High Performance Computing in Power System Applications*

Djalma M. Falcão

COPPE, Federal University of Rio de Janeiro
C.P. 68504, 21945-970, Rio de Janeiro RJ
Brazil

Abstract. This paper presents a review of the research activities developed in recent years in the field of High Performance Computing (HPC) application to power system problems and a perspective view of the utilization of this technology by the power industry. The paper starts with a brief introduction to the different types of HPC platforms adequate to power system applications. Then, the most computer intensive power system computation models are described. Next, the promising areas of HPC application in power system are commented. Finally, a critical review of the recent developed research work in the field, along with prospective developments, is presented.

1 Introduction

Power system simulation, optimization, and control can be included in the category of highly computer intensive problems found in practical engineering applications. Modern power system studies require more complex mathematical models owing to the use of power electronic based control devices and the implementation of deregulation policies which leads to operation close to the system limits. New computing techniques, such as those based on artificial intelligence and evolutionary principles, are also being introduced in these studies. All these facts are increasing even further the computer requirements of power system applications. High Performance Computing (HPC), encompassing parallel, vector, and other processing techniques, have achieved a stage of industrial development which allows economical use in this type of application. This paper presents a review of the research activities developed in recent years in the field of HPC application to power system problems and a perspective view of the utilization of this technology by the power industry. The paper starts with a brief introduction to the different types of high performance computing platforms adequate to power system applications. Then, the most computer intensive power system computation models are described. Next, the promising areas of HPC application in power system are commented. Finally, a critical review of the recent developed research work in the field, along with prospective developments, is presented.

* Invited lecture at the 2nd International Meeting on Vector and Parallel Processing (VECPAR'96), Porto, Portugal, 25-27 September, 1996. To be published in a book of the series Lecture Notes in Computer Science edited by Springer-Verlag.

2 High Performance Computing

The processing capabilities of single processors, despite the substantial increase achieved in the last years, is not high enough to cope with the rising demand observed in in several fields of science and engineering. For that reason, HPC has been relying more and more on the exploitation of concurrent tasks in the programs which can be executed in parallel on computer systems with multiplicity of hardware components. Several types of computer architectures are available for this purpose [1, 2]:

1) *Superscalar Processors* are single processors able to execute concurrently more than one instruction per clock cycle. Its efficiency depends on the ability of compilers to detect instructions that can be executed in parallel. They are used in high performance workstations and in some multiprocessor systems. Examples of superscalar processors are the IBM Power2, DEC Alpha, MIPS R10000, etc.

2) *Vector Processors* are processors designed to optimize the execution of arithmetic operations in long vectors. These processors are mostly based on the *pipeline architecture*. Almost all of the so called supercomputers, like the ones manufactured by Cray, Fujitsu, NEC, etc., are based on powerful vector processors.

3) *Shared Memory Multiprocessors* are machines composed of several processors which communicate among themselves through a global memory shared by all processors. Some of these machines have a few (2-16) powerful vector processors accessing high speed memory. Examples of such architectures are the Cray T90 and J90 families. Others, like the SGI Power Challenge, may have a larger (up to 32) number of less powerful superscalar processors.

4) *SIMD*[1] *Massively Parallel Machines* are composed of hundreds or thousands of relatively simple processors which execute, synchronously, the same instructions on different sets of data (*data parallelism*) under the command of central control unity.

5) *Distributed Memory Multicomputers* are machines composed of several pairs of memory-processor sets, connected by a high speed data communication network, which exchange information by message passing. The processors have a relatively high processing capacity and the number of processors may be large (2-1024). Owing to the possible high number of processors, this type of architecture may also be referred to as massively parallel. Examples of multicomputers are the IBM SP-2, Cray T3D/3E, Intel Paragon, etc.

6) *Heterogeneous Network of Workstations* may be used as a virtual parallel machine to solve a problem concurrently by the use of specially developed communication and coordination software like PVM and MPI [3]. From the point of view of applications development, this computer system is similar to the distributed memory multicomputers but its efficiency and reliability is usually inferior. On the other hand, the possibility of using idle workstations, already available in a company for other purposes, as a virtual parallel machine is attractive from a an economical point of view.

[1] Single Instruction Stream Multiple Data Stream

The development of applications on the HPC architectures described above may follow different programming paradigms and procedures. Parallelism can be exploited at one ore more granularity levels ranging from instruction-level (fine grain parallelism) to subprogram-level (coarse grain parallelism). Superscalar and vector processors, as well as SIMD machines, are more adequate to instruction level parallelism while multiprocessor and multicomputer architectures adapt better to subprogram parallelism. Coarse grain parallelism can be implemented using the shared memory architecture of multicomputers or the message passing paradigm used in multicomputers and network of workstations. The first model is conducive to programs easy to develop and maintain but shared memory multiprocessors are usually more expensive and less scalable than multicomputers. For that reason, HPC manufacturers have been trying to develop smart operating systems that could mimic a shared memory environment on a physically distributed memory system. The detection of parallelism in the code is mostly performed manually by the applications developers. Automatic detection of parallelism is still a challenge for HPC except in the case of superscalar and vector processors.

The gain obtained in moving an application to a parallel computer is measured in terms of the speedup and efficiency of the parallel implementation compared with the *best* available sequential code. *Speedup* is defined as the ratio between the execution time of the best available sequential code in one processor of the parallel machine to the time to run the parallel code in p processors. *Efficiency* of the parallelization process is defined as the ratio between the speedup achieved on p processors to p. In the early stages of applications development for parallel computers these two indexes were almost exclusively the determinants of the quality of parallel algorithms. As more parallel machines became commercially available, and practical applications begin to be actually implemented, other aspects of the problem started to become important. For instance, the cost/performance ratio (Mflops/\$; Mflops=10^6 floating point operations per second) attainable in real-life applications. In other cases, although speedup and efficiencies are not so high, the implementation in a parallel machine is the only way to achieve the required speed in the computations.

3 Potential Areas for HPC Applications

The major impact of HPC application in power systems may occur in problems for which conventional computers have failed so far to deliver satisfactory performance or in areas in which the requirement of more complex models will demand extra computational performance in the future. Another possibility is in the development of a new generation of analysis and synthesis tools exploiting the potential offered by modern computer technology: intelligent systems, visualization, distributed data basis, etc. Some candidate areas are commented in the following.

3.1 Real-time Control

The complexity and fast response requirement of modern Energy Management System software, particularly the components associated with security assessment, make this area a potential candidate for HPC use [4, 5]. In most of the present implementations of security control functions only static models are considered. This deficiency imposes severe limitations to their ability of detecting potentially dangerous situations in system operation. The consideration of dynamic models, associated with the angle and voltage stability phenomena, require a computational power not yet available in power system computerized control centers. Even considering only static models, problems like security constrained optimal power flow are too demanding for the present control center hardware. Taking into consideration the present trend towards a distributed architecture in control center design, the inclusion of parallel computers as number crunching servers in this architecture may offer a possibility to attend this high computing requirement. Another possibility would be the utilization of the control center network of workstations as a virtual parallel machine to solve problems requiring computational power above the one available in each of the individual workstations.

3.2 Real-time Simulation

The capacity to simulate the dynamic behavior of the power system, taking into consideration electromechanical and electromagnetic transients, in the same time scale of the physical phenomena, is of great importance in the design and testing of new apparatus, control and protection schemes, disturbance analysis, training and education, etc. [6]. Real-time simulation can be performed using analog devices (reduced model or electronic devices) or digital simulators. Hybrid simulators combine these two type of simulation technique. Digital simulators are more flexible and smaller in size due to the processors very large scale integration technology. Another advantage of digital simulators is the facility to manipulate and display results using sophisticated graphical interfaces. For a long period, analog simulation was the only way to obtain real-time performance of fast phenomena in practical size systems. The progress in digital hardware technology, however, is changing this scenario: massively parallel computer systems are nowadays able to deliver the computer power necessary to drive fully digital or hybrid real-time power system simulators. In this type of application, the high performance computer must be dedicated to the process owing to the need to interface it physically to the equipment being tested.

3.3 Optimization

Power system is a rich field for the application of optimization techniques. Problems range from the classical economic dispatch, which can be modeled straightforwardly as a non-linear programming problem and solved by gradient techniques, to the stochastic dynamic programming formulation of the multireservoir optimization problem. Other interesting and complex problems are the transmission and distribution network expansion planning and contingency constrained optimal power flow, to cite only the most reported studies in the literature. In most of these problems, a realistic formulation leads to highly nonlinear relationships, nonconvex functions, discrete and integer variables, and many other ill-behaved characteristics of the mathematical models. Some of these problems are combinatorial optimization problems with exponential increase in computer requirement. Another difficulty always present is high dimension: problems involving thousands of constraints are quite common. Most of the problems formulations are adequate for the application of decomposition techniques what allows efficient utilization of parallel computers in their solution. Recently, heuristic search optimization techniques, like Simulated Annealing, Genetic Algorithms, Evolutionary Computation, etc., have been proposed to solve some of these problems. These techniques, also, present great potential for efficient implementation on HPC platforms.

3.4 Probabilistic Assessment

This type of power system performance assessment is becoming more and more accepted as practical tools for expansion and operational planning. Some studies involving probabilistic models, like the composite reliability assessment of generation-transmission systems, require great computational effort to analyze realistic size power system even if only simplified models are used such that static representation, linearization, etc. [7, 8]. The inclusion of more realistic models make the problem almost intractable in present conventional computers. On the other hand, most of the methods used in such calculations (Monte Carlo simulation, enumeration techniques, etc) are adequate for massively parallel processing. This is one of the most promising areas of HPC application to power systems.

3.5 Intelligent Tools for Analysis and Synthesis

Power system operational and expansion planning require a time consuming and tedious cycle of scenario data preparation, simulation execution, analysis of results, and decision to choose other scenarios. A complete study involves several different types of simulation software (power flow, transient stability, electromagnetic transients, etc.) not well integrated and compatible. Present day software tools, although taking advantage of some of the modern computer facilities like graphic interfaces and integrated data bases, does not fully exploit all the hardware and software resources made available by the computer industry.

Taking into consideration the scenario depicted in the introduction of this paper, it is believed that the power system computational tools of the future will need to fulfill the following requirements:

- *Robustness* to cope with analysis of stressed systems;
- *Friendliness* to relieve engineers from routine work;

– *Integration* to broad the engineers ability of analysis;
– *Learning capability* to automatically accumulate experience;
– *Fast response* to speed up analysis and decision making.

Robustness can be achieved by better modeling and *robust algorithms* specially developed to cope with analysis of extreme operating conditions. Friendliness can be greatly improved by the use of the highly sophisticated *visualization tools* presently available provided that the power system engineers could find efficient ways to synthesize graphically the results of power system studies. Integration and learning capabilities can be achieved by the use of *intelligent systems* techniques like Expert Systems, Artificial Neural Networks, Fuzzy Logics, etc. The integration of all these computational tools to perform studies in large scale power system models would certainly need a *HPC environment* to achieve the required fast response. A visual summary of the structure of such computational tool is shown in Figure 1.

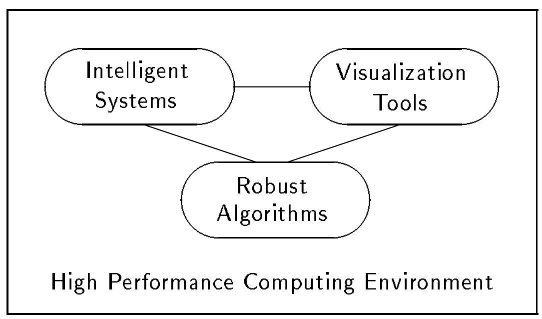

Fig. 1. Power system computational tool of the future

4 Literature Review

This section reviews the main areas of HPC application to power systems problems. The review is not meant to be exhaustive. Only areas in which a substantial amount of work has been published or areas that, in the author's opinion, has a great chance of becoming relevant in the near future, are covered.

4.1 Simulation of Electromechanical Transients

The simulation of electromechanical transients has been one of the most studied areas of application of HPC in power systems. This interest comes from the possibility it opens to real-time dynamic security assessment and the development of real-time simulators. The mathematical model usually adopted in this kind of simulation consists of a set of ordinary non-linear differential equations, associated to the synchronous machine rotors and their controllers, constrained by a set of non-linear algebraic equations associated to the transmission network, synchronous machine stators, and loads [9, 10]. These equations can be expressed as:

$$\dot{x} = f(x, z) \tag{1}$$

$$0 = g(x, z) \tag{2}$$

where f and g are non-linear vector functions; x is the vector of state variables; and z is the vector the algebraic equations variables.

In the model defined in (1) and (2), the differential equations representing one machine present interaction with the equations representing other machines only via the network equations variables. From a structural point of view, this model can be visualized as shown in Figure 2: clusters of generators are connected by local transmission and subtransmission networks and interconnected among themselves and to load centers by tie-lines. In the sequential computer context, several solution schemes have been used to solve the dynamic simulation problem. The main differences between these schemes are in the numerical integration approach (implicit or explicit) and in the strategy to solve the differential and algebraic set of equations (simultaneous or alternating). Implicit integration methods, particularly the trapezoidal rule, have been mostly adopted for this application. The most used schemes are the Alternating Implicit Scheme (AIS) and the Simultaneous Implicit Scheme (SIS)[9].

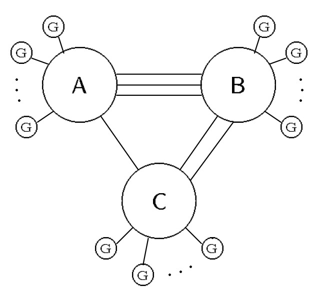

Fig. 2. Dynamic Simulation Model Decomposition

The difficulties for the parallelization of the dynamic simulation problem in the AIS are concentrated on the network solution. The differential equations associated with the synchronous machines and their controllers are naturally decoupled and easy to parallelize. On the other hand, the network equations constitute a *tightly coupled* problem requiring ingenious decomposition schemes and solution methods suitable for parallel applications. The SIS also requires the parallel solution of linear algebraic equations sets in every integration step, with difficulties similar to the ones described for the AIS.

A basic numerical problem in both simulation schemes, as well as in several other power system problems, is the parallel solution of sets of linear algebraic equations. Direct methods, like LU factorization, have been dominating this application on conventional computers. Several schemes for the efficient solution of linear algebraic equations on vector computer have also been proposed [11, 12, 13] in the context of power system simulation. In most of these schemes, only the substitution phase of the direct methods are vectorized. If parallel computers are considered, however, the hegemony of direct methods is no more guaranteed. In several other engineering and scientific fields, parallel implementations of iterative methods have shown superior performance. Among the most successful iterative methods are the ones belonging to the Conjugate Gradient (CG) category [14, 15, 16, 17, 18]. The parallelization of the network equations solution requires the decomposition of the set of equations in a number of subsets equal to the number of processors used in the simulation. An adequate decomposition is fundamental to the success of the parallel solution and need to take into consideration factors like computation load balancing, convergence rate of the iterative algorithms, etc. [19].

In the last decade or so, several parallel methods were proposed for the solution of the dynamic simulation problem. In the following sections, some of these methods are reviewed.

Spatial Parallelization Methods in this category exploit the structural properties of the the equations to be solved in each integration step of the conventional simulation schemes (AIS or SIS). Four methods are briefly described below:

1) *The Parallel VDHN* [20] consists in a straightforward parallelization of the *Very Dishonest Newton Method (VDHN)*, applied to the SIS, simply identifying tasks that can be performed concurrently and allocating them among the processors. This method was implemented on the parallel computers Intel iPSC/2 (distributed memory) and Alliant FX/8 (shared memory) and tests performed with the IEEE 118 bus and US Midwestern system with 662 buses. The results show speedups slightly superior for the iPSC/2 with a strong saturation with the increase in the number of processors. The maximum obtained speedup was 5.61 for 32 processors (efficiency = 17.5%).

2) *The Parallel Newton-W matrix Method* [21] uses a parallel version of the Sparse Matrix Inverse Factors [22] in the SIS. The method was tested on the shared memory Symmetry parallel computer and the same test systems used in the work cited in the previous item. The results show a worse performance of this method when compared to the parallel VHDN with an slowdown of 10% to 30% depending on the chosen partitions.

3) *The Parallel Real-Time Digital Simulator* [23] is based on the AIS using the trapezoidal integration method. One processor is associated to each network bus. The differential equations corresponding to each generator and its controllers are solved on the processor assigned to the bus in which the generator is connected. The network equations are solved by a Gauss-Seidel like method also allocating one equation to each processor. Therefore, the number of processors required

to perform the simulation is equal to the number of network buses. Reported results with a 261 buses network, on a 512 node nCube parallel computer, show that the required cpu time is not affected by the system dimensions. However, it is doubtful weather this property can be kept valid for larger system taking into consideration that the number of iterations required by the Gauss-Seidel algorithm increases considerably with system size. This approach exhibits low speedup and efficiency measured by the traditional indexes. However, its impact in the power system research community was considerable as it has demonstrated the usefulness of parallel processing in solving a real world problem.

4) *The Hybrid CG-LU Approach* [14, 16] is based on the AIS, the decomposition of the network equations in a Block Bordered Diagonal Form (BBDF), and a hybrid solution scheme using LU decomposition and the CG method. The equations are solved by Block-Gaussian Elimination in a two phase scheme: firstly, the interconnection equations are solved by the CG method; secondly, a number of independent sets of equations, corresponding to the diagonal blocks of the BBDF, are solved by LU factorization one in each processor. This method has the disadvantage of applying the CG method to a relatively small system of equations (interconnection block). Owing to the BBDF characteristics, the interconnection matrix is usually well-conditioned. However, the use of a Truncated Series preconditioner improves the performance of the method. Results of experiments performed with this method, as well as with other simulation methods based on the CG's methods, are presented in a later section of this paper.

5) *The Full CG Approach* [16] solves the network equations as a whole by a block-parallel version of the Preconditioned CG method. The network matrix is decomposed in such a way that the blocks in the diagonal are weakly coupled to each other, i.e., in a Near Block Diagonal Form (NBDF). The NBDF is equivalent to the decomposition of the network in subnetworks weakly coupled. A block-diagonal matrix, obtained from the NBDF neglecting the off-diagonal blocks, is used as a preconditioner.

Waveform Relaxation This method [24, 25, 26] consists in the decomposition of the set of equations describing the power system dynamics into subsystems weakly coupled and to solve each subsystem independently for several integration steps to get a first approximation of the time response. The results are, then, exchanged and the process repeated. The advantages of this method are the possibility of using different integration steps for each subsystem (multi-rate integration) and to avoid of the need to solve large sets of linear algebraic equations. However, the difficulty to obtain an efficient decomposition of the differential equations set is a major drawback in the practical application of this method.

Space and Time Parallelization This class of methods follows the idea introduced in [27] in which the differential equations are algebrized for several integration steps, called integration windows, and solved together with the algebraic equations of this window by the Newton method. Two methods are briefly described below:

1) *The Space and Time CG Approach* [16] uses two versions of the CG method (Bi-CG and Bi-CGSTAB), suitable for asymmetric sets of linear equations, to solve the resulting set of equations which presents a *stair-like* coefficient matrix. The parallelization process follows a natural choice: the equations corresponding to each integration step are assigned to different processors. Therefore, a number of integration steps equal to the number of processors available in the parallel machine can be processed concurrently. A block-diagonal preconditioning matrix, derived from the coefficient matrix, was found to be effective for both CG methods.

2) *The Space and Time Gauss-Jacobi-Block-Newton Approach* [28, 29] uses a slightly different formulation of the waveform relaxation concept. The discretization of the differential equations is performed for all integration steps simultaneously resulting in an extremely large set of algebraic equations. In a first work [28], this set of equations was solved by a parallel version of the Gauss-Jacobi method with a poor performance. In a second work [29], a method called Gauss-Jacobi-Block-Newton Approach was used with better results. This method consists, essentially, in the application of the VDHN method to the equations associated to each integration step and, then, to apply the Gauss-Jacobi globally to all integration steps. Both works present results only for simulations of parallel implementation.

Conjugate Gradient Approach Results The Hybrid CG-LU, Full CG, and Space and Time CG methods described above [14, 16] were tested using different test systems, including a representation of the South-Southern Brazilian interconnected system with 80 machines and 616 buses. The tests were performed on the iPSC/860 computer and in a prototype parallel computer using the Transputer T800 processor. Despite the difficulties in parallelizing this application, the results obtained in these tests showed a considerable reduction in computation time. The CG methods presented adequate robustness, accuracy, and computation speed establishing themselves firmly as an alternative to direct methods in parallel dynamic simulation. Moderate efficiencies and speedups were achieved, particularly in the tests performed on the iPSC/860, which are partially explained by the relatively low communication/computation speed ratio of the machines used in the tests. It is believed that in other commercially available parallel machines, the studied algorithms will be able to achieve higher levels of speedup and efficiency.

4.2 Simulation of Electromagnetic Transients

In the usual model of the power network for electromagnetic transient simulation, all network components, except transmission lines, are modeled by lumped parameter equivalent circuits composed of voltage and current sources, linear and non-linear resistors, inductors, capacitors, ideal switches, etc. These elements are described in the mathematical model by ordinary differential equations which are solved by step-by-step numerical integration, often using the trapezoidal rule, leading to equivalent circuits consisting of resistors and current sources [30].

Transmission lines often have dimensions comparable to the wave-length of the high frequency transients and, therefore, have to be modeled as distributed parameter elements described mathematically by partial differential equations (wave equation). For instance, in a transmission line of length ℓ, the voltage and current in a point at a distance x from the sending end, at a time t, are related through the following equation:

$$-\frac{\partial E(x,t)}{\partial x} = L\,\frac{\partial I(x,t)}{\partial t} + R\,I(x,t) \tag{3}$$

$$-\frac{\partial I(x,t)}{\partial x} = C\,\frac{\partial E(x,t)}{\partial t} + G\,E(x,t) \tag{4}$$

where $E(x,t)$ and $I(x,t)$ are $p \times 1$ vectors of phase voltage and currents (p is the number of phases); R, G, L and C are $p \times p$ matrices of the transmission line parameters.

The wave equation does not have an analytic solution in the time domain, in the case of a lossy line, but it has been shown that it can be adequately represented by a traveling wave model consisting of two disjoint equivalent circuits containing a current source in parallel with an impedance in both ends of the line as shown in Figure 3. The value of the current sources are determined by circuit variables computed in past integration steps (*history* terms).

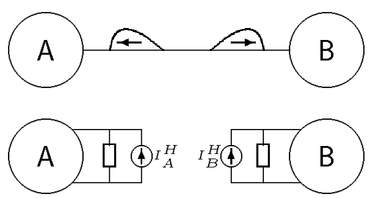

Fig. 3. Transmission Line Model

This model is nicely structured for parallel processing: subnetworks of lumped parameter circuit elements connected by transmission lines, representing a group of devices in a substation for instance, can be represented by sets of nodal equations that interface with other groups of equations by the variables required to calculate the current sources in the transmission line equivalent circuits. The exploitation of this characteristic of the network model, in the partitioning of the set of equations for parallel processing, often correspond to a geographical mapping of the power system onto the multiprocessor topology as shown below for a two subnetwork example:

$$\begin{bmatrix} G_A & 0 \\ 0 & G_B \end{bmatrix} \begin{bmatrix} E_A \\ E_B \end{bmatrix} + \begin{bmatrix} \mathcal{F}_A(E_A) \\ \mathcal{F}_B(E_B) \end{bmatrix} = \begin{bmatrix} I_A^S \\ I_B^S \end{bmatrix} + \begin{bmatrix} I_A^L \\ I_B^L \end{bmatrix} + \begin{bmatrix} I_A^C \\ I_B^C \end{bmatrix} + \begin{bmatrix} I_A^H \\ I_B^H \end{bmatrix} \tag{5}$$

where G_A and G_B are conductance matrices related to linear branch elements; $\mathcal{F}_A$ and $\mathcal{F}_B$ are non-linear functions related to non-linear branch elements; E_A and E_B are vectors of the unknown node voltages; I_A^S, I_B^S are nodal current injections corresponding to independent sources, I_A^L, I_B^L, I_A^C, I_B^C are the nodal injection currents related to the equivalent circuits of inductors and capacitors, and I_A^H, I_B^H are the nodal current injections present in the transmission line models.

Since $I^S(t)$ is known and $I^H(t)$, $I^L(t)$, and $I^C(t)$ depend only on terms computed in previous integration steps, $E_A(t)$ and $E_B(t)$ can be computed independently in different processors. The computation of the terms $I_A^S(t)$, $I_B^S(t)$, $I_A^L(t)$, $I_B^L(t)$, $I_A^C(t)$, and $I_B^C(t)$ can also be executed in parallel, since the equations related to branches in a particular subnetwork depend only on nodal voltages belonging to the same subnetwork. However, the term $I_A^H(t)$ depend on the past terms $I_B^H(t-\tau)$ and $E_B(t-\tau)$, as well as $I_B^H(t)$ depend on the past terms $I_A^H(t-\tau)$ and $E_a(t-\tau)$. Since such terms have already been evaluated in previous integration steps, the processors must exchange data in order to each one be able to compute its part of the vector $I^H(t)$.

Several parallel implementations of the electromagnetic transients simulation methodology described above are reported in the literature. In [31], [32], and [33], prototypes of parallel machines based on different networks of Transputer processors were used for these implementations, with excellent results in terms of speedup for some realistic size test systems. In [34], the implementation is performed on a workstation based on a superscalar computer architecture (IBM RISC System/6000 Model 560). The results obtained in this implementation, for medium size systems, indicate the possibility of achieving real-time simulation.

4.3 Small-Signal Stability

Power system oscillations are the result of insufficient damping torque between generators and groups of generators. This situation may arise as a consequence of heavily loaded lines, weak interconnections, high gain excitation systems, etc. Oscillations caused by small disturbances, like the normal load variation, may reach amplitudes high enough to cause protective relays to trip lines and generators which in turn causes partial or total system collapse. This type of problem can be studied using linearized versions of the power system dynamic model given by (1) and (2). The great advantage of this approach is the possibility of the performance assessment of control schemes without time simulation. This assessment is conducted through linear control systems analysis methods. A large scale numerical problem resulting from the application of these techniques is the computation of eigenvalues and eigenvectors associated with the state matrix of the linearized system model [10].

A linearized version of (1) and (2) at an operating point (x_0, z_0) is given by

$$\begin{bmatrix} \Delta\dot{x} \\ 0 \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} \tag{6}$$

where $J_1, \ldots, J_4$ are Jacobian matrices evaluated at the linearization point. The power system state transition equation can be obtained eliminating Δz from (6):

$$\Delta \dot{x} = (J_1 - J_2 J_4^{-1} J_3)\ \Delta x = A\ \Delta x \tag{7}$$

where A is the system state matrix whose eigenvalues provide information on the local stability of the nonlinear system. Efficient algorithms to obtain the dominant eigenvalues and eigenvectors of A for large scale systems do not require the explicit calculation of this matrix [35]. These algorithms can be directly applied to (6), named the *augmented* system, whose sparse structure can be fully exploited to reduce both cpu time and memory requirements. These methods require repeated solutions of linear equation sets of the form [36]:

$$\begin{bmatrix} J_1 - qI & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix}^{(k)} = \begin{bmatrix} r \\ 0 \end{bmatrix}^{(k)} \tag{8}$$

where w, v are unknown vectors; q is a complex shift used to make dominant the eigenvalues close to q; I is the identity matrix; r is a complex vector; and k is the iteration counter. These sets of equation are independent and their solution can be obtained concurrently on different processors. This property make the eigenvalue problem well suited for parallel processing.

In the work reported in [36] and [37], algorithms for the parallel solution of the eigenvalue problem for small-signal stability assessment, using the above formulation, are described and the results of tests with models of a large practical power systems are presented. A first investigatory line of research was based on the parallelization of the Lop-sided Simultaneous Iterations method [36]. The obvious parallel stratagem used was to carry out each trial vector solution on a different processor. Results obtained in tests performed on the iPSC/860 parallel computer, using two large scale representations of the Brazilian South-Southern interconnected power system, presented computation efficiencies around 50%. A second approach to the problem uses a Hybrid Method [37] resulting from the combination of the Bi-Iteration version of the Simultaneous Iteration algorithm and the Inverse Iteration method. The Hybrid algorithm exploits the fast eigenvalue estimation of the Bi-Iteration algorithm and the fast eigenvector convergence of the Inverse Iteration algorithm whenever the initial shift is close to an eigenvalue. In the Inverse Iteration stage, the Hybrid algorithm allows perfect parallelization. The results obtained indicate a superior performance of this method both in terms of computation speedup and robustness. In [38], it is described a new method for partial eigensolution of large sparse systems named the Refactored Bi-Iteration Method (RBI). A parallel version of this method was tested using the same test system and parallel computers cited above and the results indicate a possible advantage of using the RBI method in the parallel computation of eigenvalues.

4.4 Security Constrained Optimal Power Flow

The Security Constrained Optimal Power Flow (SCOPF) is usually formulated as a nonlinear programming problem of the form [4, 39]:

$$\min_{z_0, z_i} f(z_0) \tag{9}$$

subject to

$$g_i(z_i) = 0, \qquad i = 0, ..., n \tag{10}$$

$$h_i(z_i) \leq 0, \qquad i = 0, ..., n \tag{11}$$

$$\phi(u_i - u_0) \leq \theta_i, \qquad i = 1, ..., n \tag{12}$$

where $z_i = [u_i \ x_i]^T$ is a vector of decision variables, the components of which are the vectors of state or dependent variables (x_i) and the vector of control or independent variables (u_i); $z_0, ..., z_n$ corresponds to the base case (z_0) and post-contingency configurations $(z_i, \ i = 1, ..., n)$, respectively; f is the objective function which depends on the particular application; g_i is a nonlinear vector function representing the power flow constraints for the i^{th} configuration; h_i is a nonlinear vector function representing operating constraints such as limits on line flows or bus voltages for the i^{th} configuration; $\phi(.)$ is a distance metric; and θ_i is a vector of upper bounds reflecting ramp-rate limits. Typical problems involves, for each configuration, around 2000 equality constraints and 4000 inequality constraints. The number of different post-contingency configurations considered (n) may reach several hundreds. An efficient way of deal with the high dimensionality of the problem defined in (9) to (12) is by the use of decomposition techniques [39].

One of first proposed decomposition techniques for the SCOPF is based on the Benders approach [40]. In this method, the problem is divided into a master problem (base case) and subproblems (post-contingency configurations). The solution approach starts solving the base case optimization problem $(i = 0)$ and testing weather this solution satisfies the subproblems constraints $(i = 1, ..., n)$. If necessary, corrective rescheduling is performed in the subproblems. If all subproblems are feasible, then the overall problem is solved. In the case that rescheduling alone is not able to relieve constraint violations in the subproblems, then linear inequality constraints, known as Benders cuts, are incorporated to the base case and the process starts again.

In the Benders decomposition approach to SCOPF, the $n + 1$ subproblems associated with base case and the post-contingency states are independent of each other and can, therefore, be solved in parallel. These subproblems are *loosely coupled* since the amount of information exchanged between the base case and each subproblem is small compared with the local processing effort. This fact has been exploited in the work reported in [41] in synchronous and asynchronous implementations of an algorithm for the solution of a linearized version of (9) to (12). In these implementations, one of the available processors solves the base case while the others solve the subproblems. In the synchronous case, the master problem is idle when the subproblems are being solved, and vice-versa, which leads to a low efficiency use of the multiprocessor system. In the asynchronous case, the latest information available in the subproblems is communicated to the master problem enhancing the use of the processors and, therefore, the overall efficiency of the process. Efficiency up to 82 % has been reported in a test system with 504 buses, 880 circuits, and 72 controllable generators. The parallel machine used was a common-bus 16 cpu system (iAPX-286/287 processor).

In [42], an asynchronous version of a parallel solution of the SCOPF, fairly similar to the one described above, is proposed. The solution method is embedded in a general programming model for exchange of messages and data among processors which allows different problems formulation and facilitates the mapping of the application onto different computer architectures. The method was tested using two test systems: the first one with 725 buses, 1212 branches, 76 adjustable generators, and 900 post-contingency states; and the second one with 1663 buses, 2349 branches, 99 adjustable generators, and 1555 post-contingency state. Tests with the smaller system, on a shared-memory common-bus machine with 9 nodes, achieved efficiency values similar to the ones reported in [41]. In the tests with the larger system, in a 64 node distributed memory nCube machine, the achieved efficiency was around 65 %.

In [43], an asynchronous decomposed version of the SCOPF, based on the technique proposed in [44], was implemented in a network of DEC5000 workstations using PVM. The method allows the representation of soft constraints to model operating limits which need not to be enforced sharply. Reported results indicate that the accuracy of the results is not affected by the lag in communication.

4.5 State Estimation

State estimation is a basic module in the Energy Management System (EMS) advanced application software. Its main function is to provide reliable estimates of the quantities required for monitoring and control of the electric power system. In almost all state estimation implementations, a set of measurements obtained by the data acquisition system throughout the whole supervised network, at approximately the same time instant, is centrally processed by a static state estimator at regular intervals or by operator's request. Modern high speed data acquisition equipment is able to obtain new sets of measurements every 1-10 seconds but the present EMS hardware and software allow state estimation processing only every few minutes. It has been argued that a more useful state estimation operational scheme would be achieved by shortening the time interval between consecutive state estimations to allow a closer monitoring of the system evolution particularly in emergency situations in which the system state changes rapidly. Another industry trend is to enlarge the supervised network by extending state estimation to low voltage subnetworks. These trends pose the challenge of performing state estimation in a few seconds for networks with thousands of nodes.

The higher frequency in state estimation execution requires the development of faster state estimation algorithms. The larger size of the supervised networks will increase the demand on the numerical stability of the algorithms. Conventional centralized state estimation methods have reached a development stage in which substantial improvements in either speed or numerical robustness are not likely to occur. These facts, together with the technical developments on distributed EMS, based on fast data communication network technology, opens up the possibility of parallel and distributed implementations of the state estimation function.

The information model used in power system state estimation is represented by the equation

$$z = h(x) + \omega \tag{13}$$

where z is a $(m \times 1)$ measurement vector, x is a $(n \times 1)$ true state vector, $h(.)$ is a $(m \times 1)$ vector of nonlinear functions, ω is a $(m \times 1)$ measurement error vector, m is the number of measurements, and n is the number of state variables. The usual choice for state variables are the voltage phase angles and magnitudes while the measurements are active and reactive power flows and node injections and voltage magnitudes.

A distributed state estimation algorithm, based on dual recursive quadratic programming, is reported in [45]. The algorithm is aimed to perform distributed estimation at the bus level. Reported results indicate a limited computational performance. An improved version of this distributed estimator, including a distributed bad data processing scheme, is proposed in [46]. In the work reported in [47], the possibility of parallel and distributed state estimation implementation was exploited leading to a solution methodology based on conventional state estimation algorithms and a coupling constraint optimization technique. The proposed methodology performs conventional state estimation at the area level and combines these distributed estimations in a way to eliminate discrepancies in the boundary buses. The proposed method was tested on a simulated distributed environment with considerable speed up of the estimation process.

4.6 Composite Generation-Transmission Reliability Evaluation

The reliability assessment of a composite generation-transmission system consists in the evaluation of several probabilistic indices such as the loss of load probability, expected power not supplied, frequency and duration, etc., using stochastic simulation models of the power system operation. A conceptual algorithm for reliability evaluation can be stated as follows [7, 8]:

1. Select a system state x, or a system scenario, corresponding to a particular load level, equipment availability, operating conditions, etc.
2. Calculate the value of a *test function* $F(x)$ which verifies whether there are system limits violations in this specific scenario. The effect of remedial actions, such as generation rescheduling, load curtailment, etc., may be included in this assessment.
3. Update the expected value of the reliability indices based on the result obtained in 2.
4. If the accuracy of the estimates is acceptable, stop. Otherwise, go back to 1.

Step 1 in the algorithm above is usually performed by one of the following methods: enumeration or Monte Carlo sampling. In both approaches, the number of selected scenarios may reach several thousands for practical size systems. Step 2 requires the evaluation of the effect of forced outages in the system behavior for each of the selected scenarios. Static models (power flow) have been mostly used in these evaluations although some dynamic models have also been proposed. Remedial actions may be simulated by special versions of an optimal power flow program.

Step 2 of the conceptual algorithm above is by far the most computer demanding part of the composite reliability evaluation function. It requires the computation of thousands of power flow solutions. Fortunately, these computations are independent and can be carried out easily in parallel. Step 1, also, can be parallelized.

One of the first attempts to parallelize the composite reliability evaluation is the work described in [48]. In this work, a computer package developed for the Electric Power Research Institute (Palo Alto, USA), named Syrel, was adapted to run on multicomputers with hypercube topology (Intel iPSC/1 and iPSC/2). Syrel uses the enumeration approach to perform step 1 of the conceptual composite reliability algorithm. Reported tests with medium size systems (101 and 140 buses) show efficiencies around 70% on the iPSC/1 and 46% on the iPSC/2 (both machines with 16 processors). It should be pointed out that these relatively low efficiencies may be explained by the difficulty in parallelizing a large code (20,000 lines, 148 subroutines) originally developed for sequential computers without substantial changes in the code.

In [49], a parallel version of the Monte Carlo reliability evaluation algorithm was implemented in a 16 node multiprocessor system based on the iAPX 286/287 processor and a common bus shared memory architecture. Tests performed with a large scale model of an actual power system achieved an efficiency close to theoretical maximum efficiency.

In [50], an extensive investigation of topologies for scheduling processes in a parallel implementation of a composite reliability evaluation method based on Monte Carlo simulation is reported. Also, the important issue of generating independent random sequences in each processor is discussed. The schemes studied were implemented in two computer architectures: a distributed memory 64 nodes nCube 2 and a shared memory 10 nodes Sequence Balance. The power system model used in the tests is a synthetic network made up of three areas each of which is the IEEE Reliability Test System. Efficiencies around 50% was achieved in the nCube2 and closer to 100% on the Sequence Balance.

4.7 Power Flow and Contingency Analysis

Power flow is a fundamental tool in power system studies. It is by far the most often used program in evaluating system security, configuration adequacy, etc., and as a starting point for other computations such as short circuit, dynamic simulation, etc. Its efficient solution is certainly a fundamental requirement for the overall efficiency of several integrated power system analysis and synthesis programs. Therefore, it should be expected a great research effort in the parallelization of the power flow algorithms. That has not been the case, however, for two main reasons:

- The practical power flow problem is much more difficult to parallelize than other similar problems owing to the constraints added to the basic system of non-linear algebraic equations.
- Very efficient algorithms are already available which can solve large power flow problems (more than 1000 nodes) in a few seconds on relatively inexpensive computers.

More interesting investigatory lines are the parallelization of multiple power flow solutions (contingency analysis, for instance) and the speed up of power flow programs on vector and superscalar processors. In [51], it is proposed a version of the Newton-Raphson power flow method in which the linearized system of equations is solved by a variant of the Conjugate Gradient Method (Bi-CGSTAB method) with variable convergence tolerance. Results of tests performed in a Cray EL96 computer and a 616 buses model of the Brazilian power system indicates a substantial speedup when compared with the conventional approach.

4.8 Heuristic Search Techniques

The use of heuristic search techniques, such as Simulated Annealing, Genetic Algorithms, Evolutionary Computing, etc., in power system optimization problems has been growing steadily in the last few years. The motivation for the use of such techniques originates in the combinatorial nature of some problems combined with difficult mathematical models (multimodal search space, discontinuities , etc.). These technique have been applied to a variety of power system problems: generation, transmission, and distribution expansion planning, reactive power optimization, unit commitment, economic dispatch, etc. The results reported in the literature indicate that these heuristic search procedure have a great potential for finding global optimal solution to power system problems. However, the computational requirements are usually high in the case of large scale systems.

Parallel implementations of these heuristic search methods have been proposed to overcome this difficulty. In [52], it is reported an implementation of a parallel genetic algorithm for the optimal long-range generation expansion planning problem. The proposed method was tested on a network of Transputers and presented a considerable reduction in computation time in comparison with a conventional approach using dynamic programming. In [53], a parallel simulated annealing method is proposed for the solution of the transmission expansion planning problem. The results obtained show a considerable improvement in terms of reduction of the computing time and quality of the obtained solution.

5 Industrial Implementations

Most of the applications of HPC in power systems effectively used in practice are in the development of real-time simulators. In the following, some of these implementations are described.

5.1 Real-Time Digital Simulator at TEPCO

This simulator, already referred to in section 4.1 of this paper, was developed by Mitsubishi for Tokyo Electric Power Company [23]. The simulator is based on a multicomputer with 512 nodes developed by nCube with a hypercube topology. The multicomputer is interfaced with electronic apparatus through high speed A/D converters. This simulator was able to simulate in real-time the electromechanical transients of a system with 491 busses . The parallel algorithm used in this simulator allocates one processor for each network bus. In this way, the differential equations representing the dynamic behavior of system components connected to a bus are solved in the corresponding processor. The algebraic equations representing the network model are allocated one for each processor and solved by a Gauss-Seidel like procedure. The efficiency achieved in the process is very low as most of the processors time is spent in data communication. However, owing to the large number of processors available, it was possible to achieve the real-time simulation of a practical size power system.

5.2 RTDS of Manitoba HVDC Research Center

This simulator was developed with the objective of real-time simulation of electromagnetic transients in HVDC transmission systems [54, 55]. The simulator uses a parallel architecture based on state-of-the-art DSPs (Digital Signal Processors). A DSP is a processor specially designed for signal processing which is able to simulate power system transients with time steps in the order of 50ms to 100ms. This allows the simulation of high frequency transients which are almost impossible to simulate with the standard processors available in general purpose parallel machines owing to the clock speed of these processors. The software used in this simulator is based on the same mathematical formulation described in section 4.1 and used in most modern digital electromagnetic transient programs [30].

5.3 Supercomputing at Hydro-Quebec

Hydro-Quebec commissioned a Cray X-MP/216 supercomputer in its Centre d'Analyse Numerique de Reseaux in 1991 to be used as a number crunching server for a network of Sun workstations [56]. This supercomputer has been used for transient stability studies using the PSS/E package and electromagnetic transients computations using the EMTP program. In the case of transient stability, models of the Hydro-Quebec system with 12000 buses, which required up to 45 hours of cpu time for a complete study in a workstation, run in the supercomputer in less than 2 hours.

6 Conclusions

High performance computing may be the only way to make viable some power system applications requiring computing capabilities not available in conventional machines, like real-time dynamic security assessment, security constrained optimal power flow, real-time simulation of electromagnetic and electromechanical transients, composite reliability assessment using realistic models, etc. Parallel computers are presently available in a price range compatible with power system applications and presenting the required computation power.

Two main factors are still impairments to the wide acceptance of these machines in power system applications: the requirements for reprogramming or redevelopment of applications and the uncertainty about the prevailing parallel architecture. The first problem is inevitable, as automatic parallelization tools are not likely to become practical in the near future, but has been minimized by the research effort in parallel algorithms and the availability of more efficient programming tools. The second difficulty is becoming less important with the maturity of the high performance computing industry.

Likewise in the history of sequential computer evolution, a unique and overwhelming solution to the parallel computer architecture problem is not to be expected. It is more likely that a few different architectures will be successful in the next few years and the users will have to decide which one is the most adequate for their application. Moreover, it is likely that commercial processing applications, which are now turning towards parallel processing, are the ones that will shape the future parallel computer market. However, to make this scenario a little bit less uncertain, it should be pointed out the tendency in the parallel computer industry to make their products follow open system standards and the possibility of developing applications less dependent on a particular architecture.

References

1. T.G. Lewis and H. El-Rewini. *Introduction to Parallel Computing.* Prentice Hall, New York, 1992.
2. M.J. Quinn. *Parallel Computing: Theory and Practice.* McGraw-Hill, New York, 1994.
3. G.A. Geist and V.S. Sunderam. Network-based concurrent computing on the PVM system. *Concurrency: Practice and Experience*, 4(4):293–311, June 1992.
4. B. Stott, O. Alsac, and A. Monticelli. Security analysis and optimization. *Proceedings of the IEEE*, 75(12):1623–1644, December 1987.
5. N.J. Balu and et al. On-line power system security analysis. *Proceedings of the IEEE*, 80(2):262–280, February 1992.
6. Y. Sekine, K. Takahashi, and T. Sakaguchi. Real-time simulation of power system dynamics. *Int. J. of Electrical Power and Energy Systems*, 16(3):145–156, 1994.
7. M.V.F. Pereira and N.J. Balu. Composite generation/transmission reliability evaluation. *Proceedings of the IEEE*, 80(4):470–491, April 1992.
8. R. Billinton and W. Li. *Reliability Assessment of Electric Power Systems Using Monte Carlo Method.* Plenum Press, New York, 1994.

9. B. Stott. Power system dynamic response calculations. *Proceedings of the IEEE*, 67(2):219–241, February 1979.
10. P. Kundur. *Power System Stability and Control*. McGraw-Hill, New York, 1994.
11. J.Q. Wu and A. Bose. Parallel solution of large sparse matrix equations and parallel power flow. *IEEE Transactions on Power Systems*, 10(3):1343–1349, August 1995.
12. J.Q. Wu and A. Bose. A new successive relaxation scheme for the W-matrix solution method on a shared-memory parallel computer. *IEEE Transactions on Power Systems*, 11(1):233–238, February 1996.
13. G.T. Vuong, R. Chahine, G.P. Granelli, and R. Montagna. Dependency-based algorithm for vector processing of sparse matrix forward/backward substitutions. *IEEE Transactions on Power Systems*, 11(1):198–205, February 1996.
14. I.C. Decker, D.M. Falcão, and E. Kaszkurewicz. Parallel implementation of a power system dynamic simulation methodology using the conjugate gradient method. *IEEE Transactions on Power Systems*, 7(1):458–465, February 1992.
15. F.D. Galiana, H. Javidi, and S. McFee. On the application of pre-conditioned conjugate gradient algorithm to power network analysis. *IEEE Transactions on Power Systems*, 9(2):629–636, May 1994.
16. I.C. Decker, D.M. Falcão, and E. Kaszkurewicz. Conjugate gradient methods for power system dynamic simulation on parallel computers. *IEEE Transactions on Power Systems*, 11(3):1218–1227, August 1996.
17. H. Dağ and F.L. Alvarado. Computation-free preconditioners for the parallel solution of power system problems. *IEEE Transactions on Power Systems*, to appear.
18. H. Dağ and F.L. Alvarado. Toward improved use of the conjugate gradient method for power system applications. *IEEE Transactions on Power Systems*, to appear.
19. M.H.M Vale, D.M. Falcão, and E. Kaszkurewicz. Electrical power network decomposition for parallel computations. In *Proceedings of the IEEE Symposium on Circuits and Systems*, pages 2761–2764, San Diego, USA, May 1992.
20. J.S. Chai, N. Zhu, A. Bose, and D.J. Tylavsky. Parallel newton type methods for power system stability analysis using local and shared memory multiprocessors. *IEEE Transactions on Power Systems*, 6(4):1539–1545, November 1991.
21. J.S. Chai and A. Bose. Bottlenecks in parallel algorithms for power system stability analysis. *IEEE Transactions on Power Systems*, 8(1):9–15, February 1993.
22. A. Padilha and A. Morelato. A W-matrix methodology for solving sparse network equations on multiprocessor computers. *IEEE Transactions on Power Systems*, 7(3):1023–1030, 1992.
23. H. Taoka, I. Iyoda, H. Noguchi, N. Sato, and T. Nakazawa. Real-time digital simulator for power system analysis on a hypercube computer. *IEEE Transactions on Power Systems*, 7(1):1–10, February 1992.
24. M.Ilić-Spong, M.L. Crow, and M.A. Pai. Transient stability simulation by waveform relaxation methods. *IEEE Transactions on Power Systems*, 2(4):943–952, November 1987.
25. M.L. Crow and M. Ilić. The parallel implementation of the waveform relaxation method for transient stability simulations. *IEEE Transactions on Power Systems*, 5(3):922–932, August 1990.
26. L. Hou and A. Bose. Implementation of the waveform relaxation algorithm on a shared memory computer for the transient stability problem. *IEEE Transactions on Power Systems*, to appear.
27. F. Alvarado. Parallel solution of transient problems by trapezoidal integration. *IEEE Transactions on Power Apparatus and Systems*, 98(3):1080–1090, May/June

1979.
28. M. LaScala, A. Bose, D.J. Tylavsky, and J.S. Chai. A highly parallel method for transient stability analysis. *IEEE Transactions on Power Systems*, 5(4):1439–1446, November 1990.
29. M. LaScala, M. Brucoli, F. Torelli, and M. Trovato. A Gauss-Jacobi-block-Newton method for parallel transient stability analysis. *IEEE Transactions on Power Systems*, 5(4):1168–1177, November 1990.
30. H.W. Dommel and W.S. Meyer. Computation of electromagnetic transients. *Proceedings of the IEEE*, 62:983–993, July 1974.
31. D.M. Falcão, E. Kaszkurewicz, and H.L.S. Almeida. Application of parallel processing techniques to the simulation of power system electromagnetic transients. *IEEE Transactions on Power Systems*, 8(1):90–96, February 1993.
32. R.C. Durie and C. Pottle. An extensible real-time digital transient network analyzer. *IEEE Transactions on Power Systems*, 8(1):84–89, February 1993.
33. K. Werlen and H. Glavitsch. Computation of transients by parallel processing. *IEEE Transactions on Power Delivery*, 8(3):1579–1585, July 1993.
34. J.R. Marti and L.R. Linares. A real-time EMTP simulator. *IEEE Transactions on Power Systems*, 9(3):1309–1317, August 1994.
35. N. Martins. Efficient eigenvalue and frequency response methods applied to power system small-signal stability studies. *IEEE Transactions on Power Systems*, 1(1):217–226, February 1986.
36. J.M. Campagnolo, N. Martins, J.L.R. Pereira, L.T. G. Lima, H.J.C.P. Pinto, and D.M.Falcão. Fast small-signal stability assessment using parallel processing. *IEEE Transactions on Power Systems*, 9(2):949–956, May 1994.
37. J.M. Campagnolo, N. Martins, and D.M. Falcão. An efficient and robust eigenvalue method for small-signal stability assessment in parallel computers. *IEEE Transactions on Power Systems*, 10(1):506–511, February 1995.
38. J.M. Campagnolo, N. Martins, and D.M. Falcão. Refactored bi-iteration: A high performance eigensolution method for large power system matrices. *IEEE Transactions on Power Systems*, 11(3):1228–1235, August 1996.
39. S.M. Shahidehpour and V.C. Ramesh. Nonlinear programming algorithms and decomposition strategies for OPF. In *Optimal Power Flow: Solution Techniques, Requirements, and Challenges*, pages 10–25. IEEE Tutorial Course 96 TP 111-0, 1996.
40. A. Monticelli, M.V.F. Pereira, and S. Granville. Security constrained optimal power flow with post-contingency corrective rescheduling. *IEEE Transactions on Power Systems*, 2(1):175–182, February 1987.
41. H.J.C. Pinto, M.V.F. Pereira, and M.J. Teixeira. New parallel algorithms for the security-constrained dispatch with post-contingency corrective actions. In *Proceedings of the 10th Power Systems Computation Conference*, pages 848–853, Graz, Austria, August 1990.
42. M. Rodrigues, O.R. Saavedra, and A. Monticelli. Asynchronous programming model for the concurrent solution of the security constrained optimal power flow problem. *IEEE Transactions on Power Systems*, 9(4):2021–2027, November 1994.
43. V.C. Ramesh and S.N. Talukdar. A parallel asynchronous decomposition for online contingency planning. *IEEE Transactions on Power Systems*, 11(1):344–349, February 1996.
44. S.N. Talukdar and V.C. Ramesh. A multi-agent technique for contingency-constrained optimal power flows. *IEEE Transactions on Power Systems*, 9(2):855–861, May 1994.

45. S.-Y. Lin. A distributed state estimation for electric power systems. *IEEE Transactions on Power Systems*, 7(2):551–557, May 1992.
46. S.-Y. Lin and C.-H. Lin. An implementable distributed state estimation and distributed data processing scheme for electric power systems. *IEEE Transactions on Power Systems*, 9(2):1277–1284, May 1994.
47. D.M. Falcão, F.F. Wu, and L. Murphy. Parallel and distributed state estimation. *IEEE Transactions on Power Systems*, 10(2):724–730, May 1995.
48. D.J. Boratynska-Stadnicka, M.G. Lauby, and J.E. Van Ness. Converting an existing computer code to a hypercube computer. In *Proceedings of the IEEE Power Industry Computer Applications Conference*, Seattle, USA, May 1989.
49. M.J. Teixeira, H.J.C. Pinto, M.V.F. Pereira, and M.F. McCoy. Developing concurrent processing applications to power system planning and operations. *IEEE Transactions on Power Systems*, 5(2):659–664, May 1990.
50. N. Gubbala and C. Singh. Models and considerations for parallel implementation of Monte Carlo simulation methods for power system reliability evaluation. *IEEE Transactions on Power Systems*, 10(2):779–787, May 1995.
51. C.L.T. Borges, A.L.G.A. Coutinho, and D.M. Falcão. Power flow solution in vector computers using the Bi-CGSTAB method. In *Proceedings of the XI Congresso Brasileiro de Automática*, (in Portuguese), São Paulo, Brasil, May 1996.
52. Y. Fukuyama and H.-D. Chiang. A parallel genetic algorithm for generation expansion planning. *IEEE Transactions on Power Systems*, 11(2):955–961, May 1996.
53. R.A. Gallego, A.B. Alves, A. Monticelli, and R. Romero. Parallel simulated annealing applied to long term transmission network expansion planning. *IEEE Transactions on Power Systems*, to appear.
54. D. Brandt, R. Wachal, R. Valiquette, and R. Wierckx. Closed loop testing of a joint VAR controller using a digital real-time simulator. *IEEE Transactions on Power Systems*, 6(3):1140–1146, August 1991.
55. P. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht, and L. Arendt. A real-time digital simulator for testing relays. *IEEE Transactions on Power Delivery*, 7(1), January 1992.
56. G.T. Vuong, R. Chahine, and S. Behling. Supercomputing for power system analysis. *IEEE Computer Applications in Power*, 5(3):45–49, July 1992.