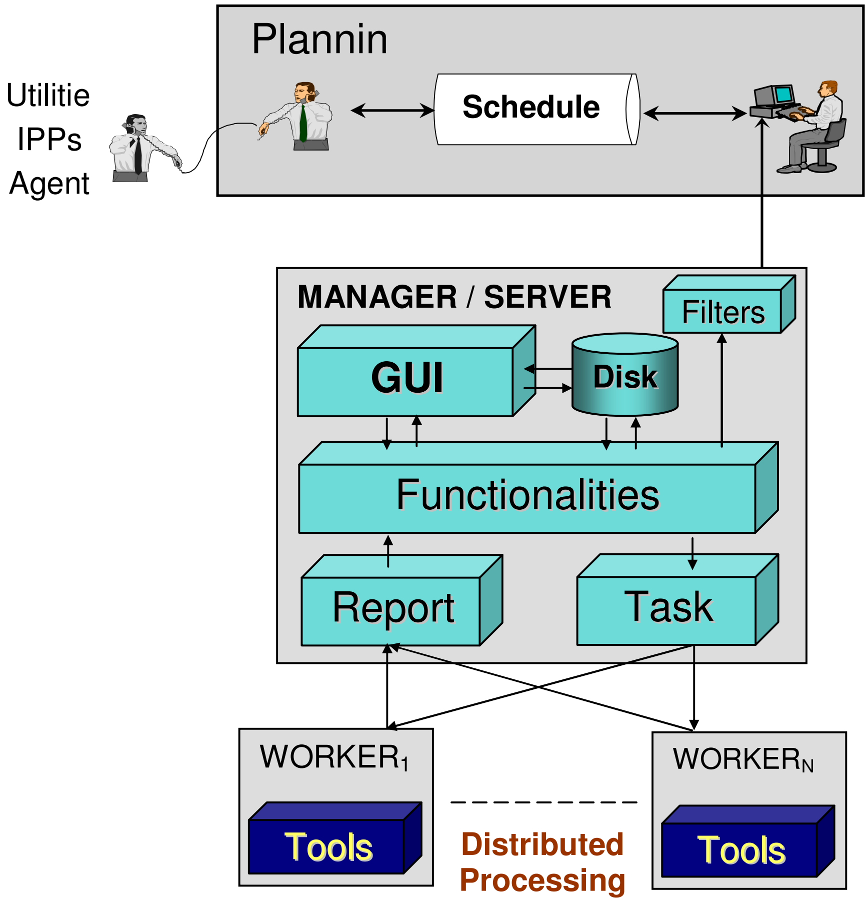

Fig. 9.3 - DSA System for Planning Studies

A typical security assessment requires a load flow (PF and CPF) to check the steady-state condition and a time simulation (TDS) to check for transient stability. The power flow application is generally used in Dynamic Security Assessment - DSA as a mean of generating initial conditions or the pre-contingency Base Case which is sent to each Slave.

For each contingency, an appropriate line circuit is removed and another power flow is applied to generate post-contingency Base Case. At this point either the continuation power flow (CPF) or the time domain simulation (TDS) can be applied to assess voltage and stability limits.

Continuation Power Flow (CPF) is used as a robust mechanism to move around the security region. An automated process is implemented in the DSA program. The procedure starts from an operating condition inside the security region and searches radially for the boundary in different directions around this point. Voltage stability, thermal limits and voltage drop are monitored along the movement in each direction.

The procedure to compute the security region using time-domain simulation is not automated yet.

9.1.1 Static VSA

In order to perform voltage stability assessment (VSA) for a given operating condition, a power flow program must be applied very extensively. It is much more effective, however, if for this type of analysis a Continuation Power Flow (CPF) method is utilized. In this method, all changes of system generation and load are subjected to a change of a single parameter λ, as follows:

For each generation unit: $P_g = P_{go} + \lambda * KP_g$ (9.1)

For each system load: $P_l = P_{lo} + \lambda * KP_l$ (9.2)

The constants KP_g's and KP_l's in Eq. (9.1) and (9.2) represent the rates of change of generation and load units respectively. They can be positive for increasing MWs or negative for decreasing MWs.

All derivatives with respect to continuation parameter λ are included in the Jacobian and solution for any point on the PV curve, including the nose point, can be achieved without running into matrix singularity problems. This is advantageous over a conventional power flow method, which often fails to produce a reasonable solution close to the nose of a PV curve.

The process of performing Contingency Analysis within DSA, is based on the CPF method applied to post-contingency cases for all possible direction of change of the system generation.

In order to allow automatic process of generation change, it is assumed that three generation groups can be distinguished in the system:

Group(1): first group of electrically close generators within the area of study

Group(2): second group of electrically close generators within the area of study

Group(3): swing group of generators outside the area of study used to balance the net change of generation in the system

The system load is kept constant while generation in Groups 1 and 2 are changing during the CPF run by means of increasing continuation parameter λ. The direction of movement is defined by growth factors KP_g's set for generators in Groups 1 and 2. Group 3 generation is simply used to balance the total generation change.

Because Group 3 generators are outside of the study area, the process of balancing the generation change results in changes of the inter-area exchange (import/export).

For each direction, the generation is changed until voltage stability limit (nose point on PV curve) or the generation limit is reached. Scanning through all possible generation patterns, achieved by moving along radial directions around the system operating point, results in generating a contour of stability region for the study area as illustrated in Figure below.

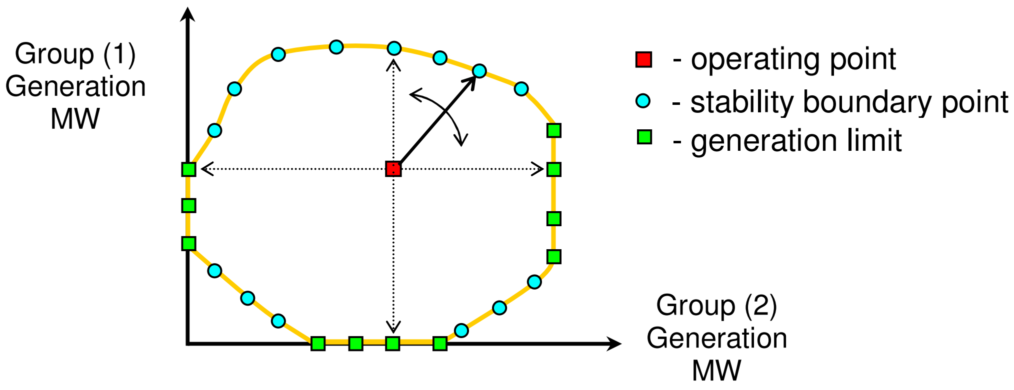

Fig. 9.4 - Voltage Stability Margin Computation in DSA

The limiting point and the reason for the limit is stored for each radial direction. If thermal limits or voltage drop criterions were violated during the movement, the violation points are also stored (indicating critical branches and critical buses) so that the diagnostics could include extensive information for different aspects of the security assessment problem.

10 Technical Specification

Hardware Platform:	Intel Pentium or AMD processors.
Operational System:	Windows NT, 2000 or XP.
Programming Languages;	Fortran 90/95, C
Distribution Processing:	MPI Standard
Project Development Technology:	Migrating to Object Oriented
Power Flow Solution Methods:	1) Full Newton with off-diagonal pivoting. 2) Full Newton with diagonal pivoting only. 3) Control System Formulation (not published yet). Note: Initialization routines available to these methods.
Power Flow Models:	OLTC (discrete & continuous); MW Phase Shift Control; Multiple generation control of a common bus voltage; Local/remote shunt voltage control (discrete/continuous); Multiple generator units per bus; Multiple load units per bus; ZIP load model.
Time Domain Simulation Method:	VMixed ABM-BDF of 1st and 2nd orders implemented with Variable-Step-Variable-Order algorithm, predictor-corrector mode to convergence, simultaneous solution of algebraic & differential equations, 5 generator models, 26 excitation system models, 7 PSS models, 16 governor models, HVDC model, 2 SVC models, 7 protection system models, 3 AGC models, TCSC model, induction motor model, 3 OLTC models, and other models. Various simulation events and plotting facilities.
Diagnosis Methods:	Numerical energy function for stability margin computation. Prony analysis for electromechanical oscillation analysis.
Graphical User Interface:	Based on Windows API. Dialog boxes for data edition. Single line diagram editor and viewer. List view report. Plotting windows.
Raw data formats:	Own format, PSS/E (Release 26) and Anarede (CEPEL).

Network sizes:	99999 buses.

11 References

11.1 Periodicals:

[1] Neal Balu, at al "On-Line Power System Security Analysis", Proceedings of The IEEE, Vol. 80, No. 2, pp 262-280, Feb 1992.
[2] Stott, B. "Review of Load Flow Calculation Methods", Proceedings of IEEE 62, pp. 916-929, 1974
[3] Peterson, N.M. and Meyer, W.S. "Automatic Adustment of Transformer and Phase-Shifter Taps in the Newton Power Flow", IEEE Trans. on PAS, pp. 103-108, Jan/Feb, 1971.
[4] Sasson, A.M., Treviño, C. and Aboytes, F. "Improved Newton's Load Flow Through a Minimization Techinique", Trans. on PAS, pp. 1974-1981, 1971.
[5] Smed,T., Andersson,G., Sheblé, G.B., and Grigsby, L.L. "A New Approach To AC/DC Power Flow", Transaction on Power Systems, Vol. 6, No. 3, Aug 1991.
[6] B. Stott, "Power System Dynamic Response Calculations", Proceedings of the IEEE, Vol. 67, No. 2, Feb 1979.
[7] J.Y.Astic, A. Bihain and M. Jerosolimski, "The mixed Adams - BDF Variable Step Size Algorithm to Simulate Transient and Long Term Phenomena In Power Systems", IEEE Trans. on PS, Vol. 9, No. 2, May 1994.
[8] Z.Q. Wu, "Single Machine Equal Area Criterion for Multimachine System Stability Assessment Based on Time Domain Simulation", IEEE Power Engineering Review, Vol. 21, No. 11, pp 51-52, November 2001.
[9] D.J. Trudnowski, J.M.Johnson, J. F. Hauer "Making Prony Analysis More Accurate Using Multiple Signals", IEEE Trans on PS, Vol. 14, No 1, Feb 1999.
[10] El.Marsafawy, M.M. and Mathur, R.M. "A New, Fast Technique for Load-Flow Solution of Integrated Multiterminal dc/ac Systems", IEEE Trans. on Power Apparatus and System, PAS-99, No 1, 1980, pp. 246-255.
[11] 'Common Format For Exchange Of Solved Load Flow Data', Working Group on a Common Format for Exchange of Solved Load Flow Data, IEEE Trans. on PAS, pp. 1916-1925, 1973.
[12] Dahlquist, G. 'A Special Stability Problem for linear Multistep Methods', BIT, Vol. 3, pp 27-43, 1963.

11.2 Books:

[13] Monticelli, A. "Fluxo de Cargas em Redes de Energia Elétrica", Editora Edgard Blucher ltda, 1983.
[14] Ramos, D.S. e Dias, E.M. "Sistemas Elétricos de Potência - Regime Permanente", Guanabara Dois, 1983.

[15] D.M. Falcão, 'High Performance Computing in Power System Applications", Lecture Notes in Computer Science, Springer-Verlag, vol. 1215, pp 1-23, 1997.
[16] A. Monticelli “State Estimation in Electric Power System: A Generalized Approach”, Klwer International Series in Engineering and Computer Science, 1999.
[17] Arrillaga, J., Arnold, C.P. and Harker, B.J. “Computer Modelling of Electrical Power Systems”, John Willey and Sons, 1983.
[18] R. L. Burden and J. D. Faires, “Numerical Analysis”, PWS-KENT Publishing Company, 1989.
[19] Luenberger, D.G. ‘Linear and Nonlinear Programming’, Addison-Wesley Publishing Company, 1989.
[20] P.M.Anderson, A A Fouad, ‘Power System Control and Stability’, Iowa State University Press, 1977.
[21] D.M. Falcão, “High Performance Computing in Power System Applications", Lecture Notes in Computer Science, Springer-Verlag, vol. 1215, pp 1-23, 1997.
[22] J. D. Lambert, "Numerical Methods for Ordinary Differential Systems: The Initial Value Problem", Wiley, 1991.

11.3 Papers from Conference Proceedings (Published)::

[23] J.L. Jardim, C.A. da S. Neto, A.P. Alves da Silva, A.C. Zambroni de Souza, D.M. Falcão, C.L.T. Borges, G.N. Taranto, “A Unified On-Line Dynamic Security Assessment System”, Cigré, Paris, France, 27 Ago – 1 Sep 2000.
[24] J.L. Jardim, W. Kwasnicki, C.A. da S. Neto, “Comparison of Static and Dynamic Approaches for Online Voltage Security Assessment”, VIII Symposium of Specialists in Electric Operational and Expansion Planning, Brasilia, DF, Brazil, 2002.
[25] J.L. Jardim, “Online Dynamic Security Assessment: Implementation Problems and Potential Use of Artificial Intelligence”, Power Engineering Society Summer Meeting, 2000. IEEE, Volume: 1, 16-20 July 2000.
[26] V.Ajjarapu, C.Christy, 'The Continuation Power Flow: A Tool for Steady State Voltage Stability Analisys', IEEE PICA, May 91, pp 304-311.
[27] J.L .Jardim, X.V. Filho “Long Term Dynamics: Its Perspective in Brazil”, V Symposium of Specialists in Electric Operational and Expansion Planning, Recife, PE, Brazil, May 1996.
[28] J.L. Jardim, B. Cory, N. Martins, “Efficient Transient Stability Assessment Using Transient Energy Function”, Power Systems Computation Conference – PSCC, Trondheim, Norway, June 1999.
[29] J.L.Jardim, “Utilização de Ferramentas de Simulação Dinâmica de Longa Durançäo na Análise de Fenômenos de Colapso de Tensão e no Treinamento de Operadores”, SNPTEE, Belém, PA, Brazil, 1997.
[30] J.F Hauer "Application of Prony Analysis to the Determination of Modal Content and Equivalent Models for Measured Power System Response”, IEEE Winter Meeting, 215-4 PWRS, 1991.

[31] C.A. Silva Neto, J.L.Jardim, “Application of a Globally Convergent Primal Dual Interior Point Algorithm to the Optimal Power Flow Problem”– VII Symposium of Specialists in Electric Operational and Expansion Planning, 2000, Curitiba, Brazil.
[32] A.M. Leite da Silva, J.L. Jardim, A.M. Rei, J.C.O. Mello “Dynamic Security Risk Assessment”, Power Engineering Society Summer Meeting, 1999 IEEE, Vol. 1, pp 198-205, 18-22 July 1999.
[33] J.L.Jardim, B.Stott “Synthetic Dynamics Power Flow”, IEEE General Meeting, San Francisco, CA, 13-16 June 2005.

11.4 Thesis

[34] J.L Jardim, "Advances in Power System Transient Stability Assessment Using Transient Energy Function Methods", Ph.D. Thesis, Imperial College of Science, Technology and Medicine, University of London, October 1994.

11.5 Technical Reports

[35] A.S. Debs, A.R. Benson “Security Assessment of Power Systems”, System Engineering for Power: Status and Prospects, US Government Printing Office, Washington, pp 144-176, DC, 1975.
[36] B. Meyer, et al, "New Trends and Requirements for Dynamic Security Assessment", Cigré Task Force 38-02-13, Jan. 1997.
[37] J.J. Dongarra, R. Hempel, A.J.G. Hey and D.W. Walker, "A proposal for a user-level message passing interface, in a distributed memory environment", Technical Report TM-12231, Oak Ridge National Laboratory, Februrary, 1993.